



CALLBACK OPTIONS

IN

CCA 5.3.0

Version 1

Friday 12th August 2011

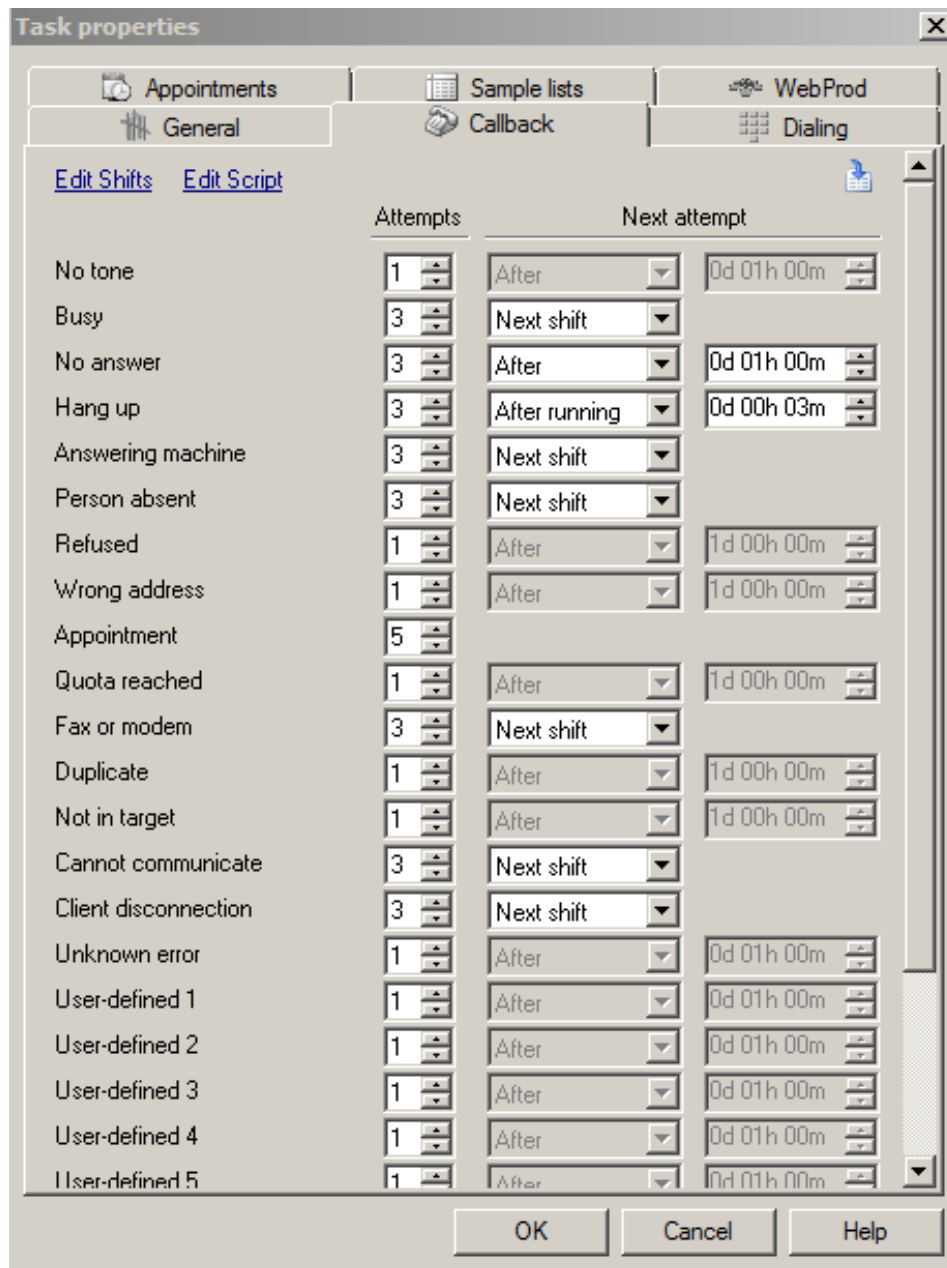
CALLBACK OPTIONS

With the 5.3.0 Cca release, we have a full scope of callbacks features which should cover most of the needs.

