

Brekeke PBX

Version 2

IVR Script Developer's Guide

Brekeke Software, Inc.

Version

Brekeke PBX Version 2 IVR Script Developer's Guide

Revised: February 2010

Copyright

This document is copyrighted by Brekeke Software, Inc.

Copyright © 2010 Brekeke Software, Inc.

This document may not be copied, reproduced, reprinted, translated, rewritten or readdressed in whole or part without expressed, written consent from Brekeke Software, Inc.

Disclaimer

Brekeke Software, Inc. reserves the right to change any information found in this document without any written notice to the user.

- 1. INTRODUCTION..... 4**

- 2. SETTING UP IVR – SCRIPT..... 4**
 - 2.1. Creating Notes with JavaScript..... 4
 - 2.2. Setting up a user extension as IVR - Script..... 4
 - 2.3. Making a call 5

- 3. IVR METHODS..... 5**
 - void answer() 5
 - void cancelTransfer() 5
 - String clearDTMFBuffer()..... 5
 - boolean connected() 5
 - int countCallsInQueue()..... 5
 - int countCallsInQueue(String queue)..... 5
 - void dropcall()..... 6
 - void exec(String note, String function, String param)..... 6
 - String getLanguage() 6
 - String getMyNumber()..... 6
 - String getOtherNumber() 6
 - String getParameter()..... 6
 - String getProperty(String key) 6
 - String getUserProperty(String key) 7
 - String play(String playlist)..... 7

- String play(string playlist, boolean ignoreDTMF) 7**
- String playAndInput(String playlist, int maxDtmfLength, int timeout, String terminateDtmf) 8**
- void record(String file, int timeout, String terminateDtmf) 8**
- void response18x(int rescode) 9**
- void response18x(int rescode, boolean bSDP)..... 9**
- void setLanguage(String lang)..... 9**
- boolean transfer(String number, int timeout)..... 9**

1. Introduction

This document explains configuration of Brekeke PBX feature "IVR Script". With IVR Script feature, Brekeke PBX administrators can write JavaScript code to customize IVR behavior as they need. To use IVR Script feature, Java 6 and Brekeke PBX license with Script option are required.

2. Setting Up IVR – Script

The example below shows the steps on how to create extension 2020 as IVR-Script user and Brekeke PBX configurations to play customized voice prompt when there is a call to the extension 2020. The customized voice prompt is saved as .ul file.

2.1. Creating Notes with JavaScript

- 1) Go to Brekeke PBX > [Options] > [Notes], and create a Note named "test"
- 2) Put the following sample JavaScript in the text field of Note "test" and save it.

```
function f_1(ivr){  
    ivr.play( "(C:\\<path_to_the_sound_file>\\<file_name>.ul)");  
    ivr.dropcall();  
}
```

2.2. Setting up a user extension as IVR - Script

In this part, an IVR-Script user extension is set up and calls the JavaScript code defined in Note "test" at step 2.1

- 1) Create a PBX user extension 2020
- 2) Select [IVR] field as Script
- 3) Click [OK] at popup window and save the settings, then the [Script Settings] fields will show at the bottom of User 2020 edit page
- 4) set [Script Settings] fields as following:

Note: test (Note name which is created in 2.1)

Function: f_1 (function name which is created in 2.1)

Parameter: no settings for this example

The value set in the field can be passed to function in Notes as the second parameter in function, such as

```
function guidance( ivr, param ){ // actions}
```

Auto Answer: yes (If set no, method answer() is needed in JavaScript)

2.3. Making a call

- 1) Call to extension 2020
- 2) Brekeke PBX will look for the sound file at the location defined in `ivr.play()` in step 2.1 and play the file
- 3) Then drop the call.

3. IVR Methods

void answer()

Description: Answer the call

This method is needed when [Auto Answer] field is set NO at IVR-Script user edit page.

void cancelTransfer()

Description: Back to the conversation with the original party

String clearDTMFBuffer()

Description: Get DTMF signals from the signal buffer

Return: DTMF characters

boolean connected()

Description: Check if the session is connected;

Returns:

Return true if the session has been established;

Return false if disconnected.

int countCallsInQueue()

Description: Get the count of calls in the queue.

This method can be called when this IVR works as a call queue

Return: The number of calls in the queue

int countCallsInQueue(String queue)

Description: Get the count of calls in the queue.

Parameters:

queue: a string of queue name

Return: The number of calls in the queue

void dropcall()

Description: Drop current call

void exec(String note, String function, String param)

Description: Execute another JavaScript function in another note.

