Forsta Plus v2022

This is revision 1 of the Forsta Plus v2022 CATI Administrator Guide published in January 2022. The information herein describes Forsta Plus CATI and its features as of Build nr. 2022.1.1972. New features may have been introduced into the product after this build. Go to www.forsta.com or check “News” on the Customer Extranet for the latest updates.

Table of Contents

What's New in this Revision?

Note: Only the latest changes to this documentation are listed here. Changes made to earlier revisions are listed in the "Changes to the User Documentation" document which can be downloaded from the Forsta Extranet at https://extranet.confirmit.com. Note that you will need to log in to the extranet to download this document.

The following changes have been made in revision 1 of the Forsta Plus v2022 CATI Administrator Guide:

- The manual is updated with new logo, company and product names. Note that images will be updated at a later date, and URLs, folder names etc. will be corrected as the changes become applicable.
- The Updating Sample with Scheduling Rule Execution topic is added to the Creating and Using Scheduling Scripts chapter (see Updating Sample with Scheduling Rule Execution on page 57 for more information).
- The Limited Access Supervisor Accounts chapter is added (see Limited Access Supervisor Accounts on page 80 for more information).

Note: The general layout and language in this document is continually being corrected, adjusted and improved to ensure the user has the best possible source of information. Only NEW information and details of functionality that has changed since the previous revision are listed here - minor corrections to the text and document layout are not listed.

Important
We need your feedback so we can improve this document and provide you with the information you require. If you have any comments or constructive criticism concerning the content or layout of this documentation, please send an email to documentation@forsta.com. Please include in your email the section number and/or heading text of the section to which your comment applies.

1. Introduction to CATI Authoring

The majority of survey authoring concepts and functionality in Forsta Plus are the same regardless of the data collection mode(s) being used. However there are some additional components required when setting up a survey for telephone interviewing that may not be relevant to other collection methods. For example in a CATI project it is necessary to have a framework in place to guide the interviewer through the process of placing the call, obtaining the respondent and assigning the appropriate outcome.

Forsta Plus also has some additional architecture in place behind the scenes to handle the demands of telephone interviewing. The majority of this architecture is completely invisible to the user, however it is recommended that Forsta Plus Professional users familiarize themselves with the survey project database structure.

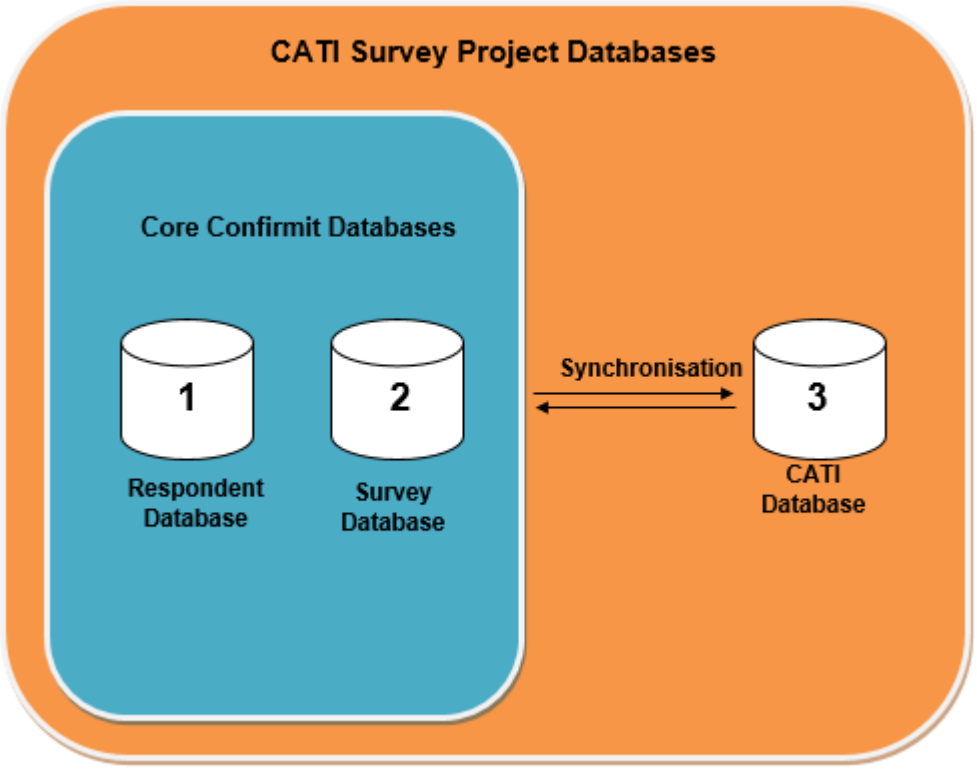


Figure 1 CATI survey project database

1. The respondent database contains background (sample) details for all respondents involved with the survey.
2. The survey database contains the data collected during interviews.
3. The CATI database is used for all aspects of Call Queue administration.

1.1. Changing Overriden Values for System Settings

System settings that configure the CATI Supervisor module are gathered in a single list. This is presented in tabular form on the **Administration menu > System Settings** tab. All Administration resources are accessible only by supervisors with Administrator rights; the Admin tab is hidden from other supervisors. The illustration below shows the System Settings list.

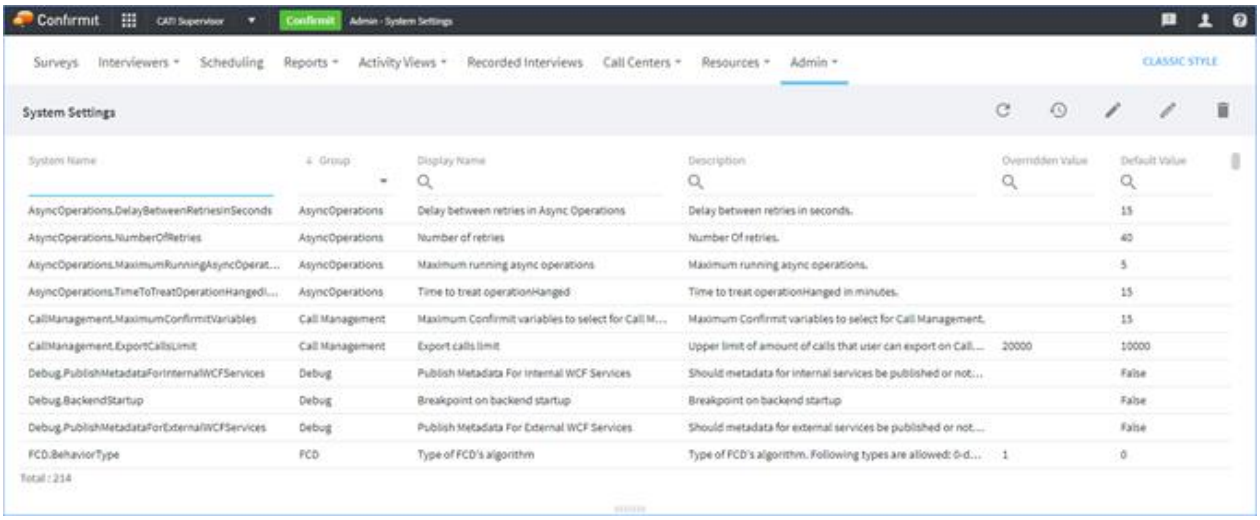


Figure 2 Example of the Admin > System Settings page

All settings are grouped according to their role in the system, with the group name being shown in the Group column. You can filter the grid by the Group name to facilitate search.

The default value for each setting is shown in the Default Value column. You can change the default value to the overridden value - to do this either double-click the required setting row in the list, or select the required row and press the **Change Overridden Value** button on the toolbar, then select the value in an overlay dialog that appears, as shown below.

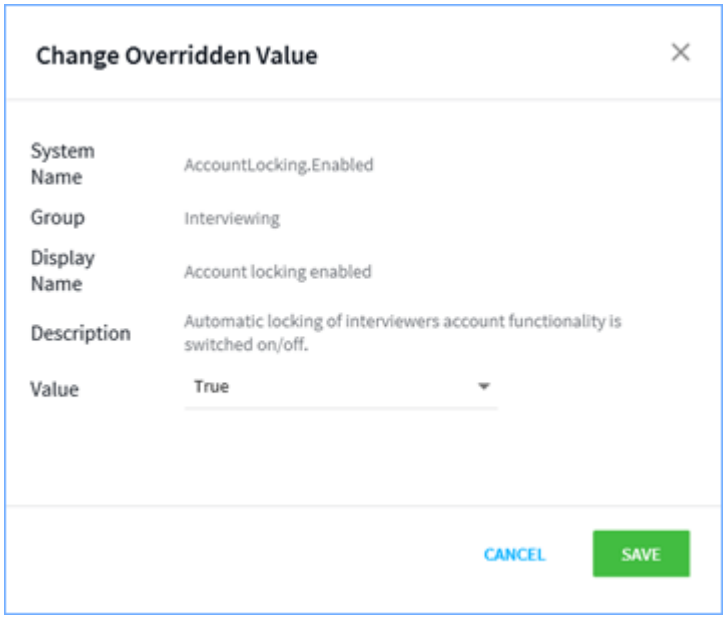


Figure 3 The Change Overridden Value dialog

The Value field format changes depending on the selected setting - use the suggested format to provide a correct value. Click Save on completion; the overlay is closed and the new value is displayed in the Overridden Value column. The new overridden value is applied in the system.

2. Getting Started with CATI Surveys

To author a survey for CATI, users must first have a project - refer to the separate Survey Designer or Professional Authoring user guides for details. After a new project is created, the collection modes for the survey are assigned in the **Project Management > Survey Settings** option. Next go to the Survey Channels tab and select the checkbox titled “CATI survey”. If the survey is to also be run in CAWI mode you can also check the “Web survey” box.

Note: Certain CATI specific functionality in the survey designer will not be available unless the CATI collection mode is enabled.

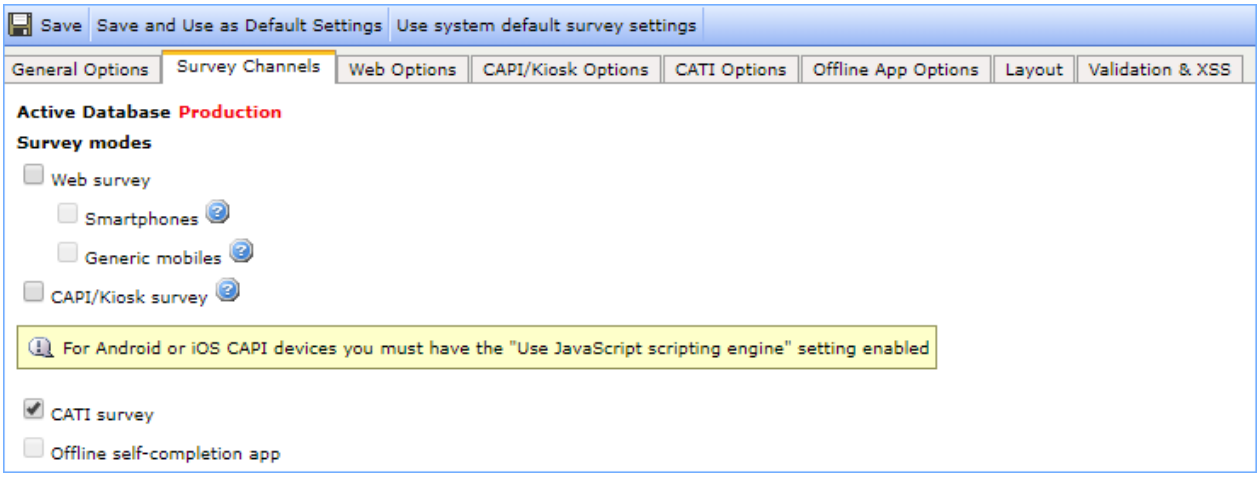


Figure 4 Applying the required survey data collection channels

With the CATI collection mode enabled it is possible to adjust some project settings which relate specifically to telephone interviewing via the CATI Options tab.

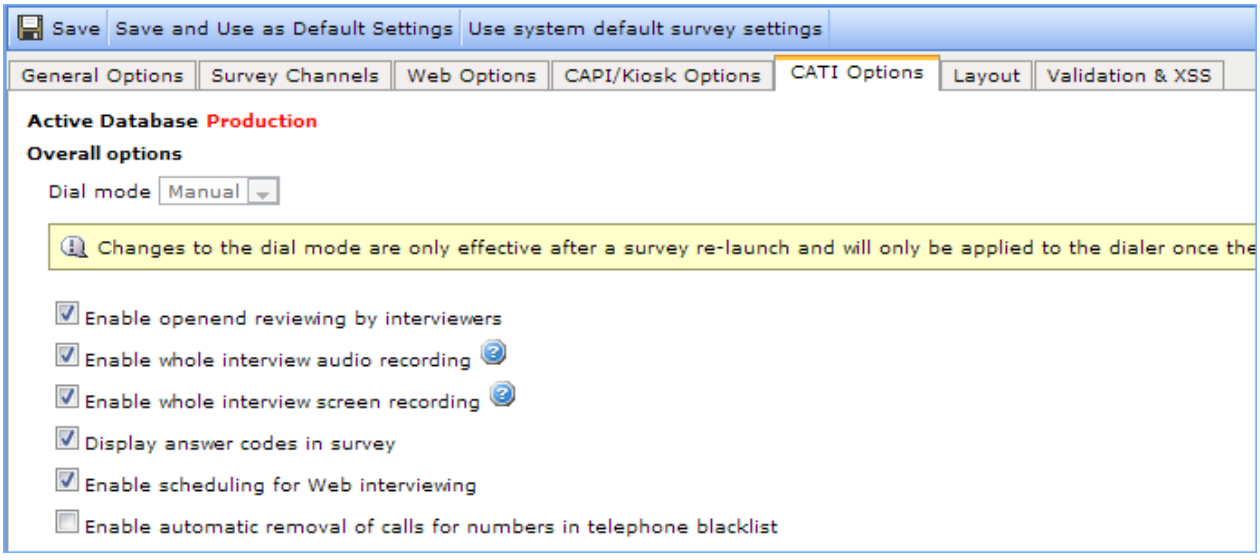


Figure 5 Assigning CATI specific project options

Important
For surveys that are to be run in CATI mode, you must always launch the survey in CATI mode before telephone sample contacts are uploaded. Any combination of modes that include CATI is acceptable, but the "CATI survey" mode must be enabled. Once sample contacts are added to a survey that is launched without the CATI mode enabled, it is then no longer possible to convert the survey to CATI mode as there is no way to copy the sample contacts across to the CATI database.

Note: Predictive dialing is only supported if the appropriate dialer add-on is used.

- **Dial Mode** : For systems enabled with an automatic dialer add-on, several automatic dialer operation modes are available:
    - **Manual** - Interviewers dial numbers manually
    - **Preview** - Interviewers can preview background information on an contact before allowing/commanding the dialer to commence auto-dialing.
    - **Automatic** - In this mode the system delivers the next connected call to the interviewers one after another.
    - **Predictive** - This dialing mode is designed to increase the productivity of the interviewers by performing "predictive" dialing. In this mode the dialer applies various algorithms that help to deliver the call to a free interviewer while keeping the number of failed ("nuisance") calls to a minimum. A dialer operating in this mode tracks and processes the call results to predict based on statistics when a number should be dialed, with the aim of delivering the interview to an interviewer immediately after the previous interview is finished.
    - **Hybrid dialing** - Some calls can be flagged to work in Preview mode, while others are dialed predictively. This is useful when calling back for appointments in cases where comments have been captured in a previous call attempt. The survey's scheduling rules can set the call to be dialed in Preview mode so that the interviewer can read the comments before starting the call.
- **Enable openend reviewing by interviewers**: When checked interviewers will be able to go back over any questions where verbatim responses have been entered at the end of each call.
- **Enable whole interview audio recording**: If supported (integrated dialer required), this option will turn on whole interview audio recording for every interview. Audio files are stored on the dialer server indefinitely.
- **Enable whole interview screen recording**: This option will enable video recording for every interview (the resulting video recordings can only be accessed by the Supervisor GUI). The monitoring console must be installed to facilitate playback. Screen recordings are held for 30 days on the Forsta Plus servers.
- **Display answer codes in survey**: This option is recommended. When enabled, answer code values will be shown next to the answer category checkboxes and radio buttons (required for keyboard support).
- **Enable scheduling for web interviewing**: Enable this option for mixed mode survey projects where there is a requirement to have calls that are transferred between modes run through the assigned scheduling rules. When this setting is enabled, completed web interviews will call the execution of the scheduling rules so that the call is dropped from the scheduled CATI calls list.
- **Enable automatic removal of calls for numbers in the telephone blacklist**: When the telephone number blacklist is enabled, the system checks each number to be dialed against the blacklist before this number is dialed. If the number to be dialed matches a number that is on the blacklist, this number is not dialed, the corresponding call is not delivered to the interviewer, this interview extended status is changed to "Blacklist", and the scheduling is not run for the call.

3. Survey Dispositions (Status Codes)

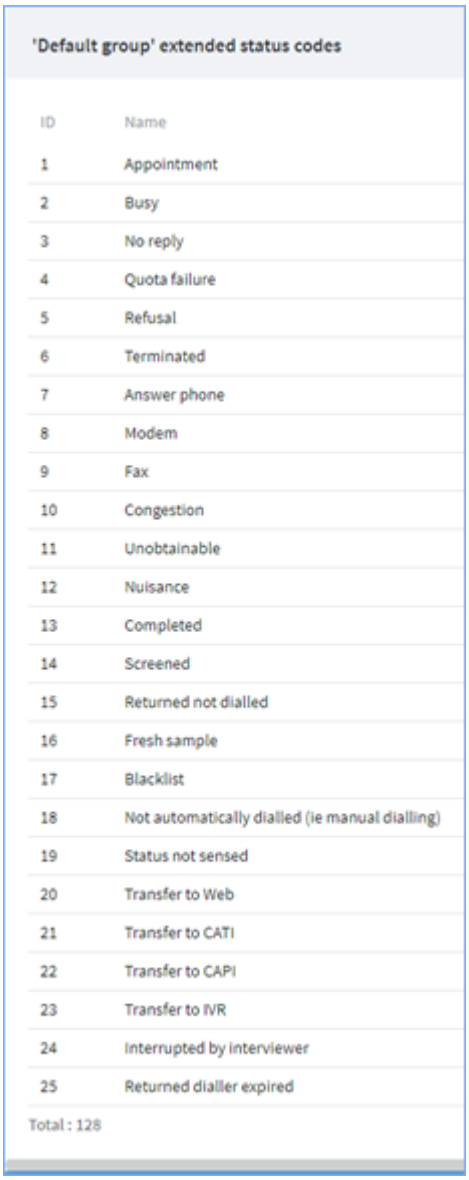
For Web surveying, three statuses are commonly used:

- Complete
- Screened
- Quota full.

The survey author can place an interview into any of these statuses by inserting a Stop node at the appropriate position within the survey. However for telephone interviewing this list of status types is often too restrictive; users prefer to be able to categorize the stopped interviews into various status categories, for example “No reply”, “Busy”, and “Answerphone”.

Note: When using Extended Status codes the regular status type may be set to “No Change”.

To overcome this restriction for CATI surveys, survey authors can choose from a much richer list of status types by applying an “Extended Status” value. In total there are 128 Extended Status types available. The first 30 of these are reserved for commonly applied outcomes, another 8 with different IDs are also applied in particular situations, and the remaining 90 are user-definable.



'Default group' extended status codes

ID	Name
1	Appointment
2	Busy
3	No reply
4	Quota failure
5	Refusal
6	Terminated
7	Answer phone
8	Modem
9	Fax
10	Congestion
11	Unobtainable
12	Nuisance
13	Completed
14	Screened
15	Returned not dialled
16	Fresh sample
17	Blacklist
18	Not automatically dialled (ie manual dialling)
19	Status not sensed
20	Transfer to Web
21	Transfer to CATI
22	Transfer to CAPI
23	Transfer to IVR
24	Interrupted by interviewer
25	Returned dialler expired

Total : 128

Figure 6 The Default Extended status codes group (as seen in the CATI Supervisor module)

When working with CATI surveys it is the Extended Status that the system observes when working in the CATI Supervisor. For example productivity reports, call management and scheduling rules are all driven by Extended Statuses. The three standard CAWI status types have corresponding (overlapping) Extended Status codes:

- Quota Full = Extended Status 4
- Screened = Extended Status 14
- Complete = Extended Status 13

It is important to ensure that an appropriate Extended Status is always given to an interview when it reaches a conclusion, and an appropriate setting for the standard status should be provided. The Extended Status can either be set via the stop node or by using the scripting function ‘SetExtendedStatus’.

A stop node in a CATI-enabled survey will have an additional entry field labeled 'Extended Status'. The Extended Status is entered as a numerical value (in the range 1..120). The first 30 Extended Status codes are reserved (the labels for these codes cannot be adjusted) whilst the remaining 90 codes are user-definable (the labels for these can be customized on a project by project basis if required).

Stop nodes should adhere to the following rules:

1. Define an Extended Status AND (if applicable) the overlapping standard status, for example
    - Standard Status = Complete
    - Extended Status = 13
2. Define an Extended Status, but if there is no overlapping standard status apply ‘No Change’, for example
    - Standard Status = No Change
    - Extended Status = 6

The only exception to either rule is when the extended status is defined with a script node using the ‘SetExtendedStatus’ function. In this case a Stop node will normally be placed when you wish to close the interview, and the Stop node must be set with the following properties:

- Standard Status = No Change (or an applicable overlapping status)
- Extended Status = Leave Blank

4. CATI Specific Scripting Functions

Several CATI specific functions are available in the system to set or retrieve system data that is used to control the CATI interviewing process. The most common functions are:

- **GetSurveyChannel()** - this is used if you are running a multimode project (CAPI, CATI, CAWI). It returns “Capi”, “Cawi” or “Cati”

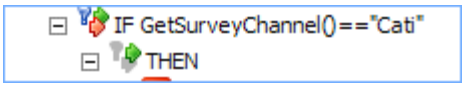


Figure 7 Using GetSurveyChannel Function

It is recommended that you always check for CATI as the survey channel even if the project is meant to be for CATI collection, in case the CAWI collection mode is required later.

- **Redo()** - can be used together with SetErrorMessage to provide a redo-context sensitive error message (this can be defined per language).

  Example:

```
Redo('q1')
SetErrorMessage(LangIDs.en, 'Error Message')
```

- **GetExtendedStatus()** and **SetExtendedStatus()** - The extended status types each have a corresponding numeric value (see Survey Dispositions (Status Codes) on page 5 for more information).

Note: The status list is also available for supervisors to view within the CATI Supervisor console (under Resources – Extended Status Codes).

GetExtendedStatus() will return the value of the current interview's extended status value (integer). If no value is set 0 will be returned.

A typical example of where this may be used is in the End block of a CATI interview. When an interviewer terminates the interview during an interview, the interview is immediately given a GetExtendedStatus() value of 6. The End block can then be used to react to this termination. In the example below, for interview terminations the type of termination is sub-classified by a termination question and then the interview is further allocated a new extended status value depending on the termination type:

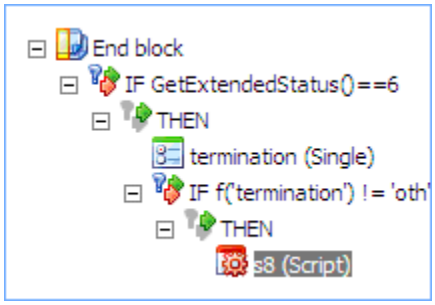


Figure 8 Using GetExtendedStatus function to classify the termination type

It is also possible to define the extended status value for an interview. This can be achieved using a script node. Here SetExtendedStatus(value) can be used to set the value of the extended status, where value is a valid integer extended status value between 1 and 120.

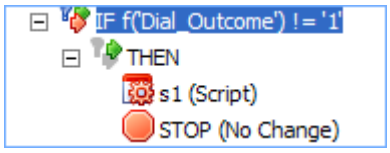
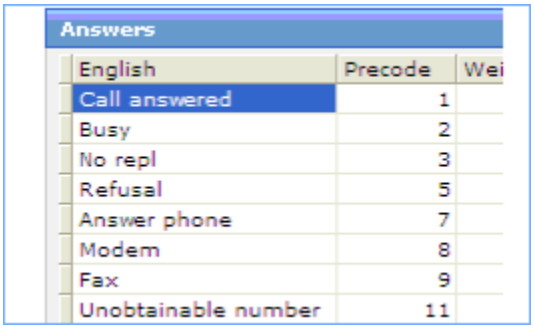


Figure 9 Using a script node to set the Extended Status

The s1 (script node) contains: SetExtendedStatus(f('Dial_Outcome'));

Where Dial_Outcome is defined as a hidden question with precodes:



Answers

English	Precode	Wei
Call answered	1	
Busy	2	
No repl	3	
Refusal	5	
Answer phone	7	
Modem	8	
Fax	9	
Unobtainable number	11	

Figure 10 Precode values are used to assign extended status codes

The initial contact screen that is used to record the call outcome of the call demonstrates the use of this function. Here we can see the Dial_Outcome question where the call outcome is recorded.

The precodes here correspond to the extended status code list within the CATI supervisor. Therefore the following script node is used to disposition the interview with a SetExtendedStatus() value corresponding to the precode for the selected category in the Dial_Outcome question.

The stop node will have a “No change” status

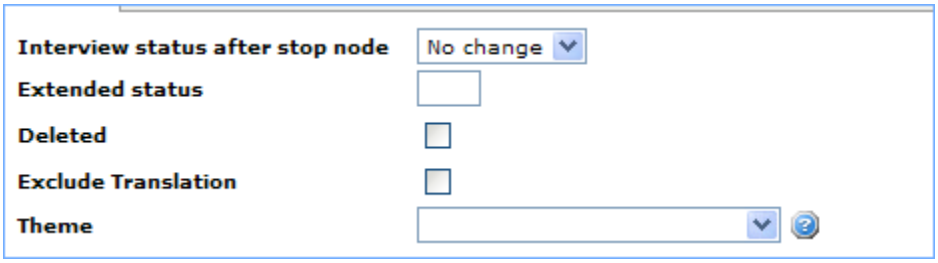


Figure 11 Applying the “No change” status to a stop node when no other default status is applicable

- **GetCatiInterviewerID()** - returns the ID corresponding to the CATI interviewer as defined by the ID column in the CATI interviewer list in the CATI supervisor. This function can be used to set (store) the interviewers unique ID to a hidden question in the survey.

  Example:

```
f('Inter_ID').set(GetCatiInterviewerId());
```

Note: If the variable “Inter_ID” is contained in a loop then the following should be used:

```
f('Inter_ID',iter).set(GetCatiInterviewerId());
```

- **GetCatiInterviewerName()** - returns the username of the current CATI Interviewer working on the interview. It can be captured to a hidden variable or displayed on screen.

  Example, to display it on the screen:

```
Hi, my name is ^GetCatiInterviewerName()^, I am calling on behalf of...
```

  Example, to set (store) the interviewers name to a hidden question in the survey named 'Inter_Name':

```
f('Inter_Name').set(GetCatiInterviewerName());
```

Note: If the variable "Inter_Name" is contained in a loop then the following should be used:

```
f('Inter_Name',iter).set(GetCatiInterviewerName());
```

- **GetCallAttemptCount()** - this is used to return the number of call attempts made to this respondent. For example, if we want to stop calling a number after 5 unsuccessful attempts, we could write

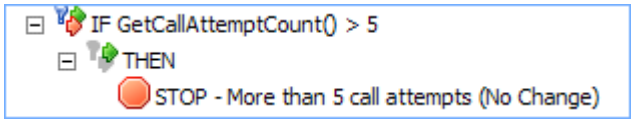


Figure 12 Disposition a call after 5 call attempts

Where the STOP node object is

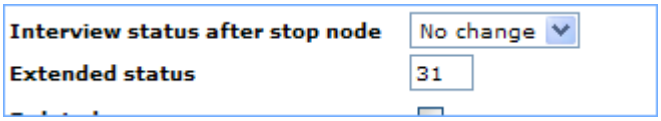


Figure 13 Applying an Extended Status using a stop node

In this case, the record will be given an Extended Status code 31, which is defined as "More than 5 attempts" in the Supervisor console.

Note: It is also possible to restrict the number of call attempts via scheduling rules. The advantage of this is that Supervisors can adjust the number of call attempts as a variable parameter on a project basis (see Restricting Call Attempts with Scheduling Parameters on page 36 for more information).

