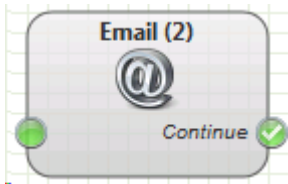


Objects

All functions in the network IVR are carried out using objects. Every object has a name, an input and (optionally) one or more outputs.

An object of a particular type is used to execute particular functions. Every instance of an object needs to be parameterised. For example, an object of type *Email* accesses the functionality to send an email. An instance of this object is parameterised, to send the email to a particular address with a particular subject and content.

Example Email object with Name, Input and Output:



Name

Every object you use should have a unique name, that defines its function and makes it uniquely identifiable by the system. Using objects of the same name is not sensible, it prevents the system from identifying them correctly. The ID in brackets next to the object name is used when the object is added to the workspace to help identify the object uniquely. You should only remove this ID if you are certain, that the object name will continue to be unique.

Input

Every object (except for Start and End Call Actions) has at least one input. This is used to connect it with previous objects in the program.

Outputs

Most objects have one or more outputs. The outputs represent different events and states that can occur when the object is executed.

Connected Outputs

If an output is connected with a further object, the program execution continues with the connected object if the corresponding event occurs during object execution.

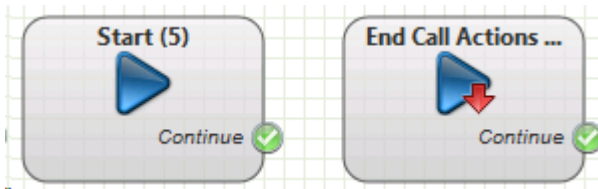
Not Connected Outputs

If an output is not connected, then the behaviour is different depending on the object. In most cases, program execution stops and the telephony connection is hung up. If the program contains *End Call Actions* then these will be executed. In menu objects a not connected output for a DTMF key signifies that this option is not active.

Start and End

The execution of every program starts with an object of type *Start*. Only one *Start* object should be used in an application.

If the system or caller ends the call by hanging up, program execution is interrupted. End Call Actions can be defined which will be executed when the call is done. To define this, the object *End Call Actions* is used. Example: Most callers who leave a message on an answering machine hang up before the maximum recording length is reached. They do not wait to see if the message was recorded or delivered successfully. The action of sending the message must therefore occur after the end of the call itself.



Parameters

All objects can be parameterised to define the objects behaviour. The dialog of object parameters is accessed by double clicking on the object.

Example Parameters for the *Email* object:

The image is a screenshot of a dialog box titled 'Email'. It contains several input fields: 'Object Name :', 'Recipient :', 'Subject :', and 'Message :'. The 'Object Name' field is filled with 'Email (2)'. The 'Recipient' and 'Subject' fields are empty. The 'Message' field is a large text area. At the bottom of the dialog is a 'Close' button.

Reference

You can find a detailed description of all objects in the section [Object Reference](#)