



DLL SC501GER

Projeto

Manual da API

Documento

Projeto	SC501GER	
Documento	Manual do API	

Sumário

Sumário.....	2
1. Inicialização da DLL	4
vInitialize.....	4
tc_startserver.....	4
tc_startserver_port.....	4
_TermGertecServer	5
vFinalize	5
dll_version	5
cInitString.....	6
2. Conversão de Tipos.....	6
TCinet_ntoa	6
TCinet_addr	6
3. Rotinas de Controle dos Terminais Conectados	7
GetTabConectados	7
GetTermConectados.....	8
iTypeTerm	8
4. Comandos de Rede	9
bEnviaVivo	9
bSendAllwaysLive	9
bSendCheckLive	9
bSendRestartSoft	10
bUpdateSoftware.....	10
bMandaConfig	11
bPedeConfig.....	11
bMandaExtConfig	12
bPedeExtConfig.....	13
bSendProdPrice	13
bSendProdPriceColor	13
bSendProdNotFound	14
bSendProdNotFoundColor.....	14
bSendDisplayMsg.....	14
bSendImageFromFile	15
bSendImageGifFromFile	15
bSendImagePrice	16
bSendImageGifPrice.....	16
bSendAudioPrice.....	17
bSendAudioPriceColor	17
bPedeParam.....	18

Projeto	SC501GER	
Documento	Manual do API	

bMandaParam	18
bPedeUpdConfig	18
bMandaUpdConfig	19
bPedeWlanConfig	19
bMandaWlanConfig	20
bPedeMacAddr	21
bSendImageBmpJpgPrice	21
bPedeListMedias	21
bPedeMediasConf	22
bRemoveMedia	22
bSaveMediasConf	22
bSendMedia	23
bMandaConfigAudio	23
bPedeAudioConf	23
bMandaHttpConfig	24
bPedeHttpConf	24
bRemoveAllMedias	25
5. ROTINAS PARA RECEBER DADOS DOS TERMINAIS	25
bReceiveBarcode	25
bReceiveConfig	25
bReceiveExtConfig	26
GetResponseCtr	27
bReceiveParam	27
bReceiveUpdConfig	28
bReceiveWlanConfig	29
bReceiveMacAddrConfig	29
bGetListMedias	30
bGetMediasConf	30
bGetAudioConf	31
bGetHttpConf	31

Projeto	SC501GER	
Documento	Manual do API	

1. Inicialização da DLL

vInitialize

Rotina de inicialização da DLL. Deve ser chamada no início da execução para alocar as estruturas necessárias.

Assinatura
<code>void __stdcall vInitialize();</code>
Parâmetros
Não possui parâmetros.
Retorno
Não retorna se houve erro ou não. Mantém compatibilidade com versões anteriores.

tc_startserver


Inicia o server escutando na porta default (6500).

Assinatura
<code>int __stdcall tc_startserver();</code>
Parâmetros
Não possui parâmetros.
Retorno
Retorna 1 se servidor foi inicializado com sucesso, ou 0 em caso de erro.

tc_startserver_port

Inicia o server escutando em uma porta escolhida.

Assinatura
<code>int __stdcall tc_startserver(int port);</code>
Parâmetros
IN: port: porta do servidor de socket
Retorno
Retorna 1 se servidor foi inicializado com sucesso, ou 0 em caso de erro.

Projeto	SC501GER	
Documento	Manual do API	

_TermGertecServer

Inicia o server escutando na porta default (6500).

Assinatura
<code>void __stdcall _TermGertecServer();</code>
Parâmetros
Não possui parâmetros.
Retorno
Não retorna se houve erro ou não. Mantém compatibilidade com versões anteriores.

vFinalize

Rotina que finaliza o servidor, encerrando as conexões e liberando a memória alocada.

Assinatura
<code>void __stdcall vFinalize();</code>
Parâmetros
Não possui parâmetros.
Retorno
Não retorna se houve erro ou não. Mantém compatibilidade com versões anteriores.

dll_version

Função que retorna versão da DLL.

Assinatura
<code>DWORD __stdcall dll_version();</code>
Parâmetros
Não possui parâmetros.
Retorno
Retorna um DWORD (4 bytes) representando o número da versão. Exemplo: Versão 3.0.0.0 (0x03000000)

Projeto	SC501GER	
Documento	Manual do API	

clnitString

Retorna a string de inicialização do terminal.

Assinatura
<code>const char* __stdcall cInitString(int id);</code>
Parâmetros
IN: id: id do socket terminal.
Retorno
Retorna um char* que aponta para a string recebida do terminal após o servidor enviar #ok.

2. Conversão de Tipos

TCinet_ntoa

Converte um endereço de rede em um endereço IP formatado por pontos.

Assinatura
<code>void __stdcall TCinet_ntoa(DWORD nIP, char *buf);</code>
Parâmetros
IN: nIP: id do socket terminal. OUT: buf: endereço de rede formatado por pontos.
Retorno
Não retorna se houve erro ou não. Mantém compatibilidade com versões anteriores.

