Auto Resize Margins to fit Terminal Size

Prepared by Sam Habiel Aug/18/2016

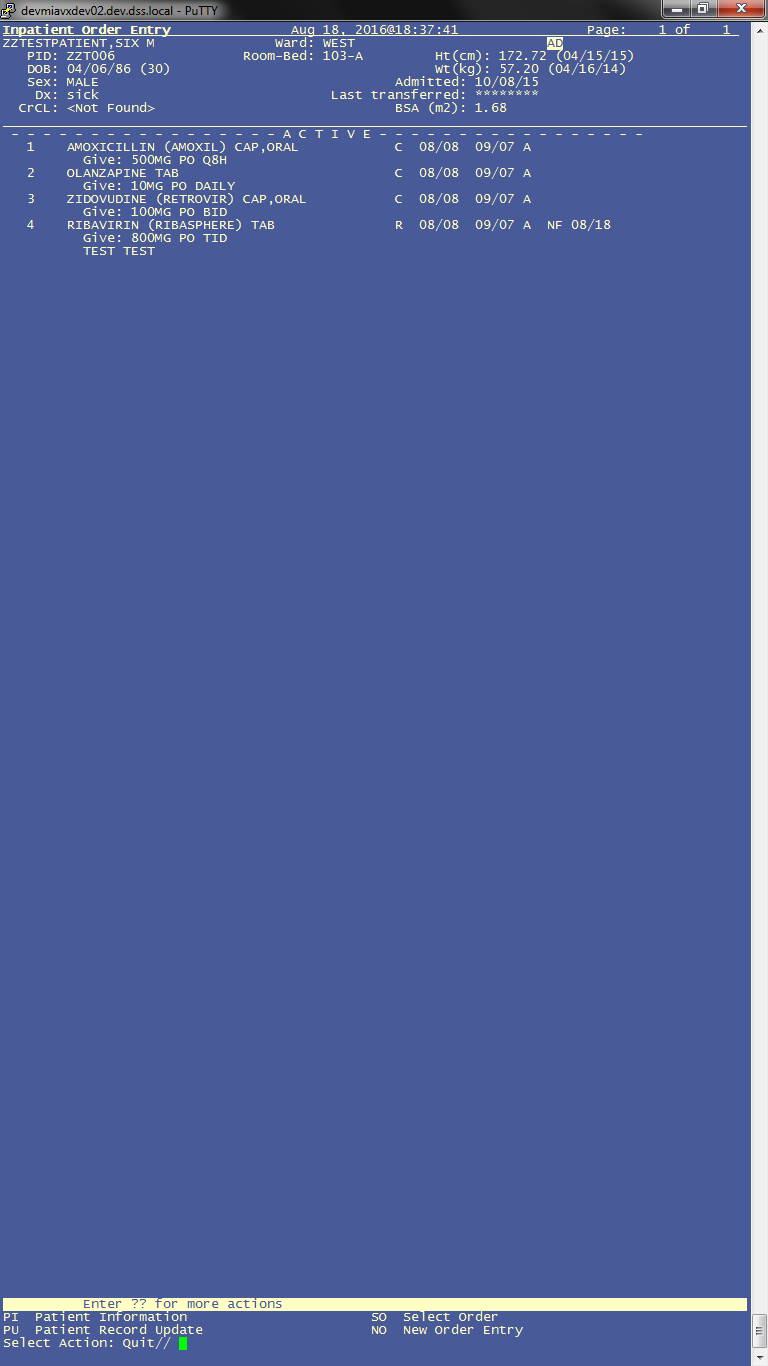


Figure - Taken from a Vertical Screen

# Introduction

Many modern applications that use terminals (e.g. Midnight Commander, vim, emacs) are able to resize themselves to take all the available real-estate on the screen. This has not been done in VISTA world by and large, except by creating special terminal types. The changes in build VFD\*15.0\*123 make a few adjustments in %ZIS3, %ZIS4, and VALM0 and VALM1 in order to be able to use all the available screen real estate.

