Vision Voice Vantage, Inc.

THINKING OUTSIDE THE BOX
IN CUSTOMER SERVICE

Volume 7 Issue 90
2/29/2012

Inside this issue:

Date and Time Can Do Alot

This month's newsletter covers the functions of Date and Time and why they may be important to the needs of your IVR application. The determination if the office is open or closed is based on the date, day of the week or time of day and is obviously needed when establishing the logic as to how a call is handled outside business hours. This can be everything from whether the IVR application allows a call transfer to a live customer service representative to how after hour emergencies are handled. The ODTVision VRU has build in setup screens for holidays and hours of operation so the script can use this VRU feature to determine if the office is open or closed. These setup screens were covered in the newsletter from June 2009 :

http://odtvision.com/includes/newsletters/0906_newsltr.pdf

Please go back and review that news-letter if you want to learn more about how the system automatically knows if the office is open or closed.

It is always important to establish the date and time of the call as they can be used by the customization scripts that run the ODTVision IVR applications.

```
dtvar = StartDateTime
dtvar = Format dtvar, "MM DD YYYY hh nn"
```

This creates a variable that can be used in so many ways and reflects the date and time that the line answered the call. Most IVR application users will want to build a log file of all activities and then use this log file to build a report of what happened during the call. Obviously, the date and time of the call will be used in any analysis of the use of the VRU system.

Retrieving Information from the DTVar

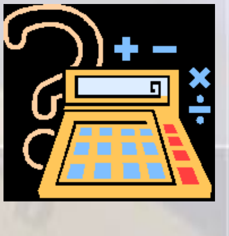
The ODTVision has syntax that allows the harvesting of information out of a date and time variable. There are a host of secondary variables that can be pulled out of the DTVar variable such as:

- ⇒ Time (hour,min.)
- ⇒ Year
- ⇒ Day of the Month
- ⇒ Day of the Week

Now let's look at how this gleaned information can be used. Say we have a payment transaction system where the user is supplying a bank card for payment on an account balance. When prompted to enter the month and year of expiration, the application can determine if the year value combined with the month value has passed. There is no reason to pass the old card information to your payment system as it would be rejected as the card has expired.

```
Today = StartDateTime
TodayYYMM = Format Today,"YYMM"
ExpDateInput = Format CExpDate,"YYMM"
If ExpDateInput < TodayYYMM then        ; Ex-
pire date entered is earlier than today, it's
invalid
        Rtn = Play "InvalidCCExp.wav"
        ; "You entered an outdated credit card
expire date...
        ErrorCt = ErrorCt + 1
        Goto EnterCCMonth
Endif
```

At this point, the system would play a voice file that explains that the card is outdated, returns them to enter a new card with acceptable date for expiration and the routine automatically continues.



Using the Harvested Variable Values to Calculate A New Variable Value

Interval

Sometimes it is important to use the current date or time variable against some other variable value to form a new variable. For example, perhaps we know the date of an invoice from the database or screen value. The application calls for the IVR system to calculate the age of that invoice. Once again, there is built in customization syntax to perform that type of function by using the "interval" syntax. Please note that your customization script will need to decide the type of interval you want to use in this calculation. In this example, the return value is in the number of days.

```
InvoiceAge = interval invoicedate, DTVar, "d"
```

This same syntax may be used to calculate the number of minutes between the beginning and end of a call so that the length of call may be used for logging purposes.

Add an Interval

If you have a date or time variable and you want to add an interval to that variable value in to create a new variable, that is done with the "AddInterval" syntax. . Let's say a serviceman is working through his job schedule and has just

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

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

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.ibmsystemsmag.com/ibmi/productnews/

2005 Products of the Year GOLD Search400.com

called in to report that he has completed the previous job. We want the IVR unit to call/text/ or email the next job location that he is on his way and give an estimated time of arrival.

```
TravelTimeToNextAppointment = 15 ;min.
PostingTimeOfLastAppointment = 8:05
```

What we want to know…..
EstArrivalTime to provide to next appointment.

```
EstArrivalTime = AddInterval TravelTimeToNextAp-
pointment, PostingTimeOfLastAppointment, "n"
```

There is also a similar "SubVariable" syntax that returns the date/time value after subtracting an interval.

dtvar = **SubInterval** *dtval, interval, type*

CreateDate/CreateTime

Any calculation on a time or date variable requires that variable to be in a time or date format. If that variable is not in the proper format, you can not do any calculations using the interval commands on it. In such cases, use the "CreateDate" or "CreateTime" syntax to accomplish that.

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.*

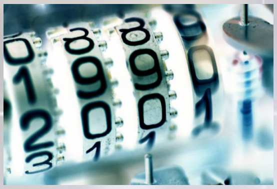


Table of Interval Types used in Date and Time Modifier Commands:

Type	**Interval**
yyyy	Year
q	Quarter
m	Month
y	Day of Year
d	Day
w	Weekday
ww	Week
h	Hours
n	Minutes
s	Seconds

ODT Vision
simplifying your business communications today and tomorrow

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

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

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

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

Advanced functions of Office Open or Closed

As mentioned earlier, there are setup screens to define if the office is open or closed based on day of the week, time of day and whether today is a holiday or not. You may want to give more specific information to your caller where the system just doesn't say the office is closed but tells what holiday it is. First it is important to know the values of the holiday table.

Holiday Number and Holiday Name

0	"New Year's Day"
1	"Martin Luther King Day"
2	"President's Day"
3	"Saint Patrick's Day"
4	Good Friday
5	"Easter"
6	"Memorial Day"
7	"Independence Day"
8	"Labor Day"
9	Rosh Hashanah
10	Yom Kippur
11	"Columbus Day"
12	"Halloween"
13	"Veterans' Day"
14	Day before Thanksgiving
15	"Thanksgiving Day"
16	Day after Thanksgiving
17	"Christmas Eve"
18	"Christmas"
19	"New Years Eve"
20-34	H1-H15 Custom Holidays

```
PassHoliday:
; HolidayNumber Command example
; Check today's date to see if it is a holiday
chkHoliday = HolidayNumber
; HolidayNumber returned a -1 so it is not a holiday.
If chkHoliday = -1 then
        Goto PassHoliday
; Custom holiday 1 (#22) is our "Closed for Inventory" day.
Elseif chkHoliday = 22 Then
        Play "CloseInv.VOC"
        Goto EndCall

; Today is a holiday, but not our custom holiday
Else
        ; Check to see if we are closed
        AreClosed = ClosedHoliday
        If AreClosed = True then

                ; We are closed today, so tell the caller the holiday.
                ; For example..........................
                ; "Today is"          (Voice file)
                ; "Christmas"   (TTS)
                ; "We are closed"  (Voice file)
                varHoliday = HolidayName
                Speak "TodayIs.VOC"
                VoiceSpeak varHoliday
                Rtn = Speak "WeClosed.VOC"
                Goto EndCall
        Endif
Endif
```

In conclusion, the ODTVision VRU system is very flexible and contains syntax and methods that allows the user to perform extensive functions or calculations on variables from the data and user's applications. This data could come from the source data fields of that user's system data or/and caller's input.◘