

# CATI MANAGEMENT: Behind the Scenes

*David Dybicki & Grant Benson, University of Michigan<sup>1</sup>*

## 1. Abstract

The Blaise CATI management system has a series of built-in features designed to maximize calling efficiencies. However, with declining response rates, it becomes important to be able to further tailor and customize the delivery algorithm to maximize response rates while controlling costs. In this paper we will describe techniques we use programmatically using Blaise manipula scripts in conjunction with Windows task Manager to maximize our sample delivery throughput for several household RDD studies.

## 2. Introduction

In the face of declining contact and response rates, it is increasingly important to be able to customize and modify delivery algorithms to different populations and study requirements. The Blaise Computer Assisted Telephone Interviewing (CATI) management system is a powerful product created by Statistics Netherlands that provides a basic set of delivery options satisfying most organizational and development needs ‘out of the box.’ However, as has been noted elsewhere (Hart and Bandeh 2003), the vast array of differing organizational needs makes it infeasible for Statistics Netherlands to generate a product that serves the requirements of all organizations. Instead, Statistics Netherlands has opted to create a basic set of features that satisfies most organizational and development needs ‘out of the box’, and allow for a substantial amount of customization through Manipula scripts to meet additional requirements.

This paper describes the implementation of specific Manipula scripts used by the University of Michigan’s Survey Research Center to increase the probability of identifying eligible household sample from a Random Digit Dial (RDD) frame, of obtaining cooperation among resistant households, and of removing sample unlikely to yield completed interviews. In addition, the paper will describe the overnight batch features we use to implement most of these techniques.

---

<sup>1</sup> The authors wish to acknowledge the significant amount of assistance we have received from Statistics Canada in learning to customize the Blaise CATI management system. Several of the programming techniques described in this paper were developed with Statistics Canada.

### **3. 'Out of the Box' Settings**

The Blaise CATI management system is hard coded to deliver cases that are most likely to yield completed interviews. In particular, the system sorts cases for delivery based on appointment status, such that 'super' appointments are delivered first, followed by 'hard' appointments that have recently reached a busy signal, hard appointments, medium-busy appointments, and so on, down to non-appointment ('default') cases. Within each of the nine priorities, the management system allows additional sorting on virtually any preloaded variable, including time zone, number of dials or calls, and whether a case has had previous contact.

The Blaise CATI management system also includes a fairly flexible tool for rotating cases without contact on the previous attempt through a series of 'time slices'. This is designed to optimize calling by restricting the number of times that an interviewer can make attempts during non-productive times, essentially keeping a case from being delivered during a particular day of the week and time of day once a specified number of attempts have already been made during that time without making contact.

In addition, the Blaise CATI management system contains a number of built-in settings that can be adjusted to allow for routing appointments back to interviewers or groups as well limiting the number of attempts that can be made on cases generally, limiting the frequency of calling cases reaching answering machines, not reaching anyone, or resulting in a busy signal.

#### **3.1 Constraints on the 'Out of the Box' Settings**

##### **3.1.1 Missed Appointments**

In the event that a hard appointment is missed, either because the respondent was not available, or because an interviewer could not make the call, the CATI management system will reset the case as a medium appointment for the start of the next day. This creates inefficiencies because it crowds out the sample at the start of the day for a time that may not be optimal for calling those cases any way. That is, if the original appointment was for 8 PM the previous evening, there is no reason to assume that calling the case at 9 AM the next day would yield a positive outcome.

##### **3.1.2 Time Slices**

By default, time slices work based on the last attempt. If the last attempt resulted in no contact, then the CATI management system will cycle through the user-defined grids until they have all been filled. Once they have been filled, the case is no longer delivered. The problem is that it does not distinguish between cases that have ever had contact on them and cases that have never had contact on them. Particularly if you are concerned about response rates, it is important to continue to make calls on a case once it has been determined that a sample line is eligible or is a working number. Alternatively, it could be helpful to distinguish between types of no contact, for example if one reaches a fax machine or computer noise, which would be potentially indicative of a phone number not being designated to a household phone.