Other useful CATI functions:

The following functions are also available. Refer to the separate Forsta Scripting Manual for details.

```
GetTelephoneNumber() and SetTelephoneNumber(value)
GetExtensionNumber() and SetExtensionNumber(value)
GetTimeZoneId() and SetTimeZoneId(value)
GetLastInterviewStart()
GetLastChannelID()
GetCatiAppointmentTime()
GetTotalAttempts()
GetTotalDuration()
GetCatiRespondentUrl()
AddToCatiBlacklist()
```

5. Call Blocks

Call Blocks are used to call or execute procedures or sub routines of questions that have been created in the Call Blocks folder. This folder is external to the main questionnaire tree. The execution of a Call Block can be called at any time from within the main questionnaire tree by inserting a “Block To Call” node. To add a new Call Block object to the tree:

1. Right click on the Call Blocks folder and choose **Insert Block**.

Figure 14 Inserting a new Call Block

2. Double-click on the Call Block object to open its properties page.

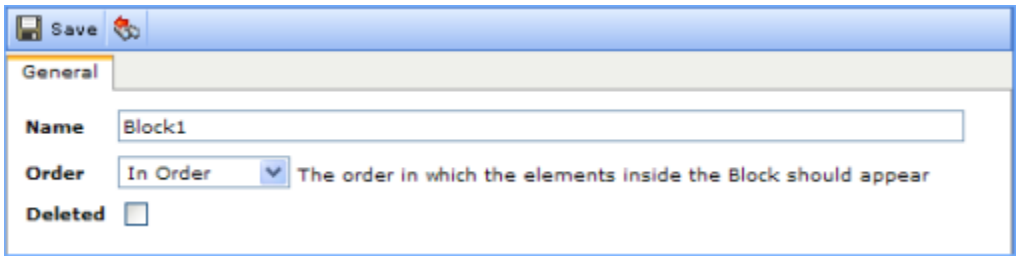


Figure 15 Call Block properties

The properties are as follows:

- **Name** - The name of the Call Block. By default the blocks will be named BlockX where X is the number in which it was created. You can rename the block as desired.
- **Order** - The order in which the elements within the block are to be presented to the respondent. Click the down-arrow beside the field and select the desired order from the list.

Note: This property only applies to the objects inside the selected block.

- **Deleted** - If the Call Block has been deleted from the tree and you wish to reinstate it, undelete it by un-checking the box.

3. Add objects (questions, loops etc.) to the Block as required, using the same methods as when adding objects to the questionnaire.

Any objects added to the Block will be executed when the Block is called by a “Block To Call” node in the questionnaire.

5.1. How to Create a “Block To Call”

Once you have created a Call Block, a “Block To Call” node can be inserted into the questionnaire tree at the point at which the routine it contains is required. To do this, right-click in the tree at the desired position and choose the “Block To Call” node for insertion or by dragging it from the new objects pane.

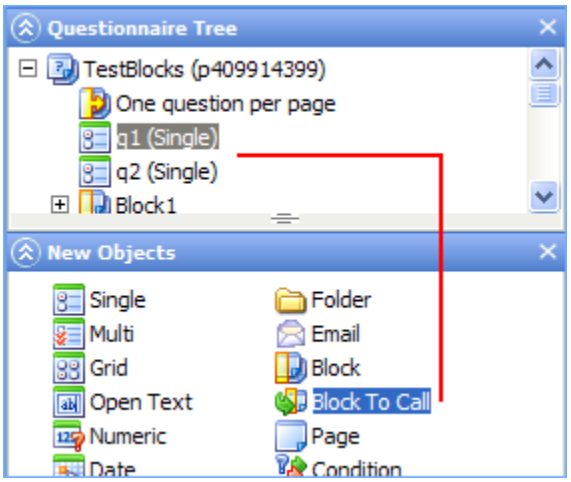


Figure 16 Inserting a Block To Call node

On creating the “Block To Call” node, a properties pane opens with a name field and a drop-down list titled “Block To Call”. A name is provided by default but you can edit this as required. In the Block To Call drop-down list, select the appropriate Call Block (in this case the one created previously, named “Block1”).

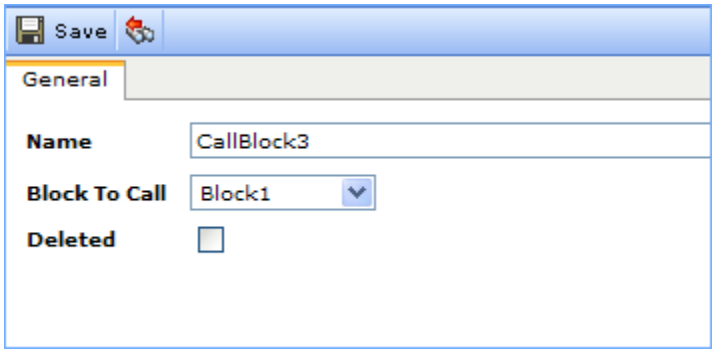


Figure 17 Selecting the “Block To Call”

The properties are as follows:

- **Name** - The name of the block. By default the blocks will be named CallBlockX where X is the number in which it was created. You can rename the block as desired.
- **Order** - The order in which the elements within the block are to be presented to the respondent. Click the down-arrow beside the field and select the desired order from the list.
  Note that this property only applies to the objects inside the selected block.
- **Deleted** - If the block has been deleted from the tree and you wish to reinstate it, undelete it by un-checking the box.

5.2. The Start Block

The Start Block is a special type of Call Block that will be activated at the beginning of an interview. Any questions or other objects located within the Start Block will be the first things that are processed when the respondent starts the interview.

The Start Block will be presented to the respondent when he/she enters the survey for the first time, and whenever the respondent returns to the questionnaire after a postponement. On returning to the questionnaire, the Start Block will present the answers that the respondent has entered previously. The respondent can then change those answers as required.

A Start Block cannot be deleted from the Questionnaire tree.

Note: If no questions are added to the Start Block, then it will simply be ignored.

A Start Block is of particular importance in CATI projects. It is here where we will define the sample information (background questions), check for the CATI collection mode, select the Extended Status (call outcomes), collect call attempt data and retrieve the interviewer name, etc.

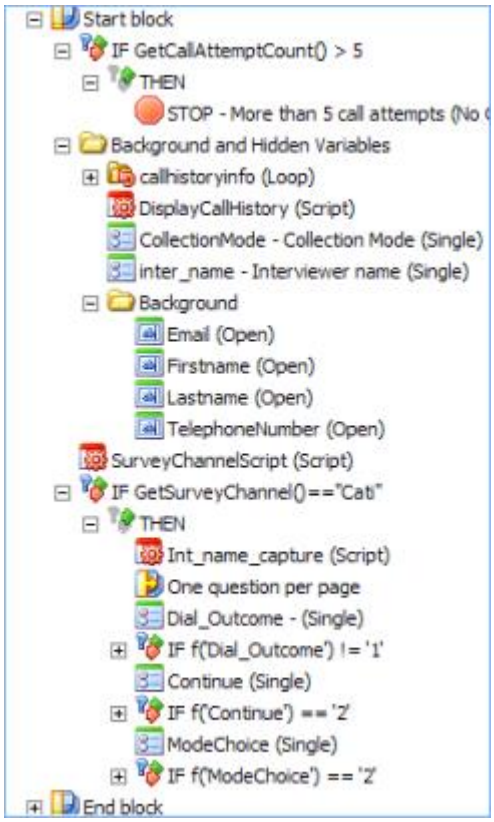


Figure 18 Example of a Start Block

5.3. The End Block

An End Block is a special case Block that will be activated at the end of an interview. Any questions or other objects located within the End Block will be the last things that are processed when the respondent completes the interview.

Any questions contained within the End Block will be presented to the respondent when he/she leaves the interview either due to a postponement or because the questionnaire is completed. If the respondent returns to the interview later and again leaves it for any reason (perhaps this time the questionnaire is completed), then any questions in the EndBlock will present the answers that the respondent has entered previously. The respondent can then change those answers as required. An End Block object cannot be deleted from the Questionnaire tree.

Note: If no questions are added to the End Block then it will be ignored.

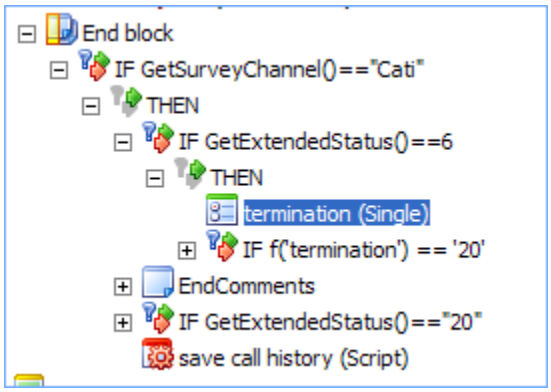


Figure 19 Example of an End Block

6. The CATI Survey Template

The main CATI call handling framework is contained in the Start block and End block of a survey. A survey template can be used so you can avoid having to redefine this framework for every new CATI survey. New CATI users are provided with an initial template, and it is strongly recommended that this is used to begin with. It contains useful functionality for CATI interviewing and can easily be re-configured. The default template is as follows:

The Start Block contains a loop named “callhistoryinfo". This is used to record call information for every call attempt made on the telephone number.

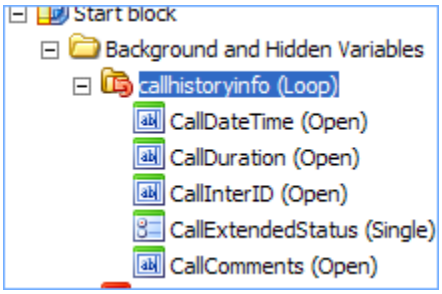


Figure 20 The Call History loop to capture background call information

When using a Start Block, background variables for respondent sample fields must be defined at the beginning of the Start Block (not at the beginning of the survey). In this example the required sample fields have been added to a “Background” folder. Typically every CATI survey should contain a field for “TelephoneNumber”. Note that “TelephoneNumber” is treated as a reserved field in Forsta Plus, so must be entered exactly as shown here. Other reserved fields include Email, TimezoneID and RespondentName.

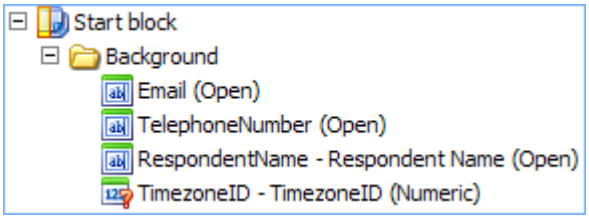


Figure 21 Background variables in the Start block

If the CATI survey channel is selected, then the interviewer will enter the CATI-specific part of the framework. During manual dialing operations they will then typically be presented with a question displaying a list of possible call outcomes - the Extended Status. If the outcome is 1 (the call is answered) then the interview will continue. A question will then be asked to check if they would like the interview on the phone or the web. If web is selected, the interviewer will then be prompted to ask for an email address.

If the outcome is not 1 (the call is not answered) then the interview ends and jumps directly to the End block. Here a question is presented to the interviewer, asking them if they wish to enter any comments.

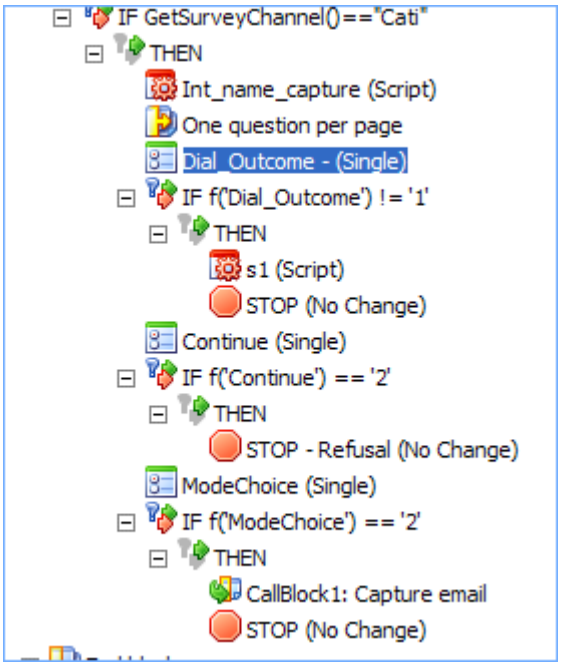


Figure 22 Example of the CATI routing based on call outcome

If the interview is interrupted at any point, the survey will move to the End Block and the interviewer will be asked to enter the reason for the interruption. Here the interviewer will have the opportunity to transfer to web, and an email address will be requested. The call history data capture and calculations will also be done here using the save call history script.

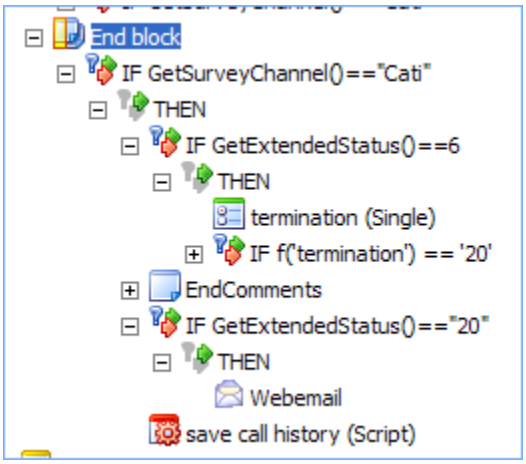


Figure 23 Using an End block to sub-classify the termination type or transfer to Web mode

The call history (for example the previous times it was engaged, no answer etc.) will be displayed at the beginning of the interview if the phone number has been tried more than once. Note the DisplayCallHistory script in the Background and Hidden variables.

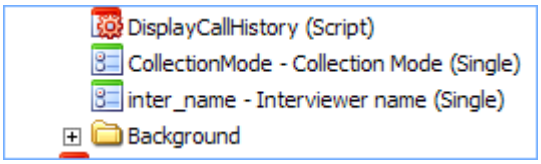


Figure 24 Script to create display call history information in a table

7. Customizing the Survey Layout - Hints and Tips

The look and feel of the CATI survey can be based on the standard public templates provided by Forsta or they may be completely custom built from a blank canvas. The public templates provided by Forsta come in both non-responsive and responsive formats. The responsive templates are recommended as they will likely provide a better experience for web survey respondents taking surveys on devices with smaller screens such as phones or tablets. The methods used to customize non-responsive and responsive templates are quite different, refer to the separate Professional Authoring User Guide for details.

There are minor differences in the template between CATI and CAWI because the CATI mode provides additional features to facilitate keyboard entry.

Answer Code Display:

Answer codes are displayed so that the interviewer can provide answers by using the keyboard. Keyboard inputs for categorical questions must be applied using the answer input area at the bottom of the interviewer screen. There is a CATI specific survey setting that determines whether or not the answer codes should be displayed. It is recommended to show them because interviewers can usually work more efficiently by entering answers via the keyboard than with the mouse.

Keyboard efficiency tips: Almost every button/action in the interviewer console has a keyboard shortcut available (see the list of the CATI Console toolbar hot keys in the CATI Supervisor User Guide for more information). When entering responses for multiple choice questions the answer codes may be supplied in the form of a comma separated list e.g. 2,4,5,7 (then press enter to apply all selections in one go). It is also recommended to make use of default and refused answer properties where applicable as this can also speed up the time it takes to input such answers (see Default and Refused Answers on page 30 for more information).

Active Question Indicator:

A vertical colored bar is displayed so the interviewer can see which question their keyboard inputs will be applied to. This feature is most useful when there are multiple questions on a survey page. You can change the highlight color for an active question indicator.

This feature uses CSS tables to apply formatting to appropriate survey layout elements. CSS formatting is applied to cells which contain data related to an active question (or sub-question). Each element is formatted by a specified class. Class properties specify the width and color of the cell border and the background color of a cell. Note that different CSS classes apply for each layout type due to differences in how the rendering functions with responsive and non-responsive layouts .

Below are examples of how these classes can be specified in either layout type.

Nonresponsive layout CSS code example:

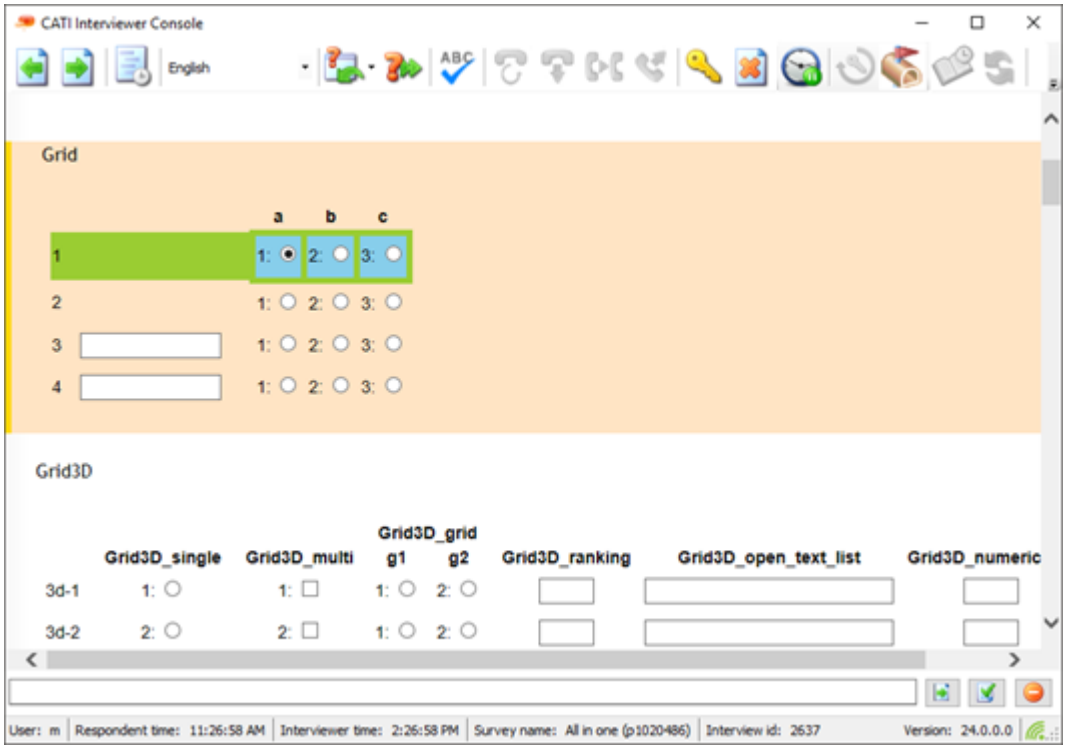


Figure 25 Example of a page with CSS highlighting applied (non-responsive layout)

The example above shows that the first question on the current page is active and highlighted (both the background and the left cell border are highlighted) plus the sub-question and answer options are highlighted.

The code example below applies this formatting. You can add this code on the Custom CSS tab when you design a survey question in the Survey Design mode in the Authoring module.

The following classes are supported:

- **.cati-current-question-container** - applies style to the cell that contains ALL question data;
- **.cati-current-subquestion-container** - applies style to the cell that contains sub-question answer option;
- **.cati-current-subquestion-container-parent** - applies style to the cell that contains sub-question name.

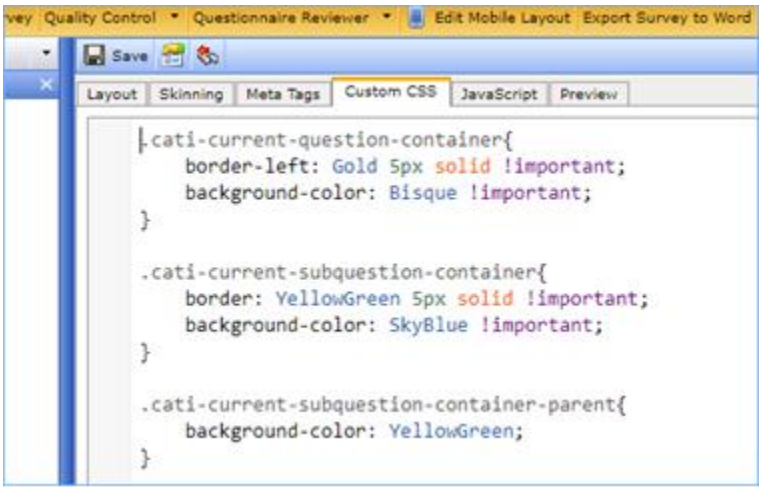


Figure 26 CSS code used to highlight a question for the non-responsive layout

It is up to the survey author to decide how the style is to look. Ensure all survey page elements are visible.

Responsive layout CSS code example:

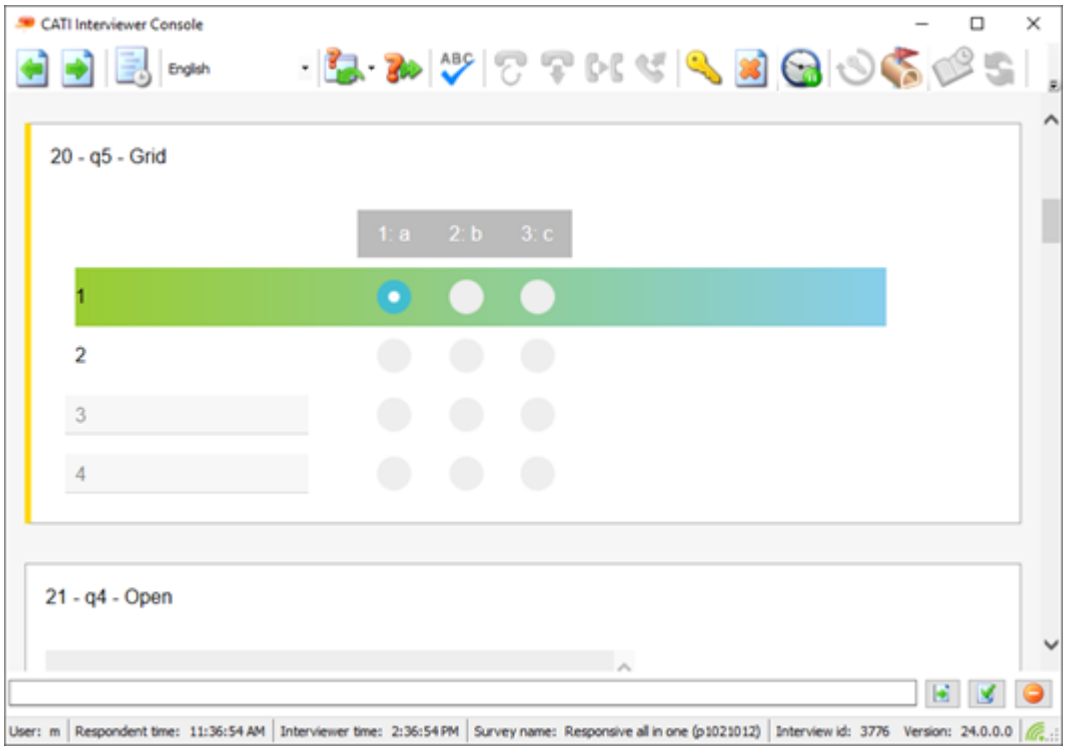


Figure 27 Example of a page with CSS highlighting applied (responsive layout)

The example above shows that the first question on the survey page is active (the left cell border is highlighted) and the sub-question is highlighted.

Figure 28 CSS code used to highlight a question for the responsive layout

The following classes are supported:

- **.cf-question.cf-question--current-cati** - applies style to the cell that contains ALL question data;
- **.cf-grid-layout__row--current-cati** - applies style to the cell that contains sub-question data.

Note that CSS used with a responsive layout does not contain classes that are applied to cells containing sub-question answer options.

It is up to the survey author to decide how the style is to look. Ensure all survey page elements are visible.

8. Quotas

Within a survey, you can use Quotas to specify the number of responses you would like to receive for different sub samples or target groups, or to limit the number of responses for the overall survey. When a quota is reached for a sub-sample, the survey can be closed automatically for respondents belonging to that sub-sample by using conditions and stop nodes, while it remains open for respondents belonging to other sub-samples that have not yet been filled. In addition, Forsta Plus can send alerts by email to specified addresses when a quota is full.

Forsta Plus allows full quota control based on both demographic and attitudinal questions, or not based on any specific questions for an overall quota (global quota) on the total number of completes. Quota control can also be achieved on compound quotas, that is quotas consisting of or built on two or more variables.

8.1. Defining Quotas

A survey can have one or more quotas, which in turn can be single or multidimensional, for example Gender (single) Gender x Age (2-dimensional)

To define a quota, right click on the quota folder in the questionnaire tree and insert a new quota. Edit the quota name as required, then drag and drop the question(s) you wish to base the quota on, into the quota folder. You can also multi-select the required questions and then right-click on one and choose Quota Wizard.

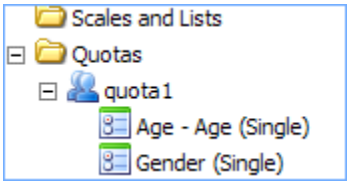


Figure 29 Quota definition added to a survey in Survey Designer

Note that to manage the defined quota in the CATI Supervisor you must enable the "Display Quota in CATI Supervisor" option on the Quota properties > Settings tab. When this option is enabled, defined quotas become available for use in the CATI Supervisor module - the Quotas tab is added to the Survey View and defined quotas are added to the list of available quotas in the Quotas tab.

The "CATI Delivery when Quota not Full" option prevents interviews from being delivered after the quota cell they fall into is full. If this option is enabled, all such "excessive" interviews are either removed from the CATI call list and assigned an extended status value of 27 ("Filtered by Call Delivery") or they are held in the scheduled list and given the status ‘Disabled by quota’ .

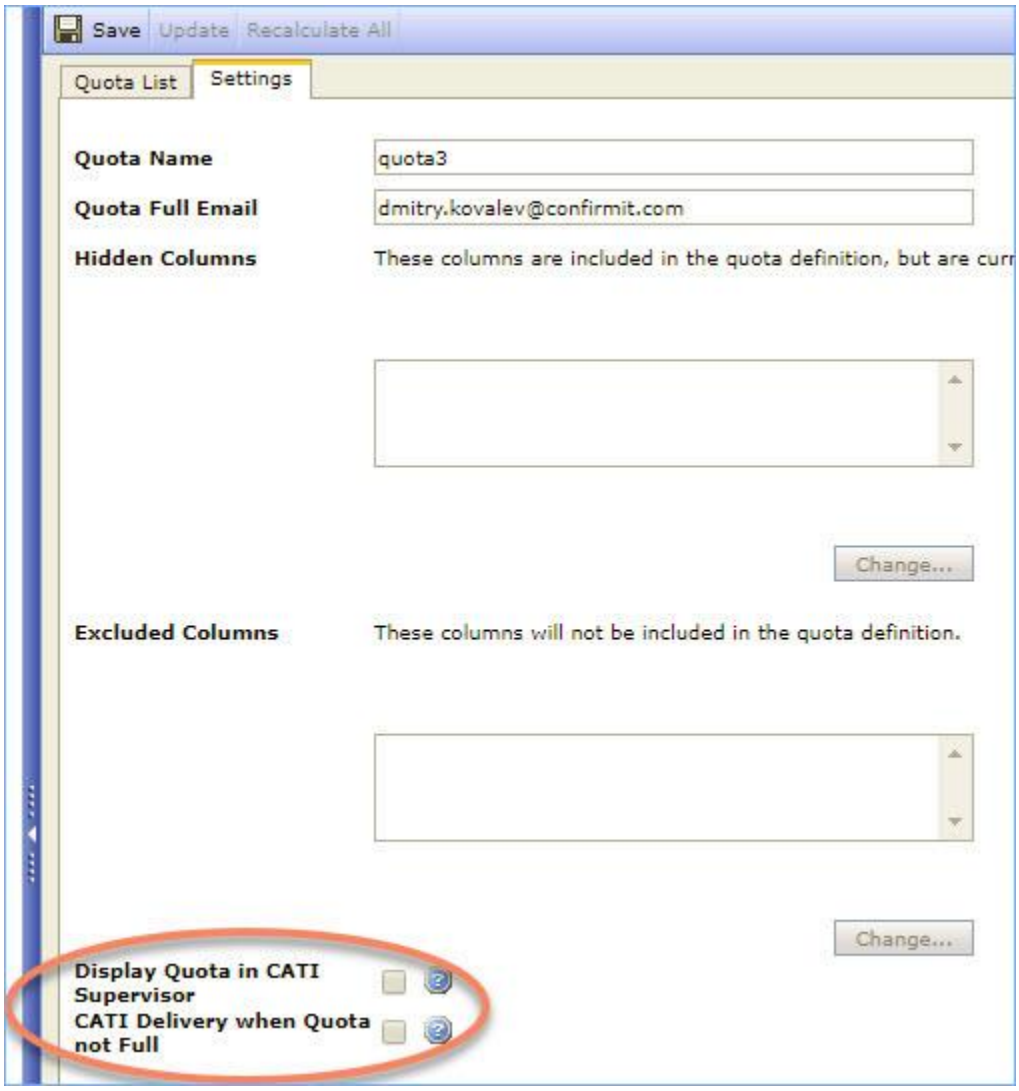


Figure 30 Enabling quota for the CATI system in Survey Designer

Note: Enabling either of these options enables the "Available as CATI filter" option (see Enabling a Variable to be used as a CATI Filter on page 28 for more information) for all variables that are currently used in this quota, so the variables chosen for CATI quotas become available in the CATI system.

Important
The Filtered Call Delivery (FCD) system must be enabled in the CATI Supervisor > Quota Management tab for the system to process the "Quota Fail" calls effectively.

8.1.1. Setting Quota Targets

To set the targets:

1. Right click on the required quota and select **Edit in Grid Mode**.

   The quota matrix appears on the right screen.

