

The background features a dark grey, low-poly geometric pattern. Overlaid on this are white line-art outlines of various electronic devices, including what appears to be a monitor or tablet, a control panel with buttons, and a device with a screen and a keypad. The TANCA logo, a stylized white 'T' with a curved bottom, is positioned to the left of the main text.

TANCA

AUTOMAÇÃO COMERCIAL

Manual de Instruções
LINHA VERIFICADORES DE PREÇO



VERIFICADORES DE PREÇO

LINHA VP



Versão da DLL: 3.0.0.4

MANUAL DE DESENVOLVIMENTO

O presente material de apoio é de cunho técnico-informativo, para que equipes de desenvolvedores recebam instruções quanto ao funcionamento de nossa DLL, necessária para o desenvolvimento de soluções que farão uso dos Verificadores de Preço produzidos pela Tanca.

Todos os direitos quanto ao conteúdo deste documento estão reservados a Tanca. Sendo proibida qualquer forma de reprodução e/ou distribuição, sem autorização prévia da Tanca. Os dados presentes neste material estão sujeitos a alterações sem aviso prévio.

A Tanca não assume responsabilidades quanto a erros, imprecisões ou omissões em seu conteúdo, bem como, qualquer outro tipo de dano, perda ou lucro cessado decorrente do uso dos dados contidos neste manual ou no arquivo da DLL disponibilizada.

SUMÁRIO

MANUAL DE DESENVOLVIMENTO.....	3
INTRODUÇÃO.....	4
ESPECIFICAÇÃO TÉCNICA E DETALHAMENTO DAS FUNÇÕES DA DLL.....	4
INICIALIZAÇÃO DO SERVIÇO.....	4
INICIALIZAÇÃO DO SERVIDOR.....	5
LISTA DE TERMINAIS.....	5
FORMATANDO ENDEREÇAMENTO.....	8
ENVIANDO MENSAGENS DO SERVIDOR PARA O TERMINAL.....	9
ROTINA DE VERIFICAÇÃO DE PRODUTO POR CÓDIGOS DE BARRAS.....	30
IDENTIFICAÇÃO DA VERSÃO DA DLL.....	32
FINALIZAÇÃO DO SERVIÇO.....	32
SUORTE TÉCNICO.....	33

INTRODUÇÃO

O presente conteúdo vem como ferramenta de apoio ao desenvolvimento de soluções para os Verificadores de Preço produzidos pela Tanca. A partir deste material, será possível reconhecer os nomes, finalidades, tipos de entradas e tipos de retornos das funções presentes na DLL (Dynamic Link Library) disponibilizada (VP_v3.dll).

Possibilitando o desenvolvimento de sistemas, independente da linguagem de programação e da base de dados, contanto que essas tecnologias permitam o consumo deste tipo de recurso externo.

ESPECIFICAÇÃO TÉCNICA E DETALHAMENTO DAS FUNÇÕES DA DLL

A DLL foi desenvolvida em linguagem Pascal, utilizando IDE Delphi vs 10. Portanto deve-se ter em mente que os tipos de retorno dessas funções devem ser analisados e o uso de recursos como bibliotecas, conversões e outros, específicos da linguagem e IDE que será utilizada no desenvolvimento da aplicação (Servidor) que consumirá essa DLL, sejam respeitados, assim como as limitações inerentes as mesmas.

Os Terminais de Verificação de Preço deverão estar devidamente instalados na rede (TCP/IP) do local em a aplicação será utilizada. Essa aplicação funcionará como Servidor, ponte entre a base de dados de produtos a serem pesquisados, que reconhecerá os Terminais conectados à rede, bem como as solicitações encaminhadas por estes, tendo como apoio as funções da DLL. O consumo da DLL, não depende do domínio do protocolo, mas do conhecimento dos nomes, entradas e saídas das funções da DLL. O Serviço proposto pela DLL consiste em gerenciar a comunicação com os Terminais de Consulta (Verificadores de Preço), através da porta 6500.

Para melhor compreensão do uso dessas funções, segue o detalhamento de FINALIDADE, ENTRADA e SAÍDA, bem como exemplos de declarações em Pascal (Delphi), C++ e C#.

