

API de Integração VisualBasic 6+

Versão 1.1.3 (25/05/2012)

Índice

Visão Geral	1
Integração de ações e eventos de telefonia	1
Ações	1
Eventos	1
Funcionamento da API	2
Formato	2
Nomenclaturas	2
Como utilizar	2
Funções	2
Aplicação de exemplo	3
Métodos Disponíveis na API	4
Métodos de Controle/Conexão	4
<i>Connect (ActionId, Host, AgentCode)</i>	4
<i>Disconnect</i>	4
<i>FreeCallBacks</i>	4
Funções de Ações	4
<i>doDial (ActionId, To, Name, Queue, BillingGroupId)</i>	4
<i>doTransfer (ActionId, CallId, To)</i>	4
<i>doHangUp (ActionId, CallId)</i>	5
<i>doPause (ActionId, Reason)</i>	5

<i>doUnpause (ActionId)</i>	5
<i>doLogin (ActionId)</i>	5
<i>doLogoff (ActionId)</i>	5
<i>doTag (ActionId, CallId, Tag)</i>	5
<i>doStatus (ActionId, Queue)</i>	5
<i>doVersion (ActionId)</i>	6

Funções de Eventos (callbacks)

<i>onConnect (Date, ActionId)</i>	6
<i>onDial (CallId, Date, Agent, Queue, From, To, CallFilename, ContactName, ActionId)</i>	6
<i>onDialAnswer (CallId, Date)</i>	6
<i>onDialFailure (CallId, Date, CauseId, CauseDescription)</i>	6
<i>onReceive (CallId, Date, Queue, From, To, CallFilename, ContactName, ActionId)</i>	7
<i>onReceiveAnswer (CallId, Date, WaitSeconds)</i>	7
<i>onReceiveFailure (CallId, Date, RingingSeconds)</i>	7
<i>onHangUp (CallId, Date, CauseId, CauseDescription)</i>	7
<i>onLogin (Date, Location, ActionId)</i>	7
<i>onLogoff (Date, Location, Duration, ActionId)</i>	8
<i>onPause (Date, Reason, ActionId)</i>	8
<i>onUnpause (Date, ActionId)</i>	8
<i>onStatus (Status, Location, ActionId)</i>	8
<i>onVersion (ActionId, Version)</i>	8
<i>onError (ActionId, Message)</i>	8

Anexo 1: Tabelas

Tabela 1 - Motivos de Pausa

Tabela 2 - Códigos de Causas de Término de Chamada

Tabela 3 - Estados dos Agentes

Visão Geral

Integração de ações e eventos de telefonia

Esta API tem a finalidade de integrar, através de ações e eventos, a utilização da telefonia pela aplicação integrante. Para realizar a integração de uma forma simples em aplicações VisualBasic 6 ou maior, a Vonix disponibiliza uma DLL universal para Windows para uso interno da aplicação, com uma série de métodos (funções e callbacks). Esta API de integração deve ser utilizada da posição de atendimento, no cliente utilizado pelo agente de callcenter para seu trabalho.

Através dela é possível que o sistema integrante faça pedidos de ações na telefonia diretamente (iniciar nova chamada para o agente, pausar o agente, deslogar o agente da fila, marcar uma chamada com uma tag, etc). Além dos pedidos de ações, através do componente, é possível receber chamadas a métodos específicos para tratamento pela aplicação dos eventos de telefonia que ocorrem na posição de atendimento (recebimento de uma chamada, desligamento de uma chamada, entrada em pausa do agente, saída do agente, etc).

O componente utiliza-se de uma conexão de rede para se conectar ao servidor de integração Vonix e mandar ações e receber eventos. Ele faz as duas direções de comunicação. A conexão é transparente para o desenvolvedor, e se faz através da porta utilizada é a 61613/tcp.