TCinet_addr

Converte um IP formatado por pontos em endereço de rede.

Assinatura
<code>DWORD __stdcall TCinet_addr(const char *buf);</code>
Parâmetros
IN: buf: string contendo endereço IP formatado por pontos
Retorno
Retorna um DWORD com o endereço convertido a partir da string fornecida.


Projeto	SC501GER	
Documento	Manual do API	

3. Rotinas de Controle dos Terminais Conectados

GetTabConectados

Retorna uma estrutura com a lista de terminais conectados.

Assinatura
TTABSOCK __stdcall GetTabConectados(int nada);
Parâmetros
IN: nada: Não utilizado, porém é necessário preencher com o valor 1. Mantido por compatibilidade com a versão anterior.
Retorno
<pre>struct TTABSOCK { int tabSock[1024]; DWORD tabIP[1024]; int numSockConec; };</pre> <p>Retorna uma estrutura do tipo TTABSOCK, onde:</p> <p>tabSock é um array com os IDs do socket dos terminais</p> <p>tabIP é um array com os endereços dos terminais</p> <p>numSockConec indica quantos terminais estão conectados</p>

Projeto	SC501GER	
Documento	Manual do API	

GetTermConectados


Retorna uma estrutura com a lista de terminais conectados.

Assinatura
TTABTERM* __stdcall GetTermConectados(int nada, TTABTERM *ptabTerm);
Parâmetros
IN: nada: Não utilizado, porém é necessário preencher com o valor 1. Mantido por compatibilidade com a versão anterior. OUT: ptabTerm: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.
Retorno
<pre>struct TTABTERM { char tabName[256][34]; int tabSock[256]; DWORD tabIP[256]; int tipo[256]; int numSockConec; };</pre> <p>Retorna uma estrutura do tipo TTABTERM, onde:</p> <p>tabName é um array com os nomes dos terminais</p> <p>tabSock é um array com os IDs do socket dos terminais</p> <p>tabIP é um array com os endereços dos terminais</p> <p>tipo é um array com os tipos dos terminais</p> <p>numSockConec indica quantos terminais estão conectados</p>

iTypeTerm

Retorna o tipo de terminal conectado.

Assinatura
int __stdcall iTypeTerm(int id);
Parâmetros
IN: id: id do socket terminal
Retorno
<p>Retorna 1 se o Terminal for um TC502.</p> <p>Retorna 2 se o Terminal for um TC505, TC506 ou TC507.</p> <p>Retorna 0 se o Terminal for um TC501 ou um tipo desconhecido.</p> <p>Retorna 3 se o Terminal for um Busca Preço G2.</p> <p>Retorna 4 se o Terminal for um TC 506 S.</p> <p>Retorna 5 se o Terminal for um Busca Preço G2 E.</p> <p>Retorna 6 se o Terminal for um TC 506 E.</p>

Projeto	SC501GER	
Documento	Manual do API	

Retorna -1 em caso de ID de socket inválido.

4. Comandos de Rede

bEnviaVivo

Envia comando de "vivo" (#live?) para o terminal.

Assinatura
<code>BOOL __stdcall bEnviaVivo(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendAllwaysLive

Envia o comando #alwayslive para o terminal. Ao enviar este comando, o terminal não tenta se reiniciar se a conexão ficar ociosa por mais de 12 segundos. Por padrão, o TC501 versão 2.0 já vem com esta opção habilitada.

Assinatura
<code>BOOL __stdcall bSendAllwaysLive(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendCheckLive

Envia o comando #checklive para o terminal. Em oposição ao comando anterior, ao enviar este comando, o terminal tentará se reiniciar se a conexão ficar ociosa por mais de 12 segundos.

Assinatura
<code>BOOL __stdcall bSendCheckLive(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bSendRestartSoft

Envia o comando #restartsoft para o terminal, causando a sua reinicialização. Pode ser enviado após um comando de configuração para que a mesma seja aplicada imediatamente.

Assinatura
<code>BOOL __stdcall bSendRestartSoft(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bUpdateSoftware

Envia o comando #updatesoft para o terminal, para que este se atualize segundo os seus parâmetros de atualização configurados.

Assinatura
<code>BOOL __stdcall bUpdateSoftware(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bMandaConfig

Envia um comando de configuração para o terminal.

Assinatura
<pre>BOOL __stdcall bMandaConfig(TCPCONFIG* conftemp);</pre>
Parâmetros
<p>OUT: conftemp: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCPCONFIG { int ID; BYTE host[64]; BYTE endereco[22]; BYTE msknet[22]; BYTE texto1[22]; BYTE texto2[22]; BYTE texto3[22]; BYTE texto4[22]; BYTE tempoexib; };</pre> <p>Recebe a referência de uma estrutura TCPCONFIG, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>host é o IP do servidor, como string. Pode ser formatado em IP ou DNS</p> <p>endereco é o IP do terminal, como string formatado por pontos</p> <p>msknet é a máscara de rede, como string formatado por pontos</p> <p>texto1-4 é o texto das linhas 1-4</p> <p>tempoexib é o tempo de exibição das mensagens</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeConfig

Envia comando para o terminal retornar sua configuração atual.