INICIALIZAÇÃO DO SERVIÇO

Para que essa comunicação seja inicializada pela aplicação, a mesma deverá fazer uso das funções da DLL "vInitialize" e "tc_startserver":

- vInitialize

Delphi: procedure vInitialize; stdcall; far;

C++: void __stdcall bInitialize(void);

C#: public static extern void vInitialize();

Finalidade: Primeira rotina que deve ser chamada para inicializar a DLL;

Entrada: N/A;

Saída: N/A;

INICIALIZAÇÃO DO SERVIDOR

- tc_startserver

Delphi: function tc_startserver: integer; stdcall; far;

C++: int __stdcall tc_startserver(void);

C#: public static extern int tc_startserver();

Finalidade: Rotina que faz com que a aplicação espere por conexões de terminais;

Entrada: N/A;

Saída: (int): Retorna 1 se a aplicação foi inicializada com sucesso ou retorna 0 se houve erro.

LISTA DE TERMINAIS

Para o controle de quais terminais estão ou não conectados na rede, pode-se utilizar da função "GetTabConectados" que faz uso de referências como a "TTABSOCK", ou, caso queira controlar essas informações pelas funções "bConnected", "bDisconnected" e "bCloseTerminal", deverá elaborar uma forma de controlar a sua lista de terminais, seja por tabela, listagem ou até mesmo enviando apenas os endereços para a grid do formulário, fica a critério, assim não terá dependência de uso da estrutura TTABSOCK, mas deve-se ter em mente o seu formato, para replicar essas informações de maneira coerente e completa.

O uso de uma abordagem de controle ou outra dependerá da tratativa de dados e compatibilidade de tipagens inerentes a linguagem escolhida para o desenvolvimento da aplicação e do comportamento da mesma em relação aos terminais.

- GetTabConectados

Delphi: function GetTabConectados(nada: integer): TTABSOCK; stdcall; far;

type

PTABCLI = ^TTABCLI;

TTABCLI = packed record

TabPort : array[0..MAX_TERMS-1] of Word;

TabSIP : array[0..MAX_TERMS-1] of String;

TabWIP : array[0..MAX_TERMS-1] of DWORD;

TabID : array[0..MAX_TERMS-1] of integer;

TabIndex : array[0..MAX_TERMS-1] of integer;

TabHandle : array[0..MAX_TERMS-1] of integer;

TabCmdSent : array[0..MAX_TERMS-1] of integer;

TabLiveTime : array[0..MAX_TERMS-1] of integer;

TabTimeOut : array[0..MAX_TERMS-1] of integer;

TabModelo : array[0..MAX_TERMS-1] of String;

```
TabVersao : array[0..MAX_TERMS-1] of String;  
TabBarCode : array[0..MAX_TERMS-1] of String;  
TabThread : array[0..MAX_TERMS-1] of Pointer;  
TabValido : array[0..MAX_TERMS-1] of Boolean;  
TabView : array[0..MAX_TERMS-1] of Boolean;  
NumSockConec:integer;  
end;
```

```
type  
PTABSOCK = ^TTABSOCK;  
TTABSOCK = packed record  
  TabSock : array[0..MAX_TERMS-1] of Word;  
  TabWIP : array[0..MAX_TERMS-1] of DWORD;  
  NumSockConec:integer;
```

Finalidade: Retornar uma estrutura com a lista de terminais conectados;

Entrada: nada (int) reservada, deve haver ao menos 1;
Saída: TTABSOCK;

- TabSock (int (array de 1024)): lista com os SOCKETs dos terminais.
- TabIP (DWORD (array de 1024)): lista com os IPs dos terminais.
- NumSockConec (int) : número de terminais conectados.

OU

- bConnected

```
Delphi: function bConnected(out ID_IP: DWORD; out ID_Socket: word): boolean; far; stdcall;  
C++: bool __stdcall bConnected(DWORD ID_IP, word ID_Socket);  
C#: public static unsafe extern bool bConnected(out uint ID_IP, out ushort Porta);
```

Finalidade: Verificar se houve a conexão de um novo terminal na aplicação;

Entrada: N/A;

Saída: (bool), ID_IP, ID_Socket : Caso um novo terminal tenha se conectado a aplicação, retornará TRUE, o IP do terminal e Porta em que o mesmo se conectou, caso contrário retornará FALSE;

- ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal
- ID_Socket (word): valor numérico da porta de comunicação com o terminal

- bDisconnected

Delphi: function bDisconnected(out ID_IP: DWORD; out ID_Socket: word): boolean; far; stdcall;

C++: bool __stdcall bConnected(DWORD ID_IP, word ID_Socket);

C#: public static extern unsafe bool bDisconnected(out uint ID_IP, out ushort ID_Socket);