### 3.1.3 Other Treatment Options

The CATI management system contains special treatment options for several result code types, including days to exclude from the daybatch (for no contact cases and answering machine cases), and number of times that busy cases can be dialed or 'tried'. However, this does not allow the user to distinguish between contact cases such that a resistant case may be treated differently from a general callback, nor distinguish between whether the interviewer left a message on an answering machine or just hung up without leaving a message. These cases potentially have different likelihood of obtaining contact or cooperation, and should therefore ideally be treated differently.

### 3.1.4 Daybatch Exclusion

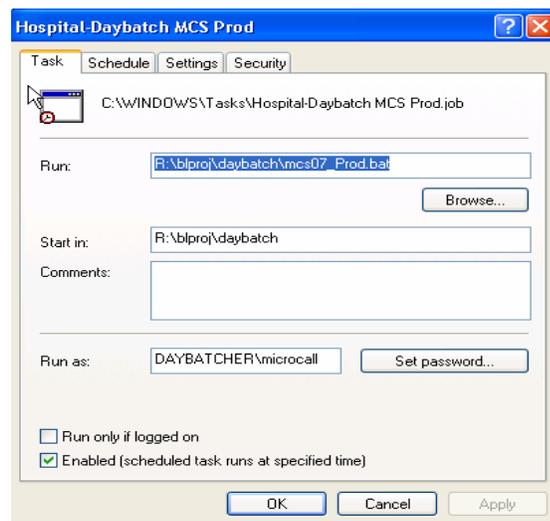
Once total call limits or time slice limits are met, cases are simply excluded from further delivery. This essentially results in the cases being unresolved until the end of a study, unless the delivery parameters are changed, with no indication for why they are excluded.

## 4. Programming Solutions to Enhance Sample Delivery

In order to make the CATI sample distribution more efficient we updated key variables in our SMS data using Blaise Manipula, and took advantage of many of the pre-existing features, including the various priority levels and appointment types. Most of the programming solutions are implemented as part of an overnight batch process.

### 4.1 Daybatch Processing

Using Windows task manager we first specify a batch file to run:



The batch file calls a .dat file that contains the following commands to run Blaise hospital, the study specific manipula scripts, the daybatch, and sends e-mails when done:

Code contained in the .dat file:

```
r:\blaise\blaise47\hospital.exe^/Lr:\blproj\daybatch\myproject.log /R
/MR:\blproj\cati\MYPROJECT\PROD\2007_08\SMS.bmi
R:\blproj\cati\MYPROJECT\PROD\2007_08\SMS.bdb

r:\blaise\blaise47\hospital.exe^/Lr:\blproj\daybatch\myproject.log /R
/MR:\blproj\cati\MYPROJECT\PROD\2007_08\MYPROJECT.bmi
R:\blproj\cati\MYPROJECT\PROD\2007_08\MYPROJECT.bdb

r:\blaise\blaise47\hospital.exe^/Lr:\blproj\daybatch\myproject.log /R
/MR:\blaise\cati\sms2006\history.bmi
R:\blproj\cati\MYPROJECT\PROD\2007_08\history.bdb

R:\blaise\blaise47\Manipula.exe^R:\blproj\cati\myproject\prod\2007_08\
ResetHardApts.msu

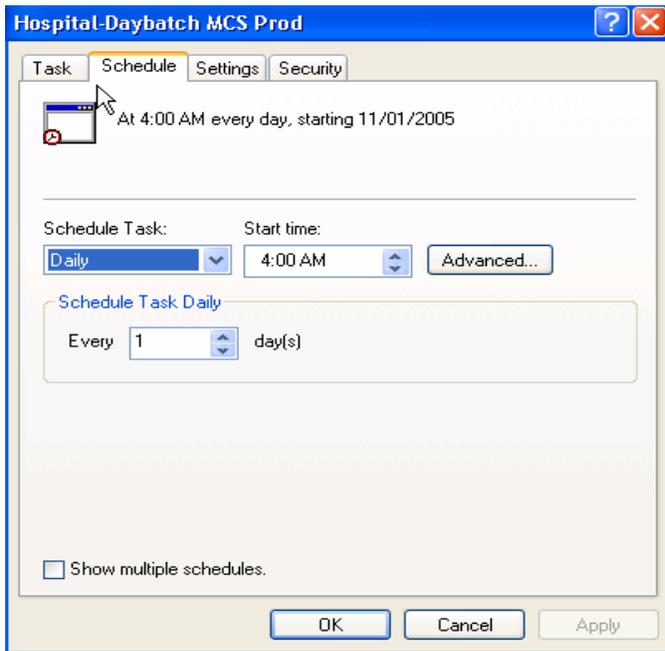
R:\blaise\blaise47\Manipula.exe^R:\blproj\cati\myproject\prod\2007_08\
SetSoftApts.msu

R:\blaise\blaise47\Manipula.exe^R:\blproj\cati\myproject\prod\2007_08\
SetSoftAptsForFax.msu

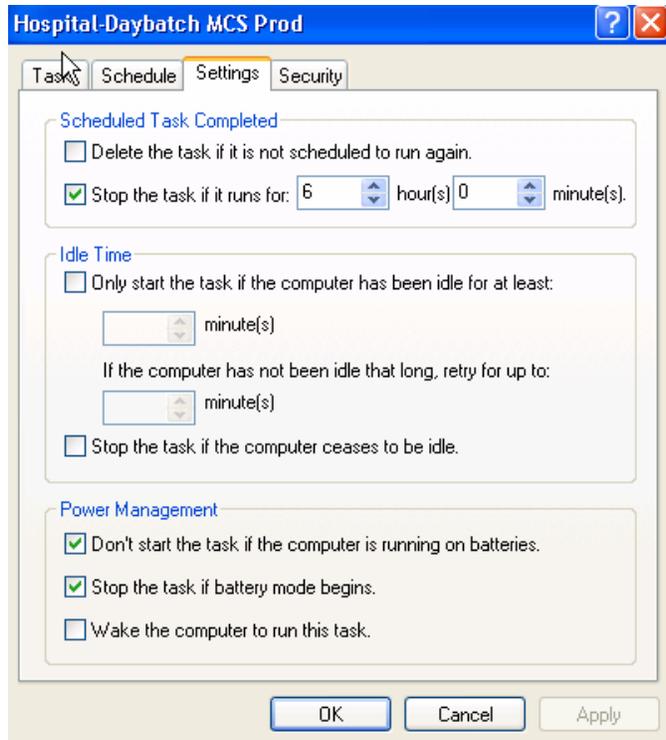
r:\blaise\blaise47\btmana^R:\blproj\cati\myproject\prod\2007_08\sms.bt
s /B /Hr:\blproj\cati\myproject\prod\2007_08\
/Wr:\blproj\cati\myproject\prod\2007_08\

R:\blproj\daybatch\batmail\batmail.exe^R:\blproj\daybatch\batmail\mypr
oject.txt R:\blproj\daybatch\batmail\message.txt
```

Then set time to run, and other settings:



Generally speaking, we try to run the daybatch at the time when it is least likely to interfere with interviewing production. We also limit how long the daybatch can continue to run to ensure that the process has been completed before our interviewing begins.



## 4.2 Resetting Missed Appointments

The scheduler default for missed hard appointments is inefficient in that when an appointment is missed (for example, the respondent is not home when we call), the case is designated as a medium appointment for the start of the next day when it is unlikely that the interviewer will reach the respondent. To address this issue, we wrote a Manipula script that resets hard appointments to soft appointments for the same general time as the original appointment for the next day. This takes advantage of knowledge that we already have about the respondent in terms of their stated availability, without requiring that interviewers go through and review all the call notes.

The script below checks for the missed hard appointments from the previous day and resets them as a soft appointment for the following day with time ranges (daypart) assigned based on the parameters below. These may be modified based on staff availability or even other historical data that may be available on a case. The key is to take advantage of knowledge that we already have on the contact.