Assinatura
<pre>BOOL __stdcall bPedeConfig(int id);</pre>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bMandaExtConfig

Envia comando de configuração para o terminal.

Assinatura
<pre>BOOL __stdcall bMandaExtConfig(TCEXTCONFIG* extconf);</pre>
Parâmetros
<p>OUT: extconf: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCEXTCONFIG { int ID; BYTE host[64]; BYTE endereco[22]; BYTE msknet[22]; BYTE gateway[22]; BYTE nameserver[22]; BYTE tcname[22]; BYTE texto1[22]; BYTE texto2[22]; BYTE updserv[100]; BYTE upduser[22]; BYTE updpass[22]; BYTE tempoexib; BYTE dinamicip; BYTE buscaserv; };</pre> <p>Recebe a referência de uma estrutura TCEXTCONFIG, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>host é o IP do servidor, como string. Pode ser formatado em IP ou DNS</p> <p>endereco é o IP do terminal, como string formatado por pontos</p> <p>msknet é a máscara de rede, como string formatado por pontos</p> <p>gateway é o IP do gateway, como string formatado por pontos</p> <p>nameserver é o IP do servidor DNS, como string formatado por pontos</p> <p>tcname é o nome do terminal</p> <p>texto1-2 é o texto das linhas 1-2</p> <p>updserv é o endereço do servidor de update</p> <p>upduser é o usuário no servidor de update</p> <p>updpass é a senha no servidor de update</p> <p>tempoexib é o tempo de exibição das mensagens</p> <p>dinamicip define se o terminal usa IP dinâmico (1 = sim, 0 = não)</p> <p>buscaserv busca servidor (1 = sim, 0 = não)</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bPedeExtConfig

Envia comando para o terminal retornar suas configurações atuais

Assinatura
<code>BOOL __stdcall bPedeExtConfig(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendProdPrice

Envia nome e preço do produto consultado.

Assinatura
<code>BOOL __stdcall bSendProdPrice(int id, char* NameProd, char* PriceProd);</code>
Parâmetros
IN: id: id do socket terminal
IN: NameProd: string com nome do produto
IN: PriceProd: string com preço do produto
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendProdPriceColor

Envia nome e preço do produto consultado com a opção de selecionar cor.

Assinatura
<code>BOOL __stdcall bSendProdPriceColor(int id,int colorProd, char* NameProd, int colorPrice, char* PriceProd);</code>
Parâmetros
IN: id: id do socket terminal
IN: colorProd: cor do nome do produto
IN: NameProd: string com nome do produto
IN: colorPrice: cor do nome do preço do produto
IN: PriceProd: string com preço do produto
Tabela de cores disponíveis: 0 - Branco, 1 - Preto, 2 - Vermelho, 3 - Amarelo, 4 - Azul, 5 - Verde, 6 - Marrom, 7 - Oliva, 8 - Marinha, 9 - Roxo, 10 - Metálico, 11 - Cinza, 12 - Prata, 13 - Lima, 14 - Rosa Púrpura, 15 - Água, 16 - Azul Gertec e

Projeto	SC501GER	
Documento	Manual do API	

17 - Amarelo Gertec
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendProdNotFound

Envia uma mensagem de produto não encontrado.

Assinatura
<code>BOOL __stdcall bSendProdNotFound(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendProdNotFoundColor

Envia uma mensagem de produto não encontrado com a opção de selecionar cor.

Assinatura
<code>BOOL __stdcall bSendProdNotFoundColor(int id, int color);</code>
Parâmetros
IN: id: id do socket terminal IN: color: cor do texto Tabela de cores disponíveis: 0 - Branco, 1 - Preto, 2 - Vermelho, 3 - Amarelo, 4 - Azul, 5 - Verde, 6 - Marrom, 7 - Oliva, 8 - Marinha, 9 - Roxo, 10 - Metálico, 11 - Cinza, 12 - Prata, 13 - Lima, 14 - Rosa Púrpura, 15 - Água, 16 - Azul Gertec e 17 - Amarelo Gertec
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendDisplayMsg

Assinatura
<code>BOOL __stdcall bSendDisplayMsg(int id, char* line1, char* line2, WORD timeExhibition, WORD typeAnimation);</code>
Parâmetros
IN: id: id do socket terminal

Projeto	SC501GER	
Documento	Manual do API	

IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada
IN: width: largura do display do terminal (128)
IN: height: altura do display do terminal (64)
IN: index: índice da imagem (0: imagem instantânea, 1-4: sequência de imagens)
IN: loop: indica se as imagens ficam sendo exibidas em loop
IN: pause: duração da exibição
IN: resize: indica se a imagem deve ser redimensionada

Retorno

Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendImageFromFile

Envia uma imagem tamanho 128x64 monocromática para o terminal (TC505 ou TC506).