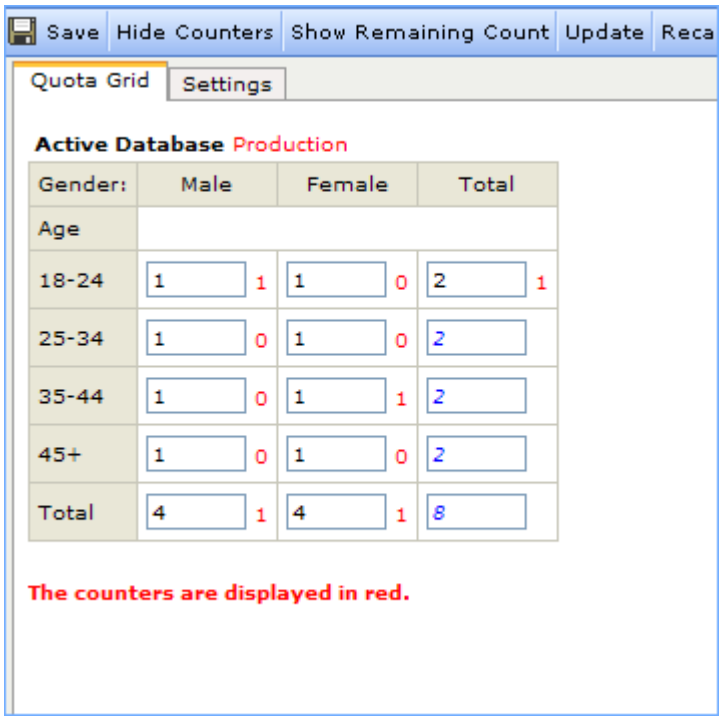


Figure 31 Defining quota cell targets

2. Edit the targets as necessary.

8.2. Checking for Quota Failure

Use the qf("quotaname") function to terminate the interview for respondents who fall into quotas that are full. The qf function returns true if the specified quota is full for the cell.

As shown in the figure below, you can stop the survey for the respondents in a full quota by using a condition and a stop node with status "Quota full".

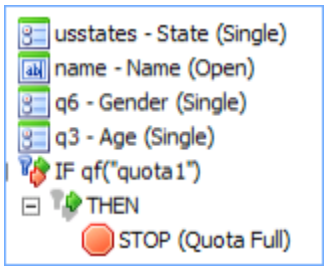


Figure 32 Stop node to disposition interviews when a quota cell is filled

The notification email for a full quota is triggered from the qf('quotaname') function. The notification email is not sent when the respondent filling the quota completes (the last to respond before the quota is full), but when the first respondent reaches the qf('quotaname') function when the quota is full.

8.3. Quota Counts and Quota Target Functions

The function qt is used to retrieve the target set for a particular quota, and qc is used to retrieve the current count.

```
qt(quotaName)
```

```
qc(quotaName)
```

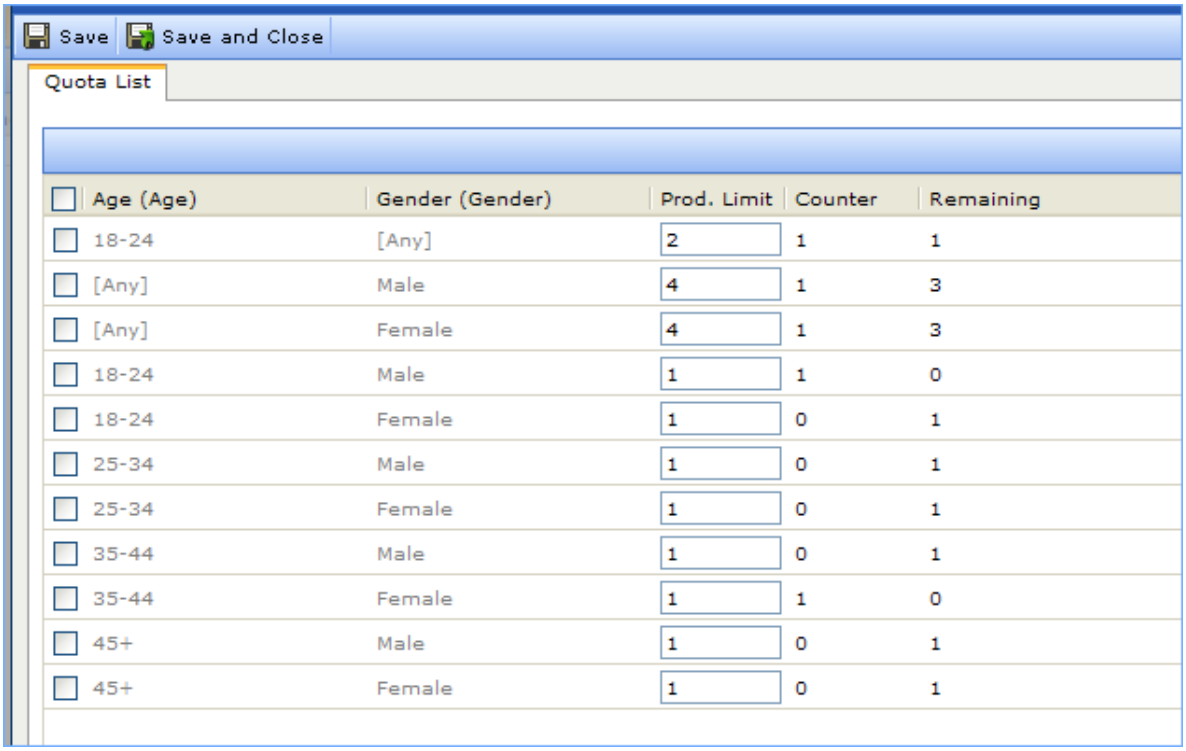
qt and qc both return an integer. They will return target and count for the quota cell in the quota quotaName corresponding with the current respondent's answers on the questions the quota is based on.

If the respondent qualifies for several quotas within quotaName, qt and qc will return -1.

8.4. Keeping Track of the Quota Counts and Targets

CATI Supervisors can keep track of the quota counts in two places:

1. From the **Project Management** menu in the main Forsta Plus UI, by selecting Quota Targets and then the required quota. The following screen appears when clicking on the quota. Here the supervisor can change the targets (Prod. Limits).



Age (Age)	Gender (Gender)	Prod. Limit	Counter	Remaining
18-24	[Any]	2	1	1
[Any]	Male	4	1	3
[Any]	Female	4	1	3
18-24	Male	1	1	0
18-24	Female	1	0	1
25-34	Male	1	0	1
25-34	Female	1	0	1
35-44	Male	1	0	1
35-44	Female	1	1	0
45+	Male	1	0	1
45+	Female	1	0	1

Figure 33 Examples of Quota targets (Prod. Limit) and achieved values (Counter)

2. From the CATI Supervisor interface, by selecting a survey from the surveys list and then by selecting the Quotas tab. Note that the Quotas tab is only visible when the quota setting 'Display Quota in CATI Supervisor' is checked.

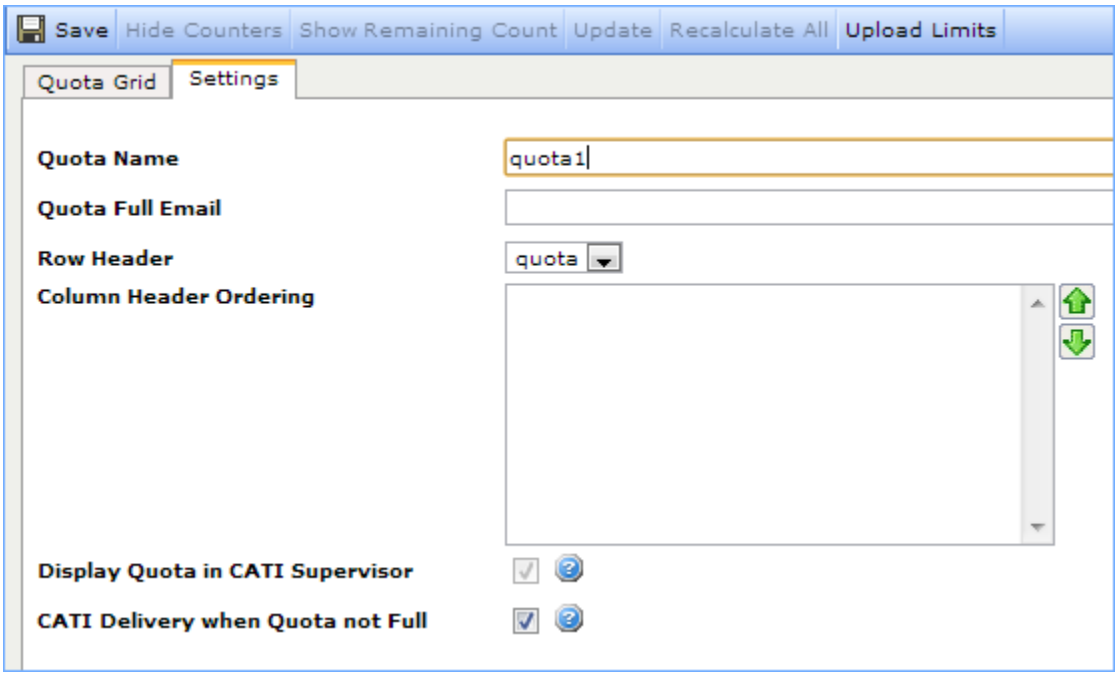


Figure 34 The Quota supervisor > Settings page

8.5. Quotas Based on Background Fields

Another CATI-specific quota setting is **CATI Delivery when Quota not Full**.

The system provides a choice of two alternative system behaviors for handling quota closures. The older (original) behavior is that when the setting ‘‘**CATI Delivery when Quota not Full**" is enabled, CATI interviews will automatically be removed from the CATI scheduled calls list and be given an extended status value of 27 ('Filtered by call delivery') when the quota cell they fall into is fulfilled.

The newer behavior will automatically set the applicable calls in the scheduled list to be flagged as disabled so the calls will remain in the scheduled list. The key benefits are:

- Disabled calls will retain their associated call properties, for example call priority, call time, user assignment and shift. This is helpful if the quota must be re-opened.
- Quota-disabled calls are color-coded in yellow for easy identification in the Call Management UI. Calls disabled manually by a supervisor in Call Management are color-coded red.
- If a cell is re-opened, for example by a supervisor increasing the limit, then the applicable disabled calls will automatically be re-enabled; there is no need to re-activate calls manually. Any manually disabled (red) calls will remain disabled.
- In optimistic quota mode, calls will automatically be disabled as soon as the desired quota limit has been met by the in-progress interviews. If the in-progress interviews fail to compete then the system will automatically re-enable the applicable calls.

As there are several benefits with the new behavior, it is recommended to switch over to this if you are not already working with it. To switch to this mode, ask a CATI administrator to go into the Resources/Settings area of the supervisor UI to make the required change. This change is a company-level setting that will affect all subsequent surveys. Note that it is possible to switch back, though it is not advisable to switch back and forth or to make the change while live interviewing is in progress.

Why is this **‘CATI Delivery when Quota not Full’** feature useful?

When a survey contains quotas that are based on background sample variables, the CATI system may become slower to deliver calls to interviewers as the quota targets in the project begin to close.

For example: Imagine a survey contains a sample-based quota for gender, and the desired quota target for male respondents has already been achieved. Despite this, there still remains a number of fresh sample records for male respondents in the call queue.The default operation is to process these respondents through the execution engine. As it does so, the male respondents will be automatically repositioned to the “Quota failure” status as expected. However, as each record is processed individually, Forsta Plus may take some time to sort through the sample to find records that are still available to deliver to the interviewer(s). The interviewer(s) would see the screen flickering as calls are being processed and the supervisor may observe a rapid increase in the number of calls being repositioned as “Quota failure”.

To avoid such delays the CATI specific quota setting in “CATI delivery when quota not full” can be applied. When this setting is enabled, all interviews in the call queue which fit into any given quota cell will be automatically disabled OR placed into the default extended status “Filtered by call delivery”.

This setting is only available for background variables that are based on single type variables; multi and numeric type questions are not supported.

Note: When a survey has this setting enabled the quota tab will always be made available to Supervisors via the main supervisor control panel in addition to the “Quota Targets” view in the main menu. The variables used in the quota will automatically be given the property ‘Available as CATI filter’.

9. Call Management with Survey Variables

The CATI system does not work directly with the respondent entries in the core Forsta Plus respondent database. Instead, it works with a dedicated CATI database which is used for the call management tasks. However, only the system variables are synchronized automatically from the core database to the CATI database (for example TelephoneNumber, RespondentName, TimeZoneID etc.).

This has some implications, especially when using quotas which are defined solely with sample variables. The Available as CATI filter property allows you to make selected variables available in the supervisors Call Management window.

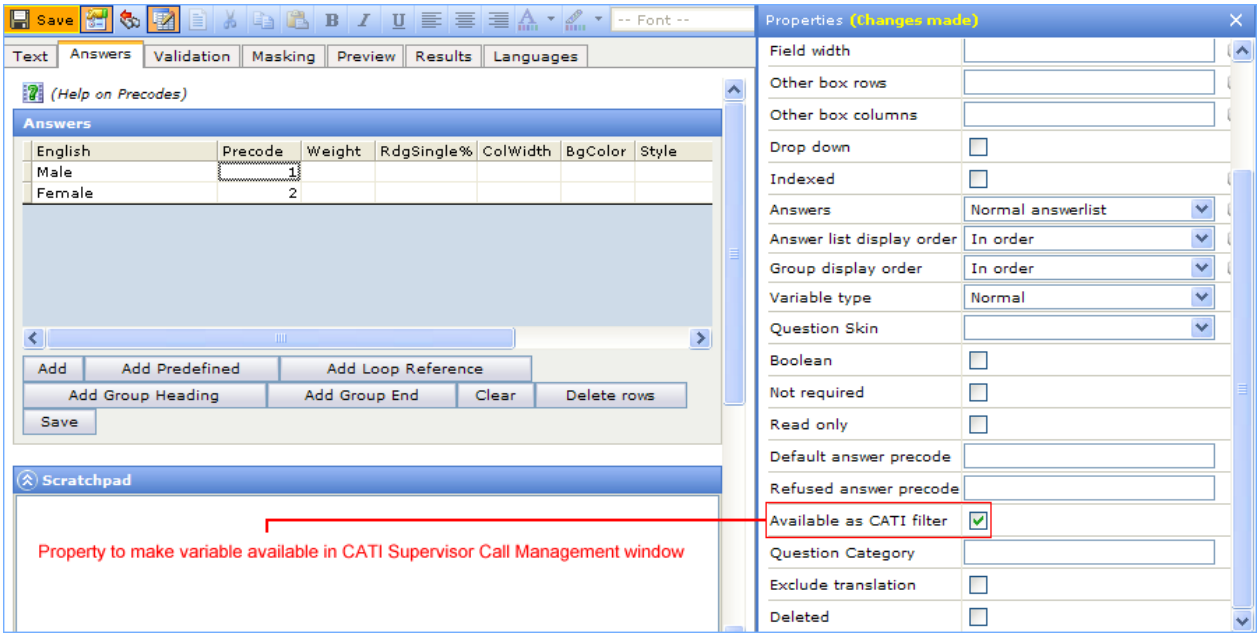


Figure 35 Assigning property to make a variable available in the CATI Supervisor

Note: A maximum of thirty variables per survey may have this setting applied. The property can be enabled or disabled at any time, but the survey will need to be re-launched after it is changed.

9.1. Enabling a Variable to be used as a CATI Filter

Any variable can be used as a filter for creating various selections in the CATI Supervisor. When designing a survey in the authoring application, for each question you wish to use as a filter, check the Available as CATI filter option.

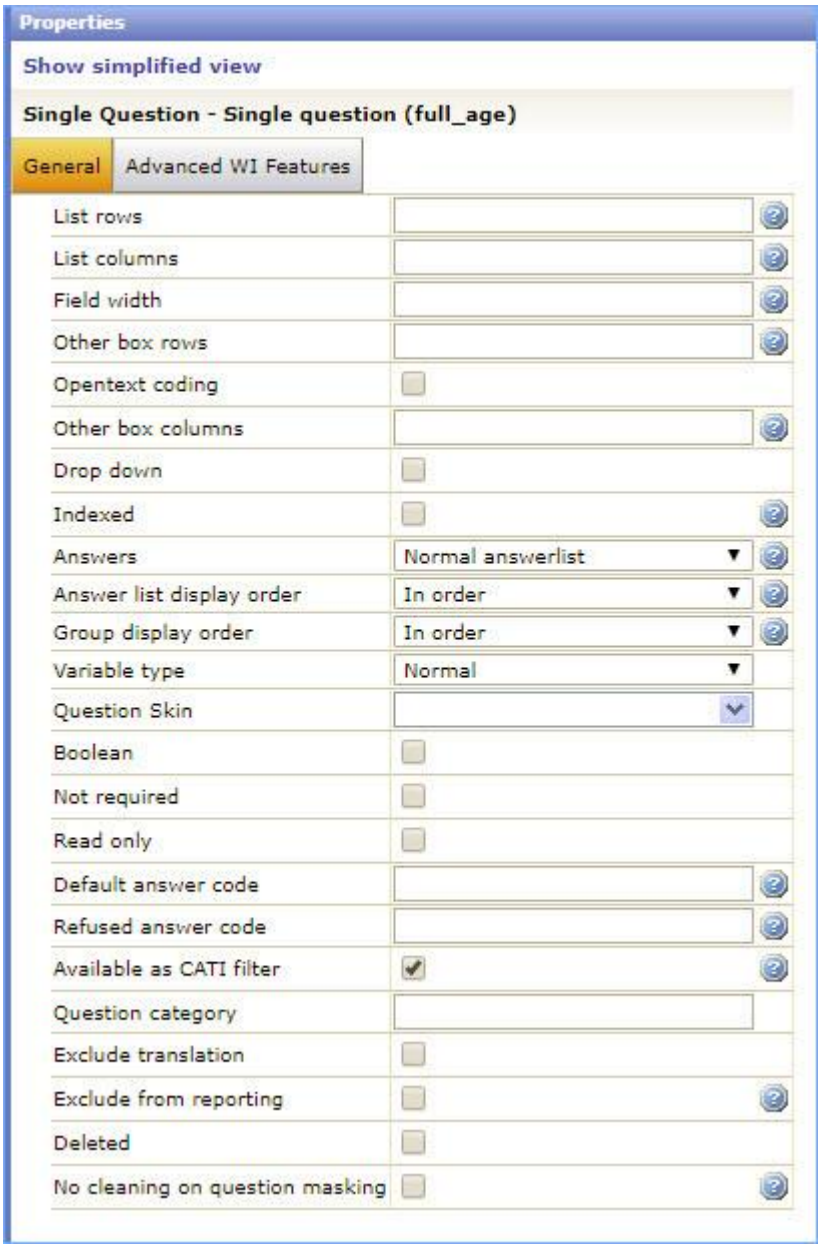


Figure 36 Enabling the "Available as CATI Filter" option for a variable in Survey Designer

Enabling the option causes the data for that variable to be replicated to the CATI database, and this makes the variable available for a number of functions in the CATI system. For example the variable will be available as custom columns in Call Management, it will be usable as a manual interview selection filter, or for filtering in the CATI reports.

Note: If the "Display Quota in CATI Supervisor", or "CATI Delivery when Quota not Full" settings are enabled for the quota, this option is turned on automatically for all variables used in the quota (see Defining Quotas on page 22 for more information).

Important
For the system to process the "Quota Fail" calls effectively, the Filtered Call Delivery system (FCD) must be enabled in the Quota Management in the CATI Supervisor UI.

9.2. Default and Refused Answers

For data consistency and interviewer data entry purposes, you can decide if you want a particular answer to be the default answer or the refuse answer. For example:

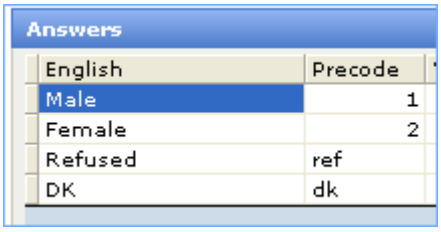


Figure 37 Applying code labels for refused and default answers

In the question properties, define:

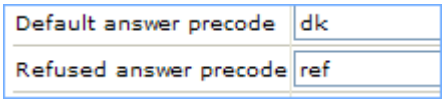


Figure 38 Setting code labels in the question properties

The interviewer will then be able to use the **Refuse** and **Default** answers during interviewing, and these can be selected from the lower-right corner of the interviewing console or by using the their respective keyboard shortcuts.

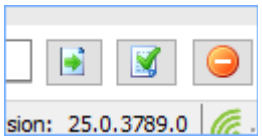


Figure 39 Interviewer console shortcut buttons

Note: The interviewer keyboard shortcuts are Ctrl+d (default) and Ctrl+r (refused).

9.3. Blacklisting Telephone Numbers

The CATI blacklist feature enables a list to be created which contains any numbers that must never be dialed, either manually or automatically.

The system will check to see if the numbers already exist in the blacklist at the time the sample data is uploaded, and again immediately before a call is delivered to an interviewer. If the number from the respondent sample is found in the Blacklist then it's Extended Status is changed to 'Blacklist' (Extended Status value 17) and the number will not be scheduled. A count of rejected numbers can be seen in the Sample Utilization report.

Note: If a predictive dialer is used, the behavior will be the same as above but checking will be invoked when numbers are about to be delivered to the dialer.

In the Professional Authoring **Survey Settings > CATI Options** tab there is a check-box to determine whether the survey is to respect the blacklist.

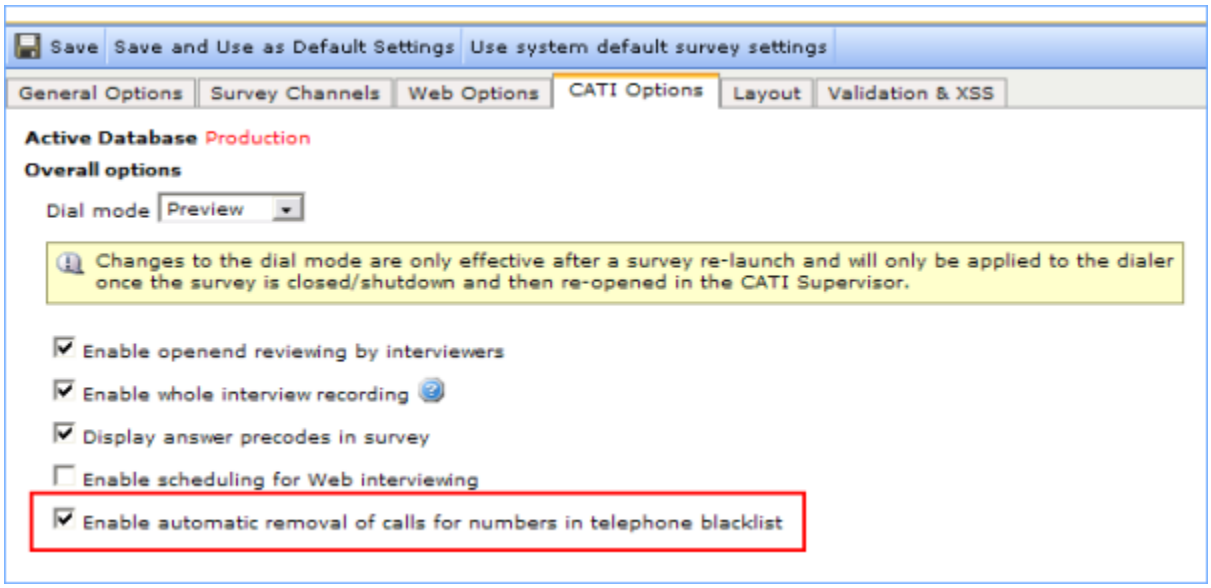


Figure 40 The Survey Settings > CATI Options tab Enable automatic removal... check-box

Numbers can be added to the blacklist as follows:

1. Via the supervisor user interface

   A supervisor can insert a number (or import a list of numbers) by going to the **Resources tab > Telephone Blacklist** option. For more information refer to the CATI supervisor guide.

2. Via the survey

   Within the survey is a function **AddToCatiBlacklist()**; use this to add the current call into the blacklist. This function can be added to a script node that will be executed if the interviewer codes a response such as 'Do not call again' on the dial outcome screen. The call should then be terminated with the extended status for black-listed numbers (code 17).

Important
Setting the extended status of a call to 17 alone is not enough to add the current telephone number into the master number Blacklist; this will just set the status of the call in the current survey. To add the number to the blacklist you must also call the AddToBlacklist() function.

9.4. Extending Appointment Expiration Timeout for the Logged-In Interviewers

You may wish to schedule return calls so that they are made by the same interviewing agent who originally made the appointment. However this can cause problems if for example the interviewing agent is unavailable when the appointment is due (they might not be working/logged in or they might be busy on another call). To help overcome this it is possible to set the expiration logic so that the system can take into account the logged-in status of the assigned interviewer. If the interviewer assigned to the appointment is not logged in, then the call can be re-assigned to other interviewers who are for example working on the same survey. Alternatively, if the interviewer is logged in but busy, then the assignment can be held to the agent for longer.

A worked example is shown below. In this case three separate sub-rules are defined for calls with the "Appointment" status.

Note: It is important that the appointment sub-rules are defined in the same order as shown in the image.

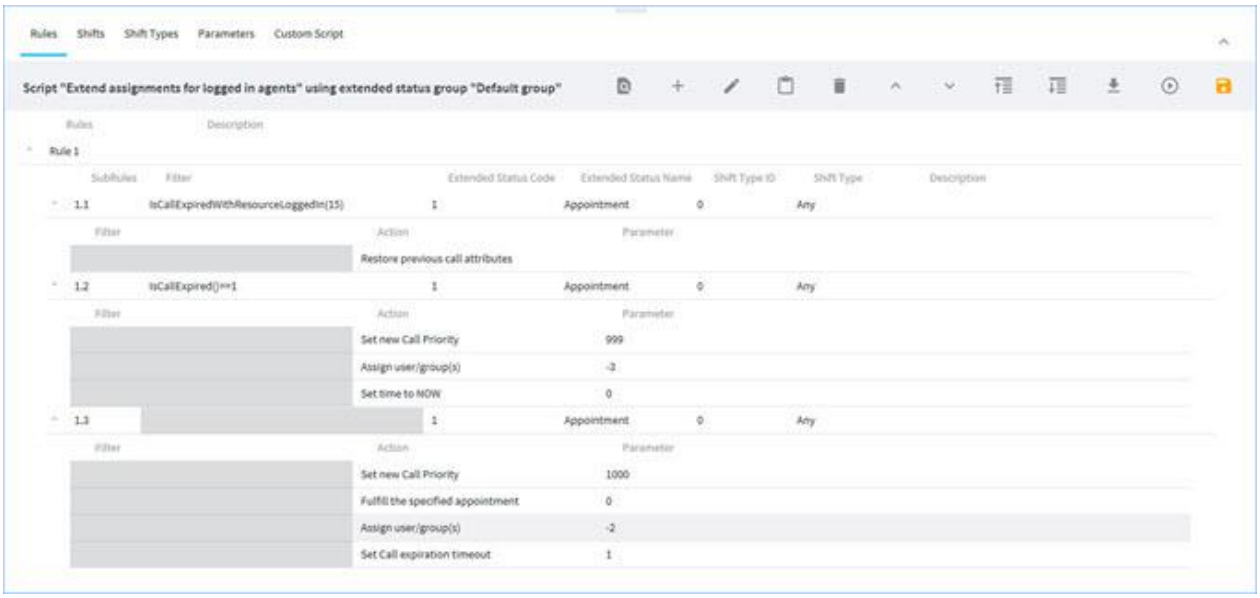


Figure 41 Example of a scheduling script used to set up the expiration time for an appointment

The three sub-rules in the image above are used to process calls with the "Appointment" extended status. Sub rules 1.1 and 1.2 use filters, so if the appointment does not meet the conditions set by these filters it is processed by sub rule 1.3.

- **Sub-Rule 1.1** handles expired appointments in the event the assigned interviewing agent is logged in but is busy with another job. To this end the "IsCallExpiredWithResourceLoggedIn" function is used in this sub rule as a filter. The function argument with the value of "15" in this example extends the appointment expiration time for another 15 minutes. During this time the system continuously checks if the status of the interviewer has changed, and if the agent becomes free then the appointed call is delivered immediately. If the extended timeout completes and the interviewer has still not picked up the appointment, then expiration sub rule 1.2 will take effect and the call will be re-assigned.

- **Sub-Rule 1.2** uses the function "IsCallExpired" as a filter. This means that if the interviewer who created the appointment is logged out at the time when the appointment expires, the system will re-assign the expired appointment so another interviewer can work with it.

    o An action “Assign user/group” with a parameter value of “-3” within the appointment sub-rule removes the appointment lock for the currently assigned user/group. Ensure that this action has the property “Filter enabled” activated.

    o The subrule should also contain two additional actions, one to set a high priority (with a value of "999" in this example) and another to make the call due with immediate effect ("Set Time to NOW" with a value of "0"). The “Filter enabled” property must also be activated for these actions.

- **Sub-Rule 1.3** sets up the appointment to be assigned to the interviewer with a short expiration of one minute. This means the appointment will expire if it is one minute overdue and the interviewer is not logged in. An action “Assign user/group” with a parameter value of “-2” within the appointment sub rule locks the call to the user/group. Other actions set the expiration period:

    o Set Call Expiration Timeout - this is the time in minutes that the system will wait for the assigned interviewing agent to pick up the appointment.

    o Additional actions are used to give the appointment a high priority value (1000 in this example) and to attempt to deliver the appointment directly on time (0 minutes before the due time).

