Making Columnar Reports in MoneyWorks 4.1.7 and later

Date: 23 Nov 05
Revised: -

The MoneyWorks report writer makes it easy to make reports where the columns contain time-based values (such as actual year-to-date, or annual budget). But what about where you want something else, such as departments, classifications or different types of general ledger code?

Departments as Columns

Consider for example where we have four departments in a group called SHOPS representing branches, say NORTH, EAST, SOUTH and WEST, and want a report with a separate column for each branch showing year-to-date values, something along the lines of:

	NORTH	EAST	SOUTH	WEST	TOTAL
Income	544.44	88.89	0.00	0.00	633.33
Less: Expenses	0.00	250.00	0.00	100.00	350.00
Profit	**544.44**	**(161.11)**	**0.00**	**(100.00)**	**283.33**

The basic approach is to have a column that calculates the year-to-date (YTD), and then 4 additional columns (one for each department) that are calculated by a formula which filters the values in each column based on a test as to whether the ledger value being printed belongs to this column or not. In our example, the first column would have a formula of:

```
if(ledger.department = "NORTH", YTD, 0) //!#
```

and the actual calculation looks like:

Note that the //!# (pronounced "slash-slash-bang-hash") is required. Basically it forces the column to accumulate down, instead of across. Thus when we print a total, the total for this column will be the sum of the preceding values, not the calculation of the total of the YTD field (which, because the ledger.department will never be NORTH, will always evaluate to zero).

This is repeated for each of the columns EAST, SOUTH and WEST.

The column parts can be any of the standard parts. Thus the following simple report:
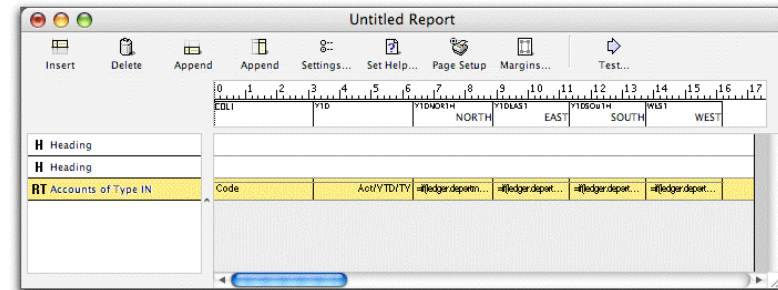
will produce:

		NORTH	EAST	SOUTH	WEST
1000	1,067.60	0.00	0.00	0.00	0.00
1200-EAST	(88.89)	0.00	(88.89)	0.00	0.00
1200-NORTH	(544.44)	(544.44)	0.00	0.00	0.00
1500	(60.00)	0.00	0.00	0.00	0.00

Note:

1. Because 1000 and 1500 are not departmentalised, they are not split into columns;
2. To isolate the departments, the report MUST be run with "Show departments"

To eliminate the non-departmental accounts, we need to use a *Find* part instead of *Accounts of Type*. In this instance, we would have a part with a Find Account:

```
Type = "IN" and Group = "SHOPS"
```
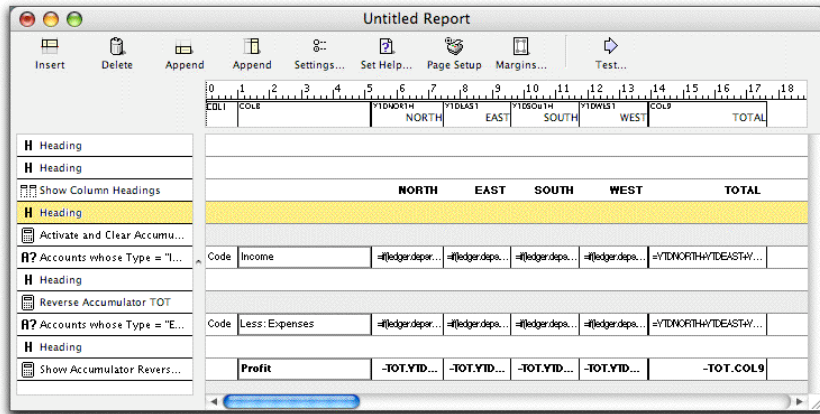
which gives the following:

		NORTH	EAST	SOUTH	WEST
1200-EAST	(88.89)	0.00	(88.89)	0.00	0.00
1200-NORTH	(544.44)	(544.44)	0.00	0.00	0.00

If we now set the "Summary Only" option we get:

		NORTH	EAST	SOUTH	WEST
	(633.33)	(544.44)	(88.89)	0.00	0.00

This implies that we are going to need a separate part for each line that we want to appear on the report.

