

Mail merge

Mail merge is, without doubt, the most powerful feature of the word processor built into your Notebook. In its simplest form it will allow you to make a simple template letter and then print multiple copies with different people's name and addresses on each. This is useful for printing things such as club newsletters.

The actual range of mail merge commands available is quite large and, in effect, forms the basis of a very simple programming language. For example, type the following lines into a new document and then Stop editing and print it. (Don't worry if you don't understand what the commands mean yet)

```
>CS This "program" prints the 7 times table
>WT Hit any key to start...
>SV count = 1
>RP
>SV result = count * 7
>SV result = result[w1]
>SV count = count[w1]
>DM &count& * 7 = &result&
>SV count = count + 1
>UN count = 13
```

That gives you just a taste of what can be achieved with mail merge but let's start at the very beginning and try to understand exactly what mail merge is.

Simple mail merging

In its simplest form mail merge is just the process of taking a document (or file) full of data such as names and addresses and slotting them into a main template letter. For example, suppose you have a document called "names" that contains the details of the members of your club in the following form (it might be an idea to type this in if you want to follow this chapter by trying the examples) - Leave a blank line after each 5 line entry:

```
John Smith
27 Acacia Avenue
Surbiton
Surrey
ABC 1HW
```

177

I write to remind you that your subscription is now due. Just to keep our records up to date please can confirm your name and address details are as shown below:

```
&name&
&addr1&
&addr2&
&addr3&
&postcode&
```

Yours sincerely,

Cliff Lawson

The "&" is a special indicator to the word processor that the word contained within the "&" symbols is the name of a mail merge "variable" which it must fill in when it prints the letter.

So that is our template letter, which can be stored in a document called "template" - type it in now if you like. Now we have to have some means of telling the word processor which document contains the list of names and addresses. We do this by using a DF stored command in the template letter. Add the line:

```
>DF names
```

to the top of the document.

However, all the DF command does is tell the word processor where the data is being kept. It has no way of knowing that the names and addresses are organised so that there is a single line with the name, 3 with the address, one with the postcode and one blank line. We must also tell the word processor how to read the data from the data file and which mail merge variables the various items should be stored in. This is achieved using the RV command to read the information from the data file into mail merge variables which can then be inserted into the template letter. Add the following command after the DF command:

```
>RV name, addr1, addr2, addr3, postcode, dummy
```

When this line is executed the word processor will read the first line from the data file and assign it to "name". The second line will go into "addr1" and so on. The reason for having an extra variable called "dummy" is that each address is followed by a blank line to

179

```
Eric Viking
26 Pownell Road
Mucklethwaite
Lancashire
PO3 9BU
```

```
Norman Nobody
1 The Avenue
Dunmow
Essex
GH6 3TU
```

Supposing you want to send each of them a letter to tell them that their subscription to your club is now due. You could write each a separate letter filling in the relevant name and address where appropriate but it would be far easier to get the word processor to do this for you. Consider what the letter would look like if you were writing to John Smith:

15th August 1992

Dear John,

I write to remind you that your subscription is now due. Just to keep our records up to date please confirm your name and address details are as shown below:

```
John Smith
27 Acacia Avenue
Surbiton
Surrey
ABC 1HW
```

Yours sincerely,

Cliff Lawson

It is clear that if we want to turn this letter into a template to be used for everyone we must replace all specific references to John Smith with some sort of "place holder" which will be filled in with different details for each different copy of the letter printed. The letter might then become:

15th August 1992

Dear &name&,

178

separate it from the next and so each time that blank line will be read into "dummy" but not used.

You could now try printing the document to a printer from the "Print document" screen. You should get three copies of the letter printed. Each will have a different name and address. While experimenting with mail merge it can be a waste of paper to repeatedly print to a printer. While editing the template you can see what the results of a mail merge will be by printing the document to the screen. Press [Print] followed by [F7]. The document is shown 16 lines at a time. Press [Page Down] to see each subsequent 15 lines. Press [Esc] to abandon printing a long document to the screen. You will see that where you have included mailmerge variables (&name&) they are filled in with the values from the data file.

More about reading data

There are two commands which can be used to read data from a data file and assign it to mail merge variables. RV is the simpler form of command. It just reads data and assigns it to the named variables until a blank line is read when it stops reading data and sets all the other "unread" variables to be blank. The problem with this is that if we had a shorter address such as:

```
Tom Brown
Rugby School
Rugby
```

```
RU3 8BY
```

The blank line would cause problems if the RV command were used.

When you print the letter you would get one in which Tom's address was used but his postcode would be missed out because the reading of the file would have finished at the blank line between Rugby and RU3 8BY. The next letter printed would have a name of "RU3 8BY"! This clearly is not what we wanted at all.