We have 4 callback options: Standard callbacks

- Standard callbacks on running lists (5.2.4)
- Callbacks by shifts (5.3.0)
- Callbacks by scripts (5.3.0)

All these options can be mixed (but be careful to the complexity...). Here the new Callback window in Cca/Supervisor. All these options can be set on survey properties and be overridden on list properties. Here is the new callback window in cca/supervisor:

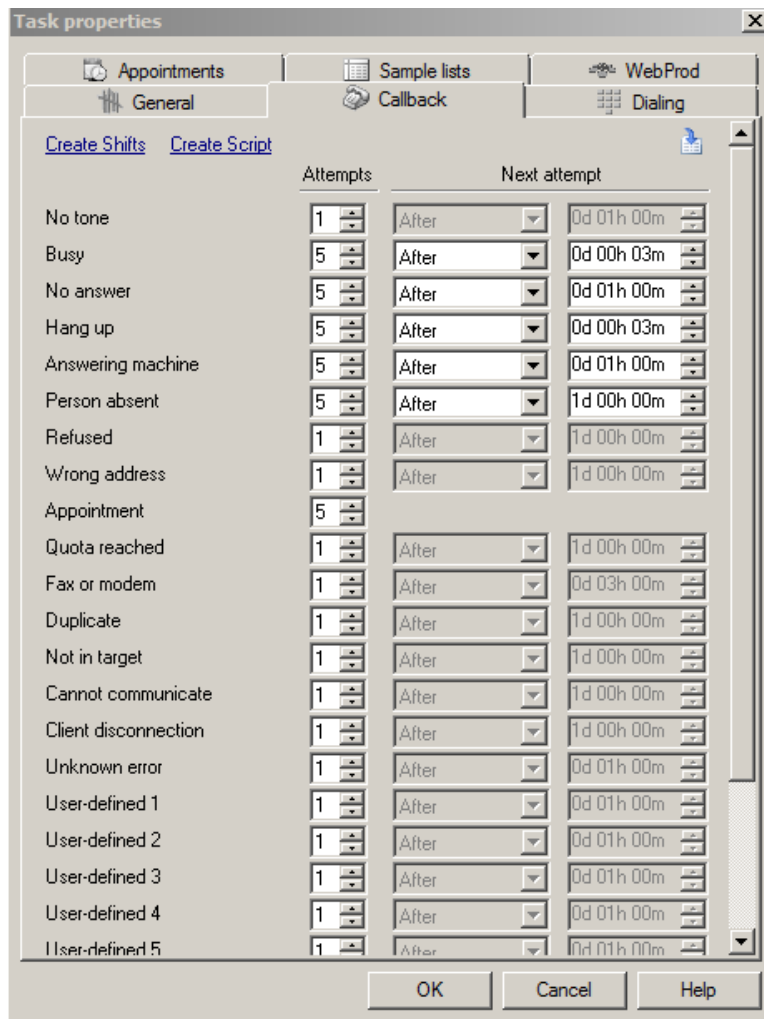


STANDARD CALLBACKS

Everyone knows the standard callback rules as these are these rules used since the beginning of the Cca (or almost).

These Callbacks are very basic:

- You specify the number of attempts for each call result codes.
- You specify the time interval between attempts.
- You specify the total maximum number of calls made for each specific address



In 5.3.0, the Callback interface has slightly changed, especially for the time between attempts which merge Day Hour Minute now. So you need to select the value in front of the letter you want to modify and enter the value manually or with the arrows.

