



# API de Integração DELPHI 7+

Versão 1.1.3 (25/05/2012)

# Índice

<b>Visão Geral</b>	<b>1</b>
<b>Integração de ações e eventos de telefonia</b>	<b>1</b>
Ações	1
Eventos	1
<b>Funcionamento da API</b>	<b>2</b>
<b>Formato</b>	<b>2</b>
<b>Nomenclaturas</b>	<b>2</b>
<b>Como utilizar</b>	<b>2</b>
Utilizando o arquivo Integration.pas em sua aplicação	2
Aplicação de exemplo	3
<b>Métodos Disponíveis na API</b>	<b>4</b>
<b>Métodos de Controle/Conexão</b>	<b>4</b>
Connect ( <i>stActionId, stHost, stAgentCode</i> )	4
Disconnect	4
FreeCallBacks	4
<b>Funções de Ações</b>	<b>4</b>
doDial ( <i>stActionId, stTo, stName, stQueue, wdBillingGroupId</i> )	4
doTransfer ( <i>stActionId, stCallId, stTo</i> )	4
doHangUp ( <i>stActionId, stCallId</i> )	5
doPause ( <i>stActionId, wdReason</i> )	5

<i>doUnpause (stActionId)</i>	5
<i>doLogin (stActionId)</i>	5
<i>doLogoff (stActionId)</i>	5
<i>doTag (stActionId, stCallId, stTag)</i>	5
<i>doStatus (stActionId, stQueue)</i>	6
<i>doVersion (stActionId)</i>	6
<b>Funções de Eventos (callbacks)</b>	6
<i>onConnect (stDate, stActionId)</i>	6
<i>onDial (stCallId, stDate, stAgent, stQueue, stFrom, stTo, stCallFilename, stContactName, stActionId)</i>	6
<i>onDialAnswer (stCallId, stDate)</i>	6
<i>onDialFailure (stCallId, stDate, wdCauseId, stCauseDescription)</i>	7
<i>onReceive (stCallId, stDate, stQueue, stFrom, stTo, stCallFilename, stContactName, stActionId)</i>	7
<i>onReceiveAnswer (stCallId, stDate, wdWaitSeconds)</i>	7
<i>onReceiveFailure (stCallId, stDate, wdRingingSeconds)</i>	7
<i>onHangUp (stCallId, stDate, wdCauseId, stCauseDescription)</i>	7
<i>onLogin (stDate, stLocation, stActionId)</i>	8
<i>onLogoff (stDate, stLocation, wdDuration, stActionId)</i>	8
<i>onPause (stDate, wdReason, stActionId)</i>	8
<i>onUnpause (stDate, stActionId)</i>	8
<i>onStatus (stStatus, stLocation, stActionId)</i>	8
<i>onVersion (stActionId, stVersion)</i>	8
<i>onError (stActionId, stMessage)</i>	9
<b>Anexo 1: Tabelas</b>	10
Tabela 1 - Motivos de Pausa	10
Tabela 2 - Códigos de Causas de Término de Chamada	10

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 7 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 apontando para a DLL.

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 **Integration.pas** 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.

### Exemplo de declaração a uma DLL

```
procedure EventConnect(clientConnect: TecConnect); stdcall external 'Vonix.dll' name 'onConnect';
```

## Utilizando o arquivo Integration.pas em sua aplicação

Para facilitar a inclusão do componente em sua aplicação, você pode usar o arquivo Integration.pas, utilizado no exemplo anexo a este manual, para centralizar a declaração dos métodos da integração. Primeiramente devemos adicionar a Unit a sua aplicação usando o atalho (Shift + F11) e selecionando o arquivo Integration.pas (é recomendado que o mesmo seja copiado para o diretório do projeto).

Depois inclua a declaração 'Uses integration;' logo após a diretiva 'Implementation'.

Agora basta chamar as ações de acordo com a regra de negócio de sua aplicação. Exemplo:

### Exemplo de uso de função de ação

```
// Esta ação inicia a conexão com o servidor de integração Vonix no IP 192.168.0.1
// para o agente com matrícula 1008
Integration.Connect('192.168.0.1', '1008');
```

Como resposta a conexão bem sucedida o sistema de telefonia dispara o evento onConnect, que em nosso exemplo executa um método em frmSample:

### Exemplo de uso de função de ação

```
procedure onConnect(stDate, stActionId: PChar);
begin
  frmSample.AddEvent('Connect', 'action_id: ' + stActionId);
end;
```

ATENÇÃO: No término da sua aplicação certifique-se de incluir a chamada FreeCallBacks para liberar os eventos vinculados à DLL. A omissão desta chamada pode gerar um erro de memória no Windows e impedir que a aplicação termine corretamente.