Assinatura

```
BOOL __stdcall bSendImageFromFile(int id, char* filename, int width, int height, int index, int loop, int pause, int resize);
```

Parâmetros

IN: id: id do socket terminal
IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada
IN: width: largura do display do terminal (128)
IN: height: altura do display do terminal (64)
IN: index: índice da imagem (0: imagem instantânea, 1-4: sequência de imagens)
IN: loop: indica se as imagens ficam sendo exibidas em loop
IN: pause: duração da exibição
IN: resize: indica se a imagem deve ser redimensionada

Retorno

Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendImageGifFromFile


Envia uma imagem no formato .GIF na dimensão 320x240 e tamanho de até 196608 bytes para o terminal (Busca Preço G2).

Assinatura

```
BOOL __stdcall bSendImageGifFromFile(int id, char* filename, int mode, int loop, int pause);
```

Parâmetros

IN: id: id do socket terminal
IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada
IN: mode: modo da imagem (0: imagem instantânea, 1: loop de imagens, 254: imagem de consulta de preço (neste modo o

Projeto	SC501GER	
Documento	Manual do API	

comando de envio da imagem funciona somente após o recebimento do evento bReceiveBarcode), 255: limpar loop de imagens)

IN: loop: número de vezes que a imagem será repetida

IN: pause: duração da exibição

Retorno

Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendImagePrice

Envia uma imagem com duas linhas de texto (nome do produto e preço) renderizadas.

Assinatura
<pre>BOOL __stdcall bSendImagePrice(int id, char* prodName, char* prodPrice, int duration, int width, int height);</pre>
Parâmetros
<p>prodName: nome do produto</p> <p>prodPrice: preço do produto</p> <p>duration: tempo de exibição</p> <p>width: largura do display do terminal (128)</p> <p>height: altura do display do terminal (64)</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendImageGifPrice

Envia uma imagem de consulta no formato .GIF na dimensão 320x240 e tamanho de até 196608 bytes para o terminal (Busca Preço G2). Esta função somente funciona após o servidor receber o evento bReceiveBarcode.

Assinatura
<pre>BOOL __stdcall bSendImageGifPrice(int id, char* filename, int loop, int pause);</pre>
Parâmetros
<p>IN: id: id do socket terminal</p> <p>IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada</p> <p>IN: loop: número de vezes que a imagem será repetida</p> <p>IN: pause: duração da exibição</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bSendAudioPrice

Envia um áudio de consulta no formato .WAV para o terminal (Busca Preço G2). Esta função somente funciona após o servidor receber o evento bReceiveBarcode. O áudio .WAV deverá ter tamanho mínimo de 16KB e máximo de 68KB. O formato do áudio deverá ser de 8Khz - Mono (1 canal) - 8 bits - PCM_U8.

Assinatura
<pre>BOOL __stdcall bSendAudioPrice(int id, char* filename, char* prodName, char* prodPrice, int duration, int volume);</pre>
Parâmetros
IN: id: id do socket terminal IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada IN: prodName: nome do produto IN: prodPrice: preço do produto IN: duration: tempo de duração do áudio IN: volume: volume do áudio, entre 0 e 3 (0:baixo – 1:médio – 2:alto – 3:muito alto)
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendAudioPriceColor

Envia um áudio de consulta no formato .WAV para o terminal (Busca Preço G2) com a opção de selecionar a cor do texto do produto e preço. Esta função somente funciona após o servidor receber o evento bReceiveBarcode. O áudio .WAV deverá ter tamanho mínimo de 16KB e máximo de 68KB. O formato do áudio deverá ser de 8Khz - Mono (1 canal) - 8 bits - PCM_U8.

Assinatura
<pre>BOOL __stdcall bSendAudioPriceColor(int id, char* filename, int colorProd, char* prodName, int colorPrice, char* prodPrice, int duration, int volume);</pre>
Parâmetros
IN: id: id do socket terminal IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada IN: colorProd: cor do nome do produto IN: prodName: nome do produto IN: colorPrice: cor do nome do preço do produto IN: prodPrice: preço do produto IN: duration: tempo de duração do áudio IN: volume: volume do áudio, entre 0 e 3 (0:baixo – 1:médio – 2:alto – 3:muito alto) Tabela de cores disponíveis: 0 - Branco, 1 - Preto, 2 - Vermelho, 3 - Amarelo, 4 - Azul, 5 - Verde, 6 - Marrom, 7 - Oliva, 8 - Marinha, 9 - Roxo, 10 - Metálico, 11 - Cinza, 12 - Prata, 13 - Lima, 14 - Rosa Púrpura, 15 - Água, 16 - Azul Gertec e 17 - Amarelo Gertec
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bPedeParam

Envia comando para o terminal retornar seus parâmetros extras de configuração (IP dinâmico e busca servidor).