Finalidade: Verificar se algum terminal se desconectou da aplicação;

Entrada: N/A;

Saída: (bool), ID_IP, ID_Socket : Caso um novo terminal tenha se desconectado da aplicação, retornará TRUE, o IP do terminal e Porta que o mesmo se desconectou, caso contrário retornará FALSE;

- ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal

- ID_Socket (word): valor numérico da porta de comunicação com o terminal

- bCloseTerminal

Delphi: function bCloseTerminal(ID_IP: DWORD; ID_Socket: Word): boolean; far; stdcall;

C++: bool __stdcall bCloseTerminal(DWORD ID_IP; word ID_Socket);

C#: public static extern unsafe bool bCloseTerminal(uint ID_IP, ushort ID_Socket);

Finalidade: Desconectar um determinado terminal, identificado através do endereço IP e porta de comunicação;

Entrada: ID_IP, ID_Socket;

- ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal

- ID_Socket (word): valor numérico da porta de comunicação com o terminal

Saída: (bool): Caso o terminal seja desconectado com sucesso, retornará TRUE, caso contrário, retornará FALSE;

FORMATANDO ENDEREÇAMENTO

A aplicação poderá consumir as funções auxiliares "Inet_NtoA" e "Inet_Addr", que formatam o endereçamento dos Terminais, seja para serem usadas em apresentações de mensagens tratadas ou em outras funções da DLL consumidas pela aplicação.

- Inet_NtoA

Delphi: `function Inet_NtoA(ID_IP: DWORD): PAnsiChar; far; stdcall;`

C++: `char* __stdcall Inet_NtoA(DWORD ID_IP);`

C#: `public static extern unsafe IntPtr Inet_NtoA(uint ID_IP);`

Finalidade: Converte o número de um IP de formato DWORD em um endereço IP no formato string (XXX.XXX.XXX.XXX);

Entrada: ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal;

Saída: (char *): ponteiro de string de endereço de rede (IP) do terminal;

- Inet_Addr

Delphi: `function Inet_Addr(sIP: PAnsiChar): DWORD; far; stdcall;`

C++: `DWORD __stdcall Inet_Addr(char *sIP);`

C#: `public static extern unsafe uint Inet_Addr(IntPtr sIP);`

Finalidade: Converte endereço IP no formato string (XXX.XXX.XXX.XXX) em número de um IP de formato DWORD;

Entrada: sIP (char *): ponteiro de string de endereço de rede (IP) do terminal;

Saída: (DWORD): valor numérico de endereço de rede (IP) do terminal;

ENVIANDO MENSAGENS DO SERVIDOR PARA O TERMINAL

Mensagens instantâneas podem ser enviadas para as linhas 1 e 2 do Terminal através da função "bSendDisplayMsg":

- bSendDisplayMsg

Delphi: function bSendDisplayMsg(ID_IP: DWORD; Linha1: PAnsiChar; Linha2: PAnsiChar; Tempo: Integer): Boolean; stdcall; far;

C++: bool __stdcall bSendDisplayMsg(DWORD ID_IP, char *Linha1, char *Linha2, int Tempo);

C#: public static extern unsafe bool bSendDisplayMsg(uint ID_IP, IntPtr Linha1, IntPtr Linha2, int Tempo);

Finalidade: Enviar mensagem para as linhas 1 e 2 do terminal;

Entrada: ID_IP (DWORD), Linha1 (char *), Linha2 (char *) e Tempo (int):

- ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal;
- Linha1 (char *): ponteiro de string que representa a primeira linha do display
- Linha2 (char *): ponteiro de string que representa a segunda linha do display
- Tempo (int): Tempo de exibição da mensagem, em segundos

Saída: (bool) : Caso a mensagem seja encaminhada com sucesso, retornará TRUE, caso contrário, retornará FALSE;

Assim que a conexão entre o Terminal e a aplicação é estabelecida, a aplicação apresentará um diálogo "#ok", confirmando que sua conexão foi aceita. O Terminal deve responder informando o prefixo "#" e os dados de modelo e versão separados por pipe, ex: "#MODELO|1.23" (Obs: esse formato permite a compatibilidade com equipamentos de outras fabricantes na mesma rede).

A partir do momento que as funções de inicialização são utilizadas e é detectado que houve conexão de um novo terminal, de maneira automática, a aplicação passará a estabelecer um diálogo com o terminal enviando o comando "#live?", a cada 10 segundos, de verificação de disponibilidade. Em resposta a aplicação, o Terminal responderá "#live". Se o intervalo de tempo, entre o comando "#live?" e a resposta "#live" exceder 30 segundos, a conexão com este terminal será fechada automaticamente. Note que todo o procedimento acima é controlado internamente pela DLL.