Ações

A primeira direção da comunicação na integração é VOCÊ mandar pedidos de ações para o sistema de telefonia Vonix (são as **ações** citadas no manual). São os métodos iniciados com "do". Quando você executa o método, o sistema de telefonia recebe os dados e executa o pedido da ação na posição de atendimento. Por exemplo, a ação de iniciar uma chamada, que é o método "**doDial**". Toda vez que você executar este método, ele vai originar a chamada para o destino passado. O softphone vai discar essa chamada automaticamente, vinda pelo servidor. Outra ação seria a doPause, para pausar o agente. Quando você executar este método, ele vai colocar o agente em pausa, com o motivo passado como parâmetro no método, para não receber mais chamadas receptivas, contabilizar o tempo de pausa no sistema, etc.

Eventos

A segunda direção da comunicação é o sistema de telefonia sinalizar os eventos que estão ocorrendo na posição para o sistema integrante. São todos os **eventos** de telefonia que ocorrem na posição de atendimento que o componente manda para o seu sistema, se você quiser tomar alguma ação no momento em que o evento ocorre. São os métodos iniciados com "on". Por exemplo, toda vez que chegar uma chamada receptiva na posição de atendimento, o método "**onReceive**" vai ser chamado pelo componente, passando os parâmetros da chamada descritos neste manual. Isto serve, por exemplo, para que você possa abrir a ficha de um cliente que está entrando na posição de atendimento (já ter a ficha do cliente aberta com todo o histórico de atendimentos, etc). Você pode colocar qualquer coisa dentro do método. Estes métodos executados em tempo real pelo componente são chamados de **callbacks**. Existem vários eventos, possibilitando que o seu sistema possa ter total controle sobre a telefonia. Alguns exemplos são quando o atendente inicia uma chamada, quando a chamada que ele está é desligada, quando entra ou sai de pausa, etc. Se você não quiser tratar um evento específico, basta não colocar nada dentro do método callback correspondente.

Funcionamento da API

Formato

A API é distribuída em uma biblioteca universal do Windows (DLL). Assim, a sua aplicação pode chamar diretamente os métodos contidos na API sem a necessidade de nenhuma codificação adicional.

Nomenclaturas

A API segue uma convenção de nomenclatura para facilitar o entendimento de seus métodos. Os métodos iniciados por **'do'** são ações, ou seja alguma solicitação que seu sistema pode fazer ao sistema de telefonia. Estes métodos você pode chamar a qualquer momento na sua aplicação. Os métodos iniciados por **'on'** são funções callbacks, que são executadas automaticamente a qualquer momento pelo componente em resposta a um evento ocorrido na telefonia. Eles devem ser preenchidos com o código que você quer que seja executado.

Como utilizar

Como toda biblioteca DLL do Windows, para utilizar a API é preciso que o arquivo Vonix.dll esteja na mesma pasta do executável da aplicação, ou na pasta /windows/system32 (recomendado). Agora basta declarar os métodos da API em sua aplicação para que elas possam ser usadas, para fazer isto utilize a instrução **Declare**. Lembrando que as declarações devem respeitar os argumentos utilizados na API para que funcionem corretamente.

Para facilitar o entendimento, juntamente com esta documentação existe uma aplicação de exemplo completamente funcional, com seu código-fonte. Além de demonstrar a utilização da API, o arquivo **VonixIntegration.bas** foi especialmente escrito para ser portado para outras aplicações, portanto basta adicioná-lo ao seu projeto e modificar somente o conteúdo dos métodos de callback de acordo com a sua necessidade. Esta prática simplifica muito o processo de integração e é altamente recomendada uma vez que dispensa codificação das chamadas da API. Nele estão contidas todas as declarações das Ações e dos Eventos.

