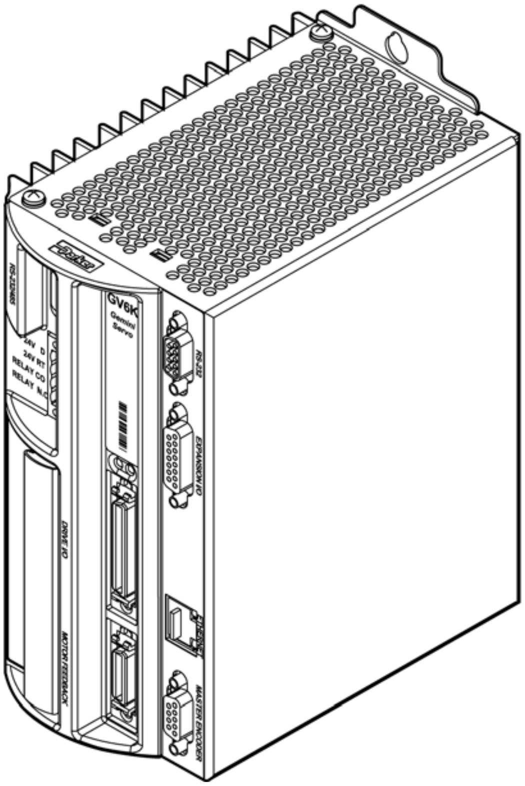
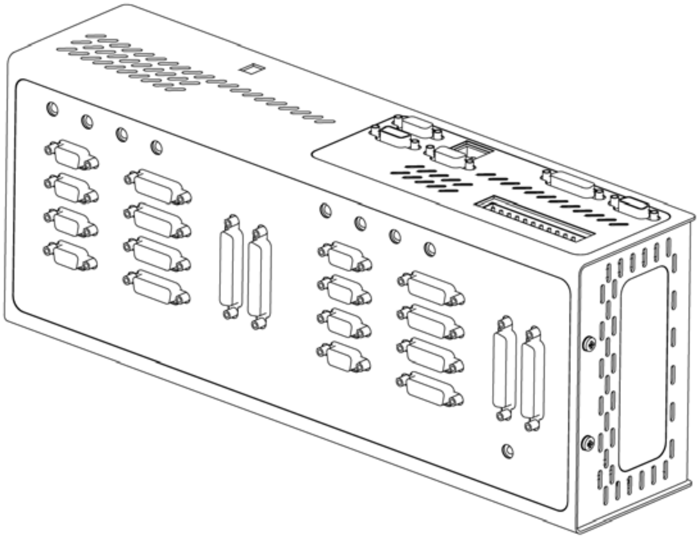
Parker Automation

COM6SRVR OCX User's Guide for Gemini & 6K Series Products





⚠ **Warning —** 6K, Gem6K, and Gemini series products are used to control electrical and mechanical components of motion control systems. You should test your motion system for safety under all potential conditions. Failure to do so can result in damage to equipment and/or serious injury to personnel.

Since Parker Hannifin constantly strives to improve all of its products, we reserve the right to change this guide, and software and hardware mentioned therein, at any time without notice.

In no event will the provider of the equipment be liable for any incidental, consequential, or special damages of any kind or nature whatsoever, including but not limited to lost profits arising from or in any way connected with the use of the equipment or this guide.

Technical Assistance

Contact your local automation technology center (ATC) or distributor.

North America and Asia
Parker Hannifin
5500 Business Park Drive
Rohnert Park, CA 94928
Telephone: (800) 358-9070 or (707) 584-7558
Fax: (707) 584-3793
Email: emn_support@parker.com
Internet: http://www.parkermotion.com

Europe (non-German speaking)
Parker Hannifin plc
Electromechanical Automation, Europe
Arena Business Centre
Holy Rood Close
Poole
Dorset, UK
BH17 7BA
Telephone: +44 (0) 1202 606300
Fax: +44 (0) 1202 606301
Email: support.digiplan@parker.com
Internet: http://www.parker-emd.com

Germany, Austria, Switzerland
Parker Hannifin
Postfach: 77607-1720
Robert-Bosch-Str. 22
D-77656 Offenburg
Telephone: +49 (0) 781 509-0
Fax: +49 (0) 781 509-176
Email: sales.hauser@parker.com
Internet: http://www.parker-emd.com

Italy
Parker Hannifin
20092 Cinisello Balsamo
Milan, Italy via Gounod, 1
Telephone: +39 02 6601 2478
Fax: +39 02 6601 2808
Email: sales.sbc@parker.com
Internet: http://www.parker-emd.com

Parker
Automation

Technical Support E-mail

emn_support@parker.com

Before You Begin

Introduction

The Com6srvrNet control is an Active X (OCX) control-a visual element that you can add to a form-to make Ethernet communications possible with the Compumotor 6K Controller. Designed primarily for Microsoft Visual Basic programmers and HMI (human-machine interface) applications that can use Active X controls, the Com6srvrNet.ocx encapsulates the Com6srvr INet COM control. Using the OCX, you can expect a small degradation in communications speed.

Installing the OCX

The Com6srvrnet.exe (We kept the same name for legacy issues) must be installed and registered on the workstation before you can install the OCX. For more information, see *Communications Server (COM6SRVR)-Programming Notes* in Help.

To include the Com6SrvrNet.ocx in a Visual Basic project:

1. Open Visual Basic.
2. On the Project menu, click Components.
3. On the Controls tab, select the Active X Control Communications Server for Compumotor check box.
4. Click OK.