Important
When working with a survey in Predictive dialing mode it is essential to ensure that any call that are assigned directly to an Interviewer are not scheduled to be dialed predictively. For any calls assigned to Interviewers explicitly it is therefore essential to set the dialing mode for the calls to Preview using a Set Dialing Mode action.

Once the rules have been configured, click **Save and Launch** on the tab's toolbar to activate the changes for any survey projects currently using this set of scheduling rules. The launch process will automatically save the scheduling rules in the current state without the need to perform a separate save step.

9.5. Working with Inbound Calls

There are several possible scenarios for managing survey projects with inbound calls. These scenarios are described in the following section.

Note: There is a variety of industry terms and abbreviations used when referring to caller ID and direct dial numbers but here we will be using CLI for Caller ID (Calling Line Identification) and DDI for Direct Dial In (also known as DID for Direct Inward Dialing).

For all inbound scenarios that involve the dialer, it is necessary to specify a DDI number association with a survey and for the survey schedule to include a rule for calls with the status 'Inbound Call'. As a minimum, this rule must contain an 'Accept inbound call' action.

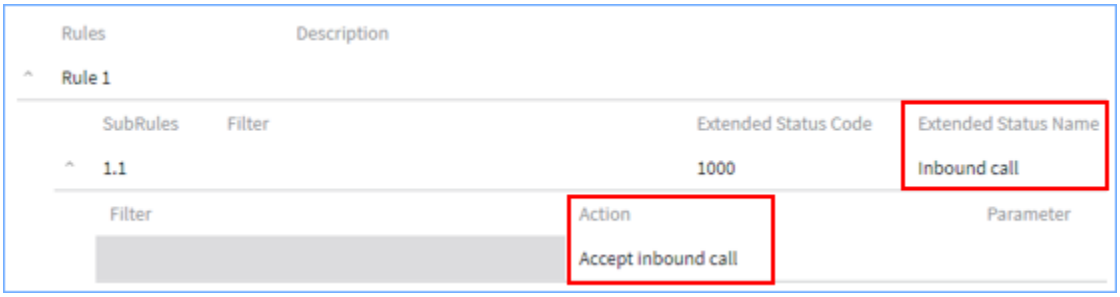


Figure 42 Example of a rule with an 'Accept inbound call' action

9.5.1. Use Case Scenarios

The following topics provide a number of use-cases that may be useful in various scenarios.

9.5.1.1. Blended Inbound/Outbound with Automatic Respondent Identification

In a blended Inbound/Outbound scenario interviewers are kept busy working on outbound calls but will be fed inbound calls when they are available. Inbound callers are automatically played a message and then queued until the dialer is able to allocate them to an available interviewer[1]. Whenever an inbound call is delivered to the interviewer the header and footer of the application will turn green to highlight that the current call is an inbound call.

DDI Number Association

A DDI number is associated with the survey so that the system can automatically route the inbound caller to the applicable survey. DDI number associations are set up in the Supervisor UI (refer to the Supervisor User Guide for more information).

Note: A DDI number can only be directly associated with one survey at a time, but it is possible to associate multiple DDI numbers to the same survey (see Language Preference). It is possible to associate one DDI number with multiple surveys, however this must be managed indirectly using a master/linked survey set up (see Blended Inbound/Outbound with manual respondent identification).

[1]Interviewers become available at the end of each call attempt before the next call is connected.

CLI Number Matching

The default behavior of the system is to look at the CLI number of the inbound caller and then attempt to match it to an identical number in the pre-loaded sample contact list of the survey (comparing the CLI number with numbers pre-loaded into the systems TelephoneNumber field). If a match is found then the inbound call will be served up directly to the first available agent (with both the survey and survey record having been automatically identified by the combination of the DDI and CLI numbers).

If no match can be found, for example because the respondent uses an unknown or withheld number to call in with or because the survey has no pre-loaded sample, then the system can dynamically create a new interview record.

Note: It is recommended to log this caller's ID to the freshly created record so that they will automatically be returned to the same record in future call attempts. This can be achieved using a Custom Script function in the Inbound Call rule of the survey schedule.

The CLI matching behavior can be adjusted on a survey basis via the 'Inbound Caller ID Matching' setting on the survey general tab. This setting has the following additional options…

1. Match CLI without creating a record.

   The system will attempt to match the CLI with existing sample records but if no match is found it will not create new records.

2. Do not match CLI.

   The system will always create a new record for an inbound call.

Having created a new record there are two possible ways to proceed:

1. The interview continues in the freshly created record, but there can be no link back to existing sample data.
2. Some questions are asked of the respondent to establish their identity, then a search is performed against the sample contacts. The interviewer can select the appropriate interview record and begin the linked interview (see Identifying and Linking to Surveys and Interviews on page 35 for more information).

Interviewer Assignment

For an inbound call to be delivered to an interviewer, they must have a valid assignment to the record (either a survey level assignment or an explicit record level assignment). It is possible for inbound calls to be delivered for a survey which the interviewer is not currently making outbound calls on. This is achieved by assigning interviewers via groups which have the property 'allow inbound calls from other surveys' enabled.

Language preference

It is possible to assign more than one DDI number to the same survey. This can be useful when there's a requirement to serve interviews up to inbound callers based on some key information in the sample data. A prime example of this is whereby the language preference of the caller is known in advance and so can be pre-loaded in the sample data. A DDI number can then be used for each language so that the inbound calls are automatically served up to the applicable language based interviewer groups. For more information on how to set this up, refer to the separate CATI Supervisor Guide, section 'Routing inbound calls through the use of a scheduling script'.

Another option is to use the Internal Call Transfer functionality to re-assign the inbound call to an appropriate language speaking group.

9.5.1.2. Blended Inbound/Outbound Calls with One DDI to Multiple Surveys

Although only one survey can directly be associated to a DDI it is possible to associate multiple surveys to a single DDI through the use of linked surveys. The basic concept here is that all inbound calls are initially delivered into a master survey and from there the interviewer is able to ask some questions of the respondent and then perform a search which will return a matching list of surveys and interview records. Selecting a survey/record will pass the interviewer directly into the appropriate linked survey and interview record. The link can either be made to another record in the same survey or to another record in a separate survey.

9.5.1.3. Inbound IVR with Transfer to Live Interviewers

On supported systems[2] it is possible to have inbound calls serviced by IVR. This could be a complete interview experience using IVR to present questions and capture responses, or IVR could be used to automate screening or routing of inbound calls to suitable surveys, interview records or interviewer groups.

9.5.1.4. Inbound Only with Automatic Call Routing

The system tries to work in outbound mode by default, so this means that as soon as an interviewer is logged in and they have valid calls assigned to them the system will begin to place outbound calls to the available sample contacts. If you do not want the interviewer to be working on outbound calls (that is you want them to be fully reserved and waiting for inbound calls) then it is necessary to assign the time to call property for the assigned calls to have a date/time in the future. When the interviewer logs into the system they will then just see a spinning icon until an inbound call is received.

9.5.1.5. Inbound Only with Manual Survey and Record Searching

In this scenario, inbound calls are routed to the interviewers without going through the CATI system or dialer. So for example calls are answered using regular desk phones and transferred using the functionality of the phone system. The interviewers will be working with the manual 'Task Choice' property which means they will be presented with a selection of all the open surveys and interview records for which they have an assignment upon logging in. To locate a contact for which sample has been pre-loaded, the interviewer can select a survey and then use the column heading filters to find the caller.

Note that it is possible to add additional search columns based on survey specific variables (refer to the separate CATI Supervisor Guide for details).

It might also be desirable to allow new interview records to be created on the fly, for example when no pre-loaded sample contacts are used or if the interviewer is unable to locate the caller using the search options but there is still a wish to conduct an interview. To support this the CATI survey setting 'Allow dynamic creation of new respondent records' must be enabled, which in turn will enable the **Create New Interview** button on the interviewer's manual selection screen.

Constraints of this method

- As the inbound calls are not routed through the dialer, dialer-based features will not be available, for example call monitoring, recording or transfers.
- Search filtering in the interviewer's manual selection screen only supports the application of filters within a given survey; it is not possible to search across all or multiple surveys in a single search operation.

9.5.2. Identifying and Linking to Surveys and Interviews

It is possible to set up an inbound survey so that it will present a custom set of questions that will be used to identify the caller against an existing respondent contact record. It could be to locate a contact record in the same survey or it could be to locate a record in another survey. Either way the basic process is the same; the interviewer feeds in the answers to the identifying questions and then performs a search of available sample records. The results of the search are displayed to the interviewer, who can then select a record to open a linked interview. At this point the interviewer will switch from the current interview record into the selected one, potentially switching to a different survey at the same time.

Note: You can move back to the master record by using the function `SetNextCatiInterviewToPrevious()`.

A summary of applicable functions:

- `GetCatiLinkedInterviews` returns a list of "parent" interview records.
- `GetCatiInterviews` returns an array of objects which should be placed into an open text list variable.

[2]A supporting system is one which has an integrated dialer that provides IVR capability. Usage of IVR is also dependent upon having license for enough concurrent virtual IVR agents to support the expected volume of inbound calls during peak times.

Then in turn a single variable is used to show the results contained in the open text list so that the user is able to select a specific record.

After selecting a single answer we have to parse the open text list item string to get the project id and resp id.

- `SetNextCatiInterview` determines which survey and interview record to start.
- `SetNextCatiInterviewToPrevious` returns interviewing activity back to the previous survey record.

9.6. Restricting Call Attempts with Scheduling Parameters

You may wish to limit the number of call attempts. The CATI system has a built-in function “CallAttemptCount” to record the number of call attempts to a number. You can add a rule to the survey schedule so that the number of attempts is limited, thereby preventing further call attempts being scheduled. This could simply be hard coded in the scheduling rule as:

```
GetRespondentValue('CallAttemptCount') >=5
```

Or it could be parametrized so that a CATI Supervisor can control the number of call attempts allowed by adjusting the parameter without the need to edit the actual schedule. In this case it is recommended to use the `GetParamNumeric` function; this will provide an integer number defining a threshold value that can be used to limit call attempts.

An example of the GetParamNumeric function usage:

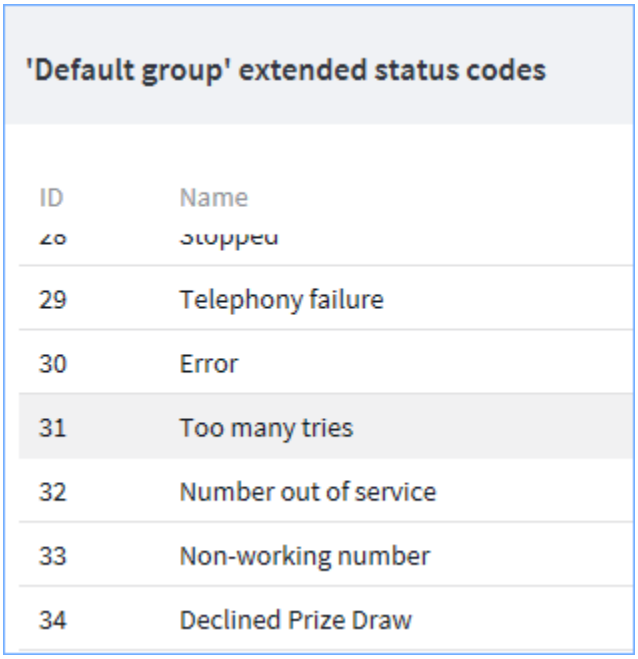
```
GetRespondentValue('CallAttemptCount') >= GetParamNumeric('MaxCall')
```

To achieve this:

1. Go to **Resources > Extended Status Codes** and choose a suitable spare status that can be relabeled.

Tip:
By modifying the ‘Default group’ this custom status could become a new default status for all surveys (assuming all surveys will stick with the Default group or a duplicate of it).

For example, here we have chosen the Custom1 status and renamed it to ‘Too many tries’…



'Default group' extended status codes

ID	Name
28	Stopped
29	Telephony failure
30	Error
31	Too many tries
32	Number out of service
33	Non-working number
34	Declined Prize Draw

Figure 43 Extended status codes

2. Go to the Scheduling tab and choose the schedule you wish to modify.

Tip:
The AllHours schedule is the default schedule applied to all surveys, so if this change is added to AllHours then it can be used in all surveys (assuming all surveys will stick with the default schedule or a duplicate of it).

3. Create a new sub-rule, ticking **Filter enabled**, and then insert the following filter expression:
```
GetRespondentValue('CallAttemptCount') >= GetParamNumeric('MaxCall')
```

4. Set the Extended Status to **Any** and then move the new rule up to the top of the rule list.

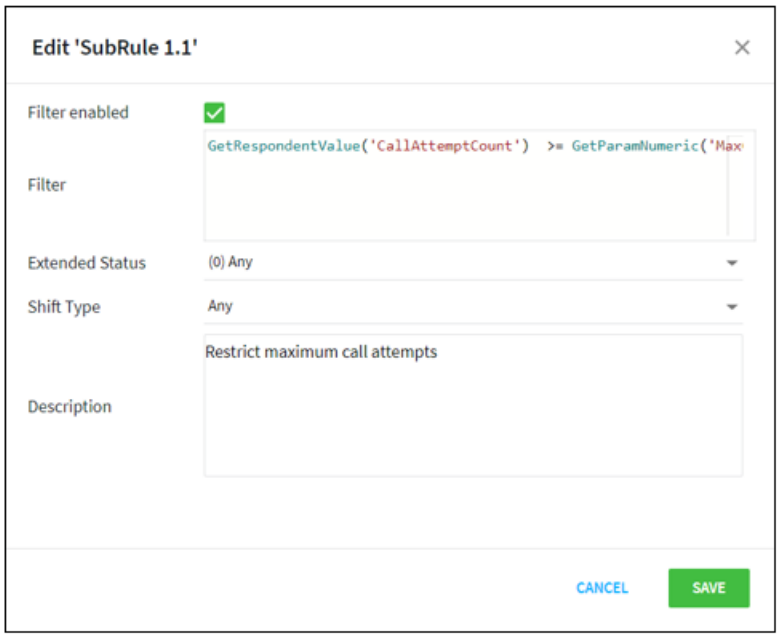


Figure 44 Creating the sub-rule

5. Add an action to the sub-rule to set the extended status to 31 (or whatever status code was chosen in step 1), and another action to suspend the interview, making sure that both actions have the 'Action enabled' and 'Filter enabled' options enabled.

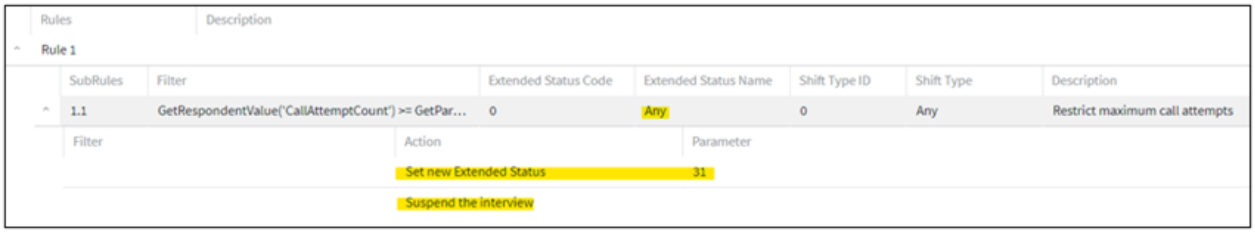


Figure 45 Adding the actions

6. Declare the parameter 'MaxCall'. To do this switch to the Parameters tab and then click on the button to add a new parameter. Set the name to 'MaxCalls' (as used in this example) and enter a description that will make it obvious to the Supervisor what the parameter is for (for example Maximum number of attempts).

   Set the type to numeric and enter a default value. This will be the default limit for call attempts in all surveys working from this schedule, unless overridden by the Supervisor.

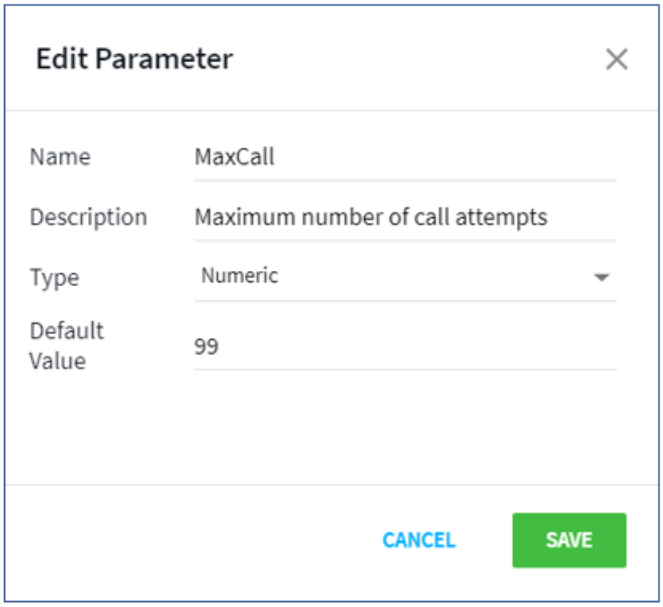


Figure 46 Setting up the parameter

7. Click **Add** to add the parameter to the sub-rule.
8. Launch the scheduling script. This will automatically save the changes and put them live.

Note: The call attempt restriction can now be adjusted (overridden) for any survey using the modified scheduling rules via the survey specific 'Scheduling Parameters' tab.

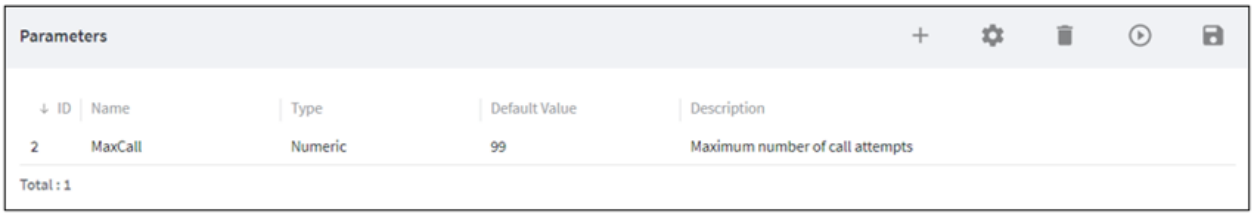


Figure 47 The Parameters tab with the new parameter

9.7. Call Transfers

The CATI system allows two types of call transfer: Internal and External.

- ***Internal call transfer*** allows the interviewer to transfer a call to another interviewer working in the same company. The Internal call transfer assumes that the call is transferred not to a particular interviewer but rather to one of the groups of interviewers to which the survey is assigned. The system then automatically picks the first available interviewer belonging to this group. Suitable interviewer groups with the appropriate skill types can therefore be used to facilitate internal transfers. To prepare for an internal call transfer operation the supervisor must:
    - Check the system setting and the corresponding UI option on the Interviewer Console tab which enable the **Internal Transfer** button on the Interviewer Console toolbar.
    - Check if the survey is assigned to at least one group of interviewers.

- Depending on the situation, enable for all or only for selected interviewer groups either the "Allow Call transfer" option only (if transferred calls are planned from a survey with which all interviewers currently work) or, additionally, the "Allow transferred calls from other surveys" option (if some transferred calls will come from surveys into which group members are not currently logged in).
- Choose the default call transfer mode (Blind or Warm).

- ***External call transfer*** means a call can be transferred to any telephone number which is specified and assigned in the CATI Supervisor External Transfer list by a supervisor. To prepare for an external call transfer operation the supervisor must:
  - Specify the corresponding target telephone number(s) in the External Transfer list (the Resources menu).
  - Check the system setting and the corresponding UI option on the Interviewer Console tab which enable the **External Transfer** button on the Interviewer Console toolbar.
  - Choose the default call transfer mode (Blind or Warm).

The CATI Supervisor module provides UI controls (available for supervisors with administrator rights) that enable or disable the Interviewer Console call transfer buttons. These options are located on the Interviewer Console tab (Settings section of the Administration menu), see the illustration below. When a system setting is turned off, the corresponding check box in the GUI is hidden. Two options are available: allow/disallow the Internal Call transfer button and allow/disallow the External Call transfer button.

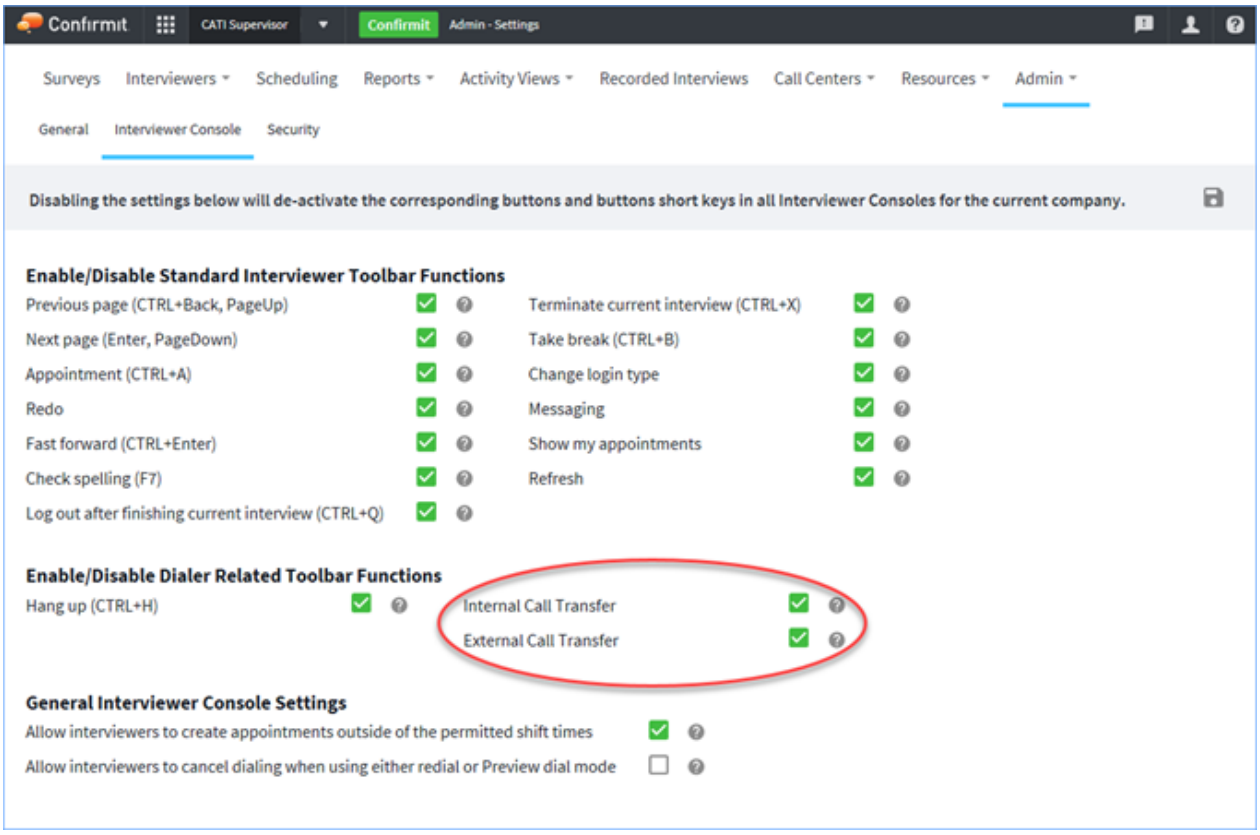


Figure 48 The Interviewer Console tab

The supervisors without Administrator rights can allow/disallow internal call transfers within a single survey - the "Allow call transfers" option - and allow/disallow internal call transfer from a survey into which the interviewer-transferee is not logged in - the "Allow transferred calls from other surveys" option. These options are located on the Properties tab of the Interviewer group Properties view (see the illustration below). The "Allow transferred calls from other surveys" option becomes available only when the "Allow call transfers" option is chosen.

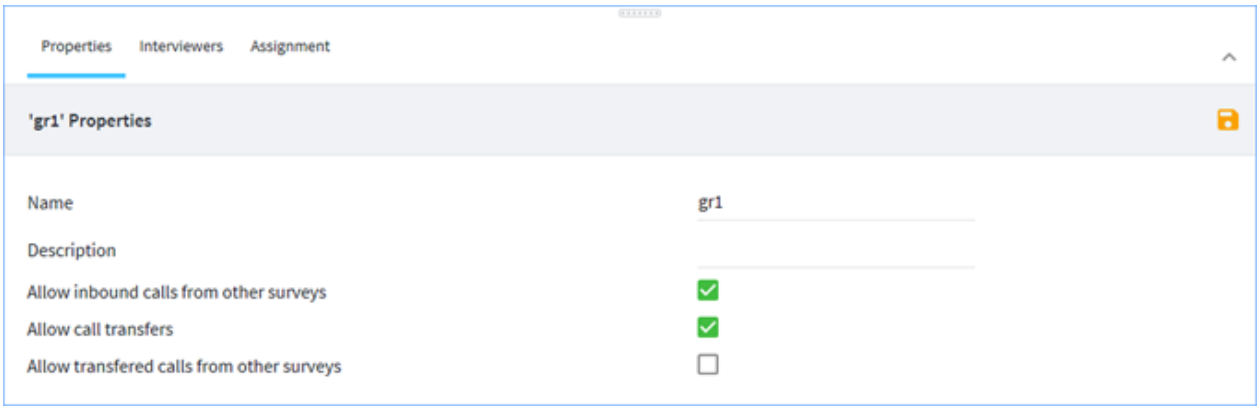


Figure 49 The Properties tab

The supervisor must specify which type of call transfer is to be initialized when the **Call Transfer** buttons on the Interviewer Console toolbar are pressed. A call transfer can be either of a Blind or Warm type.

- A "**Blind**" call transfer supposes no interaction of the initial interviewer with the target interviewer - the initial interviewer is disconnected from the call as soon as he presses the **Call Transfer** button. The system then takes over the transferred call and handles it. The initial interviewer is then free to receive another call.
- A "**Warm**" call transfer means the initial interviewer contacts the target transferee to conduct a conversation prior to transferring the call. The CATI system provides the initial interviewer with an option of establishing independent contact with a respondent, a target interviewer, or holding a three-way conversation with both of them.

The type of call transfer which is initiated by default when a **Call Transfer** button is pressed in the Interviewer Console depends on the option specified for the survey (the call transfer options are available on the Survey View > General tab). For the Internal call transfer it can be "Warm", "Blind" or "Off". If the "Off" option is selected in the CATI Supervisor module for Internal Call Transfers the **Internal Call Transfer** button on the Interviewer Console toolbar is grayed out. For the External call transfer it can be only "Warm" or "Blind".

The illustration below shows that the "Warm" transfer type is currently selected from the drop-down lists both for the Internal and External call transfer.

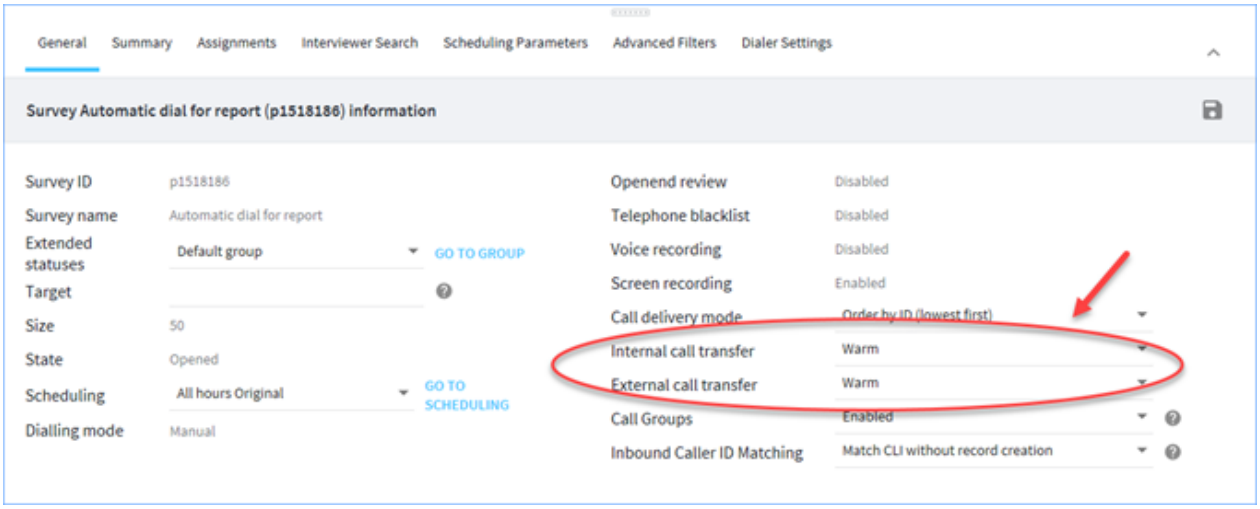


Figure 50 The "Warm" transfer type is currently selected

Only already started interviews (calls) can be transferred to another person. The interviewer should keep in mind that once a call transfer is initiated in the blind mode it cannot be canceled.