Funções

As funções de ações tem o seu nome iniciado com o prefixo "do". São sempre funções que retornam um valor numérico de acordo com a execução da solicitação. Se a resposta for 0 (zero) a execução foi concluída corretamente se for qualquer outro número ocorreu algum problema durante a execução e neste caso é executada a função callback **onError** onde é passado a descrição do problema, exemplo de chamada para uma ação:

Exemplo de chamada para uma função (ação)

```
' Declarando o metodo como uma chamada para a API
Declare Function doDial Lib "Vonix.dll" (ByVal ActionId As String, ByVal ToNumber As String, ByVal
QueueName As String, ByVal BillingId As Integer) As Long

Dim result as Long
result = doDial('1', '6733180700', 'fila_teste', 1)
```

Em contrapartida as funções do tipo callbacks são executadas a partir de chamadas feitas pela API. Neste caso, caso a função esteja declarada no seu código, ela será chamada pela API para executar o código que estiver dentro assim que o evento ocorrer. Observe o exemplo abaixo:

Exemplo de uso de função callback (evento)

```
' Declaração do Callback onReceive
Private Declare Sub onReceiveEvent Lib "Vonix.dll" Alias "onReceive" (ByVal CallbackEvent As Long)

Call onReceiveEvent(AddressOf onReceive)

Private Sub onReceive(ByVal CallId As String, ByVal strDate As String, ByVal QueueName As String,
ByVal FromNumber As String, ByVal ToNumber As String, ByVal CallFilename As String, ByVal ActionId
As String)
    ' seu código aqui que será executado quando o evento ocorrer na PA
End Sub
```

Aplicação de exemplo

Acompanhando esta documentação existe uma aplicação de exemplo. Além de servir como base para uma melhor compreensão do funcionamento do processo de integração, como já dito anteriormente, o módulo **VonixIntegration.bas** pode ser adicionado diretamente ao seu projeto. Assim, você já terá todas as funções de ação prontas para uso no seu código, e você poderá colocar todo o código do que quer que seja executado nas funções de callback a partir deste arquivo, poupando o trabalho de declarar as funções da API.

Métodos Disponíveis na API

Métodos de Controle/Conexão

Connect (ActionId, Host, AgentCode)

Inicia a conexão com o servidor de integração

Parâmetros:

ActionId: identificador da chamada a função, pode ser qualquer string

Host: endereço IP do servidor de integração

AgentCode: matrícula no sistema de telefonia Vonix do agente logado na aplicação

Disconnect

Encerra a conexão com o servidor de integração

Parâmetros:

Nenhum

FreeCallBacks

Sinaliza para que a DLL libere os eventos de callback. Certifique-se de chamar este método no fechamento da aplicação para evitar que a DLL trave o fechamento aguardando por callbacks

Parâmetros:

Nenhum

Funções de Ações

doDial (ActionId, To, Name, Queue, BillingGroupId)

Origina a discagem de uma chamada para o agente

Parâmetros:

ActionId: identificador da chamada a função, pode ser qualquer string

To: Número do telefone em formato nacional de 10 dígitos, ddd e número (ex: 1133211234, 2133141234, 6733180700)

Name: Nome do destino. Irá aparecer para o agente no softphone

Queue: Id da fila pela qual a chamada será feita

BillingGroupId: Id do centro de custo da chamada. É um inteiro. Caso não use, mandar sempre 1

doTransfer (ActionId, CallId, To)

Transfere uma chamada do agente em andamento para outro destino

Parâmetros:

CallId: Id da chamada que se deseja desligar

To: Número do destino para a chamada (geralmente 4 dígitos)

doHangUp (ActionId, CallId)

Desliga uma chamada do agente em andamento

Parâmetros:

CallId: Id da chamada que se deseja desligar

doPause (ActionId, Reason)

Solicita que o agente entre em pausa. Caso o agente esteja em chamada corrente, a pausa é agendada para ocorrer imediatamente ao desligamento da(s) chamada(s) corrente(s).

Parâmetros:

ActionId: identificador da chamada a função, pode ser qualquer string

Reason: Código do motivo da pausa, veja tabela anexa a este manual (Integer)

doUnpause (ActionId)

Retira o agente de pausa, voltando para o trabalho

Parâmetros:

Nenhum

doLogin (ActionId)

Loga o agente em suas filas de trabalho.

Parâmetros:

Nenhum

doLogoff (ActionId)

Desloga o agente de suas filas de trabalho.