The Com6srvrNet.ocx uses the Com6srvr with late binding-Com6srvr is not linked at compile time. The link is created dynamically at run-time. For more information, see *Communications Server (COM6SRVR) - Programming Notes* in Help.

Connecting & Disconnecting

At run-time, you can connect and disconnect using the Connect or DeviceConnect and Disconnect methods. Connect is a legacy interface but its still supported. Disconnect remains the same. Additionally, the Com6srvrNet OCX automatically disconnects when the application is terminated. For more information, see **Communications Server (COM6SRVR) - Ethernet Methods, Properties & Events** in Help.

Properties, Methods, & Enumerations

The names of properties and methods for Com6srvrNet.Ocx reflect the names of the underlying Com6srvr functions. For example, MOTORPOS uses the same function name in the Com6srvrNet.ocx and the Com6srvr. In fact, most properties Com6srvrNet.ocx behave similarly to the properties and methods for Com6srvr.

A few names of properties and methods used by Com6srvr create problems in the OCX environment. The following table shows the command names for Com6srvr and the modified names to use with Com6srvrNet.Ocx:

Com6srvr command	**New Com6srvrNet.Ocx command**
Write	WriteCmd
Read	ReadResponse
WriteBlocking	WriteCmdBlocking
SendVariable	SendIntegerVariable, SendBinaryVariable,
and	
	SendRealVariable
Connect,Connect2,Connect3 (6K)	DeviceConnect
Connect,Connect3 (Gem6K)	DeviceConnect

Bit Status Convention

When retrieving bit-oriented properties (e.g., AxisStatus, ErrorStatus, Limits, SystemStatus, etc.) note that the convention in the 6K programming language differs from the convention used for C and Assembly programming languages. Compumotor's 6K convention refers to the bits within a 32 bit long integer as bits 1 through 32 (left to right). The C and Assembler Programmer's convention refers to these as bits 0 through 31 (right to left). When masking these bits, you should be aware of this subtle difference when referring to 6K documentation.

6K Ethernet Functions

AlarmStatus

Description: The AlarmStatus property returns the state of the controller's alarm status.

Visual Basic: object.**AlarmStatus**.(*bit* As Long) As Long

Version: **1.0.0** 6K/Gem6K Ethernet Only

Parameter: *bit*......... Long Integer.
Specifies the status bit of the alarm status to return. It can be a number between 0 and 32. Values between 1-32 represent the alarm bits as described in the table below (refer also to the `INTHW` command). Specifying a bit value of 0 returns the entire 32 bit alarms status as a long value; otherwise a value of 1 or 0 is returned to indicate the state of any single bit. When any single bit status is retrieved using the AlarmStatus property, that bit status is automatically cleared by the Communications Server. If a bit value of 0 is used then all alarm status bits are cleared.

Return Type: Long Integer.
This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: When the **6K/Gem6K** sends an alarm packet to the COM6SRVR, the FastStatus structure is automatically updated, regardless of state of FSEnabled.

Bit #	**Function**
1	Software (forced) Alarm #1
2	Software (forced) Alarm #2
3	Software (forced) Alarm #3
4	Software (forced) Alarm #4
5	Software (forced) Alarm #5
6	Software (forced) Alarm #6
7	Software (forced) Alarm #7
8	Software (forced) Alarm #8
9	Software (forced) Alarm #9
10	Software (forced) Alarm #10
11	Software (forced) Alarm #11
12	Software (forced) Alarm #12
13	Command Buffer Full
14	ENABLE input Activated
15	Program Complete
16	Drive Fault on any Axis
17	Reserved
18	Reserved
19	Limit Hit - hard or soft limit, on any axis

20	Stall Detected (stepper) or Position Error (servo) on any axis	
21	Timer (TIMINT)	
22	Ethernet fail (RESET or ER.22 occurred) (also invokes an error dialog)	
23	Input - any of the inputs defined by INFNCi-I or LIMFNCi-I	
24	Command Error	
25	Motion Complete on Axis 1	
26	Motion Complete on Axis 2	(Gem6K – Reserved)
27	Motion Complete on Axis 3	(Gem6K – Reserved)
28	Motion Complete on Axis 4	(Gem6K – Reserved)
29	Motion Complete on Axis 5	(Gem6K – Reserved)
30	Motion Complete on Axis 6	(Gem6K – Reserved)
31	Motion Complete on Axis 7	(Gem6K – Reserved)
32	Motion Complete on Axis 8	(Gem6K – Reserved)

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

AnalogInput

Description: The AnalogInput property returns the value (in counts) of the specified analog input.

Visual Basic: object.AnalogInput(*channel* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *channel*......... Long Integer.
Specifies the analog input channel (channel 1 or 2) value to return. This property uses only the first two analog inputs detected on an I/O brick connected to the 6K/Gem6K, regardless of the `ANIEN` (analog input enable) setting.

Return Type: Long Integer.
The method returns the specified analog input value in counts.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

AxisStatus

Description: Use the AxisStatus property to retrieve the current axis status for the specified axis. This is the same as the legacy AxisStatus but if there are no Parameters supplied then the first axis is the default.

Visual Basic: object.AxisStatus(Optional *axis* As Long = 1) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *axis*......... Long Integer.
Specifies the axis about which the status pertains. The range for this value is 1-8. For Gem6K there is no need to supply an Axis Number.

