VBPHONE ActiveX

API Reference Guide

Version 1.00(Beta)

COPYRIGHTS

TRADEMARKS

DISCLAIMER

This document is provided to you for informational purposes only and is believed to be accurate as of the date of its publication, and is subject to change without notice. VbPhonex Inc. assumes no responsibility for any errors or omissions in this document and shall have no obligation to you as a result of having made this document available to you or based upon the information it contains.

Prerequisites:

The configuration depends on number of devices that will operate at the same time.
It depends on RAM heavily, especially for the buffers of played files. Each file is fully red into RAM before play.
This the most important request from the program.
The cards must be configured to have one “Voice Pack” available for each phone and line on the cards, in case you use built in macros.

Minimum hardware requirements:

CPU Pentium 500mhz class processor
RAM 256Mb
MonteCarlo 2.7(and one or more 2.7 supported PIKA cards)
Hard Disk: 1 MB + space for application.
One additional MB or RAM for each simultaneous playing/recording channel.

Optimum hardware requirements:

CPU Pentium 1000mhz class processor
RAM 512Mb
MonteCarlo 2.7(and one or more 2.7 supported PIKA cards)
Hard Disk: 1 MB + space for application.
One additional MB or RAM for each simultaneous playing/recording channel.

Methods:

The methods will return a value when there is a result. The call was success if the returned value is 0 otherwise the method returns an error.

Note : If the method returns a negative value then it is related to PIKA® API reference
If the method returns a positive value other than 0 it is a procedural error in ActiveX.
Both these error references can be found in reference material.

Call names starting with Phone are related to Phones(POTS or agents analog terminals).
Call names starting with Trunk are related to Trunks, both digital and analog.

EnableDevice

Syntax:

long EnableDevice(CardN, TrunkN, LineN, DeviceType, PhoneNumber as string)

Description:

The call is used **BEFORE** calling Initialise to select the devices that will be activated in the next call to Initialise. It must(when needed) be called for each device (Trunk or Phone) you need to use. The enumeration of Trunks/Phones will be altered by this call, unused lines/channels will not counted.
After the call to Initialise another call to EnableDevice has no effect. If case this method is not called, Initialise function will enable all devices present(and defined in MCSetup) at their full capabilities.
The first card is referred to as card 0 (see MCSetup)

Trunks entry on each Card and Lines start with 0.
Analog Cards have Trunk 0 entry and lines ranging from 0 to MaxLines-1.
If Device Type is 0 then it is called Line Entry.
If Device Type > 0 then it is called Phone Entry.

Combined devices (Lines/Phones) have separate enumeration, each having its own group.
Any Kind of card mixture is supported.

Phone Number entry is to relate an analog line to a particular number. So in case of incoming call the event will report this number as the called line number.

Terminating events:

NONE

Initialise

Syntax:

long Initialise(SetDspResources as integer)

Description:

This method allows initialisation of system resources(Trunks, Phones, DSP. MVIP, etc). The resources(like Phones, Trunks including digital ports) will be numbered starting from 1 when the initialisation method founds first card on the system.

The enumeration is separate for phones and trunks (For eg. Phone #1 and Trunk #1) Every method or event except for DSP resources is referred to such an enumeration.

DSP resources cannot exist free from phones or trunks. If they are created they are "associated" to a trunk or phone then further references to them is done by their "associated" device (Phone or Trunk).

The parameter "SetDSPResources" will have a set. If set > 0 then the initialisation process will create the default DSP resource and associate them to every phone and trunk.

In the beta version, if the chosen protocol for a digital trunk needs robbed bit signaling, the DSP port associated to the trunk is dedicated for this purpose till it is freed from this function with proper method. The trunks are then enabled for incoming call signaling.

Terminating events:

NONE

PhoneConnect

Syntax:

long PhoneConnect(Resource1, Resource2, Type)

Description:

This method allows to connect resources via CTBus. The argument value type defines the connection to do on the bus.

These are Full Duplex Connections, if a connection is active then the phone for this connection will be disconnected.

