

Bug in Subroutine Call
(PC Magazine Vol 4 No 13 June 25, 1985 Spreadsheet Clinic)

There's a little known bug in the call-and-return feature of 123's programming language. When a keyboard macro contains a subroutine call (/XClocation), 123 jumps to a new location and begins taking keystrokes from that location until it reaches a subroutine return (/XR). 123 then returns to the subroutine call and execution continues from the point immediately after the call.

Unfortunately, the location of the call is saved as an absolute storage address. If you call a subroutine that inserts (or deletes) a row (/Worksheet Insert Row or /Worksheet Delete Row) above the call, 123 fails to adjust the storage address to reflect the addition or subtraction of rows. Thus, when 123 returns from the subroutine it returns to the wrong cell and as the macro continues to execute, it will play back the wrong keystroke.

The solution? Put all macros using subroutine calls above any data areas where you will be inserting and deleting lines. This means all your databases must be near the bottom of the worksheet, not near the top. This is an ugly fix, but at least you can use the Auto-Execute macro (\O) to put the cursor at the top of the data area when you load the worksheet.

A Bug in the Subroutine Call
(PC Magazine Vol 4 No 21 Oct 15, 1985 Spreadsheet Clinic)

As pointed out in Vol 4 No 13 (above), if a macro adds or deletes lines within a worksheet, /xg or /xc commands in macros located below the additions or deletions will no longer work right if they refer to specific cell addresses. The 123 macro processor doesn't adjust cell addresses within macros when lines or columns are added or subtracted.

One way around the problem is to avoid using subroutine calls and to use named, one-cell ranges instead of specific cell addresses. Unlike cell addresses, range addresses are automatically adjusted when you change the number of columns or rows. Thus you can get the effect of a subroutine call by using a simple /xg (GOTO) command that switches control to a "subroutine" macro that begins with a named cell. Just tell the macro to branch to that named cell. If you have another named cell immediately after the original GOTO command, you can put another GOTO at the end of the subroutine to return control to the main macro at the proper location. The point is always to use named ranges instead of cell addresses.

You may, of course, end up with a lot of named ranges in your worksheets, but this is the best way to keep your macros immune to changes in the number of columns and rows. This approach is especially helpful during the development phase of a model, when you probably will be adding and deleting many rows and columns. So long as you use named ranges within macros, you can add or subtract any number of lines without affecting your macro routines.

Fixing Circular Errors in 1-2-3
(PC Magazine Vol 4 No 14 July 9, 1985 Spreadsheet Clinic)

Circular errors are one of the greatest frustrations for the 123 user. You know you've got a circular error when you see the letters CIRC highlighted at the bottom of the screen. It means that a cell formula uses its own value in a calculation or that several formulas

depend on each other to arrive at a value.

The following procedure can help to find a circular error. First save the worksheet twice; one copy for storage and one for trouble-shooting. Bring up the trouble-shooting copy and put 123 into auto-recalc mode. This way you can see the results of your hunt immediately. Now go through the spreadsheet and delete formulas. The @SUM formulas are good ones to start with because they are the usual culprits. Often the only way to make rapid progress in your search is to delete whole chunks of your worksheet -- one at a time. Check the bottom of the screen after every deletion. When the CIRC message goes away, you know you've found the part of the spreadsheet that's causing the problem. Now you can call up a fresh copy of the worksheet and narrow your search until you find the bad cell.

Editor's Note: This is a clumsy, trial-and-error way to hunt for circular errors, but it's the only one that works. When you delete whole chunks of your worksheet, you may produce ERR messages in the cells that remain, but that makes no difference; the problem that caused the circular error will still be there. Since 123 is smart enough to know a circular error when it sees one, you'd think it'd be smart enough to highlight it for you, too. Maybe next version.

Tracking Down Circular Errors

(PC Magazine Vol 4 No 21 Oct 15, 1985 Spreadsheet Clinic)

...a different way to track down circular errors in 123 from the one described in Vol 4 No 14 (above). As soon as you see the CIRC error message, save your worksheet. Next, hit F9 to recalculate. Then put the cursor in cell A1 and use the /File Combine Subtract command to read in the version you just saved. Most of the cells in your sheet should now have the value zero. Only those that changed value when you last recalculated the worksheet will have any other value. These are your circular references, along with any cells whose values depend on them. If many formulas in your worksheet depend on the value produced by the circular error, you'll have many nonzero cells after the /File Combine Subtract operation, and you should have little trouble tracking down the culprit.

Viewing Formulas Instead of Values

(PC Magazine Vol 4 No 15 July 23, 1985 Spreadsheet Clinic)

When trouble-shooting a 123 worksheet with many formulas, it's sometimes more useful to see the formulas themselves rather than the values they produce. The Lotus manual recommends printing the worksheet using the /Print Printer Options Other CellFormulas command, but that gives you the contents of only one cell per line of printout. It's hard to tell how the cells relate to each other because they all appear stacked in one long column.

Use the /Range Format Text command, and use it on only those formulas that you want displayed. The rest of the worksheet will remain intact and you can see how the formulas fit into it. However, you may have to increase column width in order to display long formulas. This procedure is in the Lotus manual, but it provides no explanation of how useful it can be.

Debugging a Spreadsheet

(PC Magazine Vol 4 No 22 October 29, 1985 Spreadsheet Clinic)

Vol 4 No 15 (July 23, 1985 above) suggests that when you are debugging a spreadsheet, it is sometimes more useful to display formulas on the screen than to display values. That's a good idea, but you can take it further with 123. Break the screen up into two windows. In one, display formulas, and in the other, display values. Then, as you adjust the formulas in one window, you will immediately be able to see their effect on the values in the other.

#####