Return Type: Long Integer.
The long integer value represents the current axis status for the specified axis. Refer to the TAS command description for a list of the status elements.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.
This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Connect

Description: The Connect method opens a connection to a 6K controller over Ethernet

Visual Basic: object.Connect(*sIpaddr* as String) As Long

Version: 1.0.0 6K Ethernet Only

Parameter: *sIpaddr*.........String.
Represents the target controller's IP address.

Return Type: Long Integer.
If the connection is successfully opened, the method returns a positive value representing the number of connected clients. If the connection is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: The Server can handle a virtually unlimited number of Ethernet connections to different IP addresses (limited by computer resources). The 6K takes up to one minute for an Ethernet connection to truly expire and be available for a new connection. This is a Legacy Interface but is still supported.
Background Commands: After a successful connection is made, the following commands are sent to the controller:

```
!PORT0
!ERRLVL4
!EOT13,0,0
!EOL13,10,0
```

ECHO mode is initially disabled (`ECHO0`) by the 6K during Ethernet communications.

DeviceConnect

Description: The Connect method opens a connection to a 6K,Gem6K, or Gemini controller over Ethernet or RS232

Visual Basic: object.DeviceConnect(Optional *iConnectInterface* As Long = 1, Optional *strNetAddress* As String = "1", Optional *bQuiet* As Boolean = False, Optional *lTimeOut* As Long = 15) As Long

Version: 1.3.0

Parameter: *iConnectInterface*…….Long

Represents the Device you would like to connect to

6K = 1

GEMINI = 2

GEM6K = 3

StrNetAddress………..String.

Represents the RS232 port number or the IP Address

BQuiet……………Boolean

Specifies whether the connection dialog will be shown. True will hide the connection dialog, false shows the connection dialog.

LTimeOut………..Long

A constant that specifies a timeout period in mS for the Ethernet connection attempt. The range for lTimeout is (0 - 60000).

Return Type: Long Integer.

If the connection is successfully opened, the method returns a positive value representing the number of connected clients. If the connection is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: This is the new connect method. You can either connect over RS232 or Ethernet. By Default (i.e. DeviceConnect()) will connect to a 6K over RS232 on port 1.The Server can handle up to two RS232 connections. The RS232 server assumes 9600 Baud operation. The last two parameters are not needed for Serial Communication. The Server can handle a virtually unlimited number of Ethernet connections to different IP addresses (limited by computer resources). The 6K takes up to one minute for an Ethernet connection to truly expire and be available for a new connection.

For a RS232 Connection for a 6K ONLY

Background Commands: After a successful connection is made, a "PORT0:" commands are sent to the controller:

For an Ethernet Connection

Background Commands: After a successful connection is made, the following commands are sent to the controller:

```
!PORT0
!ERRLVL4
!EOT13,0,0
!EOL13,10,0
```

ECHO mode is initially disabled (`ECHO0`) by the 6K/Gem6K during Ethernet communications.

Examples: DeviceConnect(1, 2)   ;Connects to a 6K over COM 2

```
DeviceConnect(3, "100.100.100.100", True)       ;Connects
to a Gem6K over
                    ;Ethernet and does this quietly
```

EncoderPos

Description: The EncoderPos property returns the current encoder position (TPE) in counts for the specified axis. This is the same as the legacy EncoderPos but if there are no Parameters supplied then the first axis is the default.

Visual Basic: object.EncoderPos(Optional *axis* As Long = 1) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *axis*……….Long Integer.
Specifies the axis number of the encoder. The range for this value is 1-8. For Gem6K there is no need to supply an Axis Number.

Return Type: Long Integer.
The value represents the current encoder position (TPE) in counts for the specified axis.

Remarks: This is a read-only property.

Inputs

Description: Use the Inputs property to check the current state of the inputs (TIN) on a specific brick.

Visual Basic: object.Inputs(*brick* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *brick*.........Long Integer.
Represents the number of the brick where the inputs reside. Range is 0-3. Brick 0 represents the onboard inputs. Bricks 1-3 represent expansion I/O bricks 1-3.

Return Type: Long Integer.
The value represents the current state of the inputs (`TIN`) for the specified brick.
This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

IsBitSet

Description: Use the IsBitSet to test the validity of a bit.

Visual Basic: object.IsBitSet(*TestValue* As Long, *TestBit* As Long) As Boolean.

Version: 1.0.0

Parameter: *TestValue*.........Long Integer
Represents the value you are testing. If the bit you are testing is a 1, the Boolean returns TRUE.
TestBit.........Long Integer
Represents the bit to test.

Return Type: Boolean

Remarks: The IsBitSet uses the 6K language convention, where the bit pattern is represented as 1 to 32, from left to right when represented as a string. To get the string value, use the LongTo 6Kbinary.

Longto6Kbinary

Description: Use the LongTo6KBinary function to convert a Long Integer to binary.

Visual Basic: object.LongTo6KBinary(*InValue* As Long) As String

Version: 1.0.0

Parameter: *InValue*.........As Long

Return Type: String

Remarks: Using a Long Integer is more efficient for testing bits, and manipulating and storing data. However, it is more convenient to show the data using the 6K language convention. Programmers familiar with the 6K language might want to see, for instance, the Axis Status represented as a string of 32 bits (1s and 0s). For example, when using the AxisStatus() method, the bit pattern 0000_0000_0000_1000_0000_0001_0000_0000 returns the value 8392704.