Type entry	Resource 1	Resource 2
1	Phone Resource #	Associated DPS resource
2	Phone Resource #	Phone Resource #
3	Phone Resource #	Trunk Resource #
0	No connection	

Terminating events:

NONE

TrunkConnect

Syntax:

long TrunkConnect(Resource1, Resource2, Type)

Description:

This method allows to connect resources via CTBus. The argument value type defines the connection to do on the bus.

These are Full Duplex Connections, if a connection is active then the Trunk for this connection will be disconnected.

Type entry	Resource 1	Resource 2
1	Trunk Resource #	Associated DPS resource
2	Trunk Resource #	Trunk Resource #
3	Trunk Resource #	Phone Resource #
0	No connection	

Note : In one case the connection method returns the same result.
For eg. PhoneConnect(7,4,3) gets the same result as TrunkConnect(4,7,3)
It connects to Phone#7 with Trunk#3.

Terminating events:

NONE

PhoneDisconnect

Syntax:

long PhoneDisconnect(Phone resource#)

Description:

Disconnects the resource from the bus and the resource previously connected. No action if not connected.
This function closes the devices opened previously by using Connect().

Terminating events:

NONE

TrunkDisconnect

Syntax:

long TrunkDisconnect(Trunk resource#)

Description:

Disconnects the resource from the bus and the resource previously connected. No action if not connected.

This function closes the devices opened previously by using Connect().

Terminating events:

NONE

PhoneGetHandle

Syntax:

long PhoneGetHandle (Phone resource#)

Description:

Gets the port handle of Phone resource as return value. This handle is used for reference purpose

Terminating events:

NONE

TrunkGetHandle

Syntax:

long TrunkGetHandle (Trunk resource#)

Description:

Gets the port handle of Trunk resource as return value. This handle is used for reference purpose

Terminating events:

NONE

PhoneGetDspHandle

Syntax:

long PhoneGetDspHandle (Phone resource#)

Description:

Gets the port handle of DSP Associated to phone resource as return value. This handle is used for reference purpose

Terminating events:

NONE

TrunkGetDspHandle

Syntax:

long TrunkGetDspHandle (Trunk resource#)

Description:

Gets the port handle of DSP Associated to Trunk resource as return value. This handle is used for reference purpose

Terminating events:

NONE

PhonePlay

Syntax:

long PhonePlay(VoiceFileName as string, Phone resource#, format as integer)

Description:

This function plays recorded voice data to the phone using the associated DSP vocal resource. If the resource is not connected it's connected automatically and eventually connected resource is disconnected. A buffer equals to the size of the file is created in memory and maintained automatically.

The audio file format is defined by the format greater than 0. By default format is equal to 0 and it is 8 Bit PCM encoding format with a sampling rate of 6KHz.

Terminating events:

PhoneEndPlay

TrunkPlay

Syntax:

long TrunkPlay(VoiceFileName as string, Trunk resource#, format as integer)

Description:

This function plays recorded voice data to the Trunk using the associated DSP vocal resource. If the resource is not connected it's connected automatically and eventually connected resource is disconnected. A buffer equals to the size of the file is created in memory and maintained automatically.
The audio file format is defined by the format greater than 0. By default format is equal to 0 and it is 8 Bit PCM encoding format with a sampling rate of 6KHz.

Terminating events:

TrunEndPlay

PhoneRecord

Syntax:

long PhoneRecord(VoiceFileName as string, maxbytes as integer, Phone resource#, format as integer)

Description:

This function records voice data from a phone # using the associated DSP vocal resource. If the resource is not connected it's connected automatically eventually connected resource is disconnected.

A buffer of proper size is created in the memory of the system and maintained automatically. If the buffer is full then recording automatically stopped and the file is closed.

The audio file format is defined by the format greater than 0. BY default format is equal to 0 and it is 8 Bit PCM encoding format with a sampling rate of 6KHz.

Terminating events:

PhoneEndRecord

TrunkRecord

Syntax:

long TrunkRecord(VoiceFileName as string, maxbytes as integer, Trunk resource#, format as integer)

Description:

This function records voice data from a Trunk # using the associated DSP vocal resource. If the resource is not connected it's connected automatically eventually connected resource is disconnected.

