R&R Report Writer Technical Bulletin

Assorted Tips and Tricks

Product: R&R Report Writer® for DOS® **Oper Sys:** DOS
Version: All

Summary

This technical bulletin includes a variety of information that may be helpful when using R&R Report Writer for DOS.

Alphabetizing Titles

Here's a tip for alphabetizing titles that applies to books, magazines, articles, films, etc. The problem you will avoid is that titles beginning with words like *A* and *The* are alphabetized under *A* and *T*.

Rather than adopting a data entry convention for entering such titles, why not let R&R Report Writer do the work? The WORD function can be used to shuffle the word order so that titles can be sorted in the desired order.

Suppose you've entered the following book titles into your database.

```
The dBASE® Programmer's Handbook
R&R User's Guide
A Guide to Database Systems
```

In the report, you would like the books to be alphabetized by title and appear as follows.

```
dBASE® Programmer's Handbook, The
Guide to Database Systems, A
R&R User's Guide
```

First, use the /Field Calculate Create command to create a field named Sort Title. Enter the following expression to transform the database field, TITLE, into the proper format to be sorted alphabetically.

```
IIF(INLIST(WORD(TITLE,1),"A","An","The"),
      RTRIM(RIGHT(TITLE,LEN(TITLE)-AT(" ",TITLE)))+
        ", "+WORD(TITLE,1),TITLE)
```

This expression says: "If the first word of the title field is *A*, *An*, or *The*, move it to the end of the title and precede it with a comma. Otherwise leave the title alone."

To complete the sort specification, use the /Sort Group/Sort Fields command to select Sort Title instead of TITLE as the sort field. Also, insert Sort Title on the report layout instead of TITLE.

Once-only Currency Signs

It is often desirable to have dollar signs, or whatever currency string you're using, print only once at the top of a column of numbers.

The R&R Report Writer currency format (/FFC) always includes the dollar sign when the field is printed. The following table illustrates the difference between printing dollar signs *always* versus *once*.

Always	Once
$1	$ 1
$10	10
$100	100

There are three ways to accomplish this, depending on whether you want a fixed or floating dollar sign, and whether you want the dollar sign touching the number or separated by spaces.

The easiest way is to have the dollar sign in a fixed column of the report. The example above shows the dollar sign at a fixed location under the *O* in *Once*.

First, use the /Field Total Create command to make a field named Count. Count any field that contains data in every record. Select the appropriate reset level depending on how frequently you want the dollar sign to print. For example, select Page to print it once on each page.

Next, use the /Field Calculate Create command to make a field named DollarSign with the following expression.

```
IIF(COUNT=1, "$", "")
```

Finally, insert DollarSign two spaces to the left of the numeric field. If you want to place the dollar sign up against the number, use the /Print Options Record-Layout command to set the option Trim adjacent fields to No. This prevents left alignment of the numbers.

You may want to make the dollar sign float, meaning have it move so that it is always the same number of spaces in front of the number regardless of the number of digits. In this example, assume the name of the numeric field is AMOUNT and that it is 3 characters wide with no decimal places. The dollar sign is placed next to the number.

Fixed		Floating
$ 1	. . .Page 1. . .	$1
$ 10	. . .Page 2. . .	$10
$100	. . .Page 3. . .	$100

To make a floating dollar sign, first make the two fields Count and DollarSign described above. Then calculate a field named DollarAmt with the following expression.

```
DollarSign + LTRIM(STR(AMOUNT,3,0))
```

Insert this field on the report in place of the AMOUNT field. Since this is a character field, use the /Field Format Right command to right align the numbers.

Spelling Dollar Amounts

The SPELLNUM function offers an easy way to spell whole numbers. But if you are printing checks, you need to print the cents also. Here's how spelled numbers typically appear on checks. (The word *Dollars* is usually printed on the check.)

```
Fifty-six and 75/100
Thirty-nine
```

In this example, the number is contained in a field named AMOUNT. Use the /Field Calculate Create command to define a field named SpellAmt with the following expression.

```
SPELLNUM(AMOUNT)+
        IIF(AMOUNT-INT(AMOUNT)," and "+
          STR((AMOUNT-INT(AMOUNT))*100,2,0)+"/100",
            "")
```

This expression first spells the integer portion of the amount. The second line tests whether there is a fraction. If there is a fraction, the word *and* is appended and the third line appends the number of cents followed by */100*. If there isn't a fraction, the fourth line appends nothing.