MotorPos

Description: The MotorPos property returns the current commanded position (TPC) for the specified axis.
Visual Basic: object.MotorPos(Optional *axis* As Long = 1) As Long
Version: 1.0.0 6K/Gem6K Ethernet Only
Parameter: *axis*......... Long Integer.
Specifies the axis number (range is 1-8).
Return Type: Long Integer.
The value represents the current commanded position (TPC) in counts for the specified axis.
Remarks: This is a read-only property.
Defaults to axis one.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

MotorVel

Description: The MotorVel property returns the current commanded motor velocity (TVEL) for the specified axis.

Visual Basic: object.MotorVel(Optional *axis* As Long =1) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *axis*......... Long Integer.
Specifies the axis number (range is 1-8).

Return Type: Long Integer.
The value represents the current commanded velocity (TVEL) in counts for the specified axis.

Remarks: This is a read-only property.
Defaults to axis one.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Outputs

Description: The Outputs property returns the state of the outputs (TOUT) on the specified brick.

Visual Basic: object.Outputs(*brick* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *brick*......... Long Integer.

Represents the number of the brick where the outputs reside. Range is 0-3. Brick 0 represents the onboard outputs. Bricks 1-3 represent expansion I/O bricks 1-3.

Return Type: Long Integer.

The value represents the state of the outputs (`TOUT`) on the specified brick.

This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Ping

Description: Pings a Gem6K or 6K at the specified IP address.

Visual Basic: object.Ping(Optional *strIPAddr* As String = "", Optional *lTimeOut* As Long = 15000) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strIPAddr* ……….String

Represents the IP address of the Device if NOT connected already.

LTimeOut ……….Long

Timeout period in mS for Ping. The range for lTimeout is (0 - 30000)

Return Type: Long.

The method returns a positive value if the operation is successful; otherwise, it returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: Allows you to ping a device with out being connected. If no IP address is given then the Current IP Address given during the Connect method is used.

ReadResponse

Description: The Read method retrieves command responses from the controller.

Visual Basic: object.ReadResponse() As String

Version: 1.0.0

Parameter: NONE

Return Type: String.

The read method does not wait for incoming responses from the controller. It returns immediately with a string containing the controller's response at the time of the request. If no response is available, this method returns an empty string. The Read method response is limited to 256 characters. If the response is longer than 256 characters, the excess characters will remain in the Com6srvr buffer. Multiple reads are necessary for long responses.

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

RequestFastStatusUpdate

Description: The RequestFastStatusUpdate method allows the Com6srvr to request a fast status update as needed, without having to enable the fast status "Streaming Mode" (FSEnabled) or set an update interval (FSUpdateRate).

Visual Basic: object.RequestFastStatusUpdate As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
If the RequestFastStatusUpdate call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: This method is one of two "On Demand" fast status update options. The other option is for the 6K to execute the `NTSFS` command. Using an On Demand update technique is more efficient for interactive PC applications than the Streaming Mode, and reduces network traffic.

SendBinaryVariable

Description: The SendBinaryVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendBinaryVariable(*varnum* As Long, *value* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer.
Specifies the one variable to be sent. Only one bit can be set in the Varnum.
value.........Variant.
Specifies the value of the variable to be sent. The Value specifies the actual variable being sent.

Return Type: Long Integer.
If the SendBinaryVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the Varnum or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: For more information about sending variable packets, see *Communications Server (COM6SRVR) - Programming Notes* in Help. The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

SendFile

Description: The SendFile method is used to download program files to the controller. (Refer also to SendFileBlocking and SetSendFileDelay.)

Visual Basic: object.SendFile(Optional *filename* As String = "") As Long

Version: 1.0.0

Parameter: *filename*.........String.
Represents the name of the program file (containing 6K programs/code) to be downloaded. If the filename is an empty string, then the user will be prompted for the filename.

Return Type: Long Integer.
If successful, the method returns a positive value; if unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFile method strips comments from the downloaded 6K code. That is, all text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.
Background Commands: At the beginning of a file download operation, these commands are sent to the controller:

```
!PORT0
!ERRLVL0
```

After the download process is completed, these commands are sent to the controller:

```
!PORT0
!ERRLVL4
!EOT13,0,0
!EOL13,10,0
```

Note: If the download process is canceled, an "END" command is sent to the controller.

SendFileBlocking

Description: This method, like the SendFile method, is used to download program files to the controller. It differs from SendFile in that it blocks the return of the method call until the 6K acknowledges that the file has been downloaded. A dialog informs the user to wait for the 6K to acknowledge.

Visual Basic: object.SendFileBlocking(Optional *filename* As String = "") As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *filename*.........String.
Represents the name of the program file (containing 6K programs/code) to be downloaded. If the filename is an empty string, then the user will be prompted for the filename.

Return Type: Long Integer.
The method returns a positive value if the operation is successful; otherwise, it returns an error code. The dialog has a Cancel button (a software specified Timeout is not provided). If the user clicks the Cancel button, the method returns error code -15 SendFileBlocking cancelled out.
Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileBlocking method strips comments from the downloaded 6K code. That is, all text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.
Background Commands: (same as SendFile)
Note: If the download process is canceled, an "END" command is sent to the controller and the method returns error code -15 SendFileBlocking cancelled out.

SendFileQuiet

Description: The SendFileQuiet method is used to download program files to the controller while suppressing the download status dialog message.

Visual Basic: object.SendFileQuiet(Optional *filename* As String = "") As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strFileName*.........String
Represents the name of the program file (containing 6K or Gem6K program/code) to be downloaded. As default the filename is an empty string so the user will be prompted for the filename.