Parâmetros:

Nenhum

doTag (ActionId, CallId, Tag)

Adiciona uma marcação (etiqueta ou tag) a chamada. É útil para marcar as chamadas para facilitar a consulta futura pela operação. Exemplos de tags: “acordo”, “desconhecido”, etc

Parâmetros:

Tag: Texto com a marcação (tag)

doStatus (ActionId, Queue)

Solicita que o sistema de telefonia informe o estado do agente em uma determinada fila. A resposta a este pedido virá na execução da função callback onStatus, que será executada em seguida a chamada desta função

Parâmetros:

Queue: Fila do agente em que se deseja saber o estado

doVersion (ActionId)

Solicita a versão do componente usado. A resposta a este pedido virá na execução da função callback onVersion, que será executada em seguida a chamada desta função

Parâmetros:

Nenhum

Funções de Eventos (callbacks)

onConnect (Date, ActionId)

Ocorre quando a conexão com o servidor de integração é estabelecida com sucesso

Parâmetros:

Date: Data da conexão

ActionId: Identificador do pedido de ação original (o mesmo enviado no Connect)

onDial (CallId, Date, Agent, Queue, From, To, CallFilename, ContactName, ActionId)

Ocorre quando o agente realiza uma chamada ativa. O evento ocorre em qualquer chamada iniciada pelo agente, sendo ela manual ou pela API

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia

Date: Data/hora do início da chamada

Agent: Matrícula do agente que está efetuando a chamada

Queue: Id da fila a qual a chamada pertence

From: Número de origem da chamada (ramal da posição de atendimento)

To: Número de destino da chamada, em formato nacional de 10 dígitos

CallFilename: Nome do arquivo de audio da gravação desta chamada no sistema de telefonia Vonix

ContactName: Nome do contato (caso tenha sido passado)

ActionId: Identificar da ação de início da chamada, caso tenha sido solicitada ou feita pelo discador

onDialAnswer (CallId, Date)

Ocorre quando a chamada em curso é atendida. O evento ocorre em qualquer chamada iniciada pelo agente, sendo ela manual ou pela API

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia Date: Data/hora do fim da chamada

onDialFailure (CallId, Date, CauseId, CauseDescription)

Ocorre quando a chamada em curso não é atendida. O evento ocorre em qualquer chamada iniciada pelo agente, sendo ela manual ou pela API

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia

Date: Data/hora do fim da chamada

CauseId: Identificador da causa de desligamento (ver tabela anexa a este manual)

CauseDescription: Descrição da causa de desligamento

onReceive (CallId, Date, Queue, From, To, CallFilename, ContactName, ActionId)

Ocorre quando o agente recebe uma chamada em sua posição de atendimento. A chamada pode ser receptiva ou vinda do discador automático. O que diferencia as duas é que no caso da chamada receptiva, não virá o parâmetro ActionId, e na chamada vinda do discador, o ActionId vai vir preenchido com o Id do contato enviado na alimentação

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia

Date: Data/hora do início da chamada

Queue: Id da fila a qual a chamada pertence

From: Número de origem da chamada, em formato nacional de 10 dígitos

To: Número de destino da chamada (ramal da posição de atendimento)

CallFilename: Nome do arquivo de audio da gravação desta chamada no sistema de telefonia Vonix

ContactName: Nome do contato (caso tenha sido passado)

ActionId: Identificador da ação de início da chamada, caso tenha sido feita pelo discador, Id do contato alimentado

onReceiveAnswer (CallId, Date, WaitSeconds)

Ocorre quando o agente atende a chamada recebida em sua posição de atendimento (tanto receptiva como vinda do discador)

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia

Date: Data/hora do fim da chamada

WaitSeconds: Segundos que a chamada ficou na fila de espera até ser atendida

onReceiveFailure (CallId, Date, RingingSeconds)

Ocorre quando o agente não atende a chamada recebida que foi oferecida a ele (tanto receptiva como vinda do discador)

