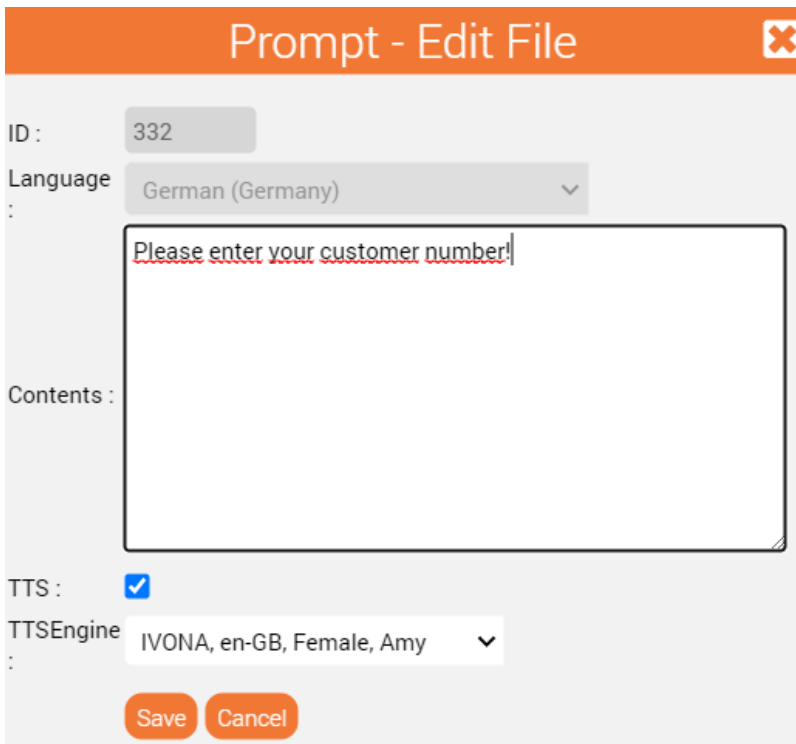# IVR - Input and Verification of a Customer Number

Introduction

This example shows you how you can request customers to enter their customer number when an incoming call is received, check this entry against a list and if entered correctly, serve the customer faster than a customer who has entered no customer number or an incorrect one.

Necessary Prompts

First create the required prompts in the menu under the item **Prompt Management ... Prompts**:

1. Request to enter the customer number.

2. Error message Input too short

New Prompt

Prompt Type :	ACD Announcements ▾ ...
Name :	Input too short
Active :	☑
Content :	The entered customer number is too short or too long.
Language :	English (United Kingdom) ▾
Wave File :	⦿ No Audio ◯ File

Save Cancel

3. Error message customer number wrong

New Prompt

Prompt Type :	ACD Announcements ▾ ...
Name :	Wrong customer number
Active :	☑
Content :	The entered customer number is false.
Language :	English (United Kingdom) ▾
Wave File :	⦿ No Audio ◯ File

Save Cancel

Lists

Create a list containing all customer numbers via the menu **System Settings ... Lists**. The list can also be exported from a CRM system and imported into the system. If you have a large amount of data, it is better to use a REST query.

Field 1 contains the phone number of the customer

Field 2 contains the customer number of the customer

Routing application

Now create a routing application in the menu **Routing ... Routing Applications:**
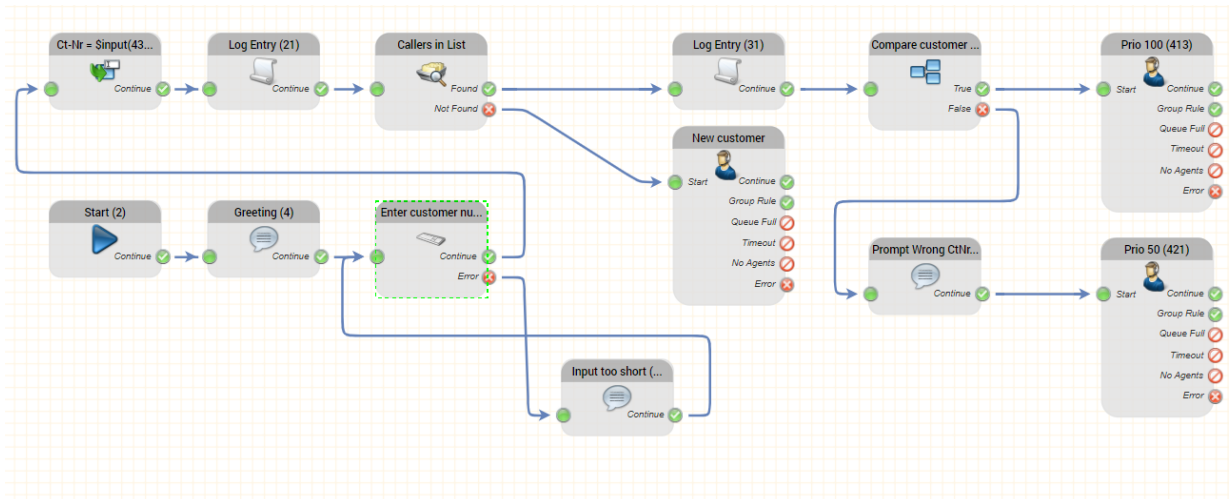
The **"Enter customer number"** object is configured to play a signal tone after the announcement.

Barge In means that the caller does not necessarily have to listen to the announcement until the end before starting the input.

The customer number must have a minimum and maximum of 4 digits and the entry of the customer number must be completed with the '#' key.

Input Digits DTMF

Object Name :	enter customer number (2)
Voice Prompt Type :	ACD Announcements ▾
Voice Prompt :	Enter customer number ▾
Play Tone :	☑
Barge-In :	☑
Minimum Number of Digits :	4 ▾
Maximum Number of Digits :	4 ▾
# stops input :	☑
Timeout no Digit :	4000 ▾
Timeout between Digits :	3000 ▾

Close

The entered sequence of numbers $input is stored in the variable Ct-Nr:

Declare variable

Object Name :	Ct-Nr = $input(431)
Variable Name :	Ct-Nr
Initial value :	$input

Close

The **"List Lookup" object** has the caller number $caller as **key value**, so that it can be determined whether the caller number matches the customer number.

The value of value column 2 from the list is stored in the variable Ct-Nr.

List Lookup

Object Name :	Callers in List (236
List :	Customer Number ▾
Key Column :	1 ▾
Key Value :	$caller
Value Column 1 :	1 ▾
Save to Variable 1 :	
Value Column 2 :	2 ▾
Save to Variable 2 :	Cu-Nr
Value Column 3 :	1 ▾
Save to Variable 3 :	

Close

If the caller $caller is not found in the list, the caller is unknown. The "Not found" output of the object is connected to the object ACD group New customer.

If the caller $caller is found in the list, the "Found" output of the object is connected to the comparison object.

Comparison

Object Name :	Compare customer number (448)
Expression 1 :	$Ct-Nr
Operator :	num == ▾
Expression 2 :	$Cu-Nr

Close

The entered value $Ct-Nr. is compared with the value from the $Cu-Nr. list. There are 2 outputs for the object True or False.

In the event of an error, an announcement is played informing the caller that the customer number was incorrect and connecting him to the group with Prio 50.

If entered correctly, the call will be transferred to the group with Prio 100.

I hope we were able to inspire you - have fun rebuilding!