

Vision Voice Vantage always assists in the design and then creates the initial client's IVR application script based on the client's specifications. This process is a great learning tool where the client can then modify that application in the future on their own or they can create new applications as needed. This gives great power to the client as they have total control of their self-service customer service application in-house but can always come back to VVV for additional help and support.

Inside this issue:

**In Design of the Application,
*Thinking Outside the Box is a***

The very purpose of this newsletter publication is to provide existing customers and potential new customers ideas of how to use our solution to improve the way you address customer service applications. Every firm wants to be more responsive to the needs of their users whether they are clients, vendors, or their own employees.

As discussed on many occasions in this monthly publication, the ODTVision VRU is a combination of software and hardware that allows the creation of an **I**nteractive **V**oice **R**esponse Application.

The telephony IVR application allows the user to interact with the company's data via a telephone keypad or by speech recognition. This is an interactive exchange between the caller and the machine. Most IVR applications either use a pre-recorded voice banner or that voice banner is dynamically generated to further direct the user about how to proceed. Basically, the logic of the user's routine is a combination of what the caller submits with the values being returned from the data.

The beauty of this customizable interface is the fact that its creation doesn't require any previous programming experience since it is more like the creation of a macro than actual program code. You just use some logic to direct the flow of the routine such as "goto" and "gosub" with the combination of

When the variable value really isn't the value to use	2
Taking a portion of the variable to create a new variable value	3
If the Variable has a character not allowed	3
Working with Date and Times	4

"If" and "Elseif" to produce call flow. For example, in a main menu you may say, "Press 1 for your balance due, press 2 to check a shipment status of an order or Press 3 to make a payment. Besides looking at the value of the user's input, you

Comparison Operator	Test for:
<	Less than
>	Greater than
<=	Less than or Equal
>=	Greater than or Equal
=	Equal
<>	Not Equal

may look at the value of the data. You can then use a comparison operator to evaluate the value of any variable from the database and determine what should be done next. For example, if you have a value for a variable that represents the balance due on an account and that value is not zero, you want to tell the caller.



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



```
If AccBalance <> ".00" then
  Play "YouHaveBal.wav" ; you have a balance of.....
  Rtn = SpeakDollars AccBalance
Endif
```

Example 2. Shipment Status Application

If there is value in the DateShipped field, we go get values for method of shipment, tracking number, etc. and then speak them back, else....Play voice file that states the order has not shipped yet. In this example our method of connectivity to the data is HLAPI screen scraping. Let's say the user has:

- ⇒ Logged onto the system
- ⇒ Entered a proper account number
- ⇒ Entered a proper order number
- ⇒ And the system is setting at the order status screen

```
If DateShipped <> "" then ; check to see if field is empty
  MethodOfShipment = SessionGetScreen 6,3,3
  TrackingNo = SessionGetScreen 7, 3, 16
  Play "YourHasBeenShipped.wav" ; your order has been shipped via
  If MethodOfShipment = "101" the by USPS
    Play "101.wav" ; your order shipped by US Post office
  Elseif MethodOfShipment = "102" the by UPS
    Play "102.wav" ; your order shipped by UPS
  Elseif MethodOfShipment = "103" the by UPS
    Play "103.wav" ; your order shipped by FedEx
  Endif
```

```
TrackingNo = Lefttrim TrackingNo
```

```
If TrackingNo <> "" then
  Rtn = Play "The Tracking Number on this shipment is...."
  Rtn = Speak TrackingNo
Endif
```

```
Rtn = Play "WasShippedON.wav" ; this
order was shipped on...
Rtn = SpeakDate DateShipped,
"MMDDYY"
else
  Rtn = Play "NoShppedYet.wav" ; has not
shipped yet
Endif
```

This all seems very simple, doesn't it. All black and white where if the user enters this, the



In Design of the Application, *How to get around issues*

script goes here or says that. If the variable value that is returned is this, the script does this or says that. Now let's look at when the data value doesn't really mean what it says.

A number of years ago we had an account that automated a customer service function where a user could call up their profile and select a monthly statement and have it faxed back to them on demand. Keep in mind that you always want to make your IVR application as friendly as possible and meaningful to what callers know. It is easy to take the date a statement was issued and then strip out the month to say which statement it is. **And here lies the problem, the month wasn't always the actual statement month.** If the day of the month was less than "11" it was actual the previous month's statement. A statement date of the 10th of January was actually the December statement so code had to be inserted into the script to allow for that logic.

```
DayOfMonth = Format IssueDate,"DD"
Msgbox "Day of month"
Msgbox dayofmonth
DayOfMonth = Value DayOfMonth
If DayOfMonth < 11 then
  MsgBox "This is the previous month's statement"
  Statementdate = SubInterval issuedate,1,"M"
  Rtn = Play "StmtToFax.vox" ; "We will fax statement for..."
  Rtn = SpeakDate StatementDate,"MMMM-YYYY"
Else
  MsgBox "This is this month's statement"
  StatementDate = IssueDate
  Rtn = Play "StmtToFax.vox" ; "We will fax statement for..."
  Rtn = SpeakDate StatementDate,"MMMM-YYYY"
Endif
```

Only Need a Part of the Return Variable Value

Sometimes you need to extract a part of the returned variable value before you do any comparison or analysis on it. Let's say we have a



PAGE 3

Vision Voice Vantage, Inc.
829 Bethel Road #213
Columbus, Ohio 43214

Phone: 888-252-2555
Email sales@ODTVision.com



THINKING OUTSIDE THE BOX Get Your Own Demo Today

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.