Return Type: Long Integer.
The method returns a positive value if the operation is successful; otherwise, it returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileQuiet method strips comments from the download code. That is, all text between the comment delimiter (semi - colon) and the command delimiter (carriage return or line feed) is removed.
Note: The SendFileQuiet method should be called when motion is *not* in progress and programs are *not* running.
Background Commands: At the beginning of a file download operation, these commands are sent to the controller:

```
!PORT0
!ECHO0
!ERRLVL0
!EOT1,0,0
!EOL10,0,0
!TDIR
```

After the download process is completed, these commands are sent to the controller:

```
!PORT0
!EOT13,0,0
!EOL13,10,0
!ERRLVL4
!ECHO1
```

Note: If the download process is canceled, an "END" command is sent to the controller.

SendFileQuietBlocking

Description: This method, like the SendFileQuiet method, is used to download program files to the controller while suppressing the download dialog. It differs from SendFileQuiet in that it blocks the return of the method call until the Gem6K acknowledges that the file has been downloaded. A dialog informs the user to wait for the Gem6K to acknowledge. The dialog has a CANCEL button (a software specified Timeout is no provided). If the user clicks the CANCEL button, the method returns the error -15.

Visual Basic: object.SendFileBlocking(Optional *filename* As String = "") As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strFileName*.............String

Represents the name of the program file (containing 6K/Gem6K programs/code) to be downloaded. By default the filename is an empty string; the user will be prompted for the filename unless one is supplied.

Return Type: Long Integer.

The method returns a positive value if the operation is successful; otherwise, it returns an error code (error code -15 is returned when the user clicks the CANCEL button). Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileQuietBlocking method strips comments from the downloads 6K/Gem6K code. That is, al text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.

Background Commands: (same as SendFile)

Note: If the download process is canceled, an "END" command is sent to the controller and the error code (-15) is returned.

SendIntegerVariable

Description: The SendIntegerVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendIntegerVariable(*varnum* As Long, *value* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer.
Specifies the one variable to be sent. Only one bit can be set in the Varnum.
value.........Variant.
Specifies the value of the variable to be sent. The Value specifies the actual variable being sent.

Return Type: Long Integer.
If the SendIntegerVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the Varnum or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: For more information about sending variable packets, see *Communications Server (COM6SRVR) - Programming Notes* in Help. The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

SendRealVariable

Description: The SendRealVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendRealVariable(*varnum* As Long, *value* As Double) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *varnum*..........Long Integer.
Specifies the one variable to be sent. Only one bit can be set in the Varnum.
value..........Variant.
Specifies the value of the variable to be sent. The Varnum specifies the actual variable being sent.

Return Type: Long Integer.
If the SendRealVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the Varnum or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: For more information about sending variable packets, see *Communications Server (COM6SRVR) - Programming Notes* in Help. The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

SendVariable

Description: The SendVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendVariable(*VariableMask* As Long, *NewValue* As Variant) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *VariableMask*.........Long Integer.

Specifies the one variable to be sent. Constants are defined for the mask bits (mask bits for Visual Basic are provided below). Only one bit can be set in the VariableMask.

NewValue..........Variant.

Specifies the value of the variable to be sent. The actual variable being sent is specified by the VariableMask. Because the SendVariable Method can be used to send integer, real or binary variables, the data type can either be a long integer or a double floating point value. Using a Variant parameter allows the flexibility of sending any integer type, while allowing the Com6srvr to cast the Variant into the appropriate data type.

Return Type: Long Integer.

If the SendVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the VariableMask or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

Variable Packet Mask Bits for Visual Basic

```
Public Const VARI1 As Long = 1
Public Const VARI2 As Long = 2
Public Const VARI3 As Long = 4
Public Const VARI4 As Long = 8
Public Const VARI5 As Long = 16
Public Const VARI6 As Long = 32
Public Const VARI7 As Long = 64
Public Const VARI8 As Long = 128
Public Const VARI9 As Long = 256
Public Const VARI10 As Long = 512
Public Const VARI11 As Long = 1024
Public Const VARI12 As Long = 2048
Public Const VAR1 As Long = 4096
Public Const VAR2 As Long = 8192
Public Const VAR3 As Long = 16384
Public Const VAR4 As Long = 32768
Public Const VAR5 As Long = 65536
Public Const VAR6 As Long = 131072
Public Const VAR7 As Long = 262144
Public Const VAR8 As Long = 524288
Public Const VAR9 As Long = 1048576
Public Const VAR10 As Long = 2097152
Public Const VAR11 As Long = 4194304
Public Const VAR12 As Long = 8388608
Public Const VARB1 As Long = 16777216
Public Const VARB2 As Long = 33554432
Public Const VARB3 As Long = 67108864
Public Const VARB4 As Long = 134217728
Public Const VARB5 As Long = 268435456
Public Const VARB6 As Long = 536870912
Public Const VARB7 As Long = 1073741824
Public Const VARB8 As Long = &H80000000
```

SetSendFileDelay

Description: SetSendFileDelay allows you to specify the delay for each character when making a call to the SendFile method. (See Remarks below for detail)

Visual Basic: object.SetSendFileDelay(*delay* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *delay*.........Long Integer.
The parameter specifies the delay for each character transmitted. Valid range is 0-100 (milliseconds).
0 = no delay.

Return Type: Long Integer.
If the specified delay is valid, the method returns zero; if the specified delay is out of range, the method returns error code -13 Specified timeout is out of range. Please see "Possible Error Codes from Com6srvr."