# Pros and Cons

The most important benefit is for users of Listman applications, which don't display enough information for the user to act on. This is especially prominent in Pharmacy. Pharmacists viewing a patient's profile in Inpatient Pharmacy can only view 4 medications at a time; and in processing an individual medication, you don't see the provider comments and CPRS order checks except by scrolling down, which is a significant risk to patient safety. Compare Figure 2 with Figure 1; and Figure 3 with Figure 4.

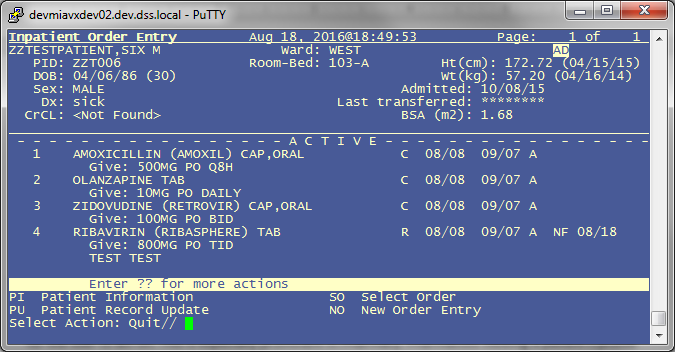
Any applications taking advantage of the any Fileman utility will see a difference: Inquire into File Entries will show more information; Screenman forms can be taller; and the Fileman Browser will take up the whole screen.

The disadvantages of this are two (which I can think of):

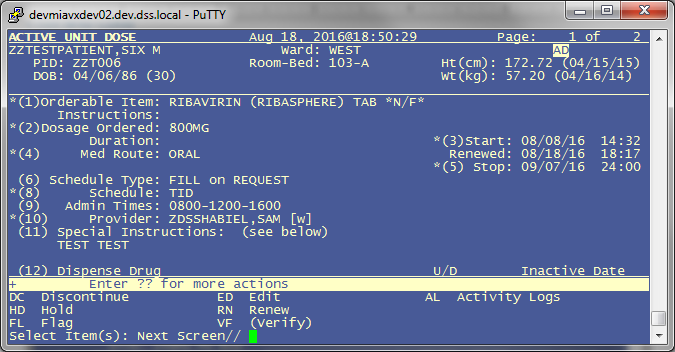
* There are possibly some menu options that may rely on a hardcoded value of 80 x 24.
* Users have to know that they can resize the terminal.

To mitigate the disadvantages, the system will run normally (without any modifications) if your terminal size is 80 x 24.

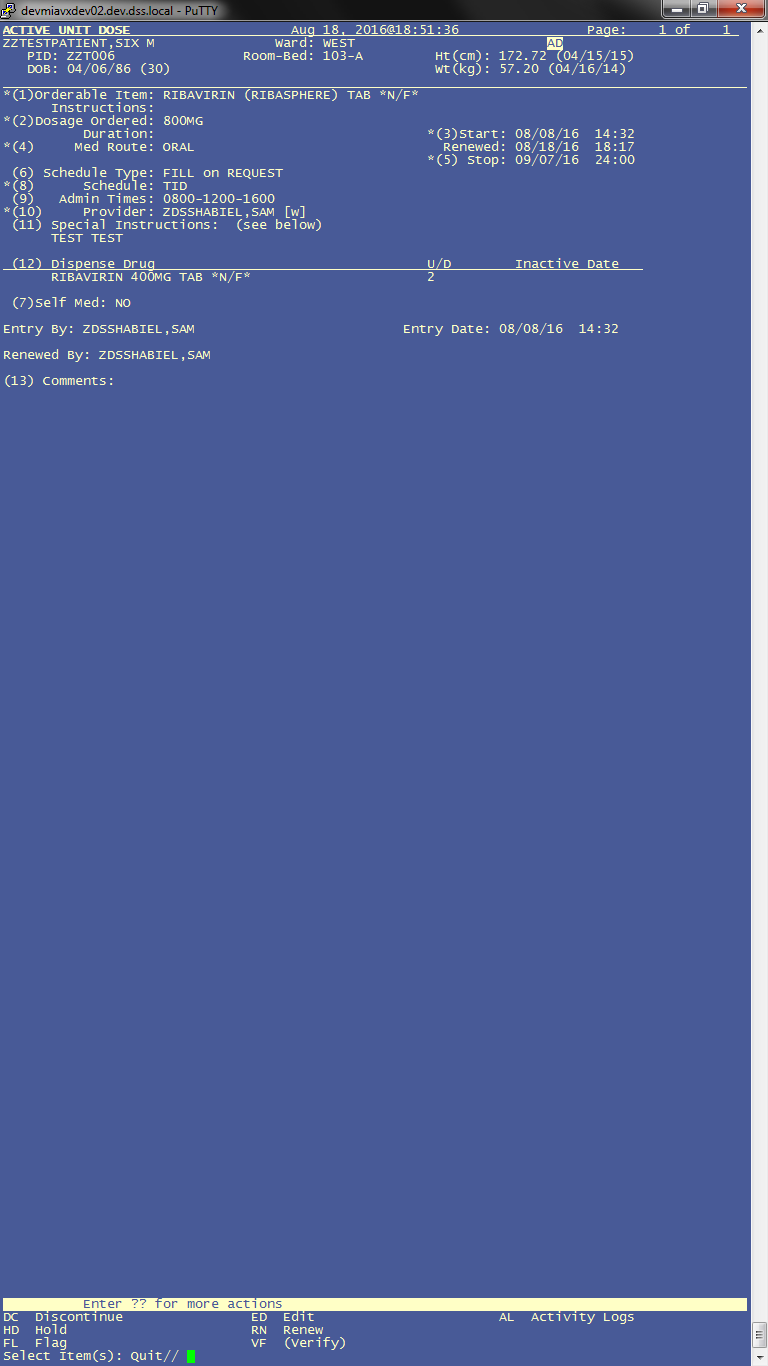
I have encountered one menu option that uses a call to reset the terminal, which changes it to 80 x 24. This is the Reminder Dialogs editor. I have a strategy for dealing with this, which will be outlined below.



