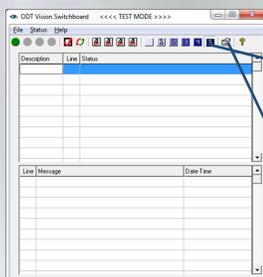


Using the System Debug Log File Feature

It has long been known that “Necessity is the Mother of Invention”. Our customers love the fact that they can create customization scripts that build their own IVR applications around their existing systems and data. Each IVR application can then be tailored to what the firm needs from their self-service applications. Sometimes a client can have issues while doing this in-house customization. Years ago we had an out of state client that was just having issues getting their customization script clean. I went to development and asked if it was possible to create a feature within our software product to track the results as each line of code within the customization script as a call was processed. What they developed was the “System Debug Log Feature” which builds a log file showing what happens as each line of code is processed.

TURNING ON SYSTEM DEBUG LOG

Open the switchboard and turn on level 5 of the system log level.



First Click on Level 5 for system debug feature

Second Click on Line Properties icon to get to log file screen

Using Message Boxes	2
Manual Walkthrough of the Script	3
CONCLUSION	3

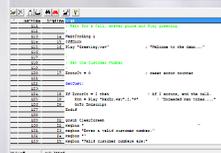
Next set up the log file type. When you click on the Line Properties Icon, the first tab is logging.



- ◆ Set number of log entries
- ◆ Click the box to Log to a file
- ◆ Determine if you want one large log file or a smaller daily date stamped file
- ◆ Click Ok
- ◆ Now start line

I like using the “Date Stamped Log File” so you can easily break your analysis down to activity for a particular day. You may also want to set the location for that file.

HOW TO RUN ANALYSIS ON THE SYSTEM DEBUG LOG FILE



Each line of code within your customization script has a line number but when you look at the raw file, that isn't apparent. To obtain the line number, take the script assigned to the line with the debug level 5 turned on and compile it. The second column in the compiler is the line number that matches the



IBM Solution Connection
Integrates solution information
with IBM eServer, software and
TotalStorage technology



log. Now open the System Debug log file and read through it to see how the script is performing for each line as the call progresses through the routine. If the script fails or a line doesn't execute as expected, look at that log entry to see what it reports. If you are testing only one line and only one line is active, this is easy to do. If you have multiple lines active, the log will contain entries for all lines so as you go through your analysis, you will have to isolate the entries by system line number.

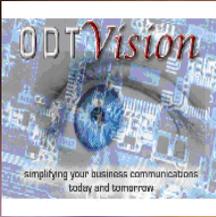
```
CALLLOG - Notepad
File Edit Format View Help
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 3 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 3 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #322 - Goto line 319
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 4 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 5 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #322 - Goto line 319
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 5 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 6 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #322 - Goto line 319
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 6 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 7 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #322 - Goto line 319
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 7 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 8 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #322 - Goto line 319
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 8 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 9 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #322 - Goto line 319
4/7/2015 10:38:02 AM Line #1 Script Line #319 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #320 z = 9 + 1
4/7/2015 10:38:02 AM Line #1 Script Line #321 If 10 < 10
4/7/2015 10:38:02 AM Line #1 Script Line #323 - Return, Resume
4/7/2015 10:38:02 AM Line #1 Script Line #94 - Message Box: -----
4/7/2015 10:38:02 AM Line #1 Script Line #95 - Message Box: Enter an inventory item to order.
4/7/2015 10:38:02 AM Line #1 Script Line #96 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #97 - Message Box: valid item numbers are:
4/7/2015 10:38:02 AM Line #1 Script Line #98 - Message box: 111, 222, 333, 444.....
4/7/2015 10:38:02 AM Line #1 Script Line #99 - gosub line 324
4/7/2015 10:38:02 AM Line #1 Script Line #324 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #325 - Message Box:
4/7/2015 10:38:02 AM Line #1 Script Line #326 - Return, Resume
4/7/2015 10:38:02 AM Line #1 Script Line #100 - Clear Digits
4/7/2015 10:38:02 AM Line #1 Script Line #101 - add file to playlist: C:\Program Files (x86)\ODT vision\Voice Files\Item.wav
4/7/2015 10:38:02 AM Line #1 Script Line #101 - Playing C:\Program Files (x86)\ODT vision\Voice Files\Item.wav
4/7/2015 10:38:02 AM Line #1 Script Line #103 - Set variable ItemInput =
4/7/2015 10:38:02 AM Line #1 Script Line #104 w = len()
4/7/2015 10:38:02 AM Line #1 Script Line #105 If 0 = 0
```

The debug log file not only gives you the line number of the script but the time this line was executed so that if you have a total failure and you know the time that occurred, you can look in the log file for that time and see what was happening.

Another benefit of this feature is you can send us the log file and script assigned when that log file was created so we can review what happen remotely.

USING MESSAGE BOXES

The System Debug Log File is a wonderful tool to debug and clean up a script, but a quicker instant method to see what is happening is to use message boxes.



Vision Voice Vantage, Inc.
1938 Zollinger Road
Columbus, Ohio 43221

Phone: 614-586-9320
Email sales@ODTVision.com



THINKING OUTSIDE THE BOX IN CUSTOMER SERVICE

Contact us to get your own demonstration of the ODTVision Voice Response Unit. This demo application is a simplistic order entry and shipment status system which is running off a Microsoft Access database. The demo is in the test mode and you will be using the "Test Phone" feature of the ODT VISION VRU to simulate a phone call to the data. Manuals and case studies are also available on the web site.

MANUAL WALK THROUGH OF THE SCRIPT

If you are getting unexpected results testing your script, use the real-time "MsgBox" syntax to show variable values during the progression of the call. What the message box syntax does is create a instant popup window over the switchboard. You can then run your analysis on what is displayed in the popup window.

There are two different reasons to use the powerful "MsgBox" syntax. The first is just to present a banner in the popup window over top of the switchboard that tells you that you have reached this part of the script. This is as simple as creating a banner that is placed within the quotes following the "MsgBox" syntax. I like to always initially put this banner within the different possibilities of an "If" statement.

```
If VarInput = 1 then
  MsgBox "User selected option 1"
Elseif VarInput = 2 then
  MsgBox "User selected option 2"
Endif
```

Not only does this tell me the results but also points out the area of the script you are in.

The second way to use MsgBox is to show variable values.

```
If VarInput = 1 then
  MsgBox "User selected option 1"
  MsgBox VarInput
Elseif VarInput = 2 then
  MsgBox "User selected option 2"
  MsgBox VarInput
Endif
```

In this method, you just put the variable behind the "MsgBox" syntax with no quotes so the value of that variable will be displayed in the popup window. MsgBox VariableName. If you combine it with the banner seen in the first example, you will see in the pop up screen the banner and the variable value. What is displayed:

Banner
Variable Value

CONCLUSION



In summary, the creation of your customization script is easy. It is more like writing a macro then actually programming. It only requires you to know your application and what is needed. The first goal is to get the customization script compiled successfully. If it fails to compile successfully, the compiler will identify the line of code by clicking on the  icon. The line that has an issue will give you a message as to what is needed to correct this issue. Once compiled successfully, if there are runtime issues with the application's script, you can use the methods outlined in this newsletter to clean that script up. You can also send the compiled script and system 5 debug log to have us assist you on the identification of the issues. ■