If the missed hard appointment was set for:

- before 1pm
- between 1pm and 5pm
- between 5pm and 9pm
- between 9pm and 10am

It is reset for the next day to:

- a range of 10am - 1pm
- a range of 1pm and 5pm
- a range of 5pm and 9pm
- a range of 9pm and 10am

```

MANIPULATE
  a_counter := 0
  SMSData.OPEN
  SMSData.RESET
  SMSData.READNEXT

  {repeats this operation for every case in the DB}

  REPEAT
    {displays count of missed hard appointments found so far}
    Display('Searching for hard appointments... ' + STR(a_counter))
    {Check for a hard appointment}
    IF ( SMSData.CatiMana.CatiAppoint.AppointType = CertainDate ) AND
      ( SMSData.CatiMana.CatiCall.Regscalls[1].DialResult = 4 ) THEN
      {Check for appointments set in the past}
      IF ((JULIAN(SYSDATE) - JULIAN(CatiMana.CatiAppoint.DateStart)) > 0) THEN
        {Set the appointment type to soft}
        SMSData.catiMana.caticall.Regscalls[1].dialResult := 4
        SMSData.CatiMana.CatiAppoint.AppointType := Period

        {if the missed hard appointment occurs before (and including) 1pm,
        set the soft appointment between 10am and 1pm}

        IF (SMSData.CatiMana.CatiAppoint.TimeStart >= (10,0,0)) AND
          (SMSData.CatiMana.CatiAppoint.TimeStart <= (13,0,0)) THEN
          SMSData.CatiMana.CatiAppoint.TimeStart := (10,0,0)
          SMSData.CatiMana.CatiAppoint.TimeEnd := (13,0,0)
          SMSData.CatiMana.CatiAppoint.DateStart := SYSDATE
          SMSData.CatiMana.CatiAppoint.DateEnd := SYSDATE

          {if the missed hard appointment occurs between 1pm and 5pm set the
          soft appointment between 1pm and 5pm}
          ELSEIF (SMSData.CatiMana.CatiAppoint.TimeStart > (13,0,0)) AND
            (SMSData.CatiMana.CatiAppoint.TimeStart <= (17,0,0)) THEN
            SMSData.CatiMana.CatiAppoint.TimeStart := (13,0,0)
            SMSData.CatiMana.CatiAppoint.TimeEnd := (17,0,0)
            SMSData.CatiMana.CatiAppoint.DateStart := SYSDATE
            SMSData.CatiMana.CatiAppoint.DateEnd := SYSDATE

            {if the missed hard appointment occurs between 9pm and 10am Drop the
            appointment}
            ELSEIF (SMSData.CatiMana.CatiAppoint.TimeStart > (21,0,0) ) AND
              ((SMSData.CatiMana.CatiAppoint.DateStart + 1 = SYSDATE) AND
              (SMSData.CatiMana.CatiAppoint.TimeStart <= (10,0,0))) THEN
              SMSData.CatiMana.CatiAppoint.TimeStart := EMPTY
              SMSData.CatiMana.CatiAppoint.TimeEnd := EMPTY
              SMSData.CatiMana.CatiAppoint.DateStart := EMPTY
              SMSData.CatiMana.CatiAppoint.DateEnd := EMPTY

              {if the missed hard appointment occurs at any other time, the soft
              appoointment is set between 5pm and 9pm}
              ELSE
                SMSData.CatiMana.CatiAppoint.TimeStart := (17,0,0)
                SMSData.CatiMana.CatiAppoint.TimeEnd := (21,0,0)
                SMSData.CatiMana.CatiAppoint.DateStart := SYSDATE
                SMSData.CatiMana.CatiAppoint.DateEnd := SYSDATE
              ENDIF
            a_counter := a_counter + 1

            SMSData.WRITE
          ENDIF
        ENDIF
        SMSData.READNEXT
      UNTIL SMSData.EOF

```

One additional modification may be to route the case back to the interviewer who set the original appointment. In general, if an interviewer has been able to set the original appointment, he or she may have established a positive relationship with the respondent, thereby increasing the likelihood of obtaining an interview. On the other hand, the interviewer may not be available the next day, but the risk can be minimized by giving the deactivation delay a low number of minutes (say, 10 minutes) in the general parameter settings.