Finding Earliest and Latest Dates

The R&R Report Writer total functions for *Highest* and *Lowest* apply only to numeric data. Using this technique, applications can calculate the earliest and latest dates in a group of records as well.

Since R&R Report Writer for DOS only allows you to total numeric fields, the trick is to convert the dates to numbers, apply the Highest and Lowest total functions, and convert the numbers back to dates to print them.

There are many ways to convert a date to a number but in this example we will use one that keeps the expressions as simple as possible. Select a *base date* that precedes all dates in the database; for example, 01/01/1900.

The following expression converts a date to the number of days since the base date. Use the /Field Calculate Create command to convert the date field DATE to a numeric field NumDate using the following expression.

```
DATE-CTOD("01/01/1900")
```

Now you can use the /Field Total Create command to create LoNum and HiNum fields which contain the lowest and highest values for the NumDate field. Rather than insert these fields on the report, convert them back to dates so they are properly formatted.

Use the /Field Calculate Create command once again to convert LoNum and HiNum from numbers back to dates named LoDate and HiDate. Enter the following calculated field expressions.

```
LoDate  = CTOD("01/01/1900")+LoNum
HiDate  = CTOD("01/01/1900")+HiNum
```

Because the expressions are almost identical, you can use a shortcut to enter them. After defining LoDate, create the HiDate field. When you are prompted to enter the expression, press [F10], select Expressions, select LoDate, and change LoNum to HiNum.

Weekday Arithmetic

Having calculated the earliest and latest dates using the tip above, it may be useful to calculate the number of elapsed weekdays between the two dates. This technique is often used in billing and cost accounting applications.

It is not as easy as it might seem to calculate the number of weekdays between two dates. The DAYSBTWN function includes weekends, so you need to implement your own algorithm. Although this type of problem would typically be solved by writing a dBASE procedure, it can also be solved using calculated field expressions.

The algorithm is based on finding the number of weekdays in the first week, all the middle weeks, and the last week, and then adding them together. To accomplish this, the following calculations are needed. Field names you will use are included for reference.

StartDay	=	day of the week of start date
EndDay	=	day of the week of end date
NextFriday	=	date of Friday after start date
LastSunday	=	date of Sunday before end date
FirstPart	=	weekdays in first partial week
MiddlePart	=	weekdays in middle full weeks
LastPart	=	weekdays in last partial week
WeekDays	=	total number of weekdays

In this example the dBASE fields are named Start and End. To calculate the number of weekdays between the dates in these two fields, use the /Field Calculate Create command to define the following calculated fields.

```
StartDay = DOW(Start)
EndDay   = DOW(End)
NextFriday = Start+(6-MIN(6,StartDay))
LastSunday = End-(EndDay-1)
FirstPart = IIF(End>NextFriday,
              NextFriday-Start,End-Start)
MiddlePart = WKSBTWN(LastSunday,NextFriday)*5
LastPart = IIF(End-Start>=7.OR.EndDay<StartDay,
              MIN(5,IIF(Start<LastSunday,
                End-LastSunday,End-Start)),0)
```

```
WeekDays = IIF(End<Start,0,
          FirstPart+MiddlePart+LastPart)
```

You should note carefully how this weekday calculation works in relation to your application. Like the DAYSBTWN function, the number of weekdays between one weekday and the next weekday is one.