Remarks: When making a call to the SendFile Ethernet method, a 2-ms per character delay is inserted to allow the commands to be transmitted and processed through the TCP/IP stack and the 6K internal buffers. In some cases the delay is not necessary, because the TCP/IP stack takes care of flow-control. In other cases, it might be desirable to allow a longer delay, such as when sending data over a very busy network. This method provides a means to control the delay, allowing a delay of 0-100 ms per character.

SetWatchdog

Description: The SetWatchdog method enables Ethernet watchdog hand-shaking between the Com6srvr and the 6K Controller.

Visual Basic: object.SetWatchdog(*Timeout* As Long, *Ticker* as Long) as Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *Timeout*.........Timeout period in seconds (see guidelines below)

Ticker.........Number of "heartbeat" packets to send during the timeout period

Return Type: Long Integer.

Returns the number of bytes sent to the watchdog if successful, or a negative error code (usually error code -11, which indicates that an invalid configuration was specified) if not successful. Please see "Possible Error Codes from Com6srvr."

Remarks: The Ethernet watchdog allows the Com6srvr and 6K Controller to gracefully recover when communication between the 6K and Com6srvr is lost. Such situations might arise from the loss of power to the 6K or to the PC while an Ethernet connection was active. Because the Com6srvr and the 6K establish a "virtual" software based connection, both parties agree to be "connected". By enabling the Watchdog, a heartbeat packet is sent periodically by the Com6srvr. The 6K detects the heartbeat and echoes it back to the Com6srvr. When the watchdog is enabled, the Com6srvr sends the heartbeat and expects an echo. Similarly, the 6K controller is expecting the heartbeat from the Com6srvr. If the Com6srvr does not detect the echoed heartbeat (within the constraints set by the Timeout and Ticker parameters), the watchdog is considered timed out. If the 6K does not receive the heartbeat (within the same Timeout and Ticker constraints), the 6K considers the watchdog timed out.

Loss or delay of a single echoed heartbeat could happen quite frequently on a busy network connection. Therefore, we provide a method whereby a number of re-tries are attempted over a specific timeout period. If all re-tries fail within the timeout period, then the watchdog is considered to have timed out. This functionality is provided by the Timeout and Ticker parameters. The constraints for these parameters are as follows:

- To enable the watchdog, set Timeout > 0 > Ticker.
- To disable the watchdog, set Timeout = 0 and set Ticker = 0.
- The Timeout/Ticker ratio must be ≤ 65.

RECOMMENDATION: Set Timeout = 100 and Ticker = 5, which provides a heartbeat once every twenty seconds (100 seconds / 5 tries = 20 seconds/attempt). If none of the 5 heartbeats are acknowledged in 100 seconds, the watchdog times out.

WHEN A WATCHDOG TIMEOUT OCCURS:

- In the 6K: When the 6K detects a watchdog timeout, it attempts to send an alarm packet to the Com6srvr (AlarmStatus bit #22). It then closes the Ethernet connection and reports "disconnected" in the TNT report. If the user has enabled error-checking bit #22 (ERROR.22-1), the 6K will execute a GOSUB branch to the ERRORP program. Within the ERRORP program, the watchdog timeout can be cleared by disabling ERROR bit #22 (ERROR.22-0).

- In the Com6srvr: When the COM6SRVR detects a watchdog timeout, the IsWatchdogTimedOut returns TRUE. (If the COM6SRVR receives the alarm packet from the 6K, it will also display an alert dialog to the user.) A client application can poll the IsWatchdogTimedOut. When a timeout is detected by the Com6srvr, the Client application should "disconnect" the Com6srvr. You can use Sub Disconnect. After the Com6srvr has been disconnected, creating a new Com6srvr object and "connecting" Ethernet will clear the watchdog timeouts. All client applications for that particular 6K Ethernet connection should be disconnected.

Var

Description: The Var property returns the value of the specified real variable (VAR). Variables VAR1 through VAR12 may be reported.

Visual Basic: object.Var(*lNum* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *lNum*.........Long Integer. Represents number of the real variable (VAR*lNum*). Range is 1-12.

Return Type: Double

The value represents the value of the specified real variable (VAR). The initial value is zero until an Extended Fast Status packet arrives.

Remarks: This is a read only property

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

VarB

Description: The VarB property returns the value of the specified binary variable (VARB).

Visual Basic: object.VarB(*varnum* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer. Represents number of the binary variable (VARB*varnum*). Range is 1-10.

Return Type: Long Integer.
The value represents the value of the specified binary variable (VARB).

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

VarI

Description: The VarI property returns the value of the specified integer variable (VARI).

Visual Basic: object.VarI(*varnum* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer. Represents number of the binary variable (VARI*varnum*). Range is 1-10.

Return Type: Long Integer.
The value represents the value of the specified integer variable (`VARI`).

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

WriteCmd

Description: The WriteCmd method is used to send commands to the controller.

Visual Basic: object.WriteCmd(*sCmd* As String) As Long

Version: 1.0.0

Parameter: *sCmd*.........String.
A string of commands to be sent. Multiple commands can be sent, but each command should be separated with a valid 6K command delimiter (colon, carriage return, or line feed). The command string should be limited to 256 characters or less. Larger command strings may cause an overflow in the 6K's command buffer.

Return Type: Long Integer.
This method returns a positive value corresponding to the number of bytes sent, or an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

WriteCmdBlocking

Description: The WriteCmdBlocking method, an alternative to the WriteCmd method, is used to send commands to the controller. The primary difference from the WriteCmd method is that WriteCmdBlocking does not return from the method call until the commands have been executed in the controller within a specified time.