The script below looks for same day or future hard appointments, and assigns them to the interviewer who made the appointment:

```
MANIPULATE

SMSData.READNEXT
REPEAT
  {Check for a hard appointment}
  IF (SMSData.CatiMana.CatiAppoint.AppointType = CertainDate) AND
    (SMSData.CatiMana.CatiCall.Regscalls[1].DialResult = 4) AND
    (SMSData.CatiMana.CatiAppoint.DateStart >= SYSDATE) THEN

    { Assign hard appointments to be interviewer who made appointment }
    SMSData.Towhom := SMSData.CatiMana.CatiAppoint.WhoMade
    SMSData.WRITE

  ENDIF
SMSData.READNEXT
UNTIL SMSData.EOF
```

### 4.3 Routing Cases Between Multiple Time Slices

The Blaise CATI sample management system allows for setting up multiple time slices. By having a Manipula script determine whether there is ever contact on a case, for example, the time slice field can be updated to move a case from a minimal time slice to a more inclusive time slice.

Often, studies will want to place a limit on the number of consecutive no contact attempts that are made on RDD cases as they are unlikely to yield contact – much less completed interviews – after a certain number of attempts (e.g., Eckman, O’Muircheartaigh, and Haggerty 2004). Thus, it makes sense to force a grid or time slice upon these cases. However, once contact has been established, the project may still find it helpful to route the case through a grid in the event of future no contact attempts, for example if the project wishes to ensure that cases are called multiple times in a day but at specific different times of day, but eliminate the restrictions on how many total attempts may be made.

To accomplish this, two or more time slices would be set up in the CATI Manager. The first time slice would be fairly restrictive, allowing 6 or 12 total no contact tries for example, and the second would be fairly generous, allowing as many as 100 total tries spread out throughout different days of the week and times of day.

Cases would start out in the first time slice. As part of the nightly process, a Manipula script would determine if contact had ever been made on a case, and would then insert the second time slice if that had occurred.

**MANIPULATE**

SMSData.READNEXT

**REPEAT**

**IF** (SMSData.StudyType = RDD) **AND**  
(SMSData.Flags.ContactMade = YES) **THEN**

SMSData.Timeslice := 'S-2'

**ENDIF**

SMSData.WRITE  
SMSData.READNEXT

**UNTIL** SMSData.EOF

## 4.4 Subsampling and Gridding

Assigning a final status to cases that have gone through grids or time slices can also greatly increase efficiency. Alternatively, projects may also subsample cases after a certain number of calls. This is done to increase efficiency for certain types of sample or to facilitate non-response analysis.

The following script writes an automated call note for a predetermined subset of “Strata 3” subsampling cases whose calls are greater than 9. For the particular project below, Strata 3 cases are RDD cases that could not be reverse matched to any name or address, suggesting that there is a significant likelihood that these are non-household (and therefore non-sample) numbers. By using the Blaise CATI sample management system to automatically code out these cases as a non-interview, we can eliminate potential interviewer or supervisor bias in assigning result codes, and subsequently be able to apply a two-phase sampling strategy.