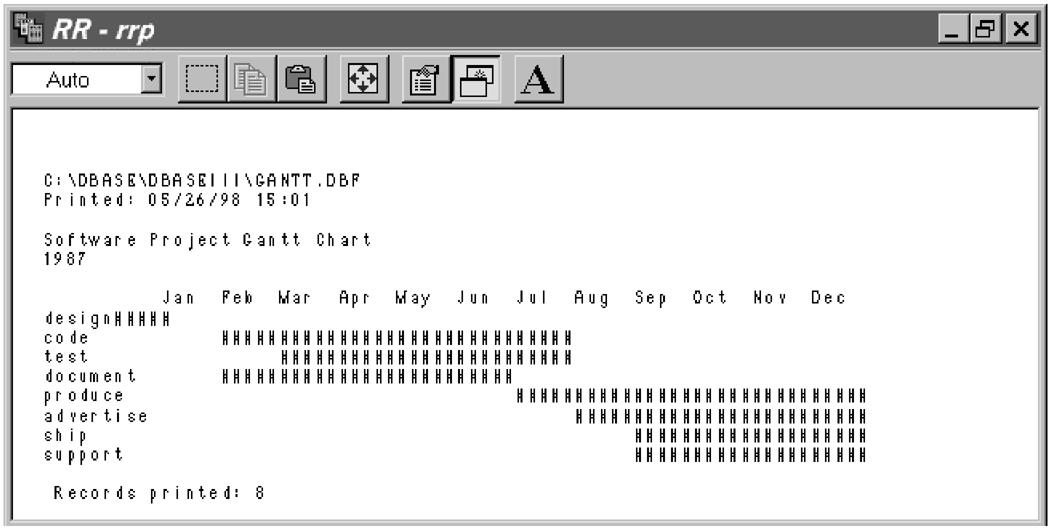
In some applications it is desirable to account for one weekday when the starting and ending day are the same. You can easily modify the FirstPart expression to account for this as follows. The additional sub-expression adds one to the count if the starting day is a weekday.

```
FirstPart = IIF(End>NextFriday,
                NextFriday-Start,End-Start)+
               IIF(StartDay>1.AND.StartDay<7,1,0)
```

Gantt Charts

A Gantt chart is useful for diagramming a series of activities in relation to a time line that extends across the page. An example based on a project schedule for the release of a software product appears below.

This chart was printed on a laser printer and makes use of the IBM extended character set. If your printer does not support this character set, substitute similar characters from those available on your keyboard. If your printer supports these characters you can use the [Shift-F3] and [F3] Draw keys to select and enter them.



```
C:\DBASE\DBASEIII\GANTT.DBF
Printed: 05/26/98 15:01

Software Project Gantt Chart
1987

          Jan  Feb  Mar  Apr  May  Jun  Jul  Aug  Sep  Oct  Nov  Dec
designHHHHH
code           HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
test                HHHHHHHHHHHHHHHHHHHHHHHHH
document       HHHHHHHHHHHHHHHHHHHHHHHHH
produce                                 HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
advertise                                    HHHHHHHHHHHHHHHHHHHHHHHHH
ship                                              HHHHHHHHHHHHHHHHHHHH
support                                           HHHHHHHHHHHHHHHHHHHH

 Records printed: 8
```

Figure 1. A Gantt chart for the release of a software product.

The data for this chart comes from the following example database. This is admittedly a simple example. A real project scheduling application would typically break down activities such as *Document* into sub-activities. It would also track the percent complete, which could be indicated in the chart with a different degree of shading. Finally, note that time units are in months rather than the more common week or day units.