Assinatura
<pre>BOOL __stdcall bPedeParam(int id);</pre>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bMandaParam

Envia um comando de configuração de parâmetros extras para o terminal.

Assinatura
<pre>BOOL __stdcall bMandaParam(TCPARAMCONFIG* conftemp);</pre>
Parâmetros
<p>OUT: conftemp: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCPARAMCONFIG { int ID; BYTE ipdinamico; BYTE buscaserv; };</pre> <p>Recebe a referência de uma estrutura TCPARAMCONFIG, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>ipdinamico indica se o terminal deve usar IP dinâmico (1) ou estático (0)</p> <p>buscaserv indica se o terminal deve buscar o servidor (1) ou não (0)</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeUpdConfig

Envia comando para o terminal retornar sua configuração de atualização.

Assinatura
<pre>BOOL __stdcall bPedeUpdConfig(int id);</pre>

Projeto	SC501GER	
Documento	Manual do API	

Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bMandaUpdConfig

Envia um comando de configuração de parâmetros de atualização para o terminal.

Assinatura
<pre>BOOL __stdcall bMandaUpdConfig(TCUPDCONFIG* conftemp);</pre>
Parâmetros
<p>OUT: conftemp: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCUPDCONFIG { int ID; BYTE gateway[22]; BYTE nameserver[22]; BYTE tcname[22]; BYTE updserv[100]; BYTE upduser[22]; BYTE updpass[22]; };</pre> <p>Recebe a referência de uma estrutura TCUPDCONFIG, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>gateway: IP do gateway, como string, formatado por pontos</p> <p>nameserver: IP do servidor de nomes, como string, formatado por pontos</p> <p>tcname: nome do terminal</p> <p>updserv: url do servidor de atualização (FTP)</p> <p>upduser: nome do usuário no servidor de atualização</p> <p>updpass: senha do usuário no servidor de atualização</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeWlanConfig

Envia comando para o terminal retornar sua configuração wireless.

Assinatura
<pre>BOOL __stdcall bPedeWlanConfig(int id);</pre>


Projeto	SC501GER	
Documento	Manual do API	

Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bMandaWlanConfig

Envia um comando de configuração de parâmetros de atualização para o terminal.

Assinatura
<pre>BOOL __stdcall bMandaWlanConfig(TCWLANCONFIG* conftemp);</pre>
Parâmetros
<p>OUT: conftemp: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCWLANCONFIG { int ID; BYTE usewifi; BYTE mode; BYTE ssid[22]; BYTE channel; BYTE wep; BYTE wepkey[22]; };</pre> <p>Recebe a referência de uma estrutura TCWLANCONFIG, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>usewifi: 0 para desabilitar a rede sem fio (habilita a rede com cabo ethernet), 1 para habilitar</p> <p>mode: 0 para infraestrutura, 1 para ad-hoc</p> <p>ssid: SSID da rede</p> <p>channel: Canal a ser utilizado, de 1 até 11 (somente no modo ad-hoc)</p> <p>wep: 0 desabilita utilização da chave WEP, 1 habilita</p> <p>wepkey: Chave WEP a ser utilizada</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bPedeMacAddr

Envia comando para o terminal retornar seu endereço MAC.

Assinatura
<pre>BOOL __stdcall bPedeMacAddr(int id, BYTE iface);</pre>
Parâmetros
IN: id: id do socket terminal IN: iface: seleciona a interface da qual se deseja obter o MAC 0 retorna endereço MAC da interface ethernet 1 retorna endereço MAC da interface sem fio 9 retorna endereço MAC da interface que estiver sendo utilizada no momento
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendImageBmpJpgPrice

Envia uma imagem no formato .BMP ou .JPEG na dimensão 320x240 e tamanho de até 46080 bytes para o terminal (Busca Preço G2 E e TC 506 E).

Assinatura
<pre>BOOL __stdcall bSendImageBmpJpgPrice(int id, char* filename, int loop, int pause);</pre>
Parâmetros
IN: id: id do socket terminal IN: filename: string com o caminho do arquivo contendo a imagem a ser enviada IN: loop: número de vezes que a imagem será repetida IN: pause: duração da exibição
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeListMedias

Envia comando para o terminal retornar a lista de mídias na memória interna do terminal.

Assinatura
<pre>BOOL __stdcall bPedeListMedias(int id);</pre>
Parâmetros
IN: id: id do socket terminal
Retorno

Projeto	SC501GER	
Documento	Manual do API	

Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeMediasConf

Envia comando para o terminal retornar a lista da grade de propaganda de mídias.

Assinatura
<code>BOOL __stdcall bPedeMediasConf(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bRemoveMedia

Remove uma mídia na memória interna do terminal

Assinatura
<code>BOOL __stdcall bRemoveMedia(int id, char* medias);</code>
Parâmetros
IN: id: id do socket terminal
IN: medias: string com nome da mídia
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSaveMediasConf

Configura a lista da grade de propaganda de mídias.