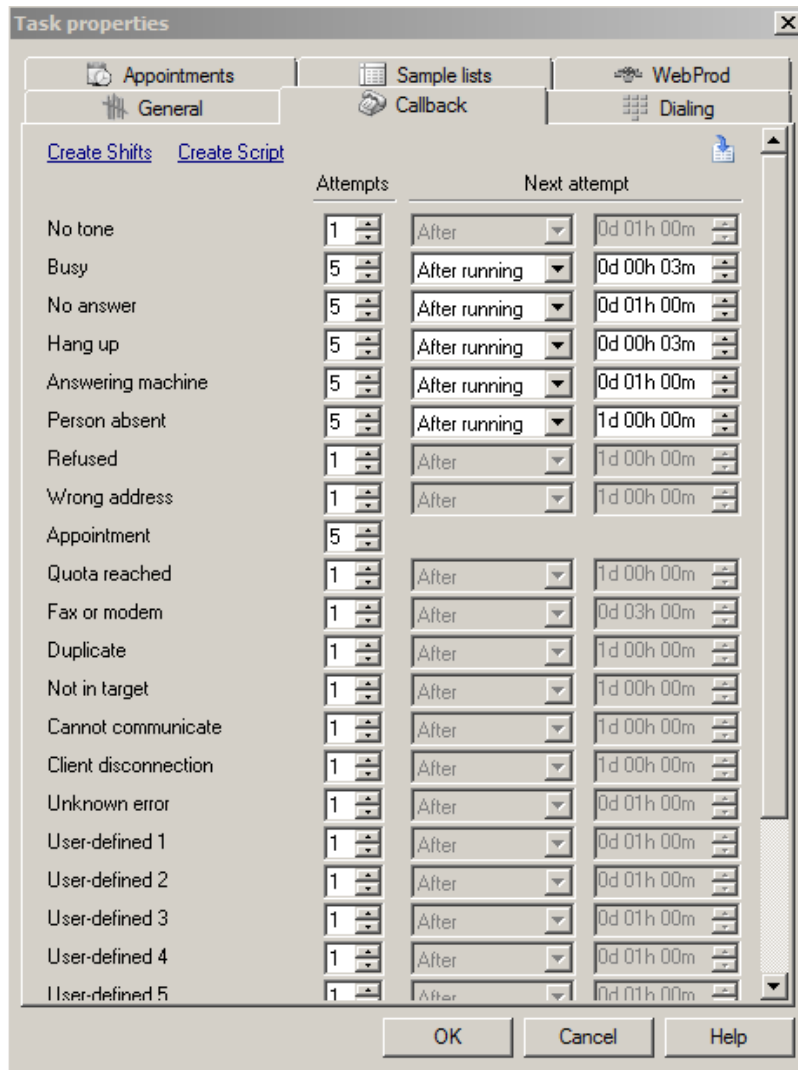
The weakness of this standard option is the time calculation based on real time. For example, if you have set a busy with a time between attempts of 2 hours, and your call of 7 PM is a busy, technically, the next attempt will be 9 PM. But if you stopped your field at 8 PM, all these busy numbers will be called at the start of the sample list the next day. So, that's for why, in 5.2.4, we have implemented the Callback on running lists.

STANDARD CALLBACKS ON RUNNING LISTS

Another big step forward concerns the callback and the new option: "After running". With this option activated, the time between each attempts is calculate on the state of the list (stopped, or "idle" when no agent is connected to the survey). Here's an example what would happen in detail:

- At 17h30 a call is made with a callback of 2h.
- At 18h all agents log out (or the list is stopped) > 30m of the necessary callback time are passed.
- At 8h the next morning the first agent logs in (or the list is restarted) > the callback time continues from now, 1h30 remaining.

- At 9h30 the contact is called again.



This option can be set for each wanted callback.

A new status has been implemented for sample list: "Running (idle)". It indicates that the list is started but no agent is connected.

CALLBACKS BY SHIFT

In 5.3.0, we have implemented the callbacks by shifts. A shift is a block of time you defined on a week calendar. So there is no longer time between each attempt, but a contact will be moved from a time shift to another, in its relative position.

The behavior is the following > we have defined 4 Time Shifts:

- TS1 > Monday 8h-12h

- TS2 > Tuesday 8h-12h
- TS3 > Wednesday 14h-18h
- TS4 > Thursday 14h-18h

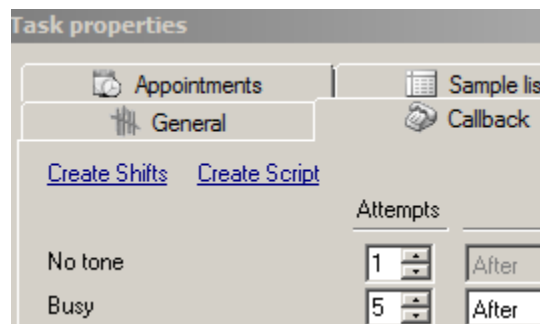
At Tuesday 9H00 (TS2), a no answer has occurred. The callback for no answer is defined as next shift (here, TS3). The contact will be called not at the beginning of TS3, but relative to the position in the shift 2:

- 9h00 is at 25% of time shift 2 (8h00-12h00).
- The callback must be done at 25% of next time shift (Wednesday 14h00-18h00)
- The callback will occurred on Wednesday at 15h00.