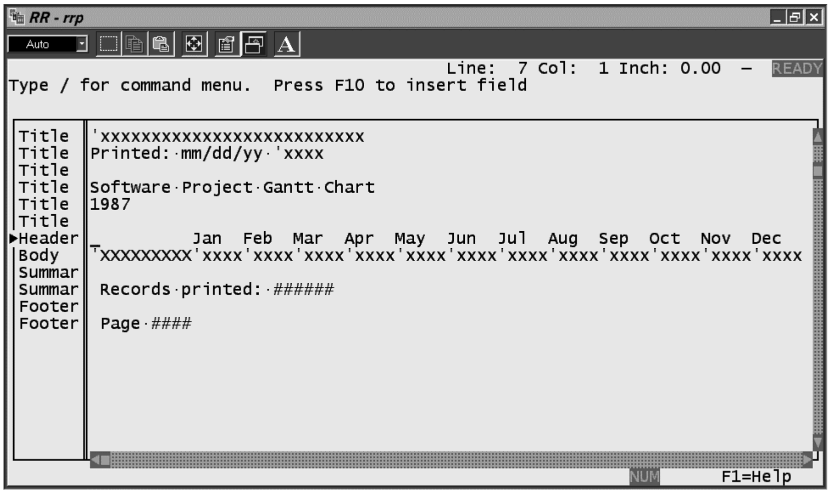
```
.display structure
    Structure for database: C:project.dbf
    Number of data records:       8
    Date of last update   : 07/01/87
    Field  Field Name  Type       Width    Dec
       1  ACTIVITY    Character     10
       2  STARTDATE   Date           8
       3  ENDDATE     Date           8
    ** Total **                     27

.list
    Record#      ACTIVITY   STARTDATE ENDDATE
         1        Design         01/01/87  01/31/87
         2        Code           02/01/87  07/31/87
         3        Test           03/01/87  07/31/87
         4        Document       02/01/87  06/30/87
         5        Produce        07/01/87  12/31/87
         6        Advertise      08/01/87  12/31/87
         7        Ship           09/01/87  12/31/87
         8        Support        09/01/87  12/31/87
```

The bar for each activity is composed of twelve calculated fields, one for each month. The INRANGE function provides a convenient method for testing whether the last day of each time period, in this case each month, is within the scheduled start and end dates.

The twelve calculated fields are created using the /Field Calculate Create command. For readability, the bar is five characters wide for each month. Enter the expressions as follows.

```
JanBar = IIF(INRANGE(CTOD("01/31/87"),
         STARTDATE,ENDDATE),"°°°°°","")
FebBar = IIF(INRANGE(CTOD("02/28/87"),
         STARTDATE,ENDDATE),"°°°°°","")
          .                  .
          .                  .
          .                  .
DecBar = IIF(INRANGE(CTOD("12/31/87"),
         STARTDATE,ENDDATE),"°°°°°","")
```

These fields are arranged in the following layout in combination with text fields. Use the /Print Options Record-Layout command to set the option Trim adjacent fields to No. Otherwise all activities appear to start in January.



```
                                             Line:   7 Col:   1 Inch: 0.00  –
Type / for command menu.  Press F10 to insert field

 Title   'xxxxxxxxxxxxxxxxxxxxxxxxxx
 Title   Printed:·mm/dd/yy·'xxxx
 Title
 Title   Software·Project·Gantt·Chart
 Title   1987
 Title
▶Header  _         Jan   Feb   Mar   Apr   May   Jun   Jul   Aug   Sep   Oct   Nov   Dec
 Body    'XXXXXXXXX'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx'xxxx
 Summar
 Summar   Records·printed:·######
 Footer
 Footer   Page·####
                                                                          F1=Help
```

Figure 2. Layout for the Gantt chart.