Figure



Figure



Figure

# Technical Approach

The main routines responsible for this change are in %ZIS3 and %ZIS4. ZIS3, ZIS4ONT, and ZIS4GTM have been modified so that when a terminal is chosen, it is automatically queried for its size. If the query returns a value, then this value is used; otherwise, the default IOSL and IOM are used from the Terminal Type. I know of the wish to parameterize this in %ZIS3, but %ZIS is the lowest part of the VISTA infrastructure and cannot rely on anything higher.

%ZIS3 modifications:

MARGN ;Get the margin and page length

S %A=$P(%Y,";",1)

I %A?1A.ANP D SUBIEN(.%A,1) I $D(^%ZIS(2,%A,1)) K %Z91 D ST(1) S %Y=$P(%Y,";",2,9),%ZISMY=$P(%ZISMY,";",2,9) G MARGN

I %A>3 S $P(%Z91,"^")=$S(%A>255:255,1:+%A)

I $P(%Y,";",2) S $P(%Z91,"^",3)=+$S($P(%Y,";",2)>65534:65534,1:$P(%Y,";",2)) ;Cache fix for $Y#65535 wrap

; DSS/SMH BEGIN MODS - Attempt to get the best margins for the terminal

I $D(^%ZOSF("ZVX")),$T(AUTOMARG^%ZIS4)]"",%ZTYPE["TRM",%ZIS'[0,$E(%ZISIOST,1,2)="C-" D

. N RMPL,RM,PL S RMPL=$$AUTOMARG^%ZIS4() ; Query terminal for best margins.

. I 'RMPL QUIT ; NB: Not always possible to query the terminal, esp if we do it multiple times in succession

. S RM=$P(RMPL,"^",1) ; Right Margin (normally IOM)

. S PL=$P(RMPL,"^",2) ; Page Length (normally IOSL)

. S %A=RM,$P(%Z91,"^")=$S(%A>255:255,1:+%A),$P(%Y,";")=RM ; copy of line above

. S $P(%Y,";",2)=PL,$P(%Z91,"^",3)=+$S($P(%Y,";",2)>65534:65534,1:$P(%Y,";",2)) ; ditto

. ; %ZIS="T" taskman (I think?), %ZISB means no "N" in %ZIS. IOP and IO("Q") are well known.