Assinatura
<code>BOOL __stdcall bSaveMediasConf(int id, char* mediasConf);</code>
Parâmetros
IN: id: id do socket terminal
IN: mediasConf: string com a lista de propaganda
Exemplo: <\nmedia_0=test1.jpg 1 1 \n>
Formato: <\nmedia_indicedaimagem=nomedamidia tempodeexibição loop \n>
Retorno

Projeto	SC501GER	
Documento	Manual do API	

Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bSendMedia

Envia uma mídia para a memória interna.

Assinatura
<pre>BOOL __stdcall bSendMedia(int id, char* nameFile, BYTE* bufferFile, long size);</pre>
Parâmetros
IN: id: id do socket terminal IN: nameFile: string com o nome da mídia IN: bufferFile: buffer da imagem IN: size: tamanho da imagem
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bMandaConfigAudio


Configura o status do áudio no terminal.

Assinatura
<pre>BOOL __stdcall bMandaConfigAudio(int id, int status);</pre>
Parâmetros
IN: id: id do socket terminal IN: status: status do áudio 0: áudio desligado 1: áudio ligado
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeAudioConf

Envia comando para o terminal retornar o status do áudio.

Assinatura
<pre>BOOL __stdcall bPedeAudioConf(int id);</pre>
Parâmetros
IN: id: id do socket terminal

Projeto	SC501GER	
Documento	Manual do API	

Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bMandaHttpConfig

Envia um comando de configuração de parâmetros do HTTP.

Assinatura
<pre>BOOL __stdcall bMandaHttpConfig(TCHTTPCONFIG* confHttp);</pre>
Parâmetros
<p>OUT: confHttp: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCHTTPCONFIG { int ID; int protocol; BYTE http[100]; };</pre> <p>Recebe a referência de uma estrutura TCHTTPCONFIG, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>protocol: protocolo do terminal</p> <p>0: Busca Preço</p> <p>1: Busca Preço G2 S/ TC 506 S</p> <p>2: HTTP</p> <p>http: url do servidor http</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

bPedeHttpConf

Envia comando para o terminal retornar o status do áudio.

Assinatura
<pre>BOOL __stdcall bPedeHttpConf(int id);</pre>
Parâmetros
<p>IN: id: id do socket terminal</p>
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

Projeto	SC501GER	
Documento	Manual do API	

bRemoveAllMedias

Remove todas as mídias na memória interna do terminal de consulta

Assinatura
<code>BOOL __stdcall bRemoveAllMedias(int id);</code>
Parâmetros
IN: id: id do socket terminal
Retorno
Retorna 0 em caso de falha ou valor diferente de zero em caso de sucesso.

5. ROTINAS PARA RECEBER DADOS DOS TERMINAIS

bReceiveBarcode

Rotina que deve ser chamada periodicamente, para receber eventos de consultas de código de barras.

Assinatura
<code>BOOL __stdcall bReceiveBarcode(int *id, int *Porta, char* buffer, int *Nbr);</code>
Parâmetros
OUT: id: Retorna o ID do socket do terminal que enviou a resposta OUT: Porta: porta que foi lido dados da serial OUT: buffer: dados recebidos da serial (alocar um buffer de até 3500 bytes para atender a modelos de TC com leitor 2D e omnidirecional) OUT: Nbr: número de dados lido da serial
Retorno
Retorna 0 em caso de falha (não há barcode no buffer) ou valor diferente de zero em caso de sucesso.

bReceiveConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração do terminal previamente requisitadas pelo servidor através do comando bPedeConfig.

Assinatura
<code>BOOL __stdcall bReceiveConfig(TCPCONFIG* conf);</code>
Parâmetros
OUT: conf: Escreve na estrutura passada por referência, onde o resultado será gravado. <pre>struct TCPCONFIG { int ID;</pre>

Projeto	SC501GER	
Documento	Manual do API	

```

BYTE host[64];
BYTE endereco[22];
BYTE msknet[22];
BYTE texto1[22];
BYTE texto2[22];
BYTE texto3[22];
BYTE texto4[22];
BYTE tempoexib;
};

```

Escreve a resposta em uma estrutura TCPCONFIG passada por referência, onde:

ID é o id do socket onde o terminal está conectado

host é o IP do servidor, como string. Pode ser formatado em IP ou DNS

endereco é o IP do terminal, como string formatado por pontos

msknet é a máscara de rede, como string formatado por pontos

texto1-4 é o texto das linhas 1-4

tempoexib é o tempo de exibição das mensagens

Retorno

Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bReceiveExtConfig

Rotina que deve ser chamada periodicamente, para receber dados de configuração do terminal previamente requisitadas pelo servidor através do comando bPedeExtConfig. Função obsoleta, mantém compatibilidade com TC501 versão 2.0.