If the RV command is changed to RU, which will happily read an entry even if it is blank, then things are restored to normal. You will notice that although the names and addresses file contains a blank line between "Rugby" and "RU3 8BY", when printed this blank does not appear. This is because when a variable appears enclosed in

180

"&" characters it does not print anything if the variable is blank. If you really wanted the blank line you could replace the "&" characters with "!" symbols. A variable name enclosed in "!" will print the contents of that variable even if it is blank.

In the data file it is probably safest to put a single dollar sign on any lines which form part of the data but would otherwise be blank (the line between Rugby and the postcode in this case). The word processor knows that a line with a single dollar character on should be read in as a blank line. Just use completely blank lines between each record (collection of data). In this case you would only have completely blank lines between the postcode of one entry and the next name.

Inputting data when printing

At the moment the template letter has the fixed date in it but we may want to use the same template several times and enter the date just before printing. This can be achieved by removing "15th August 1992" and replacing it with &date&.

Try that now and print the template letter. You will get a message saying "Unknown variable 'date'". We need the word processor to stop and ask for date to be typed in when the template is printed. This is achieved with the AV command which will Ask for a Variable to be typed in. Add a line at the top of the template document:

```
>AV "Type in today's date: ", date
```

If you now print the document you will find that there is just one slight problem with this - you are asked to type in the date for every copy of the letter that is printed. You only want to be asked once at the very start so add the following line before the AV command line:

```
>IU date
```

and add the line:

```
>EI
```

after the AV command line. This will mean that the AV command is only used if the variable "date" is undefined. (This is what IU means - If Undefined). Once you have typed the date in it will be defined on subsequent prints of the letter.

181

template, at the very end, you should see the message "Number of copies printed was 4.00". We don't really want the two decimal places to be included in the display of count. This can be achieved by including the following on the line above the CS command:

```
>SV count = count[w1]
```

Splitting data out from a variable

The [w1] is a special command to the word processor which means take only the first "word" contained in count. You might think it odd that we think of count as containing a word when really it holds a number. But the value 4.00 is thought of by the word processor as two words "4" and "00" separated by a full stop. The [w1] at the end of the line takes word 1 from count.

It is possible to split out words from a line of text in a similar way. For example, if the variable "text" was set using:

```
>SV text = "Now is the time for all good men"
text[w1] would be "Now",
text[w3] would be "the",
text[w-1] would be "men" (w-1 means the last word)
text[w4:7] would be "time for all good" (words 4 to 7)
text[w7:] would be "good men" (words 7 to the end)
```

You can also pick out a range of letters, for example text[2:8] would be "ow is t".

You can use this ability to pick out parts of a piece of text within a conditional statement. For example,

```
>IF name[w-1] = "Smith"
```

would only be true if the last word in name (the surname) was Smith.

In our example template we will currently get "Dear John Smith," printed. What we really want is just "Dear John,". You may already have guessed how we might achieve this. After the >RV command add the line:

```
>SV firstname = name[w1]
```

and then modify the line which says Dear &name&, to read:

```
Dear &firstname&,
```

183

Other ways to set variables

So far we have seen that it is possible to set the contents of a mail merge variable using either the RV/RU commands to read the value from a data file or the AV command to get the user to type in a value when the template is printed. It is also possible to set the contents of a variable directly using the SV command.

Suppose we want to keep a count of how many copies of the letter have been printed and show this at the end of printing. This could be achieved as follows. First, on the line after the AV command that asks for the date add:

```
>SV count = 1
```

This sets a variable called 'count' to be 1 on the first time the document is printed. We know this will only happen on the first occasion because we have already seen that the line within the IU and EI block is only used the once. This SV command shows another feature of the mail merge - variables can contain numbers as well as pieces of text and SV can be used to give a value to a variable (up to now we have only used name, addr1, and so on for storing pieces of text).

After each copy has been printed we must increase the value held in count by one. This is done by including the following command at the end of the document:

```
>SV count = count + 1
```

This means that count should be set to the current value of count plus one. The final thing we need is to arrange for a message to be printed on the screen once the last letter has been printed. We know when the last letter has been printed because the data file will be exhausted. We can test for this condition using the IE command. So, on the line above SV count = count + 1 add the following lines:

```
>IE
>CS Number of copies printed was &count&
>EI
```

The CS command will clear the screen and display the following message. The &count& part of the message will be filled in with the current value held in the count variable. When you print the