It will also be advisable to create shifts of more or less equal lengths in order to make sure that the amount of callbacks is more or less the same over all shifts.

How to set shifts for a survey?

You can access on this shift definition, through the callback windows on task/list properties, by clicking on “Create Shifts” (or “Edit shifts” if you already have created a shift):



You will have access to the “Shift” window:

The screenshot shows a window titled "Shifts" with a 24-hour grid. The grid has columns for 0h, 12h, and 24h, and rows for Mon, Tue, Wed, Thu, Fri, Sat, and Sun. Six shifts are shown as colored blocks: Shift 1 (orange, Mon 17:00-21:30), Shift 2 (green, Tue 17:00-21:30), Shift 3 (blue, Wed 17:00-21:30), Shift 4 (purple, Thu 17:00-21:30), Shift 5 (blue, Fri 17:00-21:30), and Shift 6* (yellow, Sat 9:30-17:00). Below the grid are two tables: "Shift Settings" and "Shift Nodes".

Shift	Description	Last Resort
1		No
2		No
3		No
4		No
5		No
6		Yes

Day	Start Time	End Time	Shift
Monday	17:00	21:30	1
Tuesday	17:00	21:30	2
Wednesday	17:00	21:30	3
Thursday	17:00	21:30	4
Friday	17:00	21:30	5
Saturday	9:30	17:00	6

Only show nodes for highlighted shift

OK Cancel

For creating a shift, you make a right click on the day wanted > “New Shift Node”> Shift [n]. By default, the system will create a shift from 0h00 to 24h00.

Then, you have two ways to set the time block wanted:

- By pointing your mouse on 1 edge of the block, you will see a horizontal double arrow you can move at your convenience.
- Or, by typing directly the value you want for Start and End Time on the shift Nodes window (lower right).

You can also select the time block and drag and drop it on another location (but always in the same day). You will keep the same interval of time but with different start/end time.

With find/edit contact feature, you're able to enter the Last Call Shift as criteria:

Number	Sample list	State	Mode	Phone number	Email Address	Survey InterviewId	Last web result	Last call	Last call shift
1	Shifts_530_20110622	Failed	Telephony	1	1	0	Not yet started	No tone	2
2	Shifts_530_20110622	Success	Telephony	2	2	1161	Not yet started	Success	2
3	Shifts_530_20110622	Failed	Telephony	3	3	0	Not yet started	Wrong address	2
4	Shifts_530_20110622	Success	Telephony	4	4	366	Not yet started	Success	1
5	Shifts_530_20110622	Success	Telephony	5	5	100	Not yet started	Success	1
6	Shifts_530_20110622	Success	Telephony	6	6	1768	Not yet started	Success	3
7	Shifts_530_20110622	Failed	Telephony	7	7	1420	Not yet started	Refused	3
8	Shifts_530_20110622	Failed	Telephony	8	8	0	Not yet started	Wrong address	1
9	Shifts_530_20110622	In use	Telephony	9	9	1758	Not yet started	Answering machine	

On List side, the information is stored on 2 places:

AskCall[ListID] > "AskShift", with values from 0 (when the call is made outside a shift) to n (the maximum number of shifts you declared on survey properties).

AskList[ListID] > 3 new fields:

- AskLastTelephonyShift: this field contains the last shift used for the last call.
- AskLastTelephonyShiftPercentage: contains the value of the position into time shift when the last call has been made.
- AskShiftAvailable: indicates in which time shift the next call will be made for this specific contact.