A buffer of proper size is created in the memory of the system and maintained automatically. If the buffer is full then recording automatically stopped and the file is closed.

The audio file format is defined by the format greater than 0. BY default format is equal to 0 and it is 8 Bit PCM encoding format with a sampling rate of 6KHz.

Terminating events:

TrunkEndRecord

PhoneRing

Syntax:

long PhoneRing(Phone resource#, ringtype, duration)

Description:

This function generates a ring on phone #. Ring type ranges from 0 to 15. The duration of the ring will be in milliseconds. If the duration is 0 then the function generates endless rings.

Terminating events:

PhoneNoAnswer

TrunkMakeCall

Syntax:

long TrunkMakeCall(Number as string, Trunk resource#)

Description:

This function allows to make a call on Trunk. If the trunk is analog then first it takes OffHook and makes a call. Once the call is terminated it will be set back to OnHook.

Terminating events:

TrunkCallConnected	**(Positive connection)**
TrunkCallFailed	**(Negative connection)**

TrunkTerminateCall

Syntax:

long TrunkTerminateCall(Trunk resource#)

Description:

This function allows to terminate a call on Trunk. If the trunk is analog then first it takes OnHook and terminates a call.

.

Terminating events:

TrunkCallTerminated

TrunkAcceptCall

Syntax:

long TrunkAcceptCall(Trunk resource#)

Description:

This function accepts an incoming call on Trunk(both digital and analog).

Terminating events:

TrunkAnswered
TrunkCallTerminated

TrunkAssociateDsp (Function not implemented)

Syntax:

long TrunkAssociateDsp(Trunk resource#)

Description:

This method enable a Trunk Associated DSP port to handle protocols requiring in band signalling.

Terminating events:

NONE

TrunkSetOnHook

Syntax:

long TrunkSetOnHook(Trunk resource#)

Description:

This method sets the Trunk status to OnHook if the device is analog. If digital it does nothing.
Normally this task is handled by the other MACROs, so it's not required in normal operation.

Terminating events:

TrunkOnHook

TrunkSetOffHook

Syntax:

long TrunkSetOffHook(Trunk resource#)

Description:

This method sets the Trunk status to OffHook if the device is analog. If digital it does nothing.
Normally this task is handled by the other MACROs, so it's not required in normal operation.

Terminating events:

TrunkOffHook

PhoneGenerateSound

Syntax:

long PhoneGenerateSound(sound #, Phone resource#)

Description:

This method generates a sound(sound #) on the phone#

.

Terminating events:

PhoneEndSound

TrunkGenerateSound

Syntax:

long TrunkGenerateSound(sound #,Trunk resource#)

Description:

This method generates a sound(sound #) on the Trunk#

.

Terminating events:

PhoneEndSound

PhoneStopVoice

Syntax:

long PhoneStopVoice(Phone resource#)

Description:

This method terminates playing the voice file on Phone#.

Terminating events:

PhoneEndVoice

TrunkStopVoice

Syntax:

long TrunkStopVoice(Phone resource#)

Description:

This method terminates playing the voice file on Trunk#.

Terminating events:

TrunkEndVoice

PhoneStopSound

Syntax:

long PhoneStopSound(Phone resource#)

Description:

This method terminates playing a sound on Phone#.

Terminating events:

TrunkEndSound

TrunkStopSound

Syntax:

long TrunkStopSound(Trunk resource#)

Description:

This method stops playing the sound on Trunk#.

Terminating events:

TrunkEndSound

PhoneDeleteDspPort

Syntax:

long PhoneDeleteDspPort(Phone resource#)

Description:

This method deletes DSP resources associated to phone# and disconnects from the bus. To use this method the resource must be idle.

Terminating events:

NONE

TrunkDeleteDspPort

Syntax:

long TrunkDeleteDspPort(Phone resource#)

Description:

This method deletes DSP resources associated to Trunk # and disconnects from the bus. To use this method the resource must be idle.

Terminating events:

NONE

PhoneEnableDtmfDetection

Syntax:

long PhoneEnableDtmfDetection(Phone resource#)

Description:

This method enables detection of DTMF digits on the associated DSP resource.