### MANIPULATE

```
SMSData.READNEXT
```

### REPEAT

```
{Strata = UV 70 SubSampling = UV 71
  IF (SMSData.UserValue[70] = '3') AND (SMSData.UserValue[71] = '0') AND
    (SMSData.CallRecord.Call_Num + 1 > 9) THEN

  IF (SMSData.CallRecord.ResultCode <> '4201') AND
    (SMSData.CallRecord.ResultCode <> '4202') AND
    (SMSData.CallRecord.ResultCode <> '4203') AND
    (SMSData.CallRecord.ResultType <> 'FI') AND
    (SMSData.CallRecord.ResultType <> 'FR') AND
    (SMSData.CallRecord.ResultType <> 'HO') AND
    (SMSData.CallRecord.ResultType <> 'NI') AND
    (SMSData.CallRecord.ResultType <> 'NS') AND
    (SMSData.CallRecord.ResultType <> 'PR') AND
    (SMSData.CallRecord.ResultType <> 'TR') THEN

    SMSData.ToWhom           := 'Home'
    SMSData.Callrecord.Call_Num := SMSData.Callrecord.Call_Num + 1
    SMSData.CallRecord.Call_date := SYSDate
    SMSData.CallRecord.Call_Time := SYSTime
    SMSData.Callrecord.Resultcode := '7002'
    SMSData.Callrecord.ResultType := 'NS'
    SMSData.Callrecord.Call_Note := 'Automated call record by
                                     system when strata = 3
                                     Subsampling flag = 0, and
                                     this is the 10th call'

    SMSData.WRITE

    HistoryData.INITRECORD
    HistoryData.SampleID           := SMSData.SampleID
    HistoryData.HistRecord.PhoneNum := SMSData.Phone
    HistoryData.HistRecord.Call_num := SMSData.Callrecord.Call_Num
    HistoryData.HistRecord.Call_Date := SYSDate
    HistoryData.HistRecord.Call_Time := SYSTime
    HistoryData.HistRecord.ResultType := 'NS'
    HistoryData.HistRecord.ResultCode := '7002'
    HistoryData.HistRecord.Call_Note := 'Automated call record by
                                     system when strata = 3
                                     Subsampling flag = 0,
                                     and this is the 10th call'

    HistoryData.HistRecord.WhoPhoned := SMSData.WhoCalled
    HistoryData.HistRecord.ToWhom    := 'Home'
```

```

HistoryData.HistRecord.IncentiveAmt := SMSData.CallRecord.IncentiveAmt
HistoryData.HistRecord.ApptStartDate := SMSData.CatiMana.CatiAppoint.DateStart
HistoryData.HistRecord.ApptEndDate := SMSData.CatiMana.CatiAppoint.DateEnd
HistoryData.HistRecord.ApptStartTime := SMSData.CatiMana.CatiAppoint.TimeStart
HistoryData.HistRecord.ApptEndTime := SMSData.CatiMana.CatiAppoint.TimeEnd

HistoryData.WRITE

SetBTHFields
WriteBTH
ENDIF

ENDIF { (SMSData.UserValue[70] = '3') AND
        (SMSData.UserValue[71] = '0') AND
        (SMSData.CallRecord.Call_Num + 1 > 9) }

SMSData.READNEXT

UNTIL SMSData.EOF

```

A similar script can be generated to automatically code out cases that are still in time slice 1 after 12 attempts, for example.

## 4.5 Other Uses of Appointments

Often projects will need to enable quicker identification of non-sample or non-household telephone sample lines, delay recontact effort on resistant sample, or create controlled follow-up effort for methodological experiments. All of these functions can easily be accommodated in the background using the built-in appointments functions in Blaise.

### 4.5.1 Faxes

For household telephone surveys, we typically code out telephone lines that are designated for fax machines. One way to validate that a phone number is designated for a fax machine, is to require that the number has never previously had contact on it, and has at least one day time and one evening call reaching a fax machine.

The script below applies to morning cases from the previous day reaching fax machines. The Manipula script checks the contact flag to ensure that there has never been any contact, and if these conditions are met, the Manipula script sets an evening soft appointment, in this case from 6:00 pm to 9:00 pm:

```

MANIPULATE

REPEAT
  IF SMSData.Flags.ContactMade <> Yes THEN
    IF SMSData.CallRecord.ResultCode IN ['1601'..'1603', '2009'..'2010'] THEN
      SMSData.catimana.caticall.Regscalls[1].dialResult := 4
      {Appointment}
      SMSData.CatiMana.CatiAppoint.AppointType := Period
      SMSData.CatiMana.CatiAppoint.TimeStart := (18,0,0)
      SMSData.CatiMana.CatiAppoint.TimeEnd := (21,0,0)
      SMSData.CatiMana.CatiAppoint.DateStart := SYSDATE
      SMSData.CatiMana.CatiAppoint.DateEnd := SYSDATE
      SMSData.WRITE
    ENDIF
  ENDIF
ENDIF
SMSData.READNEXT
UNTIL SMSData.EOF

```

If the evening call results in a fax noise again, a second script runs in the background to code out the phone number as non-sample.