É possível utilizar ainda de outros comandos da DLL para o Terminal através da aplicação, segue a tabela completa de comandos e suas referências:

TABELA DE COMANDOS DO APLICAÇÃO

COMANDO	RETORNO	REFERÊNCIA
#ok	#modelo versao	Aplicação envia para o Terminal o comando "#ok", confirmando que sua conexão foi aceita. O Terminal deve então responder informando o prefixo "#" e os dados de modelo e versão separados por pipe, ex: "#MODELO 1.23"
#live?	#live	Aplicação verificando se o terminal está "vivo" (conectado).
#alwayslive	#alwayslive_ok	Desativa desconexão com aplicação. Ao enviar este comando, o terminal não fecha a conexão com o aplicação se este deixar de enviar algum comando por mais de 30 segundos.
#checklive	#checklive_ok	Ativa desconexão com aplicação. Este comando é o inverso do anterior, ou seja, ao enviá-lo, o terminal se desconecta do aplicação se ficar mais de 30 segundos sem receber uma mensagem.
#restartsoft	#restartsoft_ok	Salva configurações e reinicia terminal. Enviando este comando, o terminal é reiniciado. Uma boa sugestão seria enviá-lo após trocar seu IP (pela configuração remota), para que a configuração seja efetuada com sucesso imediatamente. A senha é um <i>longword</i> (4 bytes) que devem ser enviados para que o terminal

		realize este comando, a senha é a seguinte (em hexa): 0xA5CC5A33.
#updatesoft + "senha"	#updatesoft_ok	Responde, mas não realiza nenhuma outra ação (compatibilidade). A senha é ignorada.
#config?	#config "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TEMPO"	<p>Retorna parâmetros de configuração incluindo apenas as linhas 1 e 2. O terminal responde com os seguintes dados:</p> <p>TAM IP APLICAÇÃO (byte): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (byte): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (byte): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1(byte): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1(<i>string</i>): texto da linha 1;</p> <p>TAM L2(byte): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TEMPO (byte): tempo de exibição;</p>

		<p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, deve-se subtrair 48 do valor de cada byte e não incluir as aspas duplas no retorno.</p>
#config02?	<pre>#config02 "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM L3" "LINHA3" "TAM L4" "LINHA4" "TEMPO"</pre>	<p>Retorna parâmetros de configuração incluindo todas as linhas. O terminal responde com os seguintes dados:</p> <p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM L3 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 3;</p> <p>LINHA3 (<i>string</i>): texto da linha 3;</p>

		<p>TAM L4 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 4;</p> <p>LINHA4 (<i>string</i>): texto da linha 4;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, deve-se subtrair 48 do valor de cada byte e não incluir as aspas duplas no retorno.</p>
#conf502?	<pre>#config502 "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM L3" "LINHA3" "TAM L4" "LINHA4" "TEMPO"</pre>	<p>Retorna parâmetros de configuração incluindo todas as linhas. O terminal responde com os seguintes dados:</p> <p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p>

		<p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM L3 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 3;</p> <p>LINHA3 (<i>string</i>): texto da linha 3;</p> <p>TAM L4 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 4;</p> <p>LINHA4 (<i>string</i>): texto da linha 4;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, deve-se subtrair 48 do valor de cada byte e não incluir as aspas duplas no retorno.</p>
#updconfig?	#updconfig "TAM GW" "GATEWAY" "TAM DNS" "DNS" "TAM TERM" "TERM" "TAM NS" "NS" "TAM NS" "NS" "TAM NS" "NS"	<p>Retorna parâmetros de configuração incluindo gateway, dns e nome do terminal:</p> <p>TAM GW (<i>byte</i>): tamanho da <i>string</i> do Gateway;</p> <p>GATEWAY (<i>string</i>): Gateway;</p> <p>TAM DNS (<i>byte</i>): tamanho da <i>string</i> do DNS;</p> <p>DNS (<i>string</i>): DNS;</p> <p>TAM TERM (<i>byte</i>): tamanho da <i>string</i> do nome do terminal;</p> <p>TERM (<i>string</i>): Nome;</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p>