Note that you can define more than 1 time period for a shift (called "shift nodes") like this:

The screenshot shows a software window titled "Shifts" with a close button in the top right corner. The main area is a grid representing a 24-hour shift schedule. The vertical axis is labeled "0h" at the top left and "24h" at the top right, with a vertical line at "12h". The horizontal axis lists the days of the week: Mon, Tue, Wed, Thu, Fri, Sat, and Sun. The grid shows two shift nodes: Shift 1 (orange boxes) from 8:00 to 12:00 and Shift 2 (green boxes) from 14:00 to 18:00, both occurring from Monday to Friday. Below the grid are two tables. The "Shift Settings" table has columns for Shift, Description, and Last Resort. The "Shift Nodes" table has columns for Day, Start Time, End Time, and Shift. At the bottom right, there is a checkbox labeled "Only show nodes for highlighted shift" and two buttons: "OK" and "Cancel".

Shift Settings:

Shift	Description	Last Resort
1		No
2		No

Shift Nodes:

Day	Start Time	End Time	Shift
Monday	8:00	12:00	1
Monday	14:00	18:00	2
Tuesday	8:00	12:00	1
Tuesday	14:00	18:00	2
Wednesday	8:00	12:00	1
Wednesday	14:00	18:00	2
Thursday	8:00	12:00	1
Thursday	14:00	18:00	2
Friday	8:00	12:00	1
Friday	14:00	18:00	2

Only show nodes for highlighted shift

OK Cancel

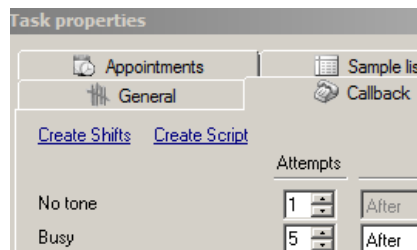
CALLBACKS BY SCRIPTS

This is the most powerful callback feature, but also the least user-friendly. While other callbacks options can be set with GUI, here, you have to type the script you want manually (or copy it from another source -txt file- or import them from another survey if you already have implemented one script).

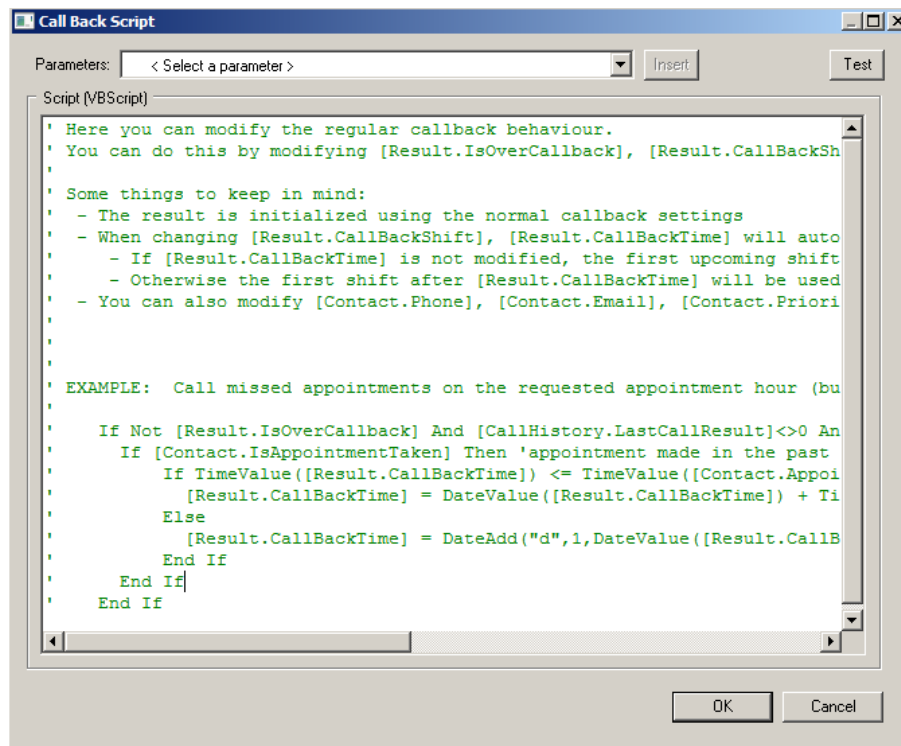
The programming language used is VBScript where there is access to several parameters. It's a relatively easy programming language but you must know it in order to not mess your callback rules up. In this way we could provide a solution for all possible requests.