Visual Basic: object.WriteCmdBlocking(*sCmd* As String, *nTimeout* As Long) As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *sCmd*……….String.

A string of commands to be sent. Multiple commands can be sent, but each command should be separated with a valid 6K command delimiter (colon, carriage return, or line feed). The command string should be limited to 256 characters or less. Larger command strings may cause an overflow in the 6K's command buffer.

Ntimeout……….Long Integer.

Specifies the time period to wait for acknowledgement from the 6K. The time is specified in seconds. A value of zero specifies an infinite period.

Return Type: Long Integer.

This method returns a positive value corresponding to the number of bytes sent, or an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: The WriteCmdBlocking method can be used as an alternative to the WriteCmd method. The discussion below outlines the benefits of the WriteCmdBlocking method, as compared to the WriteCmd method.

The WriteCmd method sends commands to the 6K by placing commands into the TCP/IP send buffer. The WriteCmd method then returns when the last 6K command has been placed into the TCP/IP buffer. If the 6K is busy performing motion or processing commands from the buffer, the WriteCmd method is very likely to return before the recently written commands are executed within the 6K.

The WriteCmdBlocking method sends commands to the 6K by placing commands into the TCP/IP send buffer, just like the WriteCmd method. However, WriteCmdBlocking does not return from the method call until the command has been executed within the 6K controller. To prevent a permanently blocked call, a timeout parameter has been added to allow the function to return in the event that the 6K controller does not execute the commands within a specified time.

The WriteCmdBlocking functions by requesting an acknowledgement from the 6K that the commands have been executed. A timeout can occur for several reasons: loss of power to the 6K, the Ethernet cable is disconnected, processing of commands took longer than expected, an error occurred within the applications (such as, a KILL occurred or the 6K program has jumped to the error routine).

Note: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

6K Ethernet Properties

Get CommandCount

Description: Use the Get CommandCount property to ascertain how many 6K commands have been executed (outside of defined programs) since the 6K controller was powered up.

Visual Basic: object.Get CommandCount As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the number of 6K commands that have been executed outside of defined programs, since the 6K controller was powered up.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

This property allows users to track when commands being sent to the controller (via the communications ports) have been executed. The value is reset to zero each time power is cycled on the 6K. The return value is affected by any background commands sent in conjunction with the Connect, GetFile, and SendFile methods.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Counter

Description: The Get Counter property returns the current Time Frame Counter value.

Visual Basic: object.Get Counter As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The values represent the current Time Frame Counter value.
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
The Time Frame Counter is a free-running timer in the controller. The Counter is updated at the System Update Rate (2 milliseconds).
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get ErrorStatus

Description: The Get ErrorStatus property returns the current error status (TER) of task 0 only.

Visual Basic: object.Get ErrorStatus As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The values represent the current error status (TER) of task 0 only.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get FsEnabled

Description: The Get FSEnabled property sets or returns the state of FastStatus polling.

Visual Basic: object.Get FSEnabled As Boolean

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE.

Return Type: Boolean.

Remarks: This is a read-only property.

The table above lists the items in the FastStatus structure. If the FSEnabled property is set to TRUE, then FastStatus information is automatically retrieved from the controller on a continual basis. Be aware that enabling automatic FastStatus polling provides fresh data from the controller on a continual basis, but this will impair the controller's processing time and create a high volume of traffic over the Ethernet network interface. If you intend to enable automatic FastStatus polling, be sure to first set the FSUpdateRate property accordingly. If the FSEnabled property is set to FALSE, automatic FastStatus polling is turned off (but the FastStatus structure will retain the values from the last update).

Get FsUpdateRate

Description: The Get FSUpdateRate property is used to set the millisecond interval on which the controller automatically updates its FastStatus information.

Visual Basic: object.Get FSUpdateRate As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE.

Return Type: Long Integer.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

This property should be set before the FSEnabled property is set to TRUE. Setting a larger value for this property means that information will be update less frequently, thereby consuming less of the controller's processing resources. A small value will provide for more frequent updates, but consume more processing time. Valid values for this property are from 10 to 65536.

VB Users: COM6SRVR interprets the FSUpdateRate as an unsigned 16-bit integer value. Visual Basic does not support use of unsigned data types. Therefore, you have to pass a signed 16-bit integer and allow the COM6SRVR to interpret it as unsigned. Thus, to allow slower update intervals than 32767 ms, a VB programmer would pass a negative value (see examples below):

Value Passed	Result
-1	65535 ms/update
-32768	+32768 ms/update
-30000	+35536 ms/update
-25536	+40000 ms/update
+32767	+32767 ms/update
+10	+10 ms/update

Get IPAddress

Description:	The Get IPAddress property returns the controller's IP Address (NTADDR).
Visual Basic:	object.Get IPAddress As Long
Version:	1.0.0 6K/Gem6K Ethernet Only
Parameter:	NONE
Return Type:	Long Integer. The value represents the controller's IP Address (NTADDR). This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."
Remarks:	This is a read-only property.

Get IsConnected

Description: The Get IsConnected returns the connection state.
Visual Basic: object. Get IsConnected() As Boolean
Version: 1.0.0
Parameter: NONE
Return Type: Boolean.
Remarks: This is a read-only property.

Get IsWatchDogTimedOut

Description: The Get IsWatchdogTimedOut method interrogates the current status of the Ethernet Watchdog. The Ethernet Watchdog is a handshake established between the Com6srvr and the 6K to monitor that the Ethernet connection is still active and "connected".