182

The complete template should read:

```
>DF names
>RV name, addr1, addr2, addr3, postcode, dummy
>SV firstname = name[w1]
>IU date
>AV "Type in today's date : " date
>SV count = 1
>EI
```

```
&date&
```

```
Dear &firstname&,
```

```
I write to remind you that you subscription is now due.
Just to keep our records up to date please can confirm
your name and address details are as shown below:
```

```
&name&
&addr1&
&addr2&
&addr3&
&postcode&
```

```
Yours sincerely,
```

```
Cliff Lawson
```

```
>IE
>SV count = count[w1]
>CS Number of copies printed was &count&
>EI
>SV count = count + 1
```

Using the IF command

As has already been shown, you can arrange for selected pieces of text and selected stored commands to be used by including them in a conditional block that starts with one of the IF commands. You can use ID to test if a variable has been defined. IU will test if a variable is undefined. The commands and text following IE will only be used if the data file is exhausted. However, most conditional tests are made using the general IF command followed by a condition. If the condition is met then the following lines are used. There are various comparisons that can be used in an IF command. they are as follows:

184

Comparison	Meaning	Example
=	equal to	>IF name = "John"
<>	not equal to	>IF (total + 3) <> 5
<	less than	>IF count < 10
<=	less than or equal	>IF vat <= 17.5
>	Greater than	>IF age > 18
>=	greater than or equal	>IF height > 5
IN	is contained in	>IF "Smi" IN name
NOTIN	is not contained in	>IF "081" NOTIN num

The condition can include expressions similar to those that might be used in an SV command. For example:

```
>IF price + delivery + vat > 200
```

Adding variables

When the addition operator is used to add to variables together, the word processor checks to see if they are both numbers. If they are, then the addition is done numerically, otherwise one piece of text is just added onto the end of another. The following examples may illustrate this:

```
>SV name1="Cliff "
>SV name2="Lawson"
>SV num1="37"
>SV num2="53"
name1 + name2 = "Cliff Lawson"
name1 + num1 = "Cliff 37"
num1 + num2 = "90"
num2 + name2 = "53 Lawson"
```

Using names and addresses from the Address Book

As you may already have a list of names and addresses entered into the address book section of your Notebook you may be wondering if you can use them in the data file for a mail merge operation. The answer is yes but because they are stored in a special way within the address book you must extract them into a new document before they can be used. The way to do this is to start a new document (call it "Addresses" maybe). Hit so that the "Start typing new text here" message disappears (this step is

```
>-----R
&descrip& &ret& &quant& &disc& &totret& &net& &vat& &totinc&
>AV "Another entry? Y/N ", yeno
>UN yeno[1]<>"Y"
-----
TOTALS &gnet& &gvat& &gtotinc&
-----
>ST
```

The first few lines just input some necessary information to be printed at the top of the invoice.

The variables **gnet**, **gvat** and **gtotinc** will be used to hold the global totals of nett price, amount of vat and total price including VAT. They are set to zero at the start.

The next few lines will actually be printed at the top of an invoice. Then everything between the RP and subsequent UN line are repeated for each line of the document until you enter something other than Yes to the "Another entry?" question.

For each line the mail merge program asks you to input a description, price, quantity and discount. A special check is made to see if was pressed when discount was asked for and, if so, the variable **disc** is set to zero. The next couple of lines do all the necessary calculations, adding VAT and reducing by any discount. The totals for this line are then added into the global totals.

The next line defines a ruler with decimal tabs so that all the numbers line up and the following line actually prints the information into the invoice.

absolutely vital!), then press to switch to the address book function. Use the keys to browse through your addresses until you find the first one that you want to use for mail merging. Now press to transfer that into your document. Repeat this process for each entry you want to add. Now make sure that each has the same number of lines and that there is a blank line after each. You may want to delete some of the lines from each one, for example, remove the line that has the fax number on. A quick way to delete a whole line in the word processor is to press while the cursor is on the line.

Mail merge without a data file

The above description of mail merge has shown the way it is used with two files, one full of names and addresses and the other a template letter with sections to be filled in. There is a second general use for the mail merge feature and this is to construct just a template which, when printed, asks you to input various pieces of information and just fills them in immediately before printing. The obvious example of this is printing invoices. The following shows an example of this:

```
>AV "Enter today's date : ", date
>AV "Enter current VAT rate % : ", vatrate
>AV "Type invoice number : ", invnum
>SV gnet=0 gvat=0 gtotinc=0
INVOICE
Invoice number: &invnum& Date: &date&
Description Retail Quant Disc Total Net VAT TOTAL
>RP
>AV "Description of goods? " descrip 12
>AV "Retail price? " ret "Quantity?" quant
>AV "Discount % ?" disc
>IF disc=""
>SV disc = 0
>EI
>SV totret=ret * quant off=totret/100*disc
>SV net=totret-off vat=net*vatrate vat=vat/100
>SV totinc=net+vat gnet=gnet+net gvat=gvat+vat
>SV gtotinc=gtotinc+totinc
```