This script is run after the calculation from all other methods. So in this way you only have to specify exceptions to rules already defined.

You can access to the callback script definition window by clicking on "Create Script" on task (or list) properties > callbacks tab:



The callback script definition window already contains some explanation and a small script example. As indicated in VB, all these lines begin with a quote (') for commentary purpose. Note that this window is resizable and you can copy/paste text to/from it.



This is the full description of the default content of the callback script window (obviously, you can delete it at any time):

```
' Here you can modify the regular callback behaviour.
' You can do this by modifying [Result.IsOverCallback], [Result.CallBackShift] and
[Result.CallBackTime] any way you like.
'
' Some things to keep in mind:
' - The result is initialized using the normal callback settings
' - When changing [Result.CallBackShift], [Result.CallBackTime] will automatically be
calculated
'   - If [Result.CallBackTime] is not modified, the first upcoming shift will be used
'   - Otherwise the first shift after [Result.CallBackTime] will be used
' - You can also modify [Contact.Phone], [Contact.Email], [Contact.Priority],
[Contact.AppointmentImportance] and all [ListField]'s
'
'
'
' EXAMPLE: Call missed appointments on the requested appointment hour (but the normal
day)
'
'   If Not [Result.IsOverCallback] And [CallHistory.LastCallResult]<>0 And
[CallHistory.LastCallResult]<>9 Then 'not over callback and no success or appointment
'   If [Contact.IsAppointmentTaken] Then 'appointment made in the past
'       If TimeValue([Result.CallBackTime]) <= TimeValue([Contact.AppointmentTime])
Then
'           [Result.CallBackTime] = DateValue([Result.CallBackTime]) +
TimeValue([Contact.AppointmentTime])
'       Else
'           [Result.CallBackTime] = DateAdd("d",1,DateValue([Result.CallBackTime])) +
TimeValue([Contact.AppointmentTime])
'       End If
'   End If
' End If
```

So, that's a good introduction to callback script. Here's a Microsoft reference page for the VBscript: [http://msdn.microsoft.com/en-us/library/sx7b3k7y\(v=vs.85\).aspx](http://msdn.microsoft.com/en-us/library/sx7b3k7y(v=vs.85).aspx)

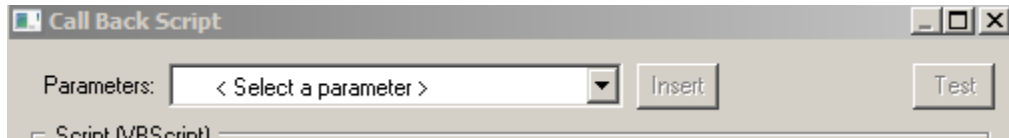
Basically, we will use condition like IF/THEN/ELSE (optional). Here is a small example regarding 3 callback rules:

```
'Specific Rules for Resultcode 2 (busy) and sex (from lst)=1
If [CallHistory.LastCallResult] = 2 and [ListFields.sex]=1 Then
    'Set the callback time in 1 day running
    [Result.CallBackTime] = DateAdd("D",1,[CallHistory.LastCallTime] +
[Contact.CallBackIdleTime])
End If

'Specific Rules for Resultcode 3 (No answer)
If ([CallHistory.LastCallResult] = 3) Then
    'Set the callback time in 5 days running
    [Result.CallBackTime] = DateAdd("D",5,[CallHistory.LastCallTime] +
[Contact.CallBackIdleTime])
End If

'Specific Rules for Resultcode 4 (Hang up)
If ([CallHistory.LastCallResult] = 4) Then
    'Set the callback time in 5 days running
    [Result.CallBackTime] = DateAdd("D",5,[CallHistory.LastCallTime] +
[Contact.CallBackIdleTime])
End If
```

On the drop down menu on the upper callback script window, you can access directly to all parameters you can use in your scripts:



Here is the list of parameters used (by group of parameters) with a small description:

Contact		
Parameters	Description	Modifiable
Contact.ID	ID of the contact in the list (also called Panel ID). (AskList.AskInterview)	No
Contact.ExternalID	ID of the contact in the external list (AskList.AskExternalUniqueid)	No
Contact.Phone	Telephone number (AskList.AskTelephone)	Yes
Contact.Email	Email address (AskList.AskEmail)	No
Contact.CallCount	Call count, included last call (AskList.AskCallCount)	No
Contact.Priority	User defined contact priority (AskList.AskPriority)	Yes
Contact.IsAppointmentTaken	True if has there ever been an appointment, in the past or now (AskList.AskAppTime NOT NULL)	No
Contact.AppointmentTime	Time of the Last appointment (AskList.AskAppTime)	No
Contact.AppointmentImportance	Importance of the last appointment (AskList.AskAppImportance)	Yes
Contact.AppointmentAgent	AgentID of the agent who took the last appointment (AskList.AskAppAgentId)	No
Contact.AppointmentAgentName	Name of the agent who took the last appointment	No
Contact.AppointmentForAgentOnly	True if the last appointment was taken "for this agent only". (AskList.AskAppForAgentOnly)	No
Contact.SurveyID	Task ID of the current survey (AskList.AskSurveyId)	No
Contact.SurveyInterviewID	ID of the current interview (AskList.AskSurveyInterviewId)	No
Contact.CallBackIdleTime	Total Time the list has been in "running (idle)" state since the last call (AskList.AskCallbackSuspendTime)	No

ListFields		
Parameters	Description	Modifiable
User-defined fields	User-defined fields which can be used on scripts.	Yes

CallHistory		
Parameters	Description	Modifiable
CallHistory.arrCallResult	Array with the call results for all calls to the current contact. See below for further details	No
CallHistory.arrCallSubResult	Array with the call sub-results for all calls to the current contact. See below for further details	No
CallHistory.arrCallAgent	Array with the agent ID for all calls to the current contact. See below for further details	No
CallHistory.arrCallAgentName	Array with the agent name for all calls to the current contact. See below for further details	No
CallHistory.arrCallShift	Array with the shift for all calls to the current contact. See below for further details	No
CallHistory.arrCallTime	Array with the time for all calls to the current contact. See below for further details	No
CallHistory.LastCallResult	Call result for the last call to the current contact	No
CallHistory.LastCallSubResult	Call sub-result for the last call to the current contact	No
CallHistory.LastCallAgent	AgentID for the last call to the current contact	No
CallHistory.LastCallAgentName	Agent name for the last call to the current contact	No
CallHistory.LastCallShift	Shift for the last call to the current contact	No
CallHistory.LastCallTime	Time of the last call to the current contact	No

All CallHistory.arr* variables contain the call history for the contact from after the latest recovery and latest derivation.

So if the call had 2 calls by agent 1, one at 12:00 (shift 1) and one at 15h (shift 2), all arrays would contain 2 elements (one for each call)

For example:

arrCallResult[0] = 3	arrCallResult[1] = 0
arrCallSubResult[0] = 0	arrCallSubResult[1] = 0
arrCallAgent[0] = 1	arrCallAgent[1] = 1
arrCallAgentName[0] = "agent 1"	arrCallAgentName[1] = "agent 1"
arrCallShift[0] = 1	arrCallShift[1] = 2
arrCallTime[0] = 12:00	arrCallTime[1] = 15:00

Configuration	
<i>These variables contain the callback settings from the main tab, they are read-only,</i>	
Configuration.arrMaxCallAttempts	callback settings: Max attempts
Configuration.arrIsCallbackDelay	callback settings: True if 'next attempt' is set to 'after'
Configuration.arrIsCallbackDelaySuspend	callback settings: True if 'next attempt' is set to 'after running'
Configuration.arrIsCallbackNextShift	callback settings: True if 'next attempt' is set to 'next shift'
Configuration.arrCallbackDelay	callback settings: Delay (in minutes), only valid if 'next attempt' is set to 'after' or 'after running'

Note that these are array with the configuration data of all possible call results.

Result		
Parameters	Description	Modifiable
Result.IsOverCallback	Set to TRUE if the contact can be called back.	Yes
Result.CallbackShift	Shift for the next call. Set to 0 or -1 to callback at CallbackTime	Yes
Result.CallbackTime	Time for the next call. If the CallbackShift is a valid value, then contact will be called back at the first available shift AFTER THIS TIME.	Yes

Please note that these values are filled in with the already calculated values from the standard callback settings.