		<p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> devemos subtrair 48 (<i>decimal</i>) do valor de cada byte.</p>
<p>#updconfig + "dados"</p>	<p>#updconfig_ok</p>	<p>Recebe e carrega parâmetros de configuração dados: "TAM GW" "GATEWAY" "TAM DNS" "DNS" "TAM TERM" "TERM" "TAM NS" "NS" "TAM NS" "NS" "TAM NS" "NS":</p> <p>TAM GW (<i>byte</i>): tamanho da <i>string</i> do Gateway;</p> <p>GATEWAY (<i>string</i>): Gateway;</p> <p>TAM DNS (<i>byte</i>): tamanho da <i>string</i> do DNS;</p> <p>DNS (<i>string</i>): DNS;</p> <p>TAM TERM (<i>byte</i>): tamanho da <i>string</i> do nome do terminal;</p> <p>TERM (<i>string</i>): Nome;</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p>

		<p>NS (string): “Não suportado” (sem as aspas);</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> devemos subtrair 48(<i>decimal</i>) do valor de cada <i>byte</i>.</p>
rupdconfig + "dados"		<p>Recebe e carrega parâmetros de configuração dados: "TAM GW" "GATEWAY" "TAM DNS" "DNS" "TAM TERM" "TERM" "TAM NS" "NS" "TAM NS" "NS" "TAM NS" "NS":</p> <p>TAM GW (byte): tamanho da <i>string</i> do Gateway;</p> <p>GATEWAY (string): Gateway;</p> <p>TAM DNS (byte): tamanho da <i>string</i> do DNS;</p> <p>DNS (string): DNS;</p> <p>TAM TERM (byte): tamanho da <i>string</i> do nome do terminal;</p> <p>TERM (string): Nome;</p> <p>TAM NS (byte): 61 (<i>decimal</i>);</p> <p>NS (string): “Não suportado” (sem as aspas);</p> <p>TAM NS (byte): 61 (<i>decimal</i>);</p> <p>NS (string): “Não suportado” (sem as aspas);</p> <p>TAM NS (byte): 61 (<i>decimal</i>);</p> <p>NS (string): “Não suportado” (sem as aspas);</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> devemos subtrair 48 (<i>decimal</i>) do valor de cada <i>byte</i>.</p>

<p>#extconfig?</p>	<pre>#extconfig "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM GW" "GATEWAY" "TAM NS" "NS" "TAM TERM" "TERM" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM NS" "NS" "TAM NS" "NS" "TAM NS" "NS" "TEMPO" "DHCP" "RESERVADO"</pre>	<p>Retorna parâmetros de configuração estendidos:</p> <p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM GW (<i>byte</i>): tamanho da <i>string</i> do Gateway;</p> <p>GATEWAY (<i>string</i>): Gateway;</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM TERM (<i>byte</i>): tamanho da <i>string</i> do nome do terminal;</p> <p>TERM (<i>string</i>): Nome;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto a linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p>
--------------------	--	--

		<p>NS (string): "Não suportado" (sem as aspas);</p> <p>TAM NS (byte): 61 (<i>decimal</i>);</p> <p>NS (string): "Não suportado" (sem as aspas);</p> <p>TAM NS (byte): 61 (<i>decimal</i>);</p> <p>NS (string): "Não suportado" (sem as aspas);</p> <p>TEMPO (byte): Tempo de Exibição;</p> <p>DHCP (byte): IP dinâmico/fixo (48 = fixo, 49 = dinâmico);</p> <p>RESERVADO (byte): 48 (<i>decimal</i>);</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, devemos subtrair 48 (<i>decimal</i>) do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
<p>#rconf + "dados"</p>		<p>Recebe e carrega parâmetros de configuração: "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "TEMPO".</p> <p>TAM IP APLICAÇÃO (byte): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (string): IP do aplicação;</p>

		<p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
<p>#reconf02 + "dados"</p>		<p>Recebe e carrega parâmetros de configuração: "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM L3" "LINHA3" "TAM L4" "LINHA4" "TEMPO".</p>

		<p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM L3 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 3;</p> <p>LINHA3 (<i>string</i>): texto da linha 3;</p> <p>TAM L4 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 4;</p> <p>LINHA4 (<i>string</i>): texto da linha 4;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, devemos subtrair 48 do</p>
--	--	--

		valor de cada <i>byte</i> . Não incluir as aspas duplas do retorno.
#rconfig + "dados"		<p>Recebe e carrega parâmetros de configuração: "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM L3" "LINHA3" "TAM L4" "LINHA4" "TEMPO"</p> <p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM L3 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 3.</p>