Another method to initialize the call transfer procedure is to add a corresponding command to the "Telephony" node in the questionnaire tree to start the internal or external call transfer operation at that point. In this case the call transfer is initiated in the blind mode only. This method can be useful for example when the interview is conducted by an IVR agent.

The commands that start the call transfer procedure are the following:

- InternalTransfer
- ExternalTransfer

The call transfer operation initiated in this way is the same as a call transfer operation initiated by pressing a corresponding button in the Interviewer Console. This means that all appropriate operation parameters such as the interviewer group assigned for transfer, transfer timeout value etc. are used in this case.

There is also a system setting **IVR.TransferedTimeout**. This regulates how long the system is to wait for a transferred call to be connected in the event it is transferred from an IVR agent to a live agent. The timeout value is specified in seconds. If the transfer to a live agent is not completed within the given timeout period it will be returned to the IVR.

10. Creating and Using Scheduling Scripts

A scheduling script is a set of rules which contain instructions detailing what the CATI Supervisor module is to do with a specific call and under which conditions.

You can create and add as many scheduling scripts to the CATI Supervisor module as you need. All these scripts will be available company-wide, and they can be used with any survey that is conducted by the company. One scheduling script can be used with a number of surveys, but each survey can use only one scheduling script.

10.1. Writing Custom Scheduling Script Code

In the event you wish the scheduling script to perform operations which cannot be configured using the regular CATI Supervisor functions available through the user interface, you can write the script code manually. This requires thorough knowledge of the JavaScript.NET language.

Note: The script code you enter using this tab is executed only when the "Run specified script" action is enabled.

To create a custom scheduling script code manually:

1. With the scheduling script opened in the View mode, choose the Custom Script tab in the bottom frame.

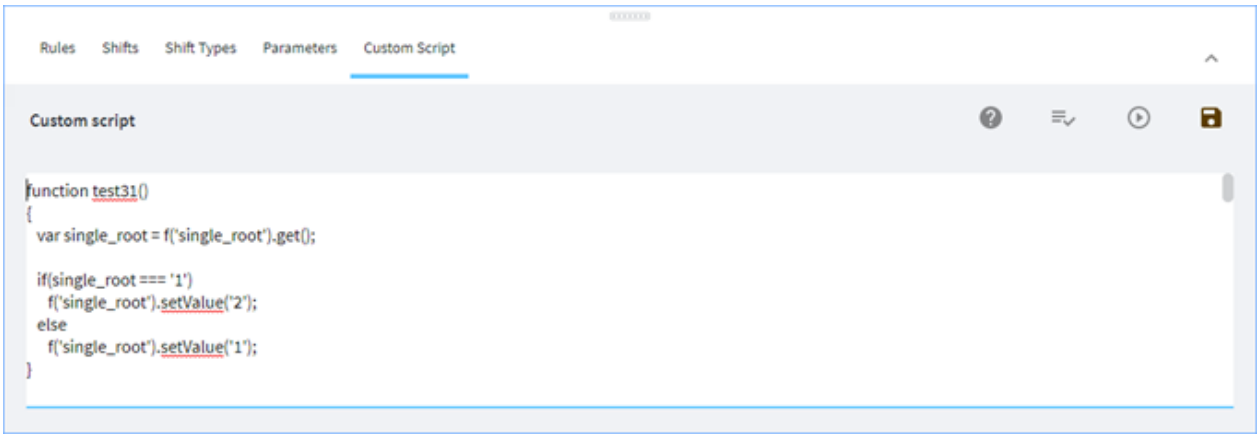


Figure 51 Adding a custom script

   This tab contains a single text field enabling you to enter the script code.

2. Enter the script code, using JavaScript.NET, in the text field.

   You can also paste the clipboard contents into this field.

3. Click the **Check** button in the frames toolbar to check the code for correctness.

A Scripting Reference for examples of custom script codes is available (see Accessing the Call Object in Custom Scripting on page 42 for more information)

10.1.1. Accessing the Call Object in Custom Scripting

This topic describes how the call object can be accessed in custom scripting within CATI scheduling. The functionality is available through the "Scheduling" object, which is available in custom scripts. The "Scheduling" object has the following properties:

Type	Name	Description

BvSurveyEntity	Survey	This object provides data for the survey which contains the current interview. ReadOnly.
BvInterviewEntity	Interview	This object provides data for interview which is scheduled. Read/Write.
BvCallEntity	LastCall	If the scheduling script is run for interview which previously had a call, then this object contains info about the call, otherwise null. ReadOnly.
BvCallEntity	NewCall	If the scheduling script creates a call, this object provides info about the new call. If the object is null, call will not be created upon scheduling completion. Read/Write.
DateTime	Time	Scheduling time.
ShiftService	Shifts	This object provides for shift functionality.

A new call is initialized within the scheduling object when it is used via a custom scheduling script: `CallShouldBeCreated()`. Once initialized, **Scheduling.NewCall** will be initialized and available. To cancel creation of the new call, set **Scheduling.NewCall** to Null.

Object breakdown:

- The **BvSurveyEntity** object provides access to survey data as follows:

Type	Name	Description
Int	SID	Internal object ID
String	Name	Project ID
String	Description	Project name
Int	ScheduleID	Scheduling script ID

- The **BvInterviewEntity** object provides access to interview data as follows:

Type	Name	Description
Int	ID	ID of interview
String	TelephoneNumber	Respondent telephone number
String	RespondentName	Respondent name
Int	TimezoneID	ID of respondent timezone
Int	TransientState	Extendend status
DateTime	LastCallTime	Last call time
Int	LastCallPersonSID	User ID of last interview
byte	DialingMode	Dialing mode

- The **BvCallEntity** object provides access to call data as follows:

Type	Name
int	CallID

int	SurveySID
int	InterviewID
int	Phase
int	RoleID
int	ShiftID
DateTime	TimeInShift
DateTime	TimeToExpire
int	Priority
int	Resource
int	ApptID
int	ResourceType
Guid	RuleNumber

Shift functionality is available through the ShiftService object supporting methods when working with shifts:

- MatchingShift GetExactShift(DateTime utcNowTime, int tzID)
- MatchingShift GetMatchingShift(DateTime utcTime, int tzID)
- DateTime GetMatchingTime(DateTime utcNowTime, int tzID)
- MatchingShift GetNextShift(MatchingShift currentShift, int tzID)
- MatchingShift GetNextShift(MatchingShift currentShift, int tzID, out int countSkipShifts)
- MatchingShift GetNextShiftByID(DateTime utcTime, int tzID, int scriptShiftID)
- MatchingShift GetNextShiftOfSpecifiedType(DateTime utcTime, int tzID, int scriptShiftTypeID)
- MatchingShift GetShiftAfterNumberOfMinutes(DateTime utcNowTime, int tzID, int countMinutes)
- MatchingShift GetShiftAfterNumberOfShifts(DateTime utcNowTime, int tzID, int numberOfShifts)
- MatchingShift GetShiftAfterNumberOfShifts(MatchingShift curentShift, int tzID, int numberOfShifts, bool isTakingExclusionIntoAccount)

Custom code examples:

1. Custom script creates new call with priority 10

```
function ScriptFunction()
{
CallShouldBeCreated();
Scheduling.NewCall.Priority = 10;
}
```

2. Custom script creates new call with priority which is taken from number variable with 'num_prior' name

```
function ScriptFunction()
{
CallShouldBeCreated();
Scheduling.NewCall.Priority = f("num_prior").get();
}
```

3. Custom script creates new call with time to call on next shift

```
function ScriptFunction()
{
CallShouldBeCreated();
var shift = Scheduling.Shifts.GetMatchingShift( Scheduling.Time,
1/*Timezone*/ );
shift = Scheduling.Shifts.GetNextShift( shift, 1/*Timezone*/ );
Scheduling.NewCall.ShiftID = shift.ShiftTypeID;
Scheduling.NewCall.TimeInShift = shift.StartDate;
}
```

4. Custom script creates new call and assigns the interviewer with the ID from variable ‘inter’

```
function ScriptFunction()
{
CallShouldBeCreated();
var name = f("inter").get();
Scheduling.NewCall.Resource = PersonRepository.GetByName( name ).SID;
}
```

5. Custom script writes the interviewer's SID to variable 'history' (for all interviewers who have conducted an interview)

```
function ScriptFunction()
{
var history= f("history").get();
var name : String = "";
if( Scheduling.Interview.LastCallPersonSID != 0 )
{
var person =
PersonRepository.GetByID(Scheduling.Interview.LastCallPersonSID);
if( person != null )
if( String.InNullOrEmpty(history))
history = person.name;
else if( !String.Split( history, ',' ).Any( x => x == person.name ) )
history = history + "," + person.name;
else
return;
f("history").setValue(history);
}
}
}
```

10.1.2. Action Examples

This section describes how the various actions can be used.

To restore previous call attributes:

You may need to restore the attributes of a call that were removed as the result of some type of event. This may occur for example when a call is assigned the "Returned Not Dialed" or "Returned Dialer Expired" extended status where calls are returned from a predictive dialer call queue. A situation may arise where you need to re-use these call attributes, so the CATI Supervisor module provides the means to restore them.

Use the "Restore previous call attributes" action in the appropriate scheduling script, and a similar function in a custom script, to achieve this.

When a scheduling or custom script action that contains the "Restore previous call attributes" function is used, all attributes of the calls which match the criteria specified in the script (by applying filters and adding the required action to the desired sub-rule) will be restored. The main difference between the two actions is that custom script allows you to create more complex and precise filters.

To restore call attributes using a scheduling script action:

1. Create a new or edit an existing scheduling script related to the survey that contains the calls for which you wish to restore the attributes.
2. Locate the desired rule or sub-rule you would like to add the "Restore..." action to.

Note the sequence in which rules, sub-rules and sub-rule actions are executed; restoring of attributes should be performed only for the calls which require this action.

3. Add the "Restore previous call attributes" action to the desired scheduling script sub-rule.

You can create a filter for the action you add. In this way you can specify the condition which the call must match for its attributes to be restored (this can be a particular extended status or any other parameter value which triggers the "Restore..." action).

An example of the scheduling script using the "Restore previous call attributes" action is shown below.

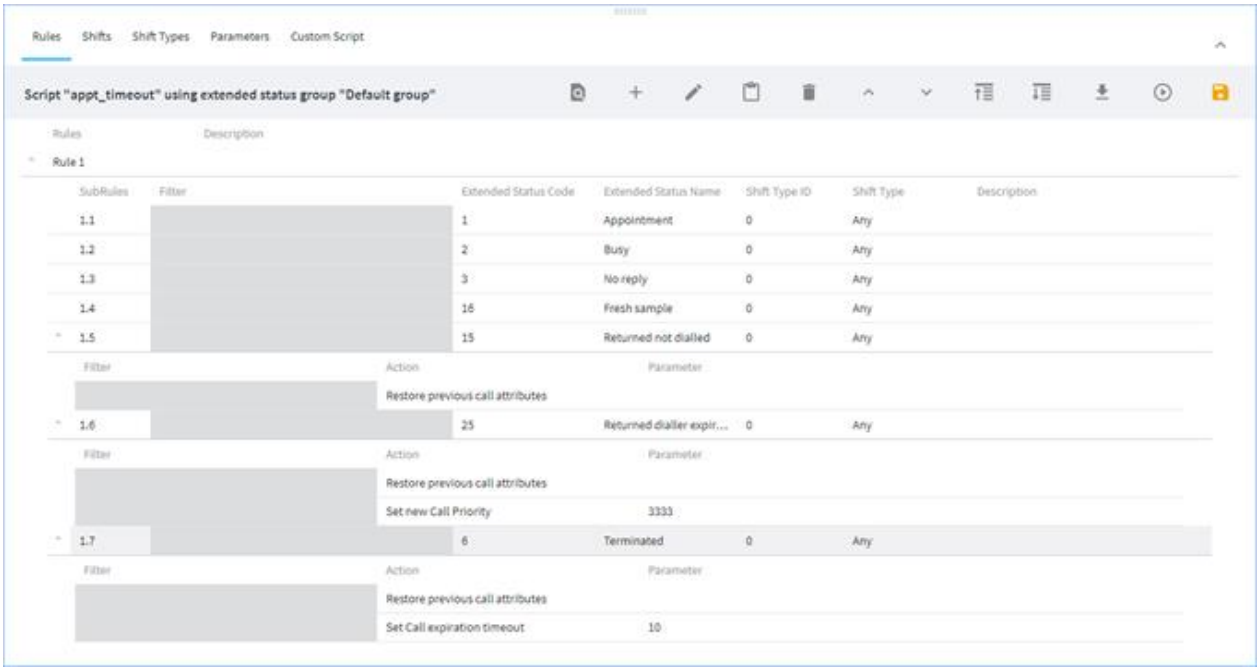


Figure 52 Example of scheduling script containing the "Restore previous call attributes" action

Note sub-rules 1.5, 1.6 and 1.7. These sub-rules use the "Restore previous call attributes" action to restore attributes of calls after they were assigned different extended statuses. The action is applied to calls with the "Returned Not Dialed", "Returned Dialer Expired" and "Terminated" statuses. In this particular case, other actions will be applied to these calls after their attributes are restored as the restored call attributes need to be edited to suit a new interviewing round.

The "Restore previous call attributes" action has no configurable parameters so this will not need to be adjusted.

4. Save the scheduling script.

To restore call attributes using a custom script function:

1. Insert the following code snippet at any location in the custom script you are editing, and save the script. The custom script may consist of this function alone.

```
function MyFunction()
{
ExecuteAction(Actions.RestorePreviousCallState);
}
```

In the example the function specified in the script is called "MyFunction".

2. Specify all the required conditions (appropriate variable names, other parameter values which will trigger the function execution etc.) by adding them to the script.

3. Save the custom script.
4. Add the Run Custom Script action to the scheduling script and specify the name of the function (MyFunction in the example above) as a parameter of the Run Custom Script action.

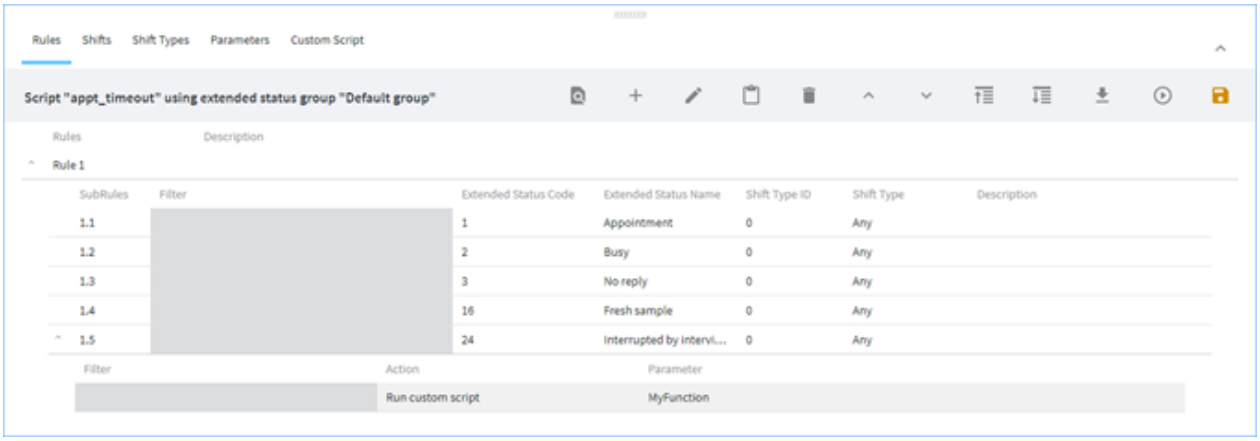


Figure 53 Example of the scheduling script containing a reference to the custom script

Note that sub-rule 1.5 uses the action which calls the custom script with the name "MyFunction". Take into account the execution sequence of rule, sub-rule and action - the Custom Script action which calls the appropriate custom script and executes the RestorePreviousCallState function should be specified for the appropriate sub-rule (all other existing rules/subrules/actions should be taken into account).

10.2. Extending Appointment Expiration Time for Explicitly Assigned Interviewers

The CATI Supervisor module allows you to set an appointment expiration timeout so that an appointment could still be delivered to a specified user while their session remains active even if they are busy when the appointment is due.

Normally, when an appointment is due but the assigned user is not free (not logged in or working on another interview), the system removes the explicit user assignment and changes properties of the call according to scheduling script rules. However, if the interviewer is known to be logged in, the supervisor can instruct the system to wait for some time without removing the explicit user assignment and changing properties of the call. The system will then wait for some time for the status of the interviewer to change. This is done by way of adding a subrule to a corresponding scheduling script.

A special function called `IsCallExpiredWithResourceLoggedIn` is used as a filter in the scheduling script subrule to this end. This function uses the "`(int timeout)`" parameter to set timeout for the call expiration time. This parameter accepts an integer value and it means "Timeout interval in minutes".

A more detailed description of the technique used to set up appointment expiration timeout can be found in the "CATI Administrator Manual".

When the timeout countdown starts, the system (in short increments of time) checks the status of the assigned user. If they become available, the system instantly delivers the appointed call. When the timeout interval ends, and the status of the user does not change, or when they log out of the system, the current scheduling script changes the properties of the call according to the specified rules.

The illustration below shows a scheduling script configured to set a timeout for an appointment. The function parameter is set to "10", which indicates the system will check the interviewer status continuously for 10 minutes until either the interviewer becomes free to receive a call, or logs out. In the first case the system delivers an appointed call to the interviewer, in the second it will change the properties of the call according to any actions which are added to this scheduling script. The same change to the call properties will be carried out if the allotted 10 minutes complete and the status of the interviewer remains unavailable.

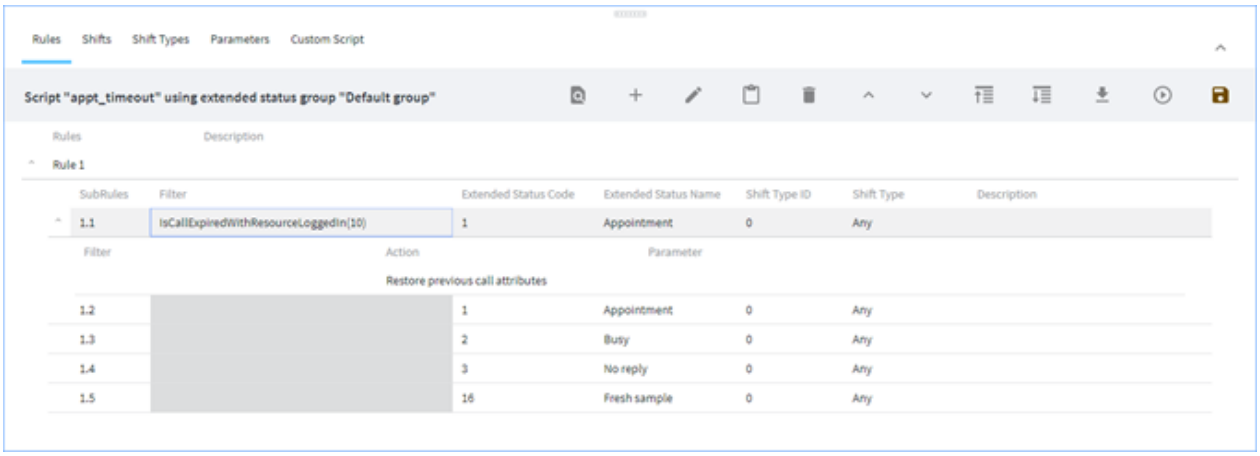


Figure 54 Scheduling script containing a subrule for setting the appointment timeout

10.3. Inbound Call Support using Scheduling Script

The CATI Supervisor module can handle inbound calls when it is connected to an integrated dialer which supports inbound call processing technology.

Inbound call processing technology relies on the DDI (Direct Dial-In) number list, and a scheduling script which may involve specific functions and methods used in custom scripting to handle inbound calls. The work of the dialer must also be configured, however most of the dialer's functionality is handled in the background and it is not usually necessary for the supervisor to edit any of these parameters.

When a call is created, the system runs a scheduling script to process inbound calls. This script must be created beforehand and must contain the "Accept Inbound Call" action. Other parameters for this script are optional.

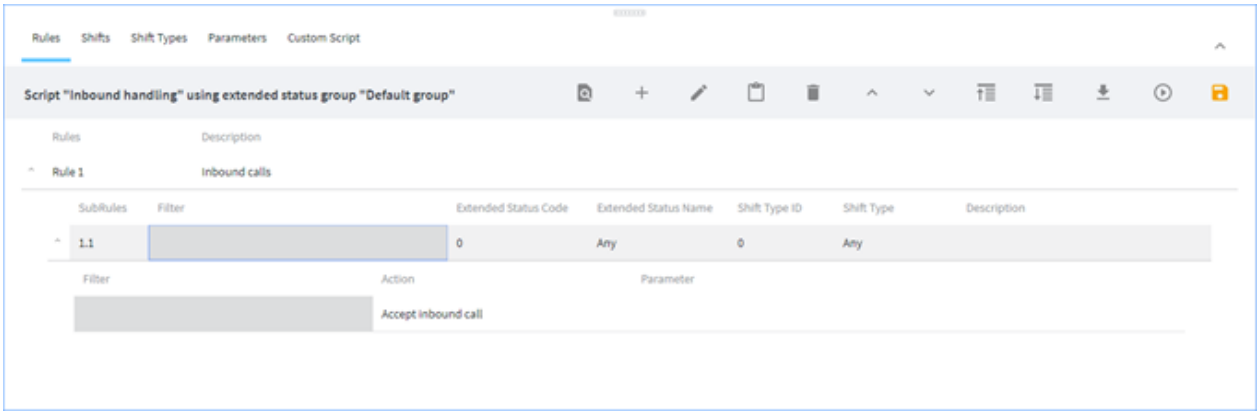


Figure 55 Example of a scheduling script for handling inbound calls

Calls have a property called the "CLI Number" (an identified telephone number from which the inbound call comes). The CLI number can be used in scheduling scripts to filter calls. A scheduling script for accepting inbound calls can contain other rules and actions which help in processing an inbound call.

It is recommended to use the "Increment priority" action for inbound calls to forcibly increase the Priority value so that the call can be delivered to an interviewer as fast as possible. This is especially important when calls are delivered in the "non-Predictive" mode.

A range of extended statuses are available that can be assigned to inbound calls when they are detected and as a result of different outcomes. These extended statuses can be used in scheduling scripts as filters for handling calls.

The CLI (Caller ID) of a calling respondent is a call parameter that can also be used as a filter in scheduling scripts.

Routing inbound calls using a scheduling script:

The CATI system provides a function that helps setting up a filter for a scheduling rule action that can route inbound calls.

The **Scheduling.DdiNumber** is a function that can be used in a scheduling rule for inbound calls that allows actions to be filtered based on the DDI number dialed to enter a survey (where more than one DDI number may be assigned to the same survey).

This example shows how to assign language groups dynamically based on the DDI number.

To instruct the system to route a calling respondent to the appropriate survey we should add (for example, one DDI number is intended for people who speak English language - they are to take an English language questionnaire, and another one - for people who speak French - they are to take a French language questionnaire. The English language questionnaire is associated with DDI number "12345" while the French language questionnaire is associated with DDI number "67890".

The supervisor should add the "Assign User/Groups" action twice to a corresponding scheduling script. Both actions should have the "Filter Enabled" option turned on!

The first added action should use the **Scheduling.DdiNumber** function telling the action should be applied to a respondent who dials a DDI number equaling '12345'. The Value for this action should be the ID of an interviewer or an interviewer group, "10" in this example, who is supposed to conduct interviews in English. If the system encounters the DDI number, it routes the call to a corresponding interviewer. An example of such an action is shown below.

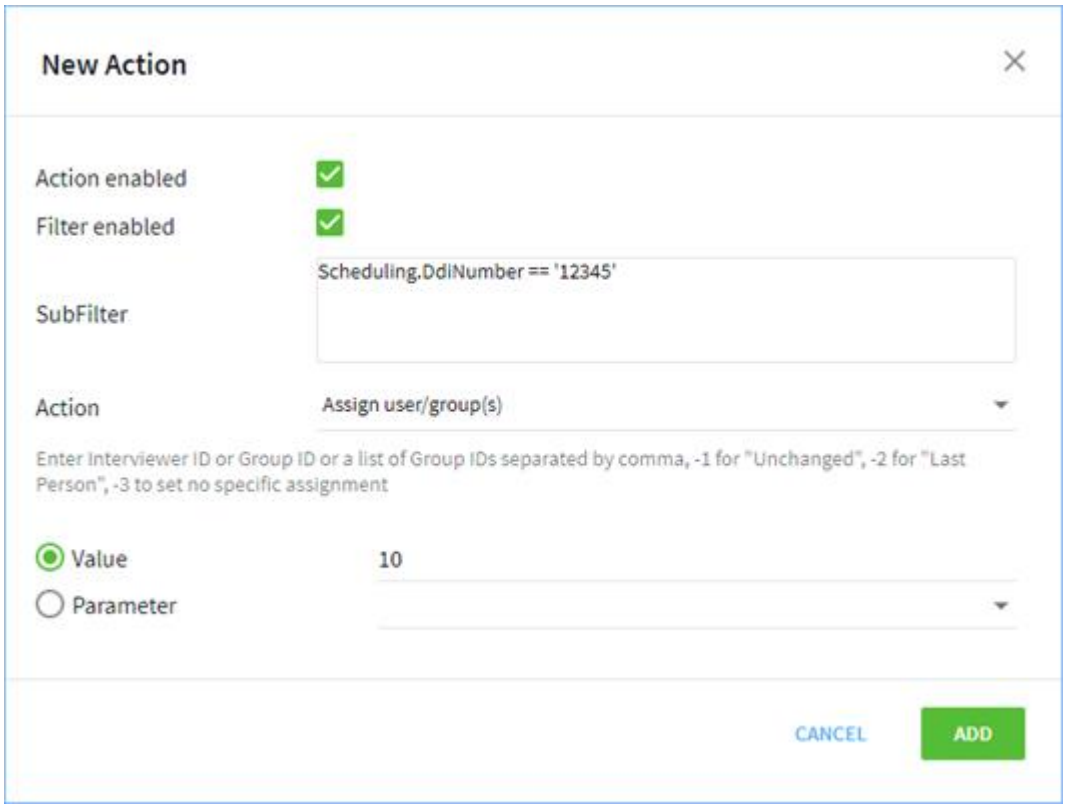


Figure 56 A scheduling rule action routing respondents based on a DDI number

The second action the supervisor should add should be applied to a respondent who dials a DDI number equaling '67890'. The Value for this action should be the ID of an interviewer or an interviewer group, "11" in this example, who is supposed to conduct interviews in French. An example of such an action is shown below.

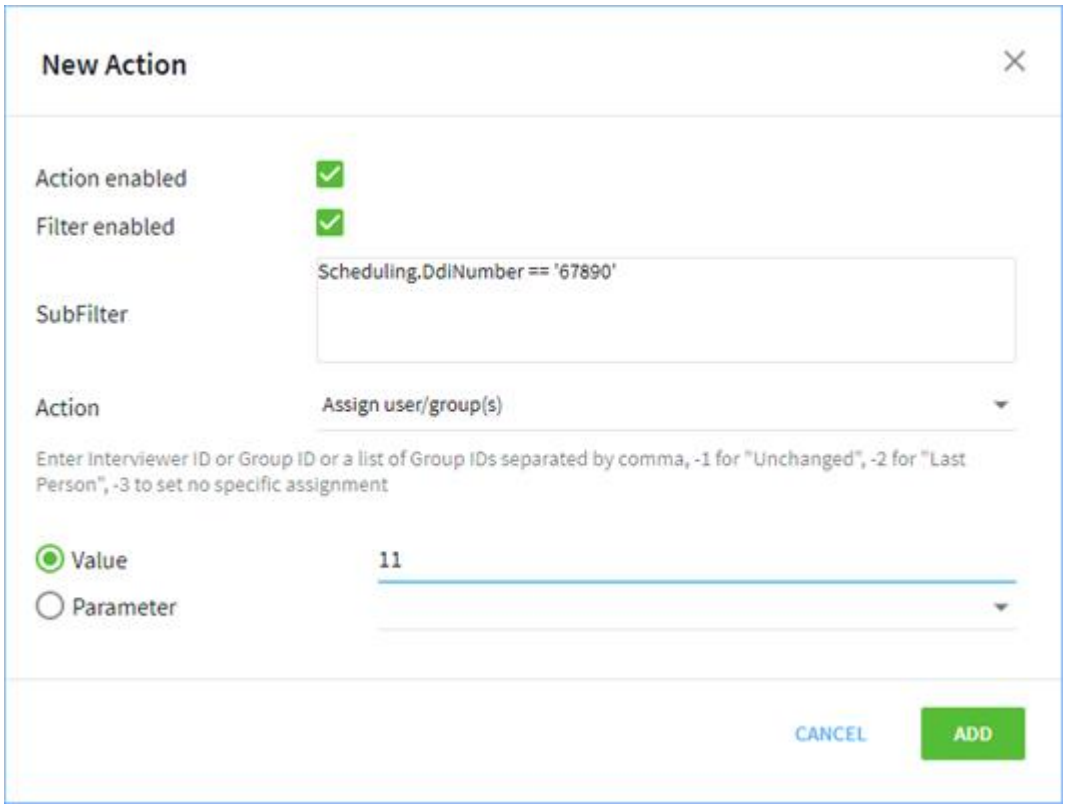


Figure 57 A scheduling rule action that routes respondents based on a DDI number

10.4. Scheduling Sample in a Mixed-Mode Project

You may need to load respondent sample data to a mixed-mode project where some records are to be scheduled and assigned for telephone interviewing whilst others will be used only for Web interviewing.