The final report specification is:

Note that the YTD figure is calculated in a hidden column to the left of the NORTH column.

Classifications as Columns

The principle is the same as for departments, except that the column search expression is modified to check on the classification of the department, using the Lookup function. E.g.

```
if(lookup(ledger.department, "department.classification") = "COM", YTD, 0) //!#
```

will pick up the values of all departments whose classification is COMM.

Combining different accounts into columns

Sometimes your ledger might be segmented into different parts, for example one section for New Zealand operations, and one for Australian.

In this case the column calculations will be based on account codes, not ledger. So if we use the NZ/Australia as an example, we need something in the account record to tell us whether the account is for NZ or Australia. This could be embedded in the account code (e.g. the first letter starts with "A" for australian accounts), but in this example we will assume that the Category4 field of the account contains an "N" for NZ and an "A" for Australia.

The NZ column (NZYTD) calculation then becomes:

```
if(account.category4 = "N", YTD, 0) //!#
```

With the Australian column (AUSYTD) being:

```
if(account.category4 = "A", YTD, 0) //!#
```

Of course if people forget to put an A or N into the category4 field, the report will be incorrect. This can be checked by seeing that the NZ plus the Australian add up to the total. The check can be a final column with the formula:

```
if(NZYTD + AUSYTD = YTD, "", "<< ERR")
```

Provided the column format is set to *None*, the message "<< ERR" will appear next to any lines where the category4 has not been set correctly.

In terms of the parts going down the report, we are going to need a separate part for each line printed (remembering that we are grouping different accounts from different parts of the ledger). In this case it seems sensible to use another of the Category fields, or probably better still the accountants code. So a

typical part might be:
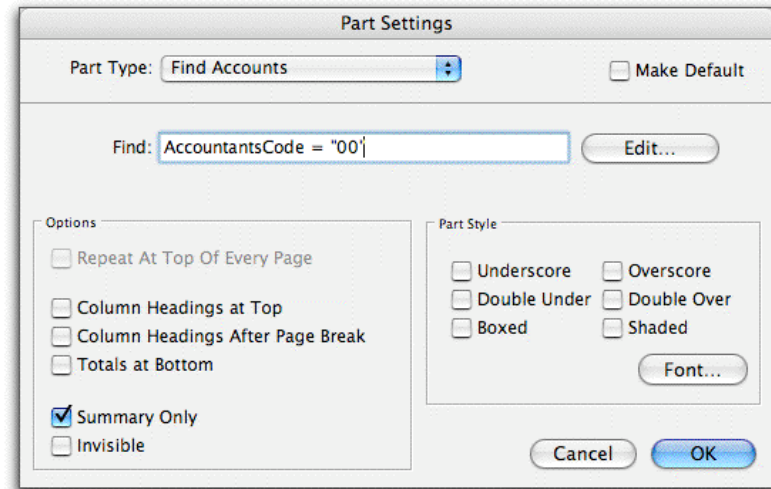
Summary

Reports can be produced with non-period values in the column by using an appropriate filter (if statement) in a column calculation. Because each line in the report is effectively a summary, a separate Part will be required. The Description on each part will probably have to be specified with a cell calculation.

It is a good idea to have a sanity check on the report, in the form of a final column that checks the column totals against the original total value, and in certain cases by additional parts that check the filtered report totals against the non-filtered values. These Parts can be invisible, and if a discrepancy is detected it can be flagged using a *Heading* part enclosed by *If … End If* Parts