		<p>LINHA3 (<i>string</i>): texto da linha 3;</p> <p>TAM L4 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 4;</p> <p>LINHA4 (<i>string</i>): texto da linha 4;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
<p>#rconf02 + "dados"</p>		<p>Recebe e carrega parâmetros de configuração: "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM L3" "LINHA3" "TAM L4" "LINHA4" "TEMPO"</p> <p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p>

		<p>MASCARA DE REDE <i>(string)</i>: máscara de rede;</p> <p>TAM L1 <i>(byte)</i>: tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 <i>(string)</i>: texto da linha 1;</p> <p>TAM L2 <i>(byte)</i>: tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 <i>(string)</i>: texto da linha 2;</p> <p>TAM L3 <i>(byte)</i>: tamanho da <i>string</i> do texto da linha 3;</p> <p>LINHA3 <i>(string)</i>: texto da linha 3;</p> <p>TAM L4 <i>(byte)</i>: tamanho da <i>string</i> do texto da linha 4;</p> <p>LINHA4 <i>(string)</i>: texto da linha 4;</p> <p>TEMPO <i>(byte)</i>: tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
<pre>#reconf502 + "dados"</pre>		<p>Recebe e carrega parâmetros de configuração: "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM L3" "LINHA3" "TAM L4" "LINHA4" "TEMPO"</p>

		<p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM L3 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 3;</p> <p>LINHA3 (<i>string</i>): texto da linha 3;</p> <p>TAM L4 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 4;</p> <p>LINHA4 (<i>string</i>): texto da linha 4;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de</p>
--	--	---

		exibição, devemos subtrair 48 do valor de cada <i>byte</i> . Não incluir as aspas duplas do retorno.
#rextconf + "dados"	#rextconf_ok	<p>Recebe e carrega parâmetros estendidos: "TAM IP APLICAÇÃO" "IP APLICAÇÃO" "TAM IP TERMINAL" "IP TERMINAL" "TAM MASC" "MASCARA DE REDE" "TAM GW" "GATEWAY" "TAM NS" "NS" "TAM TERM" "TERM" "TAM L1" "LINHA1" "TAM L2" "LINHA2" "TAM NS" "NS" "TAM NS" "NS" "TAM NS" "NS" "TEMPO" "DHCP" "RESERVADO":</p> <p>TAM IP APLICAÇÃO (<i>byte</i>): tamanho da <i>string</i> do IP do aplicação;</p> <p>IP APLICAÇÃO (<i>string</i>): IP do aplicação;</p> <p>TAM IP TERMINAL (<i>byte</i>): tamanho da <i>string</i> do IP do terminal;</p> <p>IP TERMINAL (<i>string</i>): IP do terminal;</p> <p>TAM MASC (<i>byte</i>): tamanho da <i>string</i> da máscara de rede;</p> <p>MASCARA DE REDE (<i>string</i>): máscara de rede;</p> <p>TAM GW (<i>byte</i>): tamanho da <i>string</i> do Gateway;</p> <p>GATEWAY (<i>string</i>): Gateway;</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p>

		<p>TAM TERM (<i>byte</i>): tamanho da <i>string</i> do nome do terminal;</p> <p>TERM (<i>string</i>): Nome;</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> do texto a linha 1;</p> <p>LINHA1 (<i>string</i>): texto da linha 1;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> do texto da linha 2;</p> <p>LINHA2 (<i>string</i>): texto da linha 2;</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TAM NS (<i>byte</i>): 61 (<i>decimal</i>);</p> <p>NS (<i>string</i>): "Não suportado" (sem as aspas);</p> <p>TEMPO (<i>byte</i>): Tempo de Exibição;</p> <p>DHCP (<i>byte</i>): IP dinâmico/fixo(48 = fixo, 49 = dinâmico);</p> <p>RESERVADO (<i>byte</i>): 48 (<i>decimal</i>);</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> ou do tempo de exibição, devemos subtrair 48 (<i>decimal</i>) do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
#wlanconfig?	#wlanconfig "WIFI ON" "MODO" "TAM SSID"	Retorna parâmetros de configuração do wifi.