Assinatura

```

BOOL __stdcall bReceiveExtConfig(TCEXTCONFIG* extconf);

```

Parâmetros

OUT: extconf: Escreve na estrutura passada por referência, onde o resultado será gravado.

```

struct TCEXTCONFIG {
    int ID;
    BYTE host[64];
    BYTE endereco[22];
    BYTE msknet[22];
    BYTE gateway[22];
    BYTE nameserver[22];
    BYTE tcname[22];
    BYTE texto1[22];
    BYTE texto2[22];
    BYTE updserv[100];
    BYTE upduser[22];
    BYTE updpass[22];
    BYTE tempoexib;
    BYTE dinamicip;
};

```

Projeto	SC501GER	
Documento	Manual do API	

<pre>BYTE buscaserv;</pre> <pre>};</pre> <p>Escreve a resposta em uma estrutura TTEXTCONFIG passada por referência, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>host é o IP do servidor, como string. Pode ser formatado em IP ou DNS</p> <p>endereço é o IP do terminal, como string formatado por pontos</p> <p>msknet é a máscara de rede, como string formatado por pontos</p> <p>gateway é o IP do gateway, como string formatado por pontos</p> <p>nameserver é o IP do servidor DNS, como string formatado por pontos</p> <p>tcname é o nome do terminal</p> <p>texto1-2 é o texto das linhas 1-2</p> <p>updserv é o endereço do servidor de update</p> <p>upduser é o usuário no servidor de update</p> <p>updpass é a senha no servidor de update</p> <p>tempoexib é o tempo de exibição das mensagens</p> <p>dinamicip define se o terminal usa IP dinâmico (1 = sim, 0 = não)</p> <p>buscaserv busca servidor (1 = sim, 0 = não)</p>
Retorno
Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

GetResponseCtr

Obtém eventos de resposta aos comandos enviados aos terminais.

Assinatura
<pre>BOOL __stdcall GetResponseCtr(int *pid, int *aresp);</pre>
Parâmetros
<p>OUT: pid: Retorna o ID do socket do terminal que enviou a resposta.</p> <p>OUT: aresp: Retorna do código da resposta, indicando o evento correspondente.</p>
Retorno
Retorna 0 em caso de falha (não há evento no buffer) ou valor diferente de zero em caso de sucesso.

bReceiveParam

Rotina que deve ser chamada periodicamente, para receber parâmetros extras de configuração do terminal previamente requisitados pelo servidor através do comando bPedeParam.

Assinatura
<pre>BOOL __stdcall bReceiveParam(TCPARAMCONFIG* paramconf);</pre>

Projeto	SC501GER	
Documento	Manual do API	

Parâmetros

OUT: paramconf: Escreve na estrutura passada por referência, onde o resultado será gravado.

```
struct TCPARAMCONFIG {
    int ID;
    BYTE ipdinamico;
    BYTE buscaserv;
};
```

Escreve a resposta em uma estrutura TCPARAMCONFIG passada por referência, onde:

ID é o id do socket onde o terminal está conectado

ipdinamico: indica se o terminal deve usar IP dinâmico (1) ou estático (0)

buscaserv: indica se o terminal deve buscar o servidor (1) ou não (0)

Retorno

Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bReceiveUpdConfig

Rotina que deve ser chamada periodicamente, para receber parâmetros de atualização do terminal previamente requisitados pelo servidor através do comando bPedeUpdConfig.

Assinatura

```
BOOL __stdcall bReceiveUpdConfig(TCUPDCONFIG* updconf);
```

Parâmetros

OUT: updconf: Escreve na estrutura passada por referência, onde o resultado será gravado.

```
struct TCUPDCONFIG {
    int ID;
    BYTE gateway[22];
    BYTE nameserver[22];
    BYTE tcname[22];
    BYTE updserv[100];
    BYTE upduser[22];
    BYTE updpass[22];
};
```

Escreve a resposta em uma estrutura TCUPDCONFIG passada por referência, onde:

ID é o id do socket onde o terminal está conectado

gateway: IP do gateway, como string, formatado por pontos


nameserver: IP do servidor de nomes, como string, formatado por pontos

tcname: nome do terminal

updserv: url do servidor de atualização (FTP)

upduser: nome do usuário no servidor de atualização

updpass: senha do usuário no servidor de atualização

Projeto	SC501GER	
Documento	Manual do API	

Retorno
Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bReceiveWlanConfig

Rotina que deve ser chamada periodicamente, para receber parâmetros de conexão Wi-Fi do terminal previamente requisitados pelo servidor através do comando bPedeWlanConfig.