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia

Date: Data/hora do fim da chamada

WaitSeconds: Segundos que a chamada tocou na posição do agente até que ele não atendesse

onHangUp (CallId, Date, CauseId, CauseDescription)

Ocorre no encerramento da chamada em curso, seja ela receptiva, ativa ou vinda do discador automático

Parâmetros:

CallId: Id único da chamada dentro do sistema de telefonia

Date: Data/hora do fim da chamada

CauseId: Identificador da causa de desligamento (ver tabela anexa a este manual)

CauseDescription: Descrição da causa de desligamento

onLogin (Date, Location, ActionId)

Ocorre quando o agente se loga para trabalhar

Parâmetros:

Date: Data/hora da entrada do agente

Location: Ramal da posição de atendimento que o agente se logou

ActionId: Identificador da ação que logou o agente, se ele foi logado por ação da API

onLogoff (Date, Location, Duration, ActionId)

Ocorre quando o agente se desloga do trabalho

Parâmetros:

Date: Data/hora da saída do agente

Location: Ramal da posição de atendimento que o agente estava

Duration: duração da sessão de trabalho do agente em segundos

ActionId: Identificador da ação que logou o agente, se ele foi logado por ação da API

onPause (Date, Reason, ActionId)

Ocorre quando o agente entra em pausa

Parâmetros:

Date: Data/hora da entrada em pausa do agente

Reason: Id do motivo da pausa do agente

ActionId: Identificador da ação que logou o agente, se ele foi logado por ação da API

onUnpause (Date, ActionId)

Ocorre quando o agente sai da pausa (volta ao trabalho)

Parâmetros:

Date: Data/hora da saída de pausa do agente

ActionId: Identificador da ação que logou o agente, se ele foi logado por ação da API

onStatus (Status, Location, ActionId)

Ocorre indicando o estado do agente no momento, ou em resposta a ação doStatus

Parâmetros:

Status: Estado do agente (ver tabela anexa a este manual)

Location: Posição de atendimento atual do agente

ActionId: Identificador da ação doStatus que gerou o evento, caso tenha sido resposta

onVersion (ActionId, Version)

Ocorre em resposta a ação doVersion

Parâmetros:

ActionId: Identificador da ação doVersion que gerou o evento

Version: Versão da API

onError (ActionId, Message)

Ocorre no caso de algum erro, tanto no componente como em alguma ação pedida

Parâmetros:

ActionId: Identificador da ação que gerou o erro, caso seja erro vindo de uma ação

Message: Mensagem de erro

Anexo 1 : Tabelas

Tabela 1 - Motivos de Pausa

ID	Motivo
1	Avaliação
2	Treinamento
3	Lanche
4	Toalete
5	Ginástica Laboral
6	Erro no Sistema

Tabela 2 - Códigos de Causas de Término de Chamada

ID	Causa de Desligamento
0	Cancelado
1	Número Vago
2	Sem Rota Operadora
3	Sem Rota Destino
4	Número com Problema
5	Inclusão DDD Errada
6	Canal Inválido
8	Cancelado Operadora
16	Atendida (desligamento normal)
17	Ocupado
18	Não Responde
19	Não Responde
20	Fora de Área
21	Chamada Rejeitada pela Operadora
22	Número Mudou
23	A Cobrar Rejeitado
26	Término Inesperado
27	Destino com Defeito
28	Formato Inválido
29	Serviço Indisponível
31	Desconectado da Rede
34	Sem Canal Disponível
38	Falha de Rede
41	Falha Temporária
43	Negado pela Rede
53	Destino Bloqueado

55	Destino Bloqueado
91	DDD Inválido
102	Tempo Esgotado
127	Falha de Interconexão

Tabela 3 - Estados dos Agentes

Estado	Descrição
ONLINE	Agente logado e disponível
OFFLINE	Agente deslogado
PAUSED	Agente em pausa
UNAVAILABLE	Agente indisponível
RINGING	Agente em oferecimento de chamada
ONTHEPHONE	Agente ao telefone