Reverse Indenting Memos

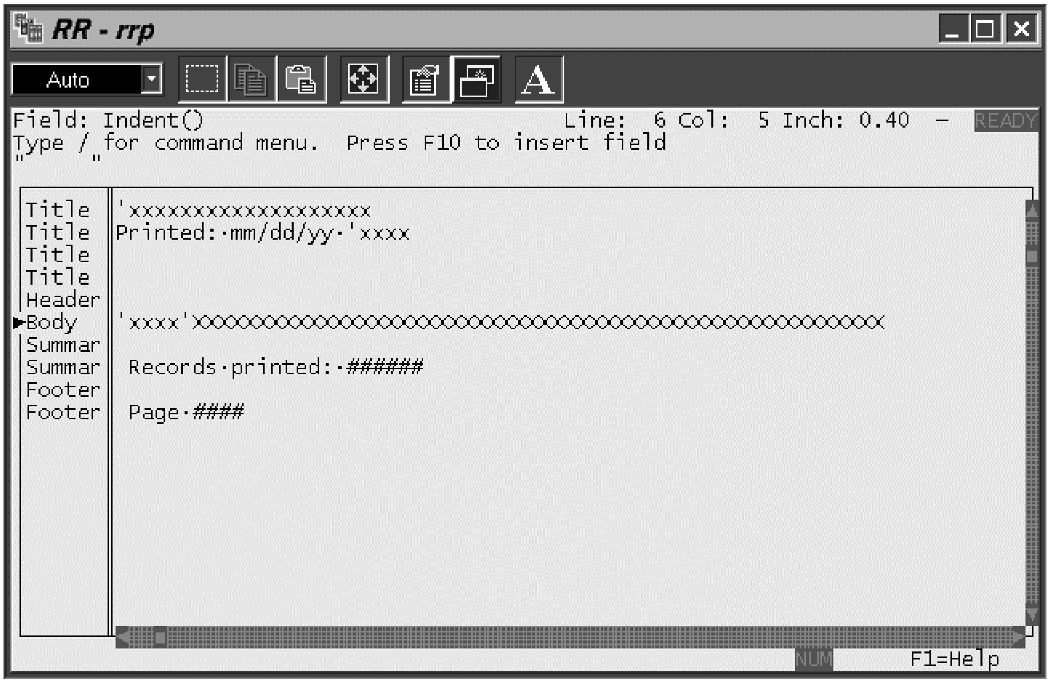
The terms *reverse indent* and *outdent* are borrowed from word-processing programs. They describe a format where all lines in a paragraph except the first line are indented as in the following example.

> You can apply the reverse indent format to any character or memo field that has the word-wrap format. This technique is useful for highlighting the beginning of each paragraph or section of a textual report.

The reverse indent technique uses the automatic trim feature of R&R Report Writer. Use the /Print Options Record-Layout command to turn on this feature by setting the option Trim adjacent fields to Yes.

Next, use the /Field Calculate Create command to create a field named Indent. When prompted to enter the expression, enter two single or double quotes to define an empty character string.

Insert the Indent field on your report in the column where you want the first line of text to begin. Use the /Field Width command to set the size of indent you want. For example, if you want a reverse indent of five characters, set the field width to five.



```
Field: Indent()                          Line:  6 Col:  5 Inch: 0.40  -  READY
Type / for command menu.  Press F10 to insert field
" "

 Title   'xxxxxxxxxxxxxxxxxxxx
 Title   Printed:·mm/dd/yy·'xxxx
 Title
 Title
 Header
►Body    'xxxx'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
 Summar
 Summar   Records·printed:·######
 Footer
 Footer   Page·####

                                                     NUM      F1=Help
```

Figure 3. Using the reverse indent field.

Insert the text field in the column immediately to the right of the Indent field. Be sure to specify the word-wrap format with the /Field Format Word-Wrap command. Automatic trim will remove the five leading spaces from the first line of text.

Standard Deviation, Variance, and Standard Error

Statistical applications often require the calculation of what are called standard deviation of a population, variance of a population, and standard error of the mean.

These statistics can be calculated for a dBASE numeric field using a combination of calculated and total fields. In a shorthand notation, the formulas are as follows.

Sum	= Sum of NUMBER
Population	= Count of NUMBER
Mean	= Average of NUMBER
Square	= NUMBER^2
MeanSquared	= Average of Square
Variance	= MeanSquared - Mean^2
Std Dev	= _Variance
Std Err	= Std Dev / _Population

You already know how to create a Sum, Count, and an Average (mean) using the /Field Total Create command. After doing so, use the /Field Calculate Create command to enter the expressions as follows.

```
Square   = NUMBER^2
Variance = MeanSquared - Mean^2
```

```
StdDev   = SQRT(Variance)
StdErr   = StdDev/(SQRT(Population))
```

All trademarks are the property of their respective owners. The information contained in this technical bulletin is subject to change without notice. Liveware Publishing Inc. provides this information "as is" without warranty of any kind, either expressed or implied, but not limited to the implied warranty of merchantability and fitness for a particular purpose. Liveware Publishing may improve or change the product at any time without further notice; this document does not represent a commitment on the part of Liveware Publishing. The software described in this document is furnished under a license agreement or nondisclosure agreement. The software may be used or copied only in accordance with the terms of the licensing agreement.