The simplest way to automate this is to create a new column sample named 'catiextendedstatus', then for CAWI-only sample records set the value to be '40'. Extended Status value ‘40’ is Custom10 and is used here as an example, but any spare custom status could be used. When loading the sample, the default ‘simple scheduling’ mode will automatically filter out the CAWI sample to the "not scheduled" list with an Extended Status value of 40 (with 40 being Custom10 or whatever status value was given).

As '**catiextendedstatus**' exists as a default field, it is not necessary to create a corresponding background variable, though you can do this if you want to make it available as a CATI filter. Now when sample is loaded the records will be identified with a value of '40', and these will be sent automatically to the 'not scheduled' list and given the extended status of 40.

It is also possible to automate this process by adding a filter to the scheduling rules. The following steps describe the set up process.

1. Create a background variable in the survey.

   In this example the question is named “Mode” and has answers “Web” and “CATI”.

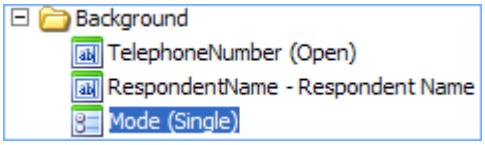


Figure 58 The “Mode” background variable

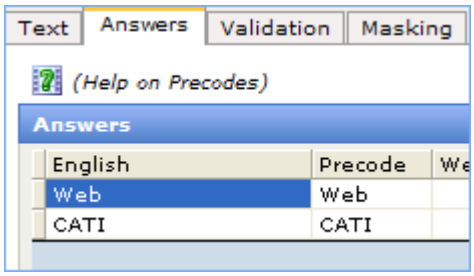


Figure 59 Create the answer categories "Web" and "CATI"

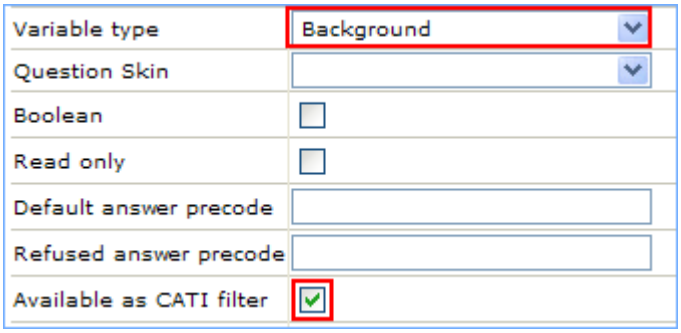


Figure 60 Set the properties to "Background" and "Available as CATI filter"

The respondent sample file will need a "Mode" column, containing "Web" for those records that are to be emailed rather than dialed, and "CATI" for those that will be assigned and scheduled for CATI interviewing.

2. Create a new sub-rule in scheduling to filter the records intended for Web interviewing.

   The filter syntax is GetRespondentValue('Mode')=="Web".

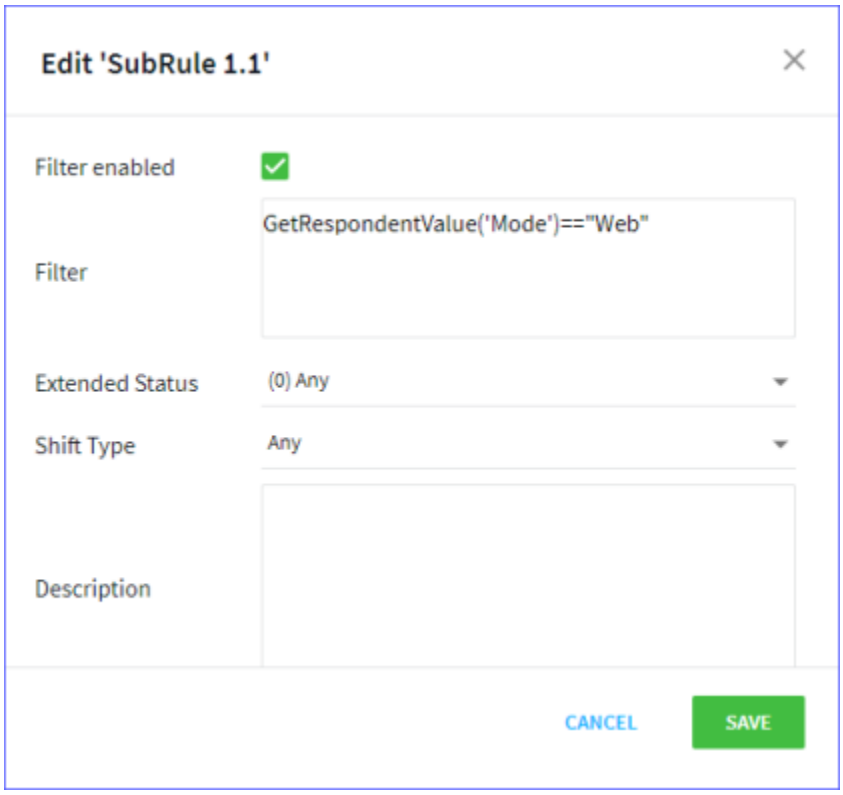


Figure 61 Creating a new sub-rule to filter the records for Web interviewing

3. Create two new actions within the sub-rule, one to suspend the interview so that Web records are not scheduled for CATI, and another to assign an extended status.

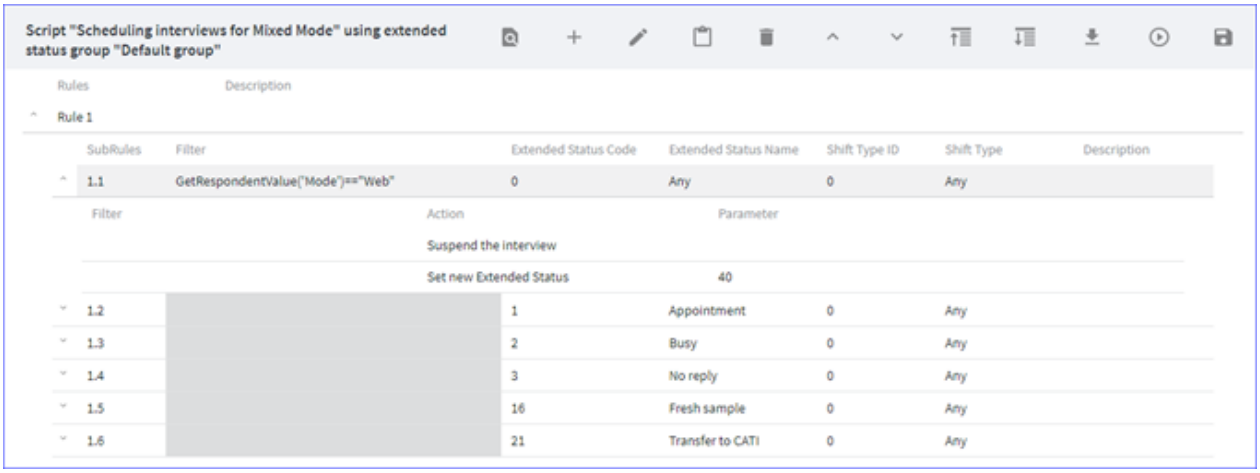


Figure 62 Sub-rule actions to suspend the interview and apply a new extended status

Any actions contained within this sub-rule must have both the “Action enabled” and “Filter enabled” checkboxes ticked.

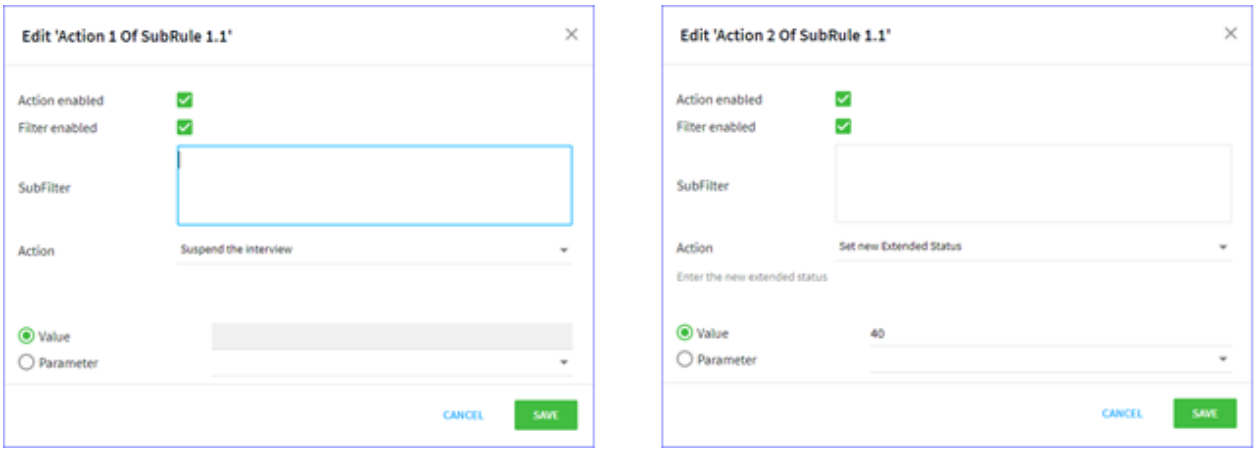


Figure 63 Actions configured to be enabled with filtering turned on

4. Once the rules are configured click **Save**.
5. To activate the changes for any survey projects currently using this set of scheduling rules, click the **Save and launch** button.

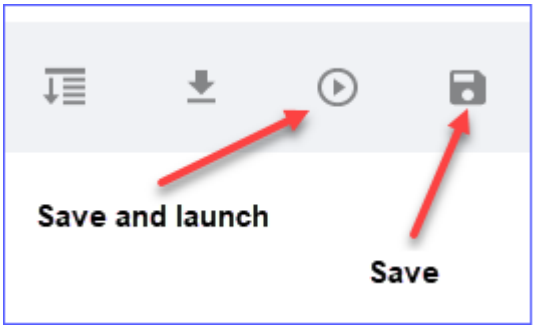


Figure 64 Save or Save and launch the created scheduling script

Note: The launch process will automatically save the scheduling rules in the current state.

6. Load the respondent sample file and set the CATI scheduling option to “Full scheduling” so that the sample loading process runs through the scheduling rules that are assigned to the project.

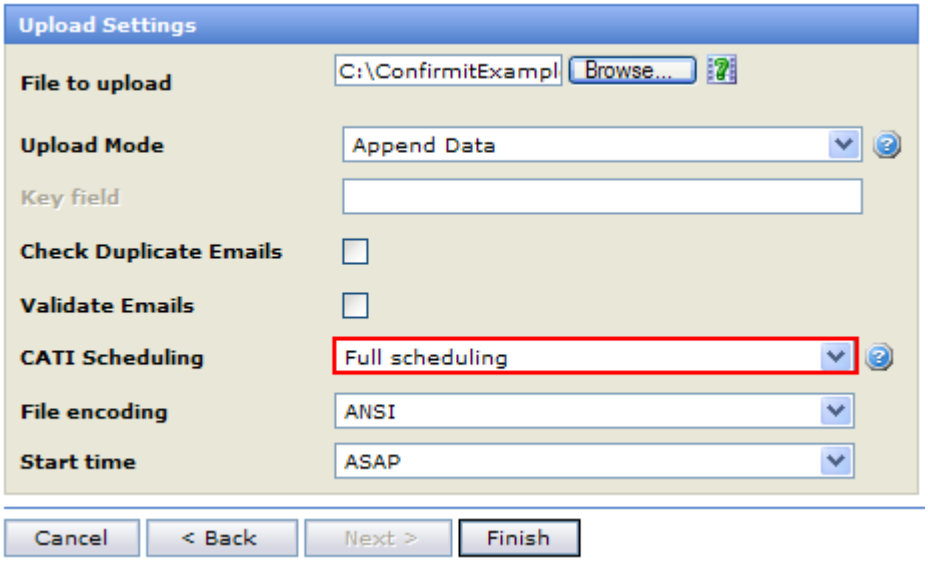


Figure 65 Load sample with "Full scheduling" selected

After loading the sample, check the Call Management window in the CATI supervisor console to ensure that the "Web" records are displayed in the "Not Scheduled" list with the appropriate extended status.

10.5. Scheduling CATI Appointments from a CAWI Interview (Transferring from CAWI to CATI)

You may want to have an interview which starts in CAWI but then transfers to CATI, capturing the essential caller information as it does so. You can use the AddRespondentToCati(status) function to create a call with corresponding Status, Respondent ID and Survey ID in CATI.

This function takes the parameter 'Status' which is the ID of the CATI Extended Status (ID value '21' represents the standard Extended Status for calls transferred to CATI). The function works only for surveys that have the CATI channel enabled, for respondents from added sample, and only when the interview is being passed in Web mode; otherwise it is ignored. For the added interview scheduling the procedure will be executed in accordance with the selected Extended Status, and the call will be updated in CATI. Quota cells will be filled in the usual way.

This example uses the AddRespondentToCati(21) function to create a CATI call with the properties captured from a CAWI interview.

1. Add a series of questions in the CAWI interview to capture the required call properties such as respondent name, telephone number, time zone and appointment date/time.

Important
This section of the survey must be defined in the Start Block to operate correctly. This is because as the interview will end at a question on the CAWI path, the survey engine will not be able to locate a suitable position at which resume the interview, and this can result in an interview which simply skips to the end without displaying any interactive questions.

For example.

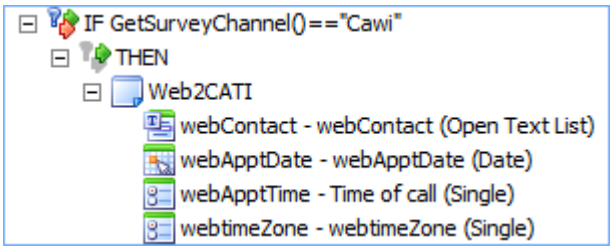


Figure 66 Example of the call property capture questions

2. Insert a script node to set the values captured from the questions defined in step 1 into the standard background fields reserved for RespondentName, TelephoneNumber and TimeZoneId.

   This script is also used to create the call in CATI with the Extended Status value '21' (Transfer to CATI) using the AddRespondentToCati(21) function.

```
SetRespondentValue("RespondentName",f("webContact")["FN"])
SetRespondentValue("TelephoneNumber",f("webContact")["PH"])
SetRespondentValue("TimeZoneId",f("webTimeZone"))
AddRespondentToCati(21)
```

3. Place a Stop node into the survey to suspend the interview and set the Extended Status to 21.

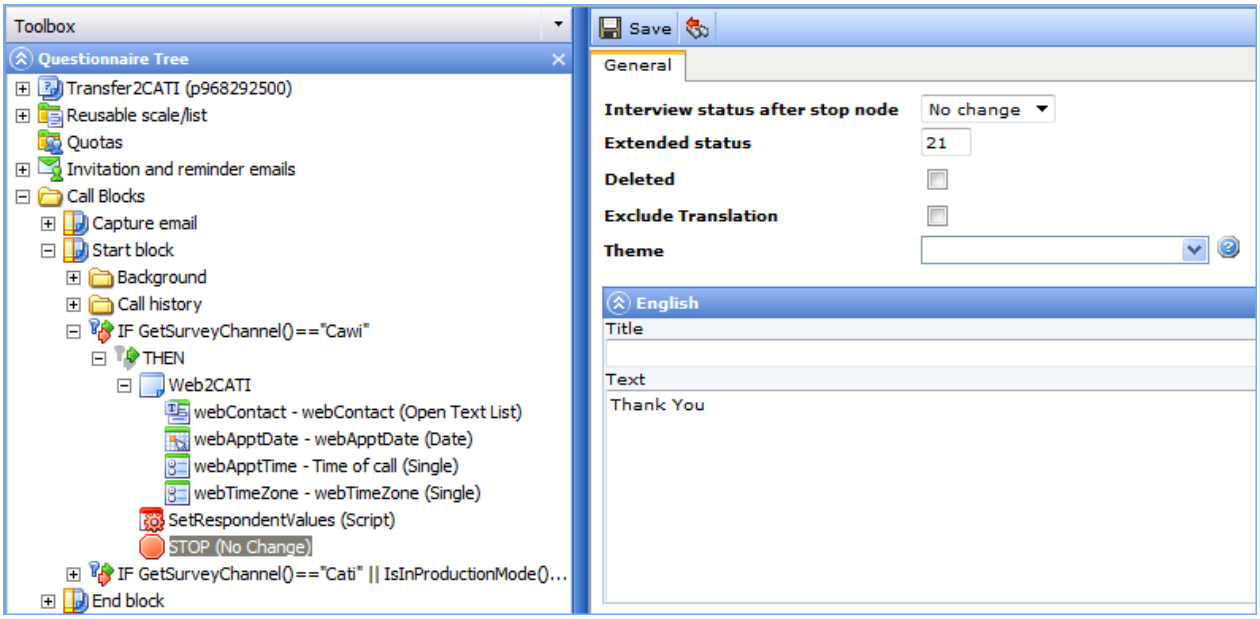


Figure 67 Adding the Stop node

4. Create or modify a scheduling script to handle calls created with 'Transfer to CATI' Extended Status.

   It is recommended to make a duplicate of the standard 'All hours' scheduling script and then to modify this duplicate. In this example the newly created script has been renamed to 'Transfer2CATI'.

5. Add a Custom Script to set the appointment date and time for the call.

   For that you can use the CreateCustomAppointment function with the appointment time specified in the respondent timezone and then click **Save**. You can paste the following code example into the Custom Script tab and click **Save**.

```
function CreateAppointment()
{
var date : DateTime = DateTime.Parse( f('webApptDate').get() );
var timeId : int = Int32.Parse( f('webApptTime').get() );
var time : DateTime = date + ( TimeSpan.FromMinutes(30 * (timeId - 1 ) )
);
CreateCustomAppointment(time);
}
```

6. On the Rules tab, add a new sub-rule for the Extended Status 21 (Transfer to CATI), then add the following three actions into the new sub-rule.

Action#	Action Enabled	Filter Enabled	Action Type	Value (Parameter)
1	Checked	Not Checked	Run custom script	CreateAppointment
2	Checked	Not Checked	Set new Call Priority	1001
3	Checked	Not Checked	Fulfill the specified appointment	5

The result should resemble:

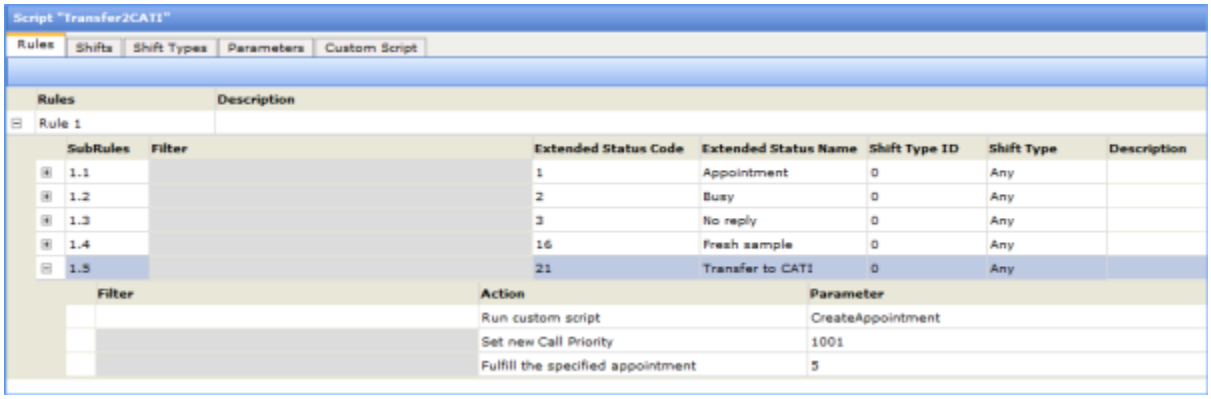
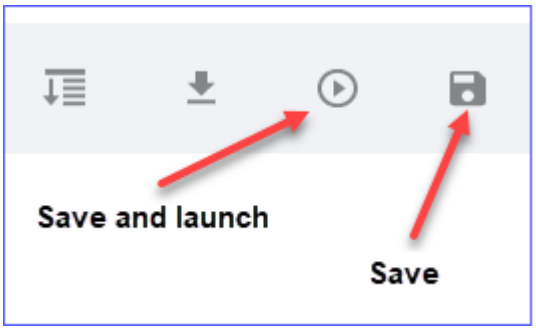


Figure 68 Example of the completed sub-rule

Once the rules have been configured, click **Save**. To activate the changes for any survey projects currently using this set of scheduling rules, click the launch button.



Note: The launch process will automatically save the scheduling rules in the current state.

7. Apply the new scheduling script to the appropriate survey using the 'Scheduling' drop down on the General tab for the given survey.

10.6. Updating Sample with Scheduling Rule Execution

In some scenarios it is necessary to update sample respondent data whilst also executing the scheduling rules of the survey. For example, perhaps some interviews were completed with an external system, and now it is necessary to update the corresponding CATI calls to set them as complete and drop them from the scheduled calls list. It is important to note that execution of the scheduling rules does not happen automatically. To update sample and call the execution of a scheduling rule it is necessary to do the following when updating the sample:

1. Modify the survey schedule in the CATI supervisor UI to include a rule that has the 'Only execute during sample update' box ticked.

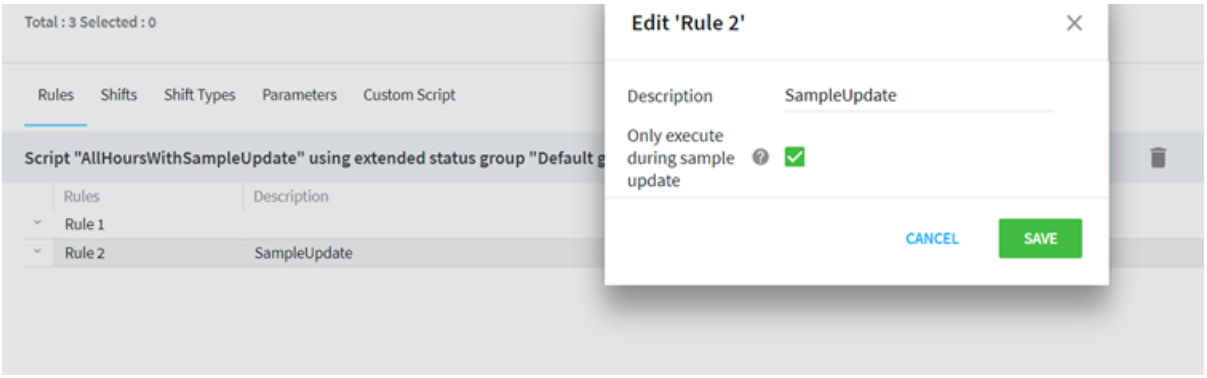


Figure 69 Adding a rule to the schedule

2. Insert a sub rule in the new rule and apply suitable actions.

   For example, here we will set the extended status to complete and drop the call from the scheduled list using a 'suspend the interview' action. Remember to launch the schedule when all changes are completed.

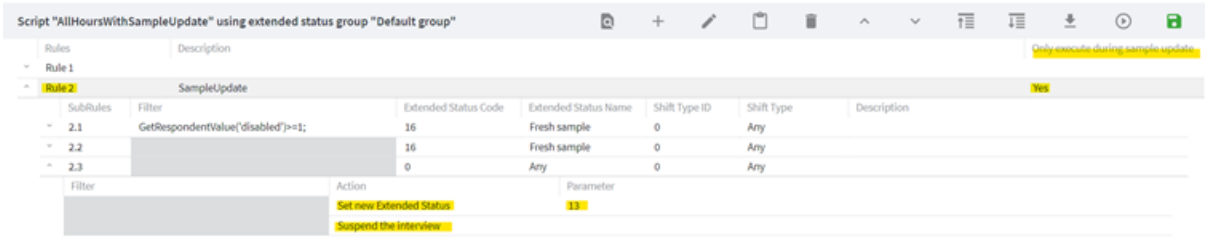


Figure 70 Inserting a sub-rule

3. Now when updating the respondent data in the survey authoring UI you must choose both the ‘update’ and ‘full scheduling’ settings to execute the update process. It will also be necessary to select a suitable key field to match the imported records with the existing records.

11. Telephone Rotation Scheduling Rules

This example of the telephone rotation scheduling rules has been set up to handle respondents who have more than one telephone number.

The rules have been implemented to work with up to six telephone numbers per respondent, but can easily be modified to cater for more numbers; there is no hard limit. They have been set up using standard rule functionality of the Scheduling interface (rules, sub-rules and actions), and Scheduling script functions to cater for specific tasks required by the phone rotation. See Appendix A for an example of the script (see Appendix A - Script Example for Telephone Rotation Scheduling Rules on page 81 for more information).

To better understand the scheduling rules and how they function in a survey, import the example rules and survey from the URLs below into your Forsta Plus account:

Telephone Rotation Scheduling Rules

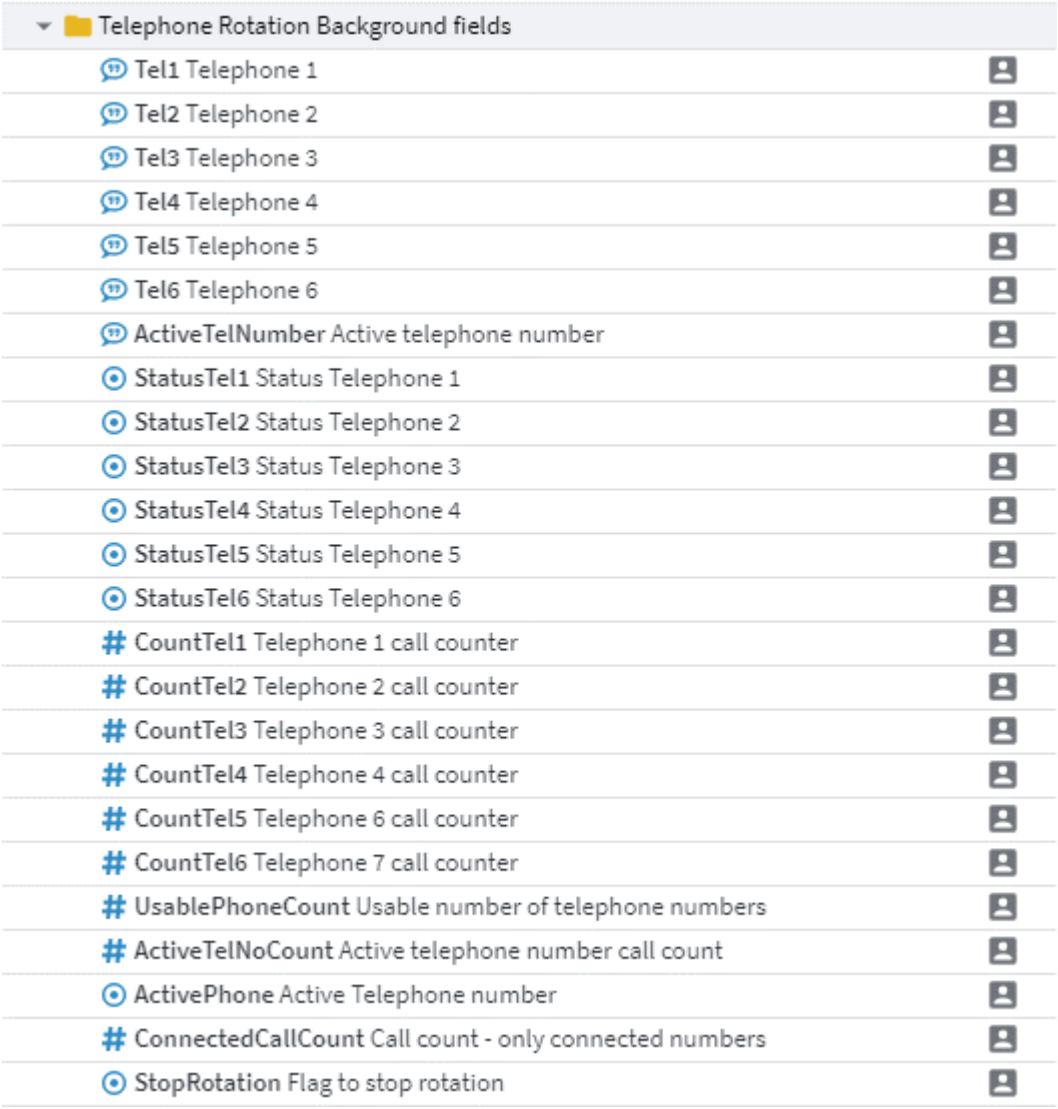
https://confirmit.box.com/s/i51io5jliesisicajbqjb5rohuthphb9

Survey

https://confirmit.box.com/s/13g3bd6ogzwhbitrbueuir8cisfdssbu

11.1. Survey Setup Requirements

The rules require that certain background questions are defined in the CATI survey that uses them, as shown below:



If more than six telephone numbers are required, simply add extra Tel, StatusTel and CountTel background questions. So, for example, if eight telephone numbers are required, Tel7, Tel8, StatusTel7, StatusTel8, CountTel7 and CountTel8 must be added.