	<p>"CANAL" "WEP ON" "TAM WEP" "CHAVE WEP"</p>	<p>WIFI ON (<i>byte</i>): 48 = Desabilita 49 = Habilita WIFI; MODO (<i>byte</i>): Modo; TAM SSID (<i>byte</i>): Tamanho da <i>string</i> do SSID; SSID (<i>string</i>): SSID. CANAL (<i>byte</i>): Canal; WEP ON (<i>byte</i>): 48 = Desabilita 49 = Desabilita WEP; TAM WEP (<i>byte</i>): Tamanho da <i>string</i> da chave WEP; CHAVE WEP (<i>string</i>): Chave WEP;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
<p>#rwlanconfig + "dados"</p>	<p>#rwlanconfig_ok</p>	<p>Recebe e carrega parâmetros de configuração do wifi: "WIFI ON" "MODO" "TAM SSID" "CANAL" "WEP ON" "TAM WEP" "CHAVE WEP"</p> <p>WIFI ON (<i>byte</i>): 48 = Desabilita 49 = Habilita WIFI; MODO (<i>byte</i>): Modo; TAM SSID (<i>byte</i>): Tamanho da <i>string</i> do SSID; SSID (<i>string</i>): SSID; CANAL (<i>byte</i>): Canal; WEP ON (<i>byte</i>): 48 = Desabilita 49 = Desabilita WEP;</p>

		<p>TAM WEP (<i>byte</i>): Tamanho da <i>string</i> da chave WEP;</p> <p>CHAVE WEP (<i>string</i>): Chave WEP;</p> <p>OBS: Para saber o tamanho real de cada <i>string</i> devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
#paramconfig?	#paramconfig "DHCP" "RESERVADO"	<p>Retorna parâmetros relativos ao DHCP</p> <p>DHCP (<i>byte</i>): 48 = DHCP desativado 49 = DHCP ativado;</p> <p>RESERVADO (<i>byte</i>) = sempre deverá ser 48;</p> <p>OBS: Para saber o valor real de cada campo, devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
#paramconfig + "dados"	#paramconfig_ok	<p>Recebe e carrega os parâmetros relativos ao DHCP: "DHCP" "RESERVADO":</p> <p>DHCP (<i>byte</i>): 48 = DHCP desativado 49 = DHCP ativado;</p> <p>RESERVADO (<i>byte</i>) = sempre deverá ser 48;</p> <p>OBS: Para saber o valor real de cada campo, devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>

<p>#rparamconfig + "dados"</p>	<p>#rparamconfig_ok</p>	<p>Recebe e carrega os parâmetros relativos ao DHCP: "DHCP" "RESERVADO":</p> <p>DHCP (<i>byte</i>): 48 = DHCP desativado 49 = DHCP ativado; RESERVADO (<i>byte</i>) = sempre deverá ser 48;</p> <p>OBS: Para saber o valor real de cada campo, devemos subtrair 48 do valor de cada <i>byte</i>. Não incluir as aspas duplas do retorno.</p>
<p>#mesg + "params"</p>		<p>Mostra mensagem recebida no display (2 linhas): "TAM L1" "LINHA1 TAM L2" "LINHA2"TEMPO" "RESERVADO":</p> <p>TAM L1 (<i>byte</i>): tamanho da <i>string</i> da mensagem da primeira linha;</p> <p>LINHA1 (<i>string</i>): mensagem da primeira linha;</p> <p>TAM L2 (<i>byte</i>): tamanho da <i>string</i> da mensagem da segunda linha;</p> <p>LINHA2 (<i>string</i>): mensagem da segunda linha;</p> <p>TEMPO (<i>byte</i>): tempo de exibição;</p> <p>RESERVADO (<i>byte</i>): reservado, deve ser = 48;</p> <p>OBS: Para saber o valor real de cada campo, devemos subtrair 48</p>

		do valor de cada <i>byte</i> . Não incluir as aspas duplas do retorno.
#macaddr? + "dados"	#macaddr "INTERFACE" "TAM MAC" "MAC"	Retorna o endereço MAC: "INTERFACE" INTERFACE (byte): 48 = Ethernet 49 = WiFi 57 = Atual; TAM MAC (byte): Tamanho da <i>string</i> do endereço MAC; MAC (string): Endereço MAC; OBS: Para saber o valor real de cada campo, devemos subtrair 48 do valor de cada <i>byte</i> . Não incluir as aspas duplas do retorno.

ROTINA DE VERIFICAÇÃO DE PRODUTO POR CÓDIGOS DE BARRAS

Ao efetuar a leitura de um código de barras, o Terminal envia para a aplicação o comando de pesquisa de código de barras, com o prefixo "#", seguido do valor de código de barras lido, ex: "#123". A aplicação deve verificar periodicamente se há novas solicitações de pesquisa de código de barras encaminhadas pelos terminais conectados. Isto é feito utilizando-se da função "bReceiveBarcode".

