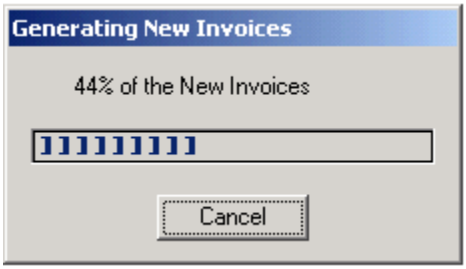
Changing The Msg() Text During Process Execution

A useful enhancement to the standard gas gauge progress bar is to change the message, showing, for example, percent of records processed. Raising the message with the changed text and then lowering it does not offer satisfactory results because of the annoying screen flicker associated with this method.

A more elegant approach to use is to manipulate the TEXT property of the MSG window **ST_TEXT** control. The Msg() function is actually updating a window called **MSG**. The MSG window has its own controls, including the **ST_TEXT** control. By loading the percentage variable along with its accompanying text into a second variable and then setting the control's TEXT property to this second variable, we can update the gas gauge message without the screen flashing every time we raise and lower the message.

The properties of the available **MSG** window controls can be viewed by selecting **Variables-Inspect Properties** menu item from the OpenInsight Debugger. Select **MSG** from the **Window Name** combo box. The list of available controls with be displayed in the **Control Name** combo box. The **ST_TEXT** control is the one that contains the text property displayed on the message. A list of its properties is available from the **Property** combo box. This control and this property continue to be available even as multiple instances of Msg() are being created and displayed.

Periodically, we would like to change the message text to indicate the percentage of completion of the process, producing a message similar to the one shown below:



The code below assumes that the **INVOICE** table is read completely. The number of rows in the table, used as the denominator for the percentage of rows read, is retrieved using the Get.RecCount() function. To prevent the invoice processing from slowing down unnecessarily, we want the percentage calculation and message update to occur only after 20 invoices have been processed. The code below illustrates how this can be accomplished:

```
declare function msg , get.reccount
declare subroutine fsmsg
$Insert msg_equates
$insert logical
open 'ORDERS' to orders_file else
   fsmsg()
   return 1
end
select orders_file
totrecs = get.reccount(orders_file, flag, 0)
if flag else
   /* unable to get record count - process error */
end
Def = ""
Def<MCAPTION$> = "Generating New Invoices"
Def<MTYPE$ > = "GC"
Def<MEXTENT$ > = totrecs
YOURTEXT = '% of the New Invoices Completed'
Def<MTEXT$> = YOURTEXT
Def<MTEXTWIDTH$> = 200
MsgUp = Msg(@window, Def)

cnt = 0
EOF = FALSE$
Loop
Readnext @ID Else EOF = TRUE$
Until EOF
   Read @Record from orders_file, @ID Else
     retval = Msg('Unable to read %1%','','',@ID)
     Goto Nextcrs
   End
   cnt += 1 ;*increment counter for gas gauge
   recs = int(totrecs/20)
   If Mod(cnt,recs) = 0 Then
     * so for every 20 records
     pct= INT((cnt/totrecs) * 100)
     If pct > 100 then pct = 100
     YOURVAR = pct : YOURTEXT
   End
   Gosub Process_Invoice
   while Msg(@window, MsgUp, CNT, MSGINSTUPDATE$, '')
     /* update the message with the new percentage completion here */
     X = Set_Property('MSG.ST_TEXT','TEXT',YOURVAR)
Nextcrs: Repeat
retval = Msg(@window, MsgUp) ;* take down the gauge
Return 0

process_invoice:
*
*  do the invoice processing here...
*
return
```