. I %ZIS'["T",%ZISB=1,'$D(IOP),'$D(IO("Q")) D

.. W " Terminal Size: "\_$P(%Z91,"^")\_" x "\_$P(%Z91,"^",3)

; DSS/SMH END MODS

%ZIS4 modifications (Cache):

; DSS/SMH - BEGIN MODS - This is new code to be called from %ZIS3

AUTOMARG() ;RETURNS IOM^IOSL IF IT CAN and resets terminal to those dimensions; Cache Version

; ZEXCEPT: APC,TERM,WIDTH - these are not really variables

N X S X=0 X ^%ZOSF("RM")

N %I,%T,ESC,DIM S %I=$I,%T=$T D

. ; resize terminal to match actual dimensions

. S ESC=$C(27)

. W ESC,"7",ESC,"[r",ESC,"[999;999H",ESC,"[6n"

. U $P:(:"+S+I":"R") R DIM:1 E Q

. W ESC,"8"

. I DIM?.APC U $P:("") Q

. S DIM=+$P(DIM,";",2)\_"^"\_+$P(DIM,"[",2)

. U $P:(+DIM:"")

; restore state

U %I I %T

Q:$Q $S(DIM:DIM,1:"") Q

; DSS/SMH - END MODS

In addition, VALM0 was modified so that VALM("BM") (bottom margin) and VALM("RM") (Right Margin) are set to values that are appropriate:

S VALM("BM")=$P(VALM0,U,6)

; DSS/SMH BEGIN MODS - Use IOSL by default for bottom margin (BM was previous written to an offset of 24)

I $D(^%ZOSF("ZVX")) D

. I $G(XQY),$$GET^XPAR("ALL","VFD VALM BM BY OPTION",XQY,"I") QUIT

. S VALM("BM")=$S($G(IOSL):IOSL-(24-$P(VALM0,U,6)),1:$P(VALM0,U,6))

; DSS/SMH END MODS

S VALM("FIXED")=$S($G(^SD(409.61,VALM("IFN"),"COL",+$O(^SD(409.61,VALM("IFN"),"COL","AIDENT",1,0)),0))]"":$P(^(0),U,2)+$P(^(0),U,3),1:0)

S VALM("RM")=$S($P(VALM0,U,4):$P(VALM0,U,4),1:80)

; DSS/SMH BEGIN MODS - Use IOM by default for right margin

I $D(^%ZOSF("ZVX")) S VALM("RM")=IOM

; DSS/SMH END MODS

Notice there this is a parameter which suppresses resizing the Bottom Margin. This exists so that the Reminders Dialog menu option [PXRM DIALOG/COMPONENT EDIT] continues to function normally. The build exports this at the package level. If there are any more options that are discovered, they can be placed here. I have tested the Problem List editor, Outpatient Pharmacy, Inpatient Pharmacy, Outpatient Pharmacy Billing, Patient Record Flags, Appointment Scheduling, and Reminders. I have only found a problem in the latter due to its use of the "soft reset" control sequence which resets the terminal to 80 x 24. (NB: The code for reminders in this patch has been updated so now it can use the whole screen as well.)

VALM1 was modified to round down DX and DY used to call positioning code in IOXY, as they contained decimal points; and an IOXY call does not work if the values to call it have decimals. They have to be integers.

PSJ200 and PSOORUT3 were both edited to disable code that resizes pharmacy screens down to 21 or 19 lines.

Post install routine VFD15123P automatically renames ZIS3 and ZIS4ONT to %ZIS3 and %ZIS4.

# Fallback after installation

As noted above, the effects of this code will be as if it wasn't installed if the terminal size is 80 x 24. In an emergency, the original copies of %ZIS3 and VALM0 can be restored.

# Recommended Testing

Because this changes the device driver, it's recommended to test everything the uses it to make sure it still functions properly. Here are recommended tests:

* CPRS Connections (Null device, Resource devices)
* Background prints from CPRS (Chart Copy/Work Copy)
* Queued prints from any roll and scroll option.
* Regular prints from any roll and scroll option.

Testing is important: I broke CPRS and Eclipse Editor when doing this, because Null device selection produced spurious control characters on the TCP connection. I changed my code to make sure 0 isn't in %ZIS and the subtype starts with "C-".

I have tested all of these; but there needs to be more testing.