IVR Example Program 2 - Lists and Variables

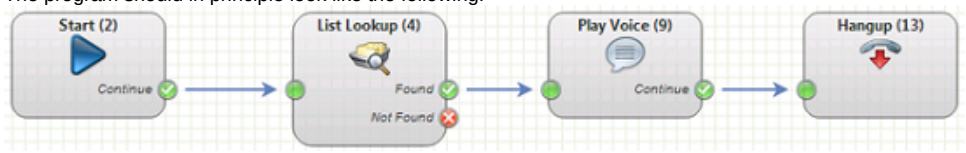
In this routing example a simple service is created which uses lists and variables. We recommend you follow and program this version. The knowledge gained in the Programming Example section is assumed.

The object is that callers should only be able to take part once in a vote. If callers ring twice, they should hear a prompt that they have already voted.

To achieve this, the hash value of the caller's number is entered into a list. The hash value is stored in the variable *$callerhash* and is a unique code representing the caller's number (in contrast with *$caller*, which depending on your rights in the system may be shortened using xxx or invisible). During every call a list is checked. If the hash value is found in the list, then the caller has already called.

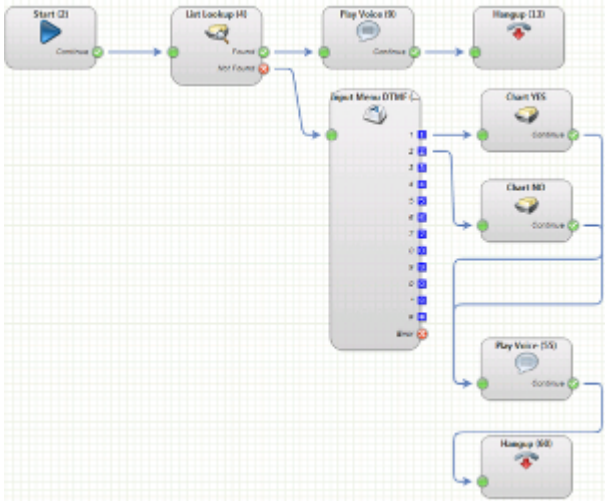
Proceed as follows:

1. First of all, create a list. Lists are created by accessing the function **System Settings - Lists** in the main menu. Use **New** to create a new list, and name it *Callers*. You will not need to create any entries - values will be added to the list when people call the system.
2. Create a new routing application. After the *Start* object, add an object *List Lookup*.
3. In the properties of this object select the *Callers* list as the Key Column *1* and enter Key Value *$callerhash*.
   In this example, the variable *$callerhash* is to be looked up in the first column of the list. If an entry is found, program execution will continue with the object output *Found* and the caller should be played a prompt that they have already called the system.
4. Connect the *Found* output to a Play Voice object containing a prompt *You have already voted*.
5. After the play voice object, add a *Hangup* object (optional).
   The program should in principle look like the following:



6. If the value is not found in the list, the output *Not Found* will be used. Add an *Input Menu DTMF* object to this output. In the parameters, setup a prompt with the following announcements: "Press one to vote yes, and press two to vote no."
7. Add two *List Functions* objects to the outputs 1 and 2 of the input menu dtmf object.
8. Call the first object *Vote YES* and parameterise it as follows:
   List: Callers
   Function: Add Value to List
   Value 1: $callerhash
   Value 2: $caller
   Value 3: $input
   Value 4: YES
9. Call the second object *Vote NO* and parameterise it as the first list, for Value 4 enter NO.
10. Add a prompt *Your vote has been counted* after the objects *List Operations* and connect it with both objects.

11. After the play voice object, add a *Hangup* object (optional).
    The program should in principle look like the following:



12. Test the program by calling it.
13. To check the results, use the function **System Settings - Lists** and view the *Callers* list. The 4 fields should be filled with values.
14. In the table of routing applications select the **Counter Statistics** function for the application. A new window opens showing the routing application. The numbers in red indicate how often a particular object has been executed. This gives you a quick way of viewing how many callers have voted *Yes* and how many *No*.