The StatusTelN single questions are used to keep track of whether a phone number has been tried, it is a valid number (i.e. connected when dialed) or failed to connect. They are defined with the following answer and code options:

Code	Item labels
0	Not tested number
1	Valid number
9	Invalid Number

The StopRotation single question with codes 0 and 1 is used to flag the respondent when an appointment is made or when a new phone number is provided. It is assumed the phone number used when the appointment is made is the correct number to continue calling and no rotation is required anymore for those respondents with more than one usable phone number. The same assumption is made when a new telephone number is supplied when calling the respondent.

11.2. Respondent File

The respondent file will include all the background data required for the survey, and the telephone numbers included under the Tel1, … Tel6 fields. There is no need to include the TelephoneNumber column as the scheduling rules take care of assigning the number on Tel1 to Telephone number on respondent upload.

	A	B	C	D	E
1	RespondentName	Tel1	Tel2	Tel3	
2	Name 1	11111	22222	33333	
3	Name 2	11111	22222	33333	
4	Name 3	11111	22222		
5	Name 4	11111			
6	Name 5	11111	22222	33333	
7	Name 6	11111	22222	33333	
8	Name 7	11111	22222	33333	
9	Name 8	11111	22222		
10	Name 9	11111			
11	Name 10	11111	22222	33333	
12	Name 11	11111	22222	33333	

Figure 71 Example of a respondent file

11.3. Respondent Upload

The telephone rotation rules require that the respondent upload task is performed using Full scheduling and with the “All background variables required” property deactivated.

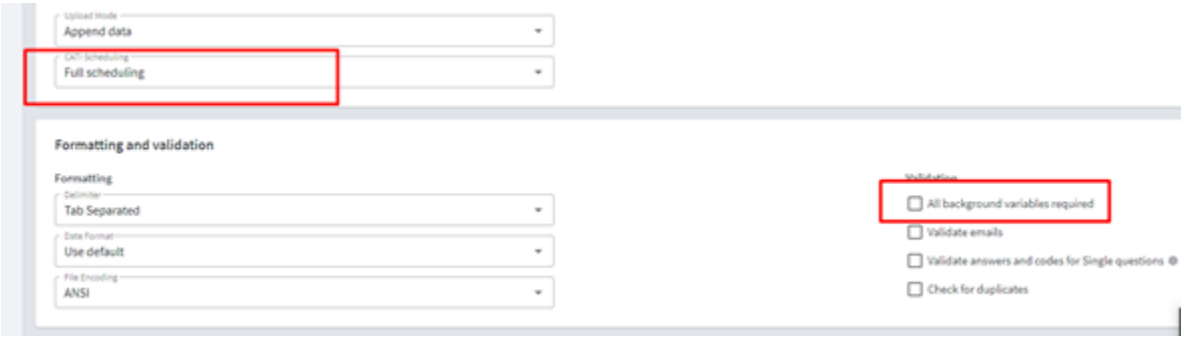


Figure 72 Upload settings

Important
The Full scheduling option is required because the Scheduling rules must be run on upload to initialize the background questions required by the telephone rotation process. Failure to do this will cause errors during the telephone rotation.

As several background questions are defined to handle the telephone rotation process (see above), an error will be caused on upload if the “All background variables required” property is active.

11.4. Scheduling Rules

A scheduling script is a set of rules which contain instructions describing what the CATI Supervisor module is to do with a specific call and under which conditions. The rules are applied to all calls that are created for a survey. Scheduling scripts use a number of parameters, some of which are predefined, and some can be configured, saved and reused in any scheduling script. This section described the rules used to manage telephone rotation scheduling.

11.4.1. Customized Extended Status

The rules use the following custom Extended Statuses:

- 31 Max number of calls attempts reached
- 32 All numbers unobtainable
- 33 Soft Appointment
- 34 Changed Tel Number

11.4.2. Rules

The scheduling set up consists of 2 rules, with several sub-rules.

Script "Telephone Rotation" using extended status group "Telephone Rotation"

Rules	Description
Rule 1	Call history
Rule 2	Extended Statuses

Figure 73 The rules

Extensive use is made throughout the Custom Scheduling Script of the following functions/call properties:

SetRespondentValue() – to set a value to a background question

GetRespondentValue() – to retrieve the content of a background question

Scheduling.Interview.TelephoneNumber – the telephone number of the current call

Scheduling.Interview.TransientState – the Extended Status (call outcome) of the current call

For example

```
SetRespondentValue('ActiveTelNumber',"Tel1")
SetRespondentValue('ConnectedCallCount', 0);
Scheduling.Interview.TransientState == 8
```

11.4.2.1. Rule 1 - Survey Data Update

This rule is used for:

1. Updating the call history fields in the survey data using the SaveCallHistory custom script.
2. Removing records with more than the maximum number of valid call attempts allowed (defined as a scheduling parameter) from the Scheduled (active) call list.

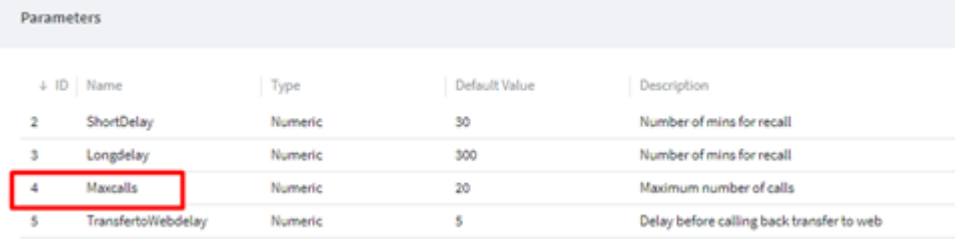


Figure 74 Rule 1 - Survey Data Update

11.4.2.2. Rule 2 - Handling Call Outcomes

Rule 2 deals with the standard call outcomes. An important point to bear in mind is that during respondent upload, two custom scripts are run (this is why the Full Scheduling option is required when uploading respondents (see Respondent Upload on page 61 for more information)).

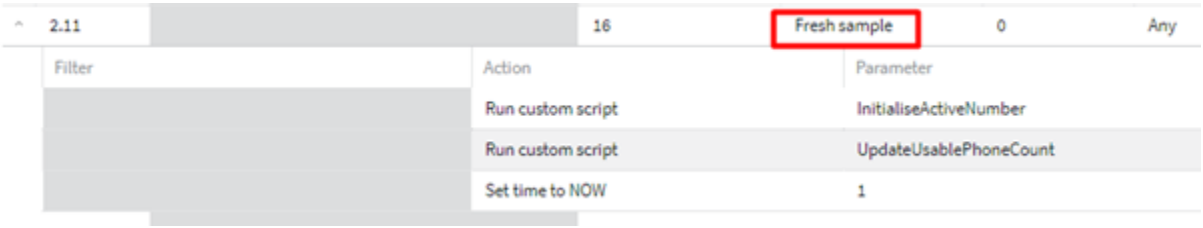


Figure 75 Rule 2 - Handling Call Outcomes

Extended Status - Fresh

The **InitialiseActiveNumber** rule sets the Tel1 number as the first number to call (Telephone Number), and the **UpdateUsablePhoneCount** counts the number of telephone numbers available per respondent and flags them as **StatusTelN 0** (Not tested number).

Important
If more than six Telephone numbers per respondent are required, the nophones var must be changed to match in the UpdateUsablePhoneCount() and the scheduling rules must be relaunched.

```
function UpdateUsablePhoneCount() {
var nophones = 6; //**** IMPORTANT Change this if there are more than 6
phone numbers *
…
```

... and add the extra Tel fields to the phoneArray on line 82 (function SelectNextNumber())

```
81      var nextphone;
82      var phoneArray = ['Tel1', 'Tel2', 'Tel3', 'Tel4', 'Tel5', 'Tel6'];
83
```

Figure 76 Adding the extra Tel fields

Important
The value of nophones and the number of elements in the phoneArray must be the same. Any extra telephone fields must be also added to the Telephony Folder with their corresponding StatusTel and CountTel fields.

Extended Status - Appointments

"Hard appointments" are locked to the interviewer who made the call. The appointment is set with an expiration time in case the interviewer is not available when the appointment time arrives.

"Soft appointments" are not locked.

It is assumed that the system should continue calling the respondent on the telephone number that connected when the appointment was made. So to stop the Telephone rotation for both hard and soft appointments, run the StopRotation custom script function.

2.3		1	Appointment	0
Filter		Action		Parameter
		Set new Call Priority		1000
		Fulfill the specified appointment		0
		Run custom script		StopRotation
		Assign user/group(s)		-2
		Set Call expiration timeout		1

Figure 77 Running the StopRotation custom script function

The StopRotation function sets the StopRotation background question to 1

Extended Status - No Reply / Busy / Answer Phone

If more than one telephone number is available, the system will attempt to move to the next available number for these three call outcomes.

2.5		2	Busy	0
Filter		Action		Parameter
GetRespondentValue("StopRotation")!="1"		Run custom script		UpdateUsablePhoneCount
GetRespondentValue("StopRotation")!="1"		Run custom script		NextNumberAction
		Recall after number of minutes		ShortDelay

Figure 78 Setting up for call outcomes

- ActivePhone

For **Busy** and **Answer Phone** (Voice Mail) outcomes, the system will attempt the same telephone number three times before it moves to the next available number. To edit this setting, modify the condition on line 54 in the function **SaveCallHistory())**.

```
51    SetRespondentValue('ActiveTelNoCount', activenocount + 1);
52
53    // IMPORTANT change the limit to the highest number of attempts for busies / answer phones
54    if (activenocount >= 2 || Scheduling.Interview.TransientState=="8" ||  Scheduling.Interview.Tra
55        SetRespondentValue('ActiveTelNoCount', 0);
56        SetRespondentValue('ActivePhone', "0");
```

Figure 79 Setting the number of times a call will be attempted

For example, >=2 will attempt the same number three times, >=1 will attempt it twice, etc.

If temporarily fixing the number for calls that are busy or go to answer phones is not required, remove the code within the red box shown below from line 138 (function NextNumberAction()).

```
137     // Remove the second part of the condition if next number should be selected
138     if (usableCount > 1 && GetRespondentValue('ActivePhone')!="1") {
139         SelectNextNumber();
140     }
```

Figure 80 Remove the code if a fix is not required

- Moving to the next available number

Three functions are used in the telephone rotation process

1. **UpdateUsablePhoneCount()** - to check if more than 1 number is available.
2. **NextNumberAction()** - if more than one number is available the next function is called.
3. **SelectNextNumber()** - to rotate through the TelN fields to find the next number available.

   The function uses the ActiveTelNumber (this will be Tel1 for on the first call)

   ```
   var currNum = GetRespondentValue('ActiveTelNumber');
   ```

   to loop through the TelN fields until it finds a valid next number (nextphone will be set)

   ```
   if (nextphone) {
   var nextNum = GetRespondentValue(nextphone);
   SetRespondentValue('TelephoneNumber', nextNum);
   Scheduling.Interview.TelephoneNumber = nextNum;
   SetRespondentValue('ActiveTelNumber', nextphone);
   }
   ```

If **nextphone** is set, the function will update the TelephoneNumber and ActiveTelNumer to this nextphone (i.e. the rotation has taken place).

12. Automated Dialing

Some additional special functions are required when working with an automatic dialer:

```
GetDialMode()
GetDialingMode()
```

These will cause the execution of the script to pause, waiting for the dialing to complete and return an indication of the outcome to the script so that suitable action may be taken. GetDialMode() will get the value from the system variable, whereas GetDialingMode() will get the value directly from the system engine. It will return a numeric value as below.

1 = Manual
2 = Preview
3 = Automatic
4 = Predictive (not supported with the TCI dialer add on)

Similarly, the dial mode can be set inside a script node in authoring as follows:

```
SetExtensionNumber(value)
```

Where value is either 1, 2, 3 or 4 based on the list above.

```
GetDialStatus()
```

When a company has the CATI telephony add on enabled, GetDialStatus() returns the outcome of a dial attempt made by the dialer. The call returns a numeric value as below.

Dial status value	Description
0	Connected call
2	Busy
3	No reply
7	Answer phone
8	Modem
9	Fax
10	Congestion
11	Unobtainable
12	Nuisance
15	Returned not dialled
18	Not automatically dialled
25	Returned dialler expired
28	Stopped

29	Telephony failure
30	Error

Typically this would then be used to control either continuing the call or finishing the interview accordingly. As shown in the image below:

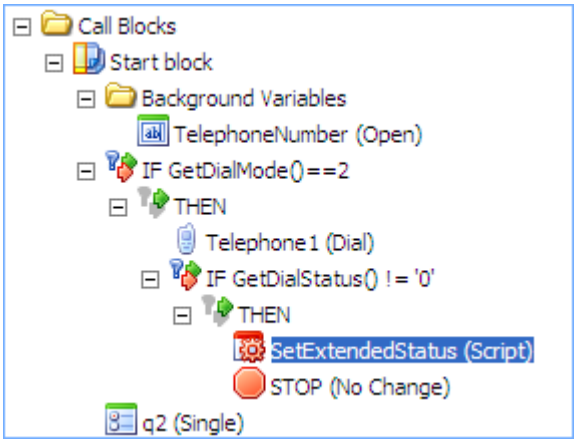


Figure 81 Setting the Extended Status with the GetDialStatus function

With the script doing the following:

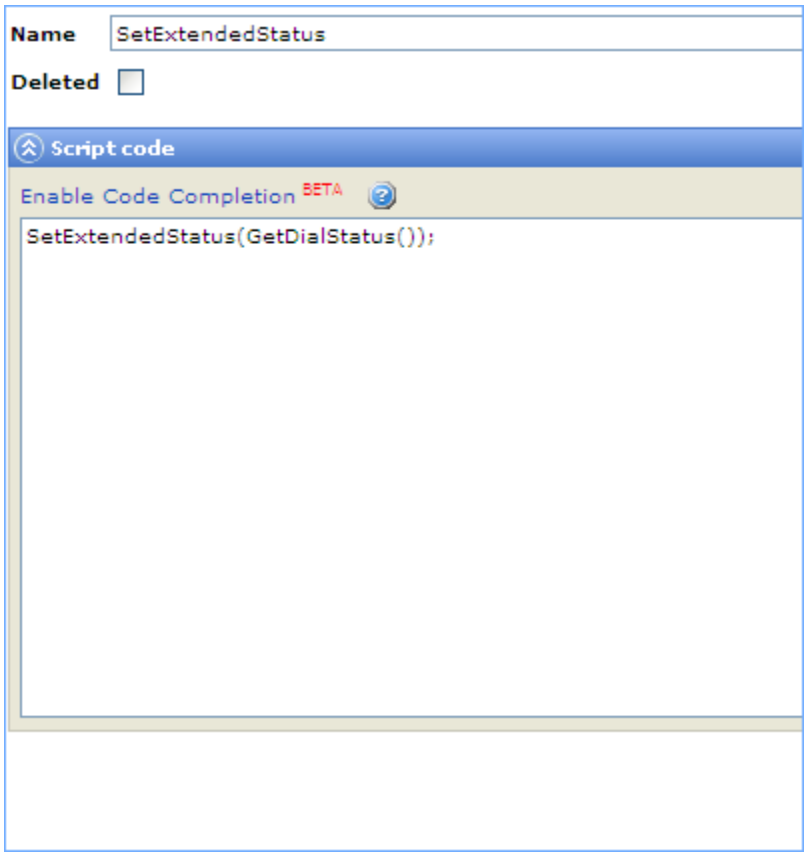


Figure 82 Script to set the Extended Status based on the value returned by "GetDialStatus"

You can have the dialer use a specified telephone number as a Caller ID, and the respondent will see this as an incoming telephone number. To accomplish this the user should enter the desired number in the corresponding sample record in the 'ExtensionNumber' column, or edit this sample record in the Authoring module.

On a supporting dialer the system field for 'ExtensionNumber' will be sent to the dialer as a call property to specify the desired Caller ID (CLID). Contact support to check if your system supports this functionality.

12.1. Hybrid Automated Dialing (Preview in Predictive mode)

In a predictive survey you can set some calls to be dialed in Preview mode. This can be achieved in the following ways:

- **Via sample** - (using the 'DialMode' column). This will assign the call to be dialed with the given mode. (for example, 2 for Preview).
- **Via scheduling** - (using the 'Set dialing mode' action). This action can be executed on sample upload by loading sample with 'Full Scheduling' set.
- **Via Call Management** - use the Dial Mode column in the Call Management view to search and filter calls. You can also modify the value set for the dial mode by right clicking on the calls to bring up the context menu. The context menu provides the following options:
    - Set Preview Dialing Mode
    - Reset Dialing Mode
- **Telephony Object** - this is a special item that can be inserted into the survey tree to initialize dialing or to hang up the call. The command has the following options

- **Dial** - place at the start of the interview to dial the number contained in the specified telephone number background variable.

Note: When dialing is enabled for a project under 'Survey Settings > CATI Options', with the dial mode specified as 'Preview', it is also possible to display some information on the screen to the interviewer whilst the dialing takes place. Text (including piping) is entered into the 'Dialing text' box of the dial command.

- **Hang up** - place at the end of the main interview to end the call.

Note: Stop nodes will automatically execute a hang up operation, so when a Stop node is defined there is no need to issue a hang up command.

13. Designing and Conducting an IVR Survey

The IVR functionality in the CATI Supervisor module enables interviews to be conducting completely or partially without human interaction. The supervisor creates and manages virtual “interviewers” – IVR agents - which “conduct” the interviews. In reality, these agents are only used to register interviews that are managed by the CATI system. Interview questions can be presented to a respondent either as a recorded human voice, or as a computer-generated voice using TTS technology if this is supported by the dialing solution. Respondents then supply answers by pressing keys on their phones. This action sends DTMF signals registered and decoded by the telecommunication provider's equipment. The CATI system fetches decoded answers and stores them in a database. The supervisor can monitor any interview conducted by a virtual IVR agent, and terminate any IVR agent’s session using the same methods as applied for human agents. Supervisors can also lock/unlock any IVR agent. Virtual IVR agent performance is included in Activity View dashboards in much the same way as for human interviewers.

Setting up an IVR survey

Set up an IVR survey the Survey Designer using the following rules:

1. IVR questions can only be based on the following four question types:
    - Single choice question
    - Numeric question
    - Open text question
    - Info node (used simply to play a message to the respondent)

    Routing logic can be applied in the usual way by inserting conditions.
2. IVR questions should strictly follow the “One question per page” directive or survey setting;
3. Special survey functions may be used to facilitate transfer to and from human interviewer mode:
    - TransferCall
4. To terminate an IVR-only interview, place a final stop node to assign the “Completed” extended status to the completed interview.

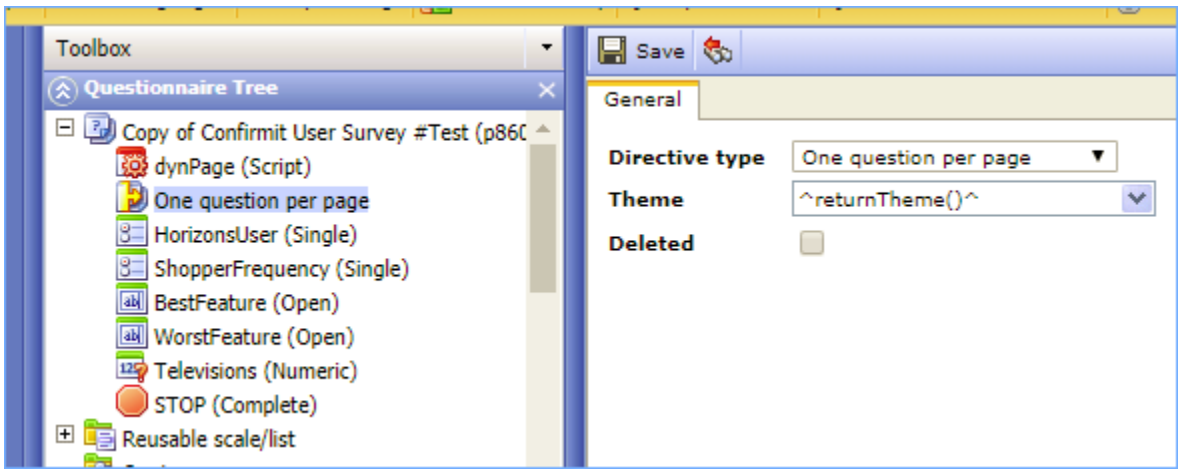


Figure 83 One question per page directive added to an IVR survey

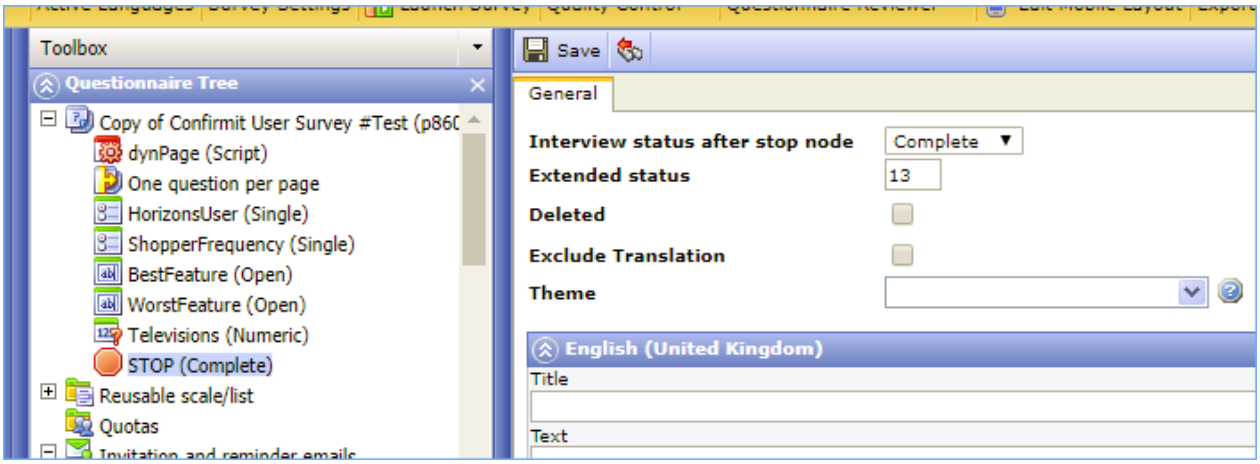


Figure 84 A Stop node assigning the Complete extended status to a completed interview

Voice Recordings

Voice recordings for the IVR questions must be uploaded to a folder on the File Library which is named to correspond with the survey's P-number, for example P1234567. The filenames of the audio messages must be set to correspond with the names of the questions to which they apply. So for example Q1 in the survey would have a file named ***Q1.wav*** uploaded into the P1234567 folder in the File Library. For multilingual surveys where for each question you need a recording for each survey language, name the files to include the applicable language ID values in the format: QID_LangID, for example ***Q1_9.wav*** for an English message.

If all the files in the survey's folder are named using only the QID, i.e. no file has a '_LangID' part in the file name, then the system will use the QID file as the default for all languages.

Note: A list of language codes is located in the Help > Language Overview menu in the Professional Authoring UI.

Audio files must be in one of the following formats:

- CCITT mu-law (aka U-law) 8k 8-bit mono wav (for North America)
- CCITT a-law 8k 8-bit mono wav (for rest of the world)

Using the TTS (Text to Speech) technology to read out question text

TTS can be defined in an IVR survey as an alternative to using voice recordings to ask the questions. To use TTS in a question, enter the full **"ivr_hidden"** div tag followed by the question text.

```
<DIV id="ivr_hidden" style="DISPLAY: none"> <audio>false</audio></DIV>
Please leave a comment
```

Creating and managing virtual IVR agents

A virtual IVR agent is in almost all aspects similar to a “normal” (human) interviewer – IVR agents are managed from the Interviewers tab, they are assigned to surveys that require IVR functionality and will be included in all corresponding Activity Views and reports. When IVR functionality is activated in the CATI Supervisor module, the **IVR Agents** check box appears on the left side of the frame toolbar in the Interviewers tab and in the Activity Views tab, and it becomes possible to view and manage the list of virtual IVR agents. Note that the IVR agent list is hidden and inaccessible in the Interviewers tab by default, even after the IVR functionality is activated.

When IVR functionality is activated, the supervisor checks the “IVR Agents” box to reveal virtual IVR agents and hide human agents. Human agents remain hidden and inaccessible whilst the supervisor manages virtual IVR agents. To revert to the list of human agents, clear the “IVR Agents” box.

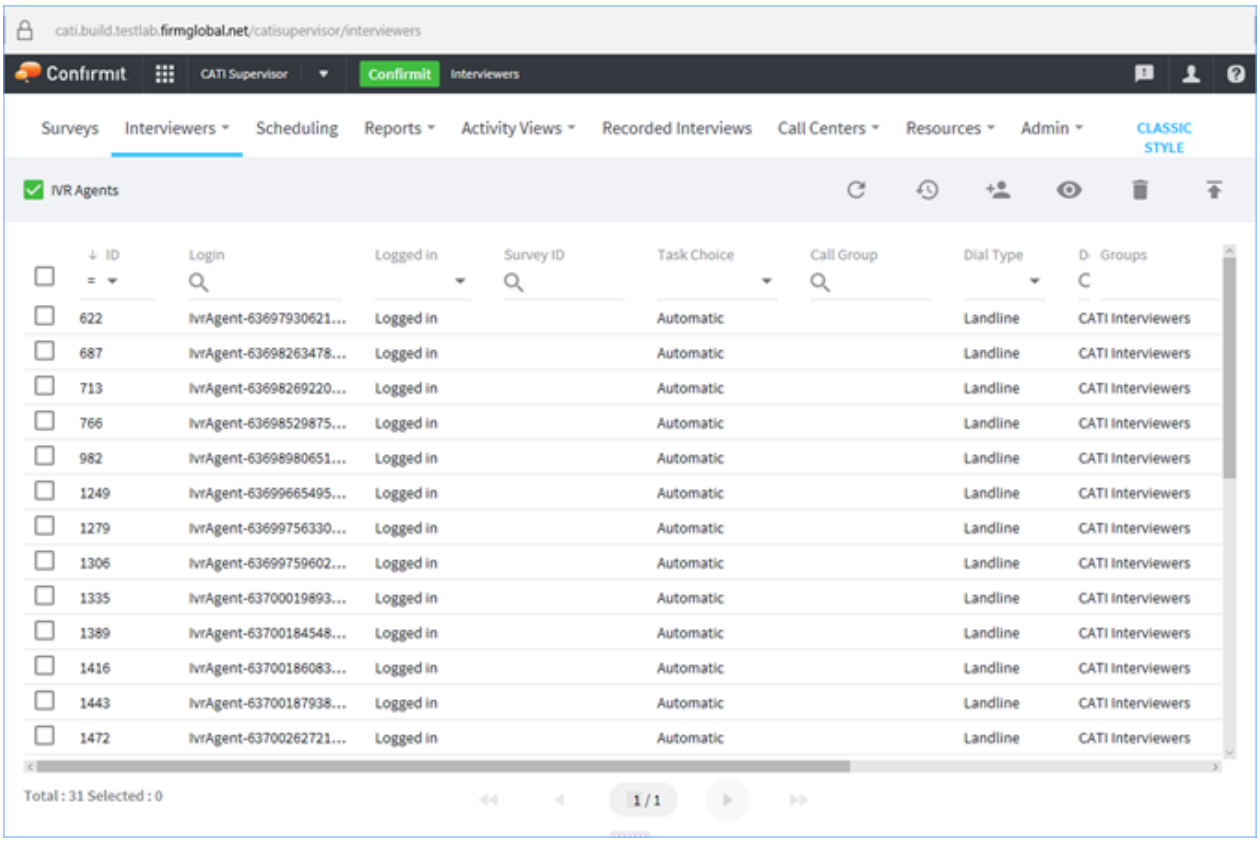


Figure 85 Browsing the list of virtual IVR agents

The same separation is applied in the Activity Views tab; check the “IVR Agents” box to hide human agents and include only virtual agents.