Ao obter o código de barras (ponteiro string) a aplicação deve procurar em sua base de dados a existência do valor referenciado pelo Terminal. Caso exista, deve retornar ao Terminal as informações de "produto" (linha 1) e "preço" (linha 2) através da função "bSendProdPrice", ex: "#Limao|2,00".

Caso esse valor não exista na base de dados, a aplicação deverá enviar uma mensagem para informar o Terminal que o produto não existe e utilizar a função "bSendProdNotFound" para validar se o terminal recebeu essa mensagem ou não.

- bReceiveBarcode

Delphi: function bReceiveBarcode(out ID_IP: DWORD; out ID_Socket: word; out Nbr: integer): PAnsiChar; stdcall; far;

C++: char * __stdcall bReceiveBarcode(DWORD ID_IP, word ID_Socket, int Nbr);

C#: public static extern unsafe IntPtr bReceiveBarcode(out uint ID_IP, out ushort ID_Socket, out int Nbr);

Finalidade: Usada periodicamente para identificar se houve solicitação de pesquisa de código de barras pendente. Caso haja, retornará o código de barras (ponteiro string), o endereço IP e a porta do terminal solicitante;

Entrada: N/A;

Saída: (char *), ID_IP, ID_Socket, Nbr

- (char *): ponteiro de string com o código de barras recebido

- ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal

- ID_Socket (word): valor numérico da porta de comunicação com o terminal

- Nbr (int): número de dados do código de barras (sempre será "1", pois o sistema analisa um código de barras por vez);

- bSendProdNotFound

Delphi: function bSendProdNotFound(ID_IP: DWORD): boolean; far; stdcall;

C++: bool __stdcall bSendProdNotFound(DWORD ID_IP);

C#: public static extern unsafe bool bSendProdNotFound(uint ID_IP);

Finalidade: Informar a aplicação que o terminal solicitante recebeu a mensagem;

Entrada: ID_IP (DWORD): valor numérico do endereço de rede (IP) do terminal;

Saída: (bool): Caso o terminal receba a mensagem enviada pela aplicação, retornará TRUE, caso contrário, retornará FALSE;

- bSendProdPrice

Delphi: function bSendProdPrice(ID_IP: DWORD; NameProd: PAnsiChar; PriceProd: PAnsiChar): Boolean; stdcall; far;

C++: bool __stdcall bSendProdPrice(DWORD ID_IP, char *NameProd, char *PriceProd);

C#: public static extern unsafe bool bSendProdPrice(UInt32 IP, IntPtr Prod, IntPtr Valor);

Finalidade: Envia para a primeira e segunda linha do display do terminal solicitante o nome e o preço do produto pesquisado;

Entrada: ID_IP, NameProd, PriceProd:

- ID_IP (DWORD): valor numérico de endereço de rede (IP) do terminal
- NameProd (char *) : ponteiro string com o nome do produto
- PriceProd (char *): ponteiro string com o preço do produto

Saída: (bool) : Caso a mensagem seja encaminhada com sucesso, retornará TRUE, caso contrário, retornará FALSE;

IDENTIFICAÇÃO DA VERSÃO DA DLL

Reforçando que algumas funções podem variar segundo a versão da DLL, caso tenha dúvidas quanto a versão da DLL utilizada, basta validar a função "dll_version":

- dll_version

Delphi: function dll_version: PAnsiChar; stdcall; far;

C++: char * __stdcall dll_version(void);

C#: static extern IntPtr dll_version();

Finalidade: Retornar a versão da DLL;

Entrada: N/A;

Retorno: (char *): ponteiro de string com o valor da versão da DLL.

FINALIZAÇÃO DO SERVIÇO

Por fim, para finalizar o serviço basta utilizar a função "bTerminate":

- bTerminate

Delphi: function bTerminate: boolean; far; stdcall;

C++: bool __stdcall bTerminate(void);

C#: public static extern bool bTerminate();

Finalidade: Última rotina chamada antes de finalizar serviço;

Entrada: N/A;

Retorno: (bool) : Caso tenha sucesso, retornará TRUE, caso contrário, retornará FALSE;

SUPORTE TÉCNICO TANCA

Em caso de dúvidas entrar em contato através de solicitação de suporte técnico no canal <https://bit.ly/3KqYn8L>, nossa equipe está a disposição.

