

ScanServ OCX

Description

Scanserv.ocx is a library module that implements the ProductInfo protocol. It is meant primarily to support Visual Basic but can also be used with Visual C++ and .NET. It is single-threaded, which means it will only handle one terminal *at a time* but there is no limit to the number of terminals that can be supported in a sequential manner.

It would not be a problem for a Visual Basic program using scanserv.ocx to support 6 or more terminals depending upon the frequency of use of the terminals. By default, the terminals wait 3 seconds for a response from the host. As most actions on the host side take a small fraction of this time there is only a small chance that the response to a terminal would be delayed, and a minuscule chance that the terminal would time out because of a busy host. This assumes that the host is getting the query, looking up the response, and sending that single response to the terminal. If the host-terminal interaction is more sophisticated than that, the chances of being a negative influence on other terminals goes up according to the time spent on each terminal.

Scanserv is demonstration software intended to show the programmer how to create a server rather than to be used as an actual application. The same applies to all the utilities. While all efforts have been made to make the software bug-free and utilitarian, it is up to the end user to validate the software as appropriate for their enterprise. The sources to scanserv.ocx are provided and are well documented should the end user wish to make changes.

Other example code includes demoserv in both single-threaded and multiple-threaded versions, again with well-documented source. These programs have been compiled and run on various Windows systems, as well as Linux, Unix, and VAX/VMS. Demoserve programs are stand-alone ProductInfo servers rather than libraries to be used with other programs, and provide examples of the type of functionality possible. Also provided is nanoserv, a minimalist example of a ProductInfo server.

Purpose

The purpose of **ScanServ OCX** is to build a simple Visual Basic server based on the **ProductInfo** protocol. It supports only a single connection at a time. This software is provided **AS-IS** and not certified for production use.

Following is a list of methods provided by **ScanServ OCX**, followed by a detail description of each function.

ScanServ.**Connect** (port As Long) As Long
ScanServ.**Listen** As Long
ScanServ.**GetPacketType** As Long
ScanServ.**GetProductID** As String

ScanServ.**GetProductQuery** As String
ScanServ.**GetTrivialPacket** As String
ScanServ.**SendFile** (Name As String) As Long
ScanServ.**SockSend** (Buffer As String, Size As Long)
ScanServ.**Close**
ScanServ.**Exit**

Connect

Connect function creates a socket and place it in a state where it is listening for an incoming connection (listening socket).

Connect
(port As Long)
As Long

Parameters

Port [in] Descriptor identifying a port to create the socket for (SLOPPI Port).

Return Values

If no error occurs, **Connect** returns a descriptor referencing the listening socket. Otherwise, a value of **-1** is returned.

Remarks

On exit, **Exit** should be called in order to close the listening socket.

Listen

Listen function waits for an incoming connection on a listening socket, read and store the data packet.

Listen

As Long

Parameters

None

Return Values

- 1** : Fatal Error
- 0** : Error, Try again
- 1** : Packet received is a **Trivial** Packet
- 2** : Packet received is a **Nominal** Packet

Remarks

GetPacketType

GetPacketType waits for an incoming packet.

GetPacketType

As Long

Parameters

None

Return Values

GetPacketType returns the Packet Token.

PacketToken: TERM, CAPQ, CAPR, MDSQ, MODR,
 ERR_, PROD, DATA, RGST, EVNT

Remarks

GetProductID

GetProductID returns product ID.

GetProductID

As String

Parameters

None

Return Values

GetProductID returns an string specifying the ProductID.

Remarks

GetProductQuery

GetProductQuery returns product query.

GetProductQuery

As String

Parameters

None

Return Values

GetProductQuery returns an string specifying the Product Query.

GetTrivialPacket

GetTrivialPacket returns the trivial data packet received.

GetTrivialPacket
As String

Parameters

None

Return Values

GetTrivialPacket returns the trivial packet, or NULL If the packet received was nominal.

SendFile

SendFile send the file specified.

SendFile

(Name As String)
As Long

Parameters

Name [in] Specifies the name of the file to send.

Return Values

0: Error
1: Success

Remarks

SendFile sends the file name, the four-bytes file size, and then the file.

SockSend

SockSend sends raw data to ScanVue/ShopVue.

```
SockSend  
    ( Buffer As String,  
      Size As Long  
    )  
As Long
```

Parameters

Buffer [in] Raw Data.

Size [in] Size of data.

Return Values

None

Close

Close Closes the active connections.

Close

Parameters

None

Return Values

None

Remarks

Exit

Exit Closes the listening socket.

Exit

Parameters

None

Return Values

None

Remarks

On exit, **Exit** should be called in order to close the listening socket.