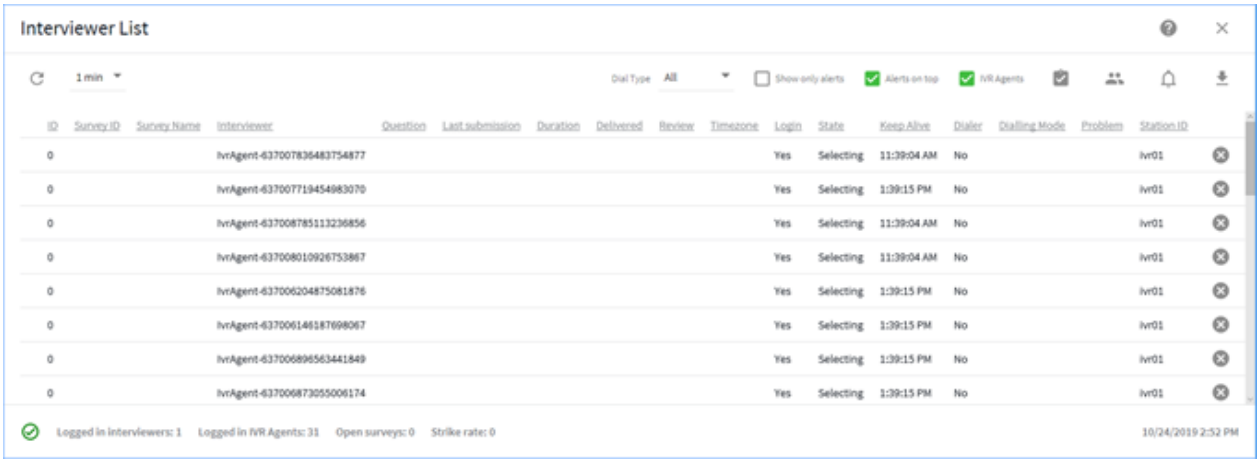


Figure 86 Virtual agents in the Activity Views tab

Before beginning to use IVR for data collection, a number of IVR agents must be created. The number of IVR agents determines the number of concurrent calls that can be conducted using IVR at any given time. Click “Create Interviewer” to create a number of virtual IVR agents simultaneously. The total number of virtual IVR agents that can be created for a company is defined by the Forsta Plus license. When creating IVR agents, the supervisor can choose how many agents they will create at one time. Note that there is no requirement to use all of the available limit; you can create as many IVR agents as required in one pass, potentially leaving some spare capacity that can be used when required.

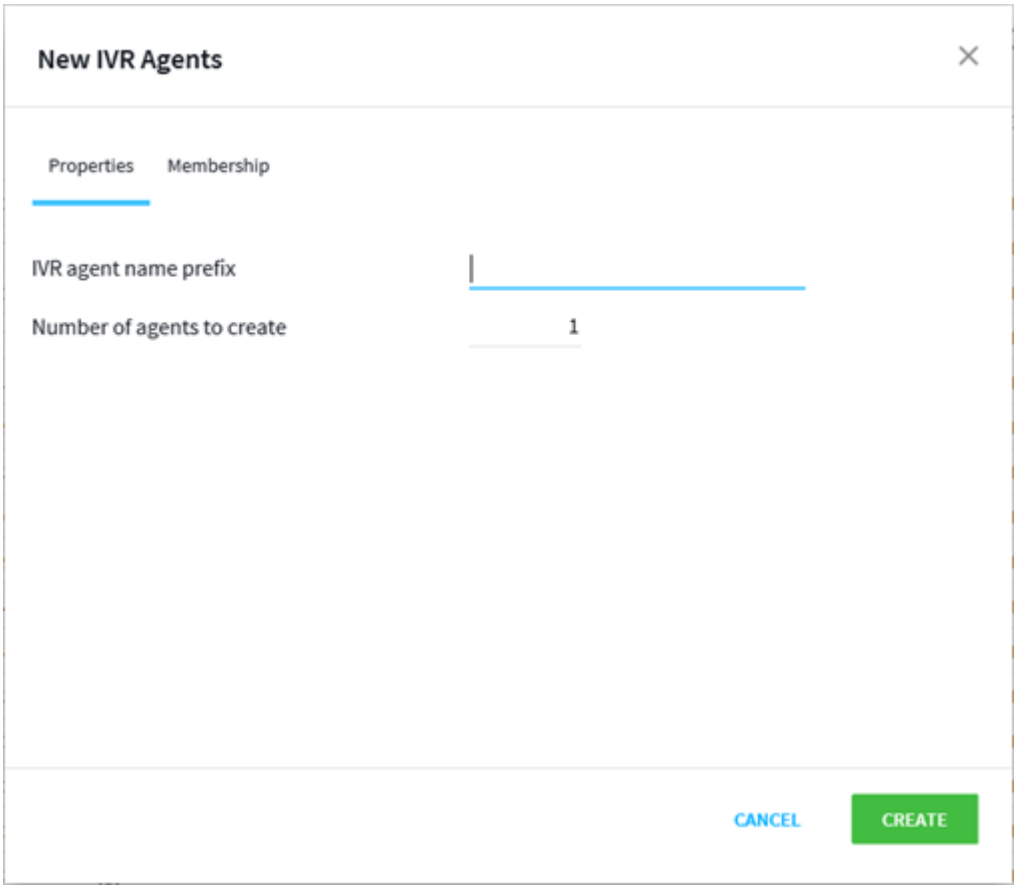


Figure 87 Creating new IVR agents

Note: All virtual IVR agents work ONLY in the Automatic Choice mode.

Automated log in procedures for IVR agents

IVR agents can provide a 24/7 service, so shortly after the virtual IVR agents are created they will be logged into the system automatically. The CATI system periodically checks whether the designated dialer is connected and activated, and if it is then an IVR agent is assigned and the login procedure for newly created IVR agents will commence.

If an IVR agent is assigned to an interview and this interview exists in the system, then the interview is immediately delivered to the IVR agent since the agent resource is already logged in. If there is no interview assigned to an IVR agent, the agent will remain “idle” – meaning that they stay logged in and waiting for a call.

The IVR agent’s property set differs from the property set of a human interviewer as follows:

- It does not have the Properties tab since it relies on a CATI system and a dialer for logging in/out and it conducts interviews in an Automatic mode only.

- Just as for a human interviewer, a virtual IVR agent can be locked and unlocked and their session can be terminated. The terminate command performed for an IVR agent terminates the current interview instantly and logs the IVR agent out. The Lock command prevents the CATI system from logging the locked interviewer back in. The next time the CATI system checks the availability of the dialer, it will try to log all IVR agents back in if they are not locked. In the event an IVR agent is locked they cannot log back in automatically. The automatic login procedure for an IVR agent is re-started as soon as they are unlocked (given that the designated dialer is connected and activated).

Supported question types and IVR survey logic

When an interview is started, questions are presented to a respondent in one of two ways:

- An audio track containing the question text and all related information which can be used by a respondent to provide an answer, is played back.
- The “Text-to-speech” technology is used. In this case the question text and all the related information is read by a computer-generated voice. A respondent provides answers by pressing suggested numeric key combination on their phone.

Info questions normally do not require an answer. Open text questions require a “free text” answer and the only way to capture such an answer is to record the respondent’s speech. Sound recording is activated automatically, and the recorded answer is stored in the database. Numeric and a single choice questions support only integer digits provided as the answer.

Call Transfer functionality

A survey scripting function enables a call to be transferred from an IVR agent to a human interviewer, or to an interviewer group. The script is added to a question in the Survey Designer, and it should be positioned immediately after a question which prompts a respondent to switch to a human agent.

The script is triggered if a respondent chooses the answer which activates switching to a live agent. In the example below it is answer '1' to the question 'q1'.

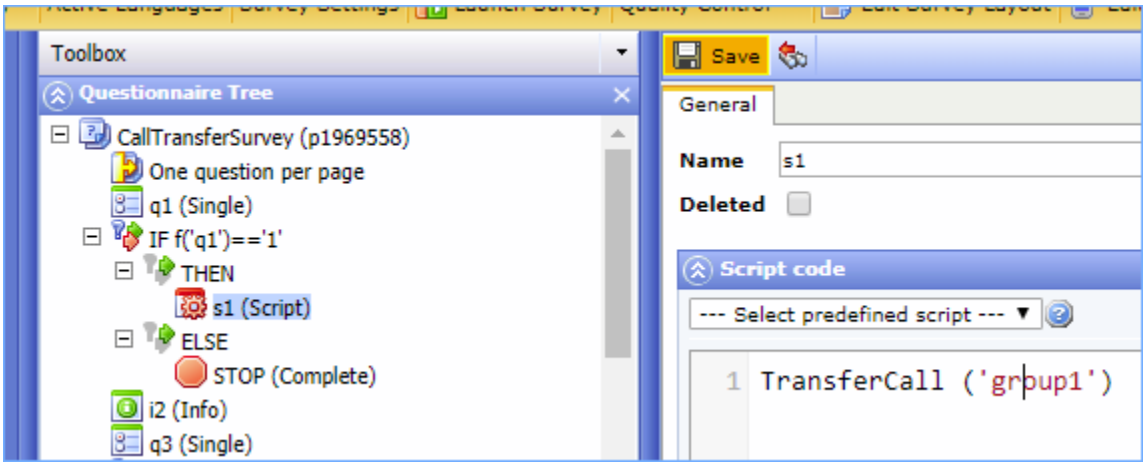


Figure 88 The TransferCall function used in a script to switch a respondent from an IVR to a live agent

The script must call the 'TransferCall' function . This function can be used both in Predictive and Non-predictive surveys and also with or without parameters. When the function is used without parameters, like this:

```
TransferCall ()
```

it tells the system to deliver the transferred call to the first live interviewer who is currently free.

The function also accepts the name of an interviewer group as a parameter:

```
TransferCall ('group1')
```

In this case the call is transferred to the first live interviewer belonging to the specified group. The interview will not be delivered to an interviewer who does not belong to that group.

When the function is triggered, the transferred interview is assigned the "Internal Transfer" ITS with the ID '1010' and the priority value '2000'. A corresponding record is added to the Call History log.

14. Interview Voice Recording

When using an automatic dialer you can create voice recordings using the .wav file format. The recordings are stored on the physical dialer hardware*.

* With TCI the default path for recordings is: C:\BellviewTCI\Interviews\<Project ID number>\

Note: If multiple call attempts are made to the same number, each recording will be made in a separate file. A number is added to the end of the file name to indicate the attempt number.

A CATI project-level option can be set to enable whole interview recordings. Recording will start each time the dialer begins to call a number, and will stop each time the call is terminated (when the interview reaches a hangup command in the survey script).

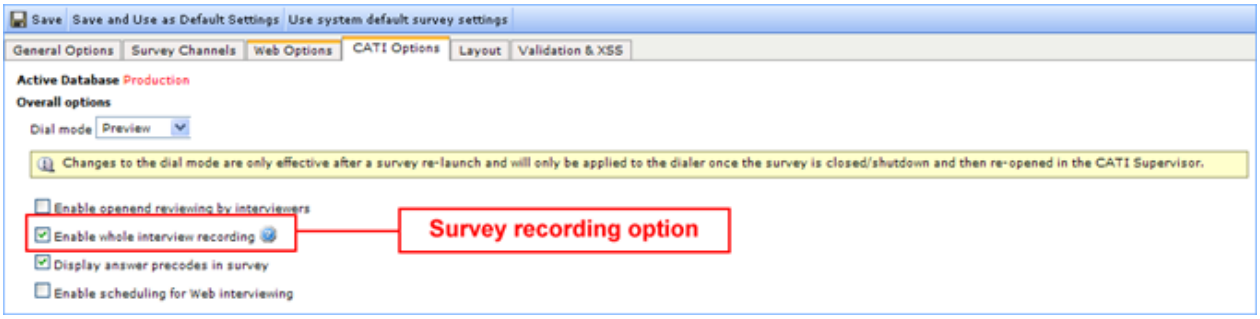


Figure 89 CATI Options tab: Enable whole interview recording

14.1. Sectional Voice Recording

You can record sections of an interview by placing start and stop functions at the desired locations within the survey.

StartVoiceRecording('label')

To begin a sectional recording, call StartVoiceRecording in a script node. The recording will continue until either a StopVoiceRecording() function call is made or the interview reaches a hangup command.

When the StartVoiceRecording script is declared, a label must also be included within parenthesis. The label specified is then incorporated into the name of the .wav file for the recorded section.

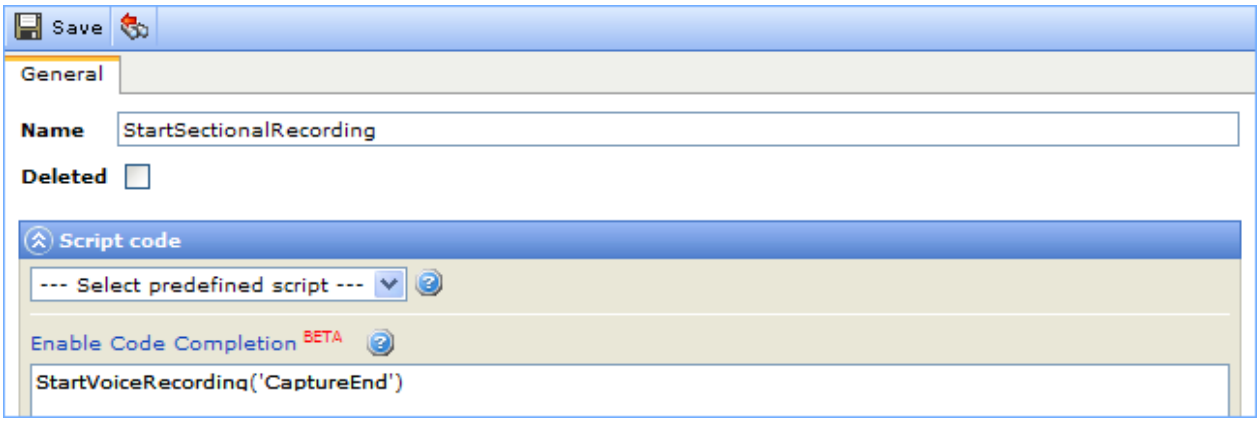


Figure 90 Script to call StartVoiceRecording function

StopVoiceRecording()

StopVoiceRecording is a function that may be called inside of a script node to end either whole interview or sectional recording.

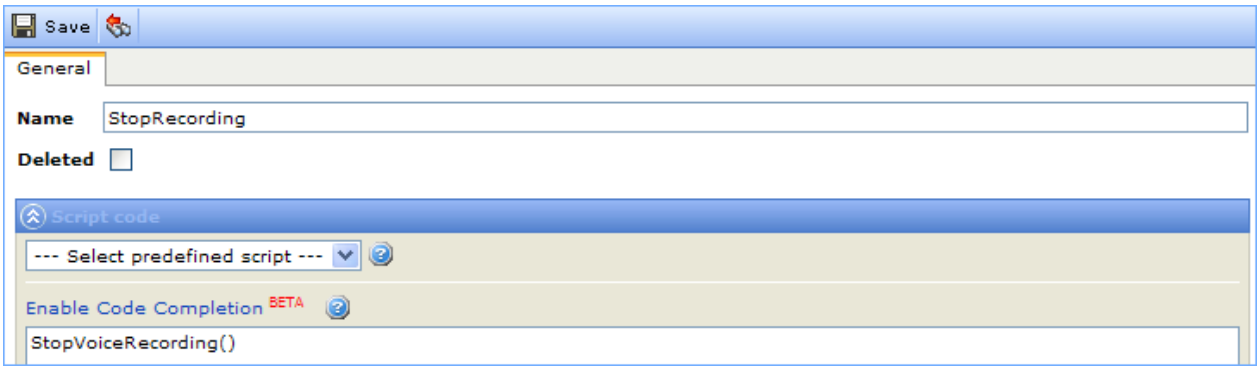


Figure 91 Script to call StopVoiceRecording function

Although it is necessary to supply a label when starting a new sectional recording, this is not required when stopping the sectional recording. Instead you must use the StopVoiceRecording command with one of the three supported strings… "WholeInterview", "Sectional" or "Both" as:

```
StopVoiceRecording('WholeInterview')
StopVoiceRecording('Sectional')
StopVoiceRecording('Both')
```

Note that these strings are not case sensitive.

Note: It is not always possible to have concurrent whole-interview recordings and sectional recordings for the same interview. This functionality is supported by the Forsta API, but it is down to the dialer vendor to make use of the capability. Neither the Forsta TCI dialer nor the Marketing Systems Group, PRO-T-S® dialer currently support concurrent whole interview + sectional recording. If whole interview recording is enabled then a StartVoiceRecording('label') or StopVoiceRecording() function call will toggle the whole interview recording at the point of execution.

14.2. User Privacy and Monitoring Consent

To protect user privacy, legislation in some countries prohibits live monitoring of respondent's answers until they provide their explicit consent for such monitoring. To comply with these rules the Forsta Plus software provides the means to enable and disable the interview live monitoring and recording functions.

Enable and disable live and deferred interview monitoring using of controls provided:

- On the **CATI Options** tab of the **Survey Management > Survey Settings** menu in Professional Authoring.
- By scripting functions which can be added to a question in the Survey Designer.

The "Monitoring and recording options" section on the **Survey Settings > CATI Options** tab provides controls which either allow live interview monitoring and/or screen recording from interview start, or prohibit such activity.

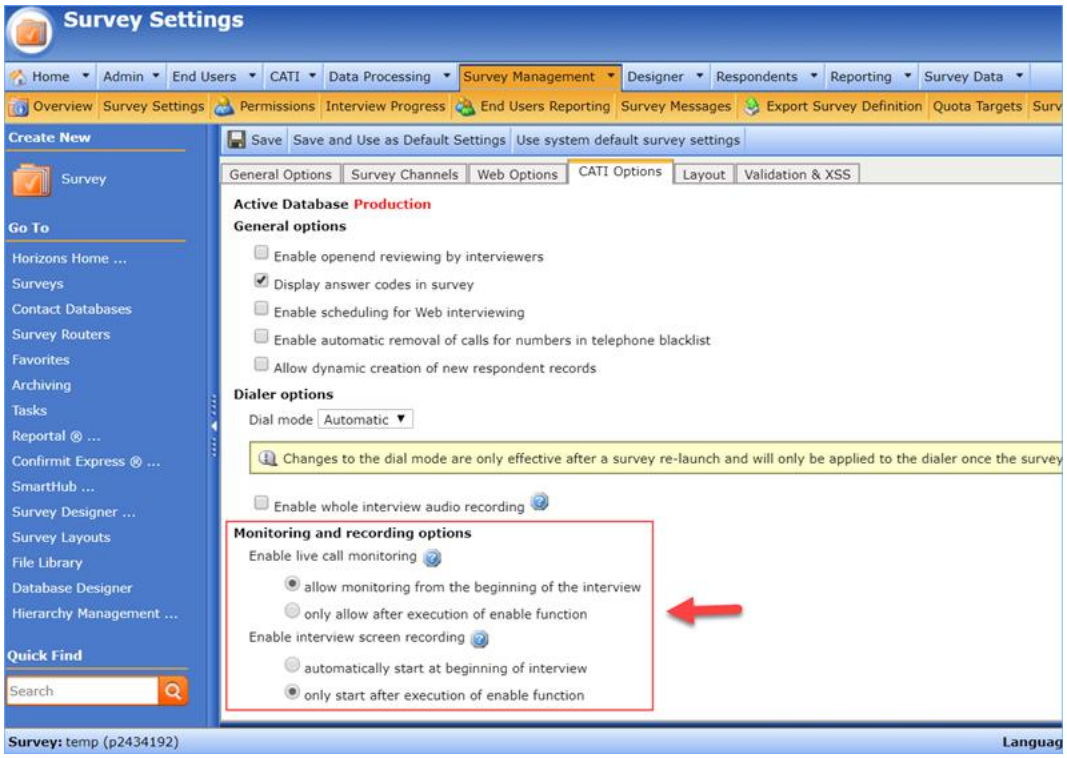


Figure 92 CATI Monitoring and recording options

In addition to these settings, two scripting functions are available: **EnableLiveMonitoring()** and **StartScreenRecording()**. Both are used in conjunction with the "Only allow/start after execution of enable function" options. When one or both of these options is enabled, the call monitoring and/or video recording features will only be permitted after explicit consent has been obtained in the survey. This means that the survey programmer must include the corresponding script functions - EnableLiveMonitoring() and StartScreenRecording() - in the appropriate node of the questionnaire tree. See the example in the illustration below.

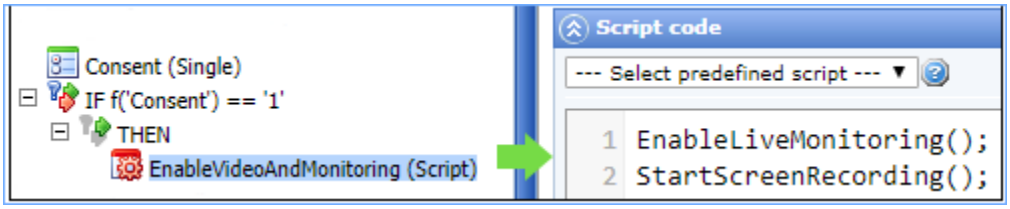


Figure 93 The script functions

To specify a point where the respondent's consent is obtained, add a script object to the survey and type in the relevant function or select one from the list of the predefined functions. This will then be the start point from which live monitoring and/or screen recording is allowed.

Live monitoring cannot be prohibited again once it is allowed for any given interview - this interview remains available for monitoring/recording until it finishes with a status.

When live monitoring is prohibited for the interview, the **Monitoring Not Permitted** icon is shown in the Interviewer Activity view for the current interview. When this is the case, the supervisor cannot monitor the current interview until the point in the interview logic is reached which allows to start live monitoring (one of the scripting functions that are mentioned above). At this point the **Start Monitoring** button becomes available.

A situation is possible when the previous interview finishes and the supervisor keeps the Monitoring Console window open when the next interview is delivered. Live monitoring is not allowed for this interview, and the monitoring console will show a blank screen and a notification message in the status bar. The monitoring console will start to show the interviewing process when the interview reaches the point where live monitoring is permitted (when a corresponding function is triggered in the questionnaire tree).

Audio is also not available for interviews where live monitoring activity is banned. If you are preparing to record audio through the use of an integrated dialer, you should use separate functions that control the audio recording functionality from within the survey. They are added to the questionnaire tree in the same way as the functions described above.

15. Limited Access Supervisor Accounts

In some cases, it may be desirable to set up supervisor accounts that have access permissions restricted to the point that only specific operations may be performed. For example, an external client may wish to monitor live calls and/or playback recorded interviews for work conducted on a particular survey.

In this case an account should be created that has the following base permissions:

Note: If you are working with a SaaS based Forsta Plus system user, contact your account manager for this to be set up.

Permission	Description
SYSTEM_CONFIRMIT_ACCESS	Allows access to the main Forsta Plus user interface.
FEATURE_CATI	Allows access to the CATI menu in Forsta Plus.

This will create a user who can access the supervisor UI but who does not have the ability to perform any operations (all menus will be hidden). It will therefore be necessary to grant at least one additional permission to make the account useful for some purpose. In the example of the external client, this would mean granting either (or both) of the following permissions…

Permission	Description
FEATURE_CATI_SUPERVISE_RECORDED	*Allows access to the Recorded Interviews tab.
SYSTEM_CATI_SUPERVISE_MONITOR	*Allows access to the Interviewers List, activity view and to select live calls to monitor.

*Access is only permitted for surveys that the supervisor user has been given explicit permission to see (where the CATI survey permission setting is enabled in the survey authoring application).

16. Appendix A - Script Example for Telephone Rotation Scheduling Rules

Procedures and descriptions are provided in the Telephone Rotation Schedule Rules chapter (see Telephone Rotation Scheduling Rules on page 59 for more information).

```
// To initialize 2 required background fields on upload
function InitialiseActiveNumber()
{
 var telnumber=GetRespondentValue("Tel1");
 Scheduling.Interview.TelephoneNumber=telnumber;
 SetRespondentValue('TelephoneNumber',telnumber)
 SetRespondentValue('ActiveTelNumber',"Tel1")
 SetRespondentValue('ConnectedCallCount', 0);
}

// To stop rotation - used for appointment and new number outcomes
function StopRotation() {
 SetRespondentValue('StopRotation', "1");
}

// to update the call history fields after each connected call
// required as the dialer does not go to the survey engine for outcomes
that don't connect to an interviewer
(e.g. no reply, busy, fax, etc)
function SaveCallHistory() {
 var callstatus = Scheduling.Interview.TransientState;
 var dt: DateTime = DateTime.Now;
 var iter = int(GetRespondentValue('CallAttemptCount'));
 var currentnumber = GetRespondentValue('ActiveTelNumber');

// to increment call count for connected calls only
if (Scheduling.Interview.TransientState == 1 ||
Scheduling.Interview.TransientState == 33 || Scheduling.Interview.
TransientState == 2 || Scheduling.Interview.TransientState == 3 ||
Scheduling.Interview.TransientState
 == 7 || Scheduling.Interview.TransientState == 7) {
 SetRespondentValue('Status' + currentnumber, "1");
 SetRespondentValue('ConnectedCallCount',
int(GetRespondentValue('ConnectedCallCount')) + 1);
}

// to incremement count for current number
 var currentnumbercount = int(GetRespondentValue('Count' +
currentnumber))
 SetRespondentValue('Count' + currentnumber, currentnumbercount + 1);
 SetRespondentValue('LastExtendedStatus', callstatus);
 f('CallDateTime', iter).setValue(dt);
 f('CallExtendedStatus', iter).setValue(callstatus);

 var telno = GetRespondentValue('TelephoneNumber')
 f('CallPhoneNumber', iter).setValue(telno);
 Scheduling.Interview.TelephoneNumber = telno;

// To keep track of how many times the number has been called for
outcomes Busy / Answer Phone
 if (Scheduling.Interview.TransientState == "2" ||
Scheduling.Interview.TransientState == "7")
  SetRespondentValue('ActivePhone', "1");

 var activenocount = int(GetRespondentValue('ActiveTelNoCount'))
 SetRespondentValue('ActiveTelNoCount', activenocount + 1);
```

```
// IMPORTANT change the limit to the highest number of attempts for
busies / answer phones
if (activenocount >= 2 || Scheduling.Interview.TransientState=="8" ||
Scheduling.Interview.TransientState=="9" ||
Scheduling.Interview.TransientState == "11") {
 SetRespondentValue('ActiveTelNoCount', 0);
 SetRespondentValue('ActivePhone', "0");
}
}

// to keep track of how many numbers are still available to call (it
excludes unobtainables, faxes, modems)
function UpdateUsablePhoneCount() {

 var nophones = 6; //**** IMPORTANT Change this if there are more than 6
phone numbers *****
 var count: int = 0;
 for (var i: int = 1; i <= nophones; i++) {
  if (GetRespondentValue('Tel' + i) != null &&
GetRespondentValue('StatusTel' + i)!= 9) {
   count += 1;
   if (GetRespondentValue('StatusTel' + i) == null) {
    SetRespondentValue('StatusTel' + i, '0');
   }
  }
 }
 SetRespondentValue('UsablePhoneCount', count);
}

// To select to next number available
// Used for rotation when call outcome is a no reply
function SelectNextNumber() {
 var currNum = GetRespondentValue('ActiveTelNumber');
 var nextphone;
 var phoneArray = ['Tel1', 'Tel2', 'Tel3', 'Tel4', 'Tel5', 'Tel6'] /****
IMPORTANT Add to this if there are more than 6 telephone number
background fields */

  // to go through all the next numbers
  var flagCurr = false;
  var rLength = phoneArray.length;

  var currArray;
  SetRespondentValue('ActiveTelNoCount',0)
  for (var j: int = 0; j < rLength; j++) {
   var chkNum = phoneArray[j];
   currArray = j;
   if (chkNum == currNum) {
    flagCurr = true;
    continue;
   }
   if (flagCurr) {
    // test next number to see if TelStatus value is 1 or 9
    var testVal = 'Status' + chkNum;

    if (GetRespondentValue(testVal) != null &&
(GetRespondentValue(testVal) == "0" || GetRespondentValue(testVal) ==
"1")) {
     nextphone = chkNum;
     break;
    }
   }
```

```
  }

  //end of available phone numbers reached
  if (currArray == rLength - 1 && nextphone == null) {
   for (var jj: int = 0; jj < rLength; jj++) {
    var chkNum1 = phoneArray[jj];
    // test next number to see if TelStatus value is 1 or 9
    var testVal1 = 'Status' + chkNum1;

    if (GetRespondentValue(testVal1) != null &&
(GetRespondentValue(testVal1) == "0" || GetRespondentValue(testVal1) ==
"1")) {
     nextphone = chkNum1;
     break;
    }
   }
  }

  if (nextphone) {
   var nextNum = GetRespondentValue(nextphone);
   SetRespondentValue('TelephoneNumber', nextNum);
   Scheduling.Interview.TelephoneNumber = nextNum;
   SetRespondentValue('ActiveTelNumber', nextphone);
  }
 }

// To flag record for rotation
function NextNumberAction() {
 var usableCount = int(GetRespondentValue('UsablePhoneCount'));

 // Select next number if ActivePhone is not set to 1 (i.e. for busies
and answer phones)
 // Remove the second part of the condition if next number should be
selected for busies and no replies
 if (usableCount > 1 && GetRespondentValue('ActivePhone')!="1") {
  SelectNextNumber();
 }
}

// to flag invalid numbers
function InvalidNumberAction() {
 var currTelAct = GetRespondentValue('ActiveTelNumber');
 var tag = 'Status' + currTelAct;
 SetRespondentValue(tag, '9');

 UpdateUsablePhoneCount();
 var usableCount = int(GetRespondentValue('UsablePhoneCount'));
 if (usableCount > 0) {
  SelectNextNumber();
 } else {
  SetRespondentValue('ActiveTelNumber', '');
  SetRespondentValue('TelephoneNumber', '');
  Scheduling.Interview.TransientState="32" // all numbers are
unobtainable
 }
}
```

Index

L

M

N

O

P

Q

R

S

T

U

V

W