#### 4.5.2 Days to Delay

In order to increase response rates, projects must address how they work with resistant cases. Research indicates that if a case “rests” for a period of time, interviewers are more likely to obtain subsequent cooperation from respondents than otherwise (Triplett, Scheib, and Blair 2001). To ensure that cases did receive the required exclusion from delivery after an initial resistance, we programmed a feature to allow projects to determine how soon a case should be called back. In addition, we assigned delivery of the case such that it would randomly be assigned to different times of day, with a higher probability of returning in the evening.

By allowing managers to set the number of days to exclude resistant cases ‘on the fly’, we were able to accommodate not only changes in study protocols, such as increased incentives, but also approaching the conclusion of the study period. For example, even if it is optimal to wait for two weeks after an initial resistance before attempting recontact, that guideline is useless if the study ends in one week. Additionally, by randomly assigning when the case is redistributed, we avoid overloading one particular shift with these resistant cases.

The following code delays resistant cases by a pre-determined amount of days for a randomly selected appointment time for result codes equal to 4301, 4302, 4303 or 4304 which are our codes for initial resistance:

##### **MANIPULATE**

```
ProjectData.GET('SRC.SRO.MyProject)
NumOfDays := ProjectData.ProjDaysToDelay
```

##### **REPEAT**

```
IF SMSData.CallRecord.ResultCode IN ['4301'..'4304'] THEN
  SMSData.CatiMana.Caticall.Regscalls[1].dialResult := 4
  {Appointment}
  SMSData.CatiMana.CatiAppoint.AppointType := Period
  SMSData.CatiMana.CatiAppoint.DateStart :=
    SMSData.CallRecord.Call_date + (0,0, NumOfDays)
  SMSData.CatiMana.CatiAppoint.DateEnd :=
    SMSData.CallRecord.Call_date + (0,0, NumOfDays)
  RandomNumber := RANDOM (7)
  IF RandomNumber = 0 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (10,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd := (13,0,0)
  ELSEIF RandomNumber = 1 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (13,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd := (16,0,0)
  ELSEIF RandomNumber = 2 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (16,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd := (18,0,0)
  ELSEIF RandomNumber = 3 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (18,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd := (20,0,0)
  ELSEIF RandomNumber = 4 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (18,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd := (20,0,0)
  ELSEIF RandomNumber = 5 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (20,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd := (21,0,0)
```

```
ELSEIF RandomNumber = 6 THEN
    SMSData.CatiMana.CatiAppoint.TimeStart := (20,0,0)
    SMSData.CatiMana.CatiAppoint.TimeEnd   := (21,0,0)
ENDIF

SMSData.WRITE
ENDIF
SMSData.READNEXT

UNTIL SMSData.EOF
```

Similar code may be written for other procedures, including follow-up on cases where voice messages are left on answering machines, or methodological experiments.

## 5. Conclusion

Our organization has enjoyed a great deal of success with the Blaise CATI call scheduler thanks to the control afforded by the manipulation of critical data. We have written hundreds of manipula scripts over the years that have enhanced the features of the call scheduler. If we don't get it right at first we continue to tweak the scripts until we get the desired outcome and manipulation of the daybatch. It's through this constant perseverance that efficiency is achieved.

## 6. References

Stephanie Eckman, Colm O'Muircheartaigh, and Catherine Haggerty, "Effects of Gridout Procedures on Response Rates and Survey Quality." *Journal of Survey Methodology* (2004), pp. 4762-4765.

Leonard Hart and Linda Bandeh, "Blaise CATI at Mathematica." Paper presented at the 2003 International Blaise Users Conference.

Timothy Triplett, Julie Scheib, and Johnny Blair, "How Long Should You Wait Before Attempting to Convert a Telephone Refusal?" *Proceedings of the Annual Meeting of the American Statistical Association*, August 5-9, 2001.

Westat (2004) *Blaise CATI Guide*. Rockville, MD.