## Aplicação de exemplo

Acompanhando esta documentação existe uma aplicação de exemplo em Delphi completamente funcional, com seu código-fonte. Além de servir como base para uma melhor compreensão do funcionamento do processo de integração, como já dito anteriormente, o módulo **Integration.pas** 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 (stActionId, stHost, stAgentCode)**

Inicia a conexão com o servidor de integração

*Parâmetros:*

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

stHost: endereço IP do servidor de integração

stAgentCode: 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 (stActionId, stTo, stName, stQueue, wdBillingGroupId)**

Origina a discagem de uma chamada para o agente

*Parâmetros:*

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

stTo: Número do telefone em formato nacional ddd e número (ex: 1133211234, 2133141234, 6733180700)

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

stQueue: Id da fila pela qual a chamada será feita

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

### **doTransfer (stActionId, stCallId, stTo)**

Transfere uma chamada do agente em andamento para outro destino. Este método termina a chamada na posição atual, mas sem derrubar a chamada, que vai ser redirecionada para o destino solicitado

**Parâmetros:**

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

stCallId: Id da chamada que se deseja desligar (pode ser passado em branco, transferindo assim qualquer chamada em andamento no agente)

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

**doHangUp (stActionId, stCallId)**

Desliga uma chamada do agente em andamento

**Parâmetros:**

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

stCallId: Id da chamada que se deseja desligar (pode ser passado em branco, desligando assim qualquer chamada em andamento no agente)

**doPause (stActionId, wdReason)**

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:**

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

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

**doUnpause (stActionId)**

Retira o agente de pausa, voltando para o trabalho

**Parâmetros:**

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

**doLogin (stActionId)**

Loga o agente em suas filas de trabalho.

**Parâmetros:**

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

**doLogoff (stActionId)**

Desloga o agente de suas filas de trabalho.

**Parâmetros:**

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

**doTag (stActionId, stCallId, stTag)**

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:**

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

stTag: Texto com a marcação (tag)

### **doStatus (stActionId, stQueue)**

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:*

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

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

### **doVersion (stActionId)**

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:*

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

## **Funções de Eventos (callbacks)**

### **onConnect (stDate, stActionId)**

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

*Parâmetros:*

stDate: Data da conexão

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

### **onDial (stCallId, stDate, stAgent, stQueue, stFrom, stTo, stCallFilename, stContactName, stActionId)**

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

*Parâmetros:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do início da chamada

stAgent: Matrícula do agente que está efetuando a chamada

stQueue: Id da fila a qual a chamada pertence

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

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

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

stContactName: Nome do contato (caso tenha sido passado)

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

### **onDialAnswer (stCallId, stDate)**

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

*Parâmetros:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do fim da chamada

#### **onDialFailure (stCallId, stDate, wdCauseId, stCauseDescription)**

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:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do fim da chamada

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

stCauseDescription: Descrição da causa de desligamento

#### **onReceive (stCallId, stDate, stQueue, stFrom, stTo, stCallFilename, stContactName, stActionId)**

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:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do início da chamada

stQueue: Id da fila a qual a chamada pertence

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

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

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

stContactName: Nome do contato (caso tenha sido passado)

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

#### **onReceiveAnswer (stCallId, stDate, wdWaitSeconds)**

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

##### *Parâmetros:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do fim da chamada

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

#### **onReceiveFailure (stCallId, stDate, wdRingSeconds)**

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

##### *Parâmetros:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do fim da chamada

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

#### **onHangUp (stCallId, stDate, wdCauseId, stCauseDescription)**

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

##### *Parâmetros:*

stCallId: Id único da chamada dentro do sistema de telefonia

stDate: Data/hora do fim da chamada

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

stCauseDescription: Descrição da causa de desligamento

**onLogin (stDate, stLocation, stActionId)**

Ocorre quando o agente se loga para trabalhar

*Parâmetros:*

stDate: Data/hora da entrada do agente

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

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

**onLogoff (stDate, stLocation, wdDuration, stActionId)**

Ocorre quando o agente se desloga do trabalho

*Parâmetros:*

stDate: Data/hora da saída do agente

stLocation: Ramal da posição de atendimento que o agente estava

wdDuration: duração da sessão de trabalho do agente em segundos

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

**onPause (stDate, wdReason, stActionId)**

Ocorre quando o agente entra em pausa

*Parâmetros:*

stDate: Data/hora da entrada em pausa do agente

wdReason: Id do motivo da pausa do agente

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

**onUnpause (stDate, stActionId)**

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

*Parâmetros:*

stDate: Data/hora da saída de pausa do agente

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

**onStatus (stStatus, stLocation, stActionId)**

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

*Parâmetros:*

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

stLocation: Posição de atendimento atual do agente

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

**onVersion (stActionId, stVersion)**

Ocorre em resposta a ação doVersion

*Parâmetros:*

stActionId: Identificador da ação doVersion que gerou o evento

stVersion: Versão da API

**onError (stActionId, stMessage)**

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

*Parâmetros:*

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

stMessage: 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**

<b>Estado</b>	<b>Descrição</b>
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