Parameters:

note: Note name

function: Function name

param: Only one parameter

String getLanguage()

Description: Get a language code of the current language

Return: Current language code

"en" for English, "ja" for Japanese or another language code if you have custom

String getMyNumber()

Description: Get phone number that is used for current call

Return: The phone number

String getOtherNumber()

Description: Get phone number of the other party of current call

Return: The other party's phone number

String getParameter()

Description: Get a parameter of this call

Return: The parameter of the call in string

String getProperty(String key)

Description: Retrieve a property value from Brekeke PBX.

The parameters may set at [Options] > [Advanced]

Parameters:

Key: a string of a property name

Return: A string containing the property value.

String getUserProperty(String key)

Description: Retrieve a user property value.

Parameters:

Key: a string of property name

Return: A string containing the property value.

String play(String playlist)

Description: Play a sequence of sound files

Parameters:

```
playlist = *play-resource
           play-resource = dtmf-character / prompt / voice-lib / ulaw-file
           dtmf-character = DIGIT / "A" / "B" / "C" / "D" / "*" / "#"
           prompt = "{" prompt-name "}"
           voice-lib = "{" voice-lib-name ":" voice-lib-param "}"
           voice-lib-name = "name" / "date" / "time" / "number"
           voice-lib-param = 1*(ALPHA / DIGIT)
           ulaw-file = "(" fullpath-ulaw-file ")"
```

Returns: DTMF character if a button was pressed

Example:

```
ivr.play("{ring}");
```

Play the file named “ring” which is uploaded to Brekeke PBX from [VoicePrompts]>[System]

```
ivr.play("{ring}1234");
```

First play the file named “ring” which is uploaded to Brekeke PBX from [VoicePrompts]>[System], then play 1234 as DTMF.

String play(string playlist, boolean ignoreDTMF)

Description: Play a sequence of sound files

Parameters:

```
playlist = *play-resource
           play-resource = dtmf-character / prompt / voice-lib / ulaw-file
           dtmf-character = DIGIT / "A" / "B" / "C" / "D" / "*" / "#"
           prompt = "{" prompt-name "}"
           voice-lib = "{" voice-lib-name ":" voice-lib-param "}"
           voice-lib-name = "name" / "date" / "time" / "number"
           voice-lib-param = 1*(ALPHA / DIGIT)
           ulaw-file = "(" fullpath-ulaw-file ")"
```


ignoreDTMF = true or false; Stop playing with DTMF

Returns: DTMF characters if button were pressed. Always empty if "ignoreDTMF" is true.

String playAndInput(String playlist, int maxDtmfLength, int timeout, String terminateDtmf)

Description: Play a sequence of sound files and retrieve DTMF signals.

Parameters:

playlist = *play-resource

play-resource = dtmf-character / prompt / voice-lib / ulaw-file

dtmf-character = DIGIT / "A" / "B" / "C" / "D" / "*" / "#"

prompt = "{" prompt-name "}"

voice-lib = "{" voice-lib-name ":" voice-lib-param "}"

voice-lib-name = "name" / "date" / "time" / "number"

voice-lib-param = 1*(ALPHA / DIGIT)

ulaw-file = "(" fullpath-ulaw-file ")"

maxDtmfLength:

Maximum length of DTMF signals

timeout:

Length of time (milliseconds) for detecting DTMF signals

terminateDtmf:

DTMF characters used to terminate to retrieve DTMF input

Returns: Detected DTMF signals

Example:

```
ivr.playAndInput("{call}", 10, 5000, "#");
```

Play the file named "call" and detect DTMF input within 5000 millisecond (5s) after playing sound file. Input "#" at the end to terminate DTMF detection if the number of input digits is less than 10.

void record(String file, int timeout, String terminateDtmf)

Description: Record sound data

Parameters:

file: File name to store the recorded sound

timeout: length of record time (milliseconds)

terminateDtmf: DTMF characters input to terminate recording

void response18x(int rescode)

Description: Send 18x response without SDP

This method can be called before method answer()

Parameters:

rescode: Response code. Set 180 or 183.

void response18x(int rescode, boolean bSDP)

Description: Send 18x response.

This method can be called before method answer()

Parameters:

rescode: Response code. Set 180 or 183.

bSDP: Add SDP or not to the 18x response

void setLanguage(String lang)

Description: Set a language code for current language

Parameters:

lang: Language code. such as "en", "ja" or another language code if you have custom

boolean transfer(String number, int timeout)

Description: Start attended transfer

Parameters:

number: Destination phone number

timeout: the length of time to ring the transfer recipient destination (Milliseconds)

Return: Return true if get connection for the transfer destination; otherwise, return false.