Terminating events:

NONE

TrunkEnableDtmfDetection

Syntax:

long TrunkEnableDtmfDetection (Trunk resource#)

Description:

This method enables detection of DTMF digits on the associated DSP resource.

Terminating events:

NONE

PhoneDisableDtmfDetection

Syntax:

long PhoneDisableDtmfDetection(Phone resource#)

Description:

This method enables detection of DTMF digits on the associated DSP resource.

Terminating events:

NONE

TrunkDisableDtmfDetection

Syntax:

long TrunkDisableDtmfDetection(Phone resource#)

Description:

This method enables detection of DTMF digits on the associated DSP resource.

Terminating events:

NONE

Events for Phone and associated DSP ports:

PhoneOnHook (Phone resource#)

This event informs Phone # went OnHook.

PhoneOffHook (Phone resource#)

This event informs Phone # went OffHook.

PhoneHookFlash (Phone resource#)

This event informs Phone # send HookFlash.

PhoneEndPlay (Phone resource#)

This event informs Phone # ended playing a voice file (or voice file play is cancelled)

PhoneEndSound (Phone resource#)

This event informs Phone # ended playing a sound (or sound is cancelled)

PhoneEndRecord (Phone resource#)

This event informs Phone # ended recording a voice file (or voice file recroding is cancelled)

PhoneNoAnswer (Phone resource#)

This event informs Phone # timed out while ringing with no answer.

PhoneBusy (Phone resource#)

This event returns if phone # is busy.

PhoneDtmfDetected (Phone resource#, digit)

This event is generated when a DTMF Digit is detetected. The “digit” is the character digited.

Events for Trunk and associated DSP ports:

TrunkInCall (Trunk resource#, CallingNo as String, CalledNo as String)

This method informs Trunk # has an incoming call. The calling number and called number are reported if available, otherwise returned blank.

TrunkAnswered (Trunk resource#)

This method informs the Trunk # answered an incoming call.

TrunkCallTerminated (Trunk resource#)

This method informs Trunk # has terminated a call.

TrunkSyncOff (Trunk resource#)

This method informs Trunk # loose its sync with CO.

TrunkSyncOn (Trunk resource#)

This method informs Trunk # is sync with CO.

TrunkOffHook (Trunk resource#)

This method informs Trunk # gone OffHook.

TrunkOnHook (Trunk resource#)

This method informs Trunk # gone OnHook.

TrunkEndPlay (Trunk resource#)

This method informs Trunk # ended playing a voice file(or voice file play cancelled)

TrunkEndSound (Trunk resource#)

This method informs Trunk # ended playing a sound (or sound is cancelled)

TrunkEndRecord (Trunk resource#)

This method informs Trunk # ended Recording a voice file(or voice file recording cancelled)

TrunkDtmfDetected (Trunk resource#, digit)

This method generates a DTMF Digit.

TrunkCallFailed(Trunk Resource#, Reason as integer)

This method informs the Trunk # didn't complete the call dialing.

TrunkCallConnected(Trunk Resource#)

This method informs the Trunk # completed the call dialing successfully.

Return codes from methods' calls:

PARAM1_NOT_OK (2)

Parameter 1 in call non correct or out of bound

NO_RESOURCE_ASSOCIATED (3)

The action requested needs an associated resource but this is absent

PARAM2_NOT_OK (4)

Parameter 2 in call non correct or out of bound.

INVALID_CONNECTION_PARAMETER (5)

Parameter for connection type is wrong or resource is non correctly routed.

RESOURCE_IS_NOT_ROUTED (6)

Attempt to disconnect an unrouted resource.

VOCAL_FILE_NOT_FOUND (10)

The requested vocal file cannot be found

VOCAL_RESOURCE_BUSY (11)

The vocal resource is busy.

UNABLE_TO_ADD_BUFFER (12)

The record/play buffer cannot be added to the pool(should never happen)

UNABLE_TO_START_VOICE (13)

Not able to start a play/record.

PLAY_RECORD_FORMAT_NOT_VALID (15)

Format for play and record

FATAL_GENERAL_ERROR (99)

Unrecoverable error

Summary