Improving Customer Service Affordability

Get free project analysis regarding your telephony application or submit technical questions at:
TechSupport@ODTVision.com
Or Call: 888-252-2555

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

IBM Server Proven
Solutions with experience



<http://www-304.ibm.com/partnerworld/gsd/solutiondetails.do?&solution=25001&lc=en>
<http://search400.techtarget.com/review/Ohio-Data-Transfers-DTT1000-Voice-Response-Unit-60>
<http://www.ibmssystemsmag.com/ibmi/productnews/>

CallerID value that contains both the phone number and the name. We only need the phone number so there are actually two ways to achieve this.

Method 1: Just trim off the side of the variable.

If the phone number is on the left or right side, you could just pull off the ten digits that is the phone number and then create a new variable value based on what is returned. Of course, you need to know whether the phone number is on the left or right side but it would look like this...

CallerID = Right CallerID, 10 ; when Phone number is the 10 digits to the right

Or

CallerID = Left CallerID, 10 ; when Phone number is the 10 digits to the left

Method 2: Using the Split, Part Syntax

Let's say we are looking at a variable that contains three names for a person and we just need the first and last name

```
NameParts = Split Name           ; NameParts is 3
FirstName = Part 1                ; FirstName is "JOHN"
LastName = Part 3                 ; LastName is "SMITH"
```

What if the variable has a value not allowed

Both the "SpeakDollars" and "SpeakNumbers" syntax verbs require the data to be numeric in nature.

It doesn't allow commas which your variable string may have if it has been formatted in that way. What is required is to strip the comma out of the variable string so it is purely numeric. Here is a sub-routine you can use to do this. Sub-routines are used when that logic is used multiple times throughout the script. This example is for a currency variable:

First, set the value used in the sub-routine to be equal to the variable that will be used in the sub-routine.

```
AmtInput = LastPaymentAmount
gosub StripComma
```

Now in the subroutine, the commas are stripped out.

```
-----
; Stripping commas from amounts
;-----
StripComma:

c = ","
msgbox "Type of variable working with"
msgbox AmtInputtype
msgbox "Value of AmtInput"
msgbox amtinput
questionOnVariable = "/" & AmtInput
msgbox "Value of QuestionOnVariable"
msgbox QuestionOnVariable

amtInput = lefttrim AmtInput
```



PAGE 4

Vision Voice Vantage, Inc.
829 Bethel Road #213
Columbus, Ohio 43214

Phone: 888-252-2555
Email sales@ODTVision.com



THINKING OUTSIDE THE BOX
IN CUSTOMER SERVICE

Get Your Own Demo Today

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 ODTVISION VRU to simulate a phone call to the data. Manuals and case studies are also available on the web site.

Improving Customer Service Affordability

Get free project analysis regarding your telephony application or submit technical questions at:
TechSupport@ODTVision.com
Or Call: 888-252-2555



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



<http://www.ibmssystemsmag.com/ibmi/productnews/productreviews/Product-Review--DTT1000-from-Ohio-Data-Transfer/>

```
msgbox "Value of AmtInput"  
msgbox AmtInput
```

```
c = ","  
charpos = Search 1 AmtInput, c  
msgbox "position of comma"  
msgbox charpos
```

```
If Charpos = 0 then  
    msgbox "no comma found"  
    msgbox "Value of AmtInput with no comma & then return"  
    msgbox AmtInput  
    return  
Endif
```

```
varvalue = value charpos  
L = varvalue - 1  
E = varvalue + 1  
Startvar = left AmtInput, L  
msgbox "Value of Start String"  
msgbox startvar
```

```
Endvar = middle AmtInput, E  
msgbox "Value of end of string"  
msgbox Endvar
```

```
x = Startvar & Endvar  
msgbox "Value of AmtInput after strip comma"  
msgbox x  
amtinput = x  
return
```

Note that upon the return from the sub-routine, the variable you were working with needs to be set to the return value of the "AmtInput".

```
LastPaymentAmount = AmtInput
```

Working with Dates and Times If the Caller Needs to Create a Date or Time

Sometimes a caller needs to create a date/ time variable which we will compare against today's date/time or some date/time variable obtained from the database. This is simple by using the create Date/time syntax.

Syntax:

dvar = **CreateDate** month, day, year Returns the date value from month, day, and year.

tvar = **CreateTime** hours, minutes, seconds Returns the time value from hours, minutes, and seconds.

You can also modify an existing Date or Time

dtvar = **AddInterval** dtval, interval, type Returns the date/time value after adding an interval.

dtvar = **SubInterval** dtval, interval, type Returns the date/time value after subtracting an interval.

You can also extract values from Date or Time

numvar = **ExtractPart** dtval, type Returns the part of a date/time value specified by type.

numvar = **Interval** dtval1, dtval2, type Returns the interval between two date/time values.

Bottom line, you may need to modify the values of variables the user supplies or that you get from the database but the ODTVision VRU has the syntax to allow you to do this easily. If you know what you need to do, the script can be written to do it. ■