Assinatura
<pre>BOOL __stdcall bReceiveWlanConfig(TCWLANCONFIG* wlanconf);</pre>
Parâmetros
<p>OUT: wlanconf: Escreve na estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCWLANCONFIG { int ID; BYTE usewifi; BYTE mode; BYTE ssid[22]; BYTE channel; BYTE wep; BYTE wepkey[22]; };</pre> <p>Escreve a resposta em uma estrutura TCWLANCONFIG passada por referência, onde:</p> <p>ID é o id do socket onde o terminal está conectado</p> <p>usewifi: 0 para desabilitar a rede sem fio (habilita a rede com cabo ethernet), 1 para habilitar</p> <p>mode: 0 para infraestrutura, 1 para ad-hoc</p> <p>ssid: SSID da rede</p> <p>channel: Canal a ser utilizado, de 1 até 11 (somente no modo ad-hoc)</p> <p>wep: 0 desabilita utilização da chave WEP, 1 habilita</p> <p>wepkey: Chave WEP a ser utilizada (string ASCII)</p>
Retorno
Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bReceiveMacAddrConfig

Rotina que deve ser chamada periodicamente, para receber dados de MAC do terminal previamente requisitados pelo servidor através do comando bPedeMacAddr.

Assinatura
<pre>BOOL __stdcall bReceiveMacAddrConfig(TCMACADDRCONFIG* macconf);</pre>
Parâmetros

Projeto	SC501GER	
Documento	Manual do API	

OUT: macconf: Escreve na estrutura passada por referência, onde o resultado será gravado.

```
struct TCMACADDRCONFIG {
    int ID;
    BYTE iface;
    BYTE macaddr[17];
};
```

Escreve a resposta em uma estrutura TCMACADDRCONFIG passada por referência, onde:

ID é o id do socket onde o terminal está conectado

iface é a interface para a qual foi solicitado o mac (0: ethernet, 1: wi-fi, 9: interface em uso atual)

macaddr é o mac do dispositivo, codificado como string ASCII

Retorno

Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bGetListMedias

Rotina que deve ser chamada periodicamente, para receber a lista de mídias da memória interna do terminal previamente requisitados pelo servidor através do comando bPedeListMedias.

Assinatura

```
BOOL __stdcall bGetListMedias(TCGETLISTMEDIAS* listMedias);
```

Parâmetros

OUT: listMedias: Escreve na estrutura passada por referência, onde o resultado será gravado.

```
struct TCGETLISTMEDIAS {
    int TermSock;
    BYTE listMedias[300];
};
```

Escreve a resposta em uma estrutura TCGETLISTMEDIAS passada por referência, onde:

TermSock é o id do socket onde o terminal está conectado

listMedias é a lista de mídias da memória interna do terminal

Retorno


Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bGetMediasConf

Rotina que deve ser chamada periodicamente, para receber a lista de propaganda de mídias do terminal previamente requisitados pelo servidor através do comando bPedeMediasConf.

Assinatura

```
BOOL __stdcall bGetMediasConf(TCGETMEDIASCONF* mediasConf);
```

Projeto	SC501GER	
Documento	Manual do API	

Parâmetros
<p>OUT: mediasConf: Escreve na estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCGETMEDIASCONF { int TermSock; BYTE mediasConf[500]; };</pre> <p>Escreve a resposta em uma estrutura TCGETMEDIASCONF passada por referência, onde: TermSock é o id do socket onde o terminal está conectado mediasConf é a lista de propaganda de mídias do terminal</p>
Retorno
Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bGetAudioConf


Rotina que deve ser chamada periodicamente, para receber o status do áudio do terminal previamente requisitados pelo servidor através do comando bPedeAudioConf.

Assinatura
<pre>BOOL __stdcall bGetAudioConf(TCGETAUDIOCONF* audioConf);</pre>
Parâmetros
<p>OUT: audioConf: Escreve na estrutura passada por referência, onde o resultado será gravado.</p> <pre>struct TCGETAUDIOCONF { int TermSock; int status[4]; };</pre> <p>Escreve a resposta em uma estrutura TCGETAUDIOCONF passada por referência, onde: TermSock é o id do socket onde o terminal está conectado status é o status do áudio do terminal</p>
Retorno
Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.

bGetHttpConf

Rotina que deve ser chamada periodicamente, para receber o status do áudio do terminal previamente requisitados pelo servidor através do comando bPedeHttpConf.

Assinatura
<pre>BOOL __stdcall bGetHttpConf(TCHTTPCONFIG* httpConf);</pre>

Projeto	SC501GER	
Documento	Manual do API	

Parâmetros

OUT: confHttp: Ponteiro para a estrutura passada por referência, onde o resultado será gravado.

```
struct TCHTTPCONFIG {
    int ID;
    int protocol;
    BYTE http[100];
};
```

Recebe a referência de uma estrutura TCHTTPCONFIG, onde:

ID é o id do socket onde o terminal está conectado

protocol: protocolo do terminal

0: Busca Preço

1: Busca Preço G2 S/ TC 506 S

2: HTTP

http: url do servidor http

Retorno

Retorna 0 em caso de falha (não há resposta no buffer) ou valor diferente de zero em caso de sucesso.