Visual Basic: object.Get IsWatchdogTimedOut As Boolean

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Boolean.

A TRUE indicates that the Ethernet connection has been lost (possible causes: the 6K was reset, or the Ethernet connection was broken). The property is cleared when a new Ethernet connection is established.

Remarks: This is a read-only property.

For further information, refer to the SetWatchdog method.

Get LastCom6OcxError

Description: The Get LastCom6OcxError gets the Com6Ocx Error code
Visual Basic: object.Get LastCom6OcxError() As Com6OcxError
Version: 1.0.0
Parameter: NONE
Return Type: Com6OcxError (see table below)
Remarks: This is a read-only property.
The property returns a number (0 through -20) that refers to the Com6OcxError code.
The Com6srvrNet.ocx performs additional range checking because some parameters pass to the Com6srvrNet.Ocx as Long data and are then converted to Integer or Short data before being passed to Com6srvr. If parameters are determined to be out of range, the function stores an error code within the OCX. To display the last error, use the command LastCom6OcxError.
Note: By calling the last error, you also clear the internal error code in the OCX.

Com6srvrNet.ocx Error Codes

Error Code	Description
0	No errors are present. This error clears on successful connect or by calling LastCom6OcxError
-1	Specified axis out of range. Used Axis one.
-2	Specified brick out of range. Used Brick zero (on board)
-3	Specified analog channel out of range. Used Channel one
-4	Specified variable number out of range. Used VarI/B1
-5	Specified alarm bit out of range. Used zero.
-6	Specified TestBit in the IsBitSet function is out of range. Returns FALSE.
-7	Unable to perform operation because not connected.
-8	Unable to create Com6srvr object.
-9	Connect attempted failed.
-10	Called Property/Method while not connected.
-11	Invalid Internet Protocol Address or Invalid Parameter.
-12	Specified delay is out of range.
-13	Specified timeout is out of range.
-14	WriteBlocking timed out.
-15	SendFileBlocking cancelled out.
	The Following error codes were added in version 1.3.0
-16	Invalid Device for Function or Property
-17	Already connect to device
-18	Not Connected to fast status
-19	Ethernet Function call only
-20	RS232 Function call only

Get Limits

Description: The Get Limits property returns the current limit status (`TLIM`).

Visual Basic: object.Get Limits As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the current limit status (`TLIM`).
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get SystemStatus

Description: The Get SystemStatus property returns the system status (TSS) for task 0 only.

Visual Basic: object.Get SystemStatus As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the system status (TSS) for task 0 only.
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Timer

Description: The Get Timer property returns the current Timer value (TTIM) for task 0 only.

Visual Basic: object.Get Timer As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the current Timer value (`TTIM`) for task 0 only.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Get Triggers

Description: The Get Triggers property returns the Trigger Interrupt Status (TTRIG).

Visual Basic: object.Get Triggers As Long

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the current state of the Trigger Interrupt Status (`TTRIG`).

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get UpdateID

Remarks: This property is no longer supported.

Get UserStatus

Description: The Get UserStatus property returns the current state of the user status register (TUS).

Visual Basic: object.Get UserStatus As Long

Parameter: NONE

Return Type: Long Integer.

The value represents the current state of the user status register (TUS).

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Visible

Description: Use Get Visible to get the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.

Visual Basic: object. Get Visible() As Boolean

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Boolean

Remarks: This is a read-only property.
When the Boolean value is 1, the visible property tests as TRUE.

Let FsEnabled

Description: The FSEnabled property sets or returns the state of FastStatus polling.

Visual Basic: object.Let FSEnabled (By Val *bNew Value* As Boolean)

Version: 1.0.0 6K/Gem6K Ethernet Only

Parameter: *bNew Value* ……….As Boolean.

Return Type: NONE.

Remarks: This is a write-only property.

The table above lists the items in the FastStatus structure. If the FSEnabled property is set to TRUE, then FastStatus information is automatically retrieved from the controller on a continual basis. Be aware that enabling automatic FastStatus polling provides fresh data from the controller on a continual basis, but this will impair the controller's processing time and create a high volume of traffic over the Ethernet network interface. If you intend to enable automatic FastStatus polling, be sure to first set the FSUpdateRate property accordingly. If the FSEnabled property is set to FALSE, automatic FastStatus polling is turned off (but the FastStatus structure will retain the values from the last update).

Let FsUpdateRate

Description: The FSUpdateRate property is used to set the millisecond interval on which the controller automatically updates its FastStatus information.

Visual Basic: object.Let FSUpdateRate (ByVal *lNewValue* As Long)

Parameter: lNewValue……….Long Integer.

Return Type: NONE

Remarks: This is a write-only property.

This property should be set before the FSEnabled property is set to TRUE. Setting a larger value for this property means that information will be update less frequently, thereby consuming less of the controller's processing resources. A small value will provide for more frequent updates, but consume more processing time. Valid values for this property are from 10 to 65536.

VB Users: COM6SRVR interprets the FSUpdateRate as an unsigned 16-bit integer value. Visual Basic does not support use of unsigned data types. Therefore, you have to pass a signed 16-bit integer and allow the COM6SRVR to interpret it as unsigned. Thus, to allow slower update intervals than 32767 ms, a VB programmer would pass a negative value (see examples below):

Value Passed	Result
-1	65535 ms/update
-32768	+32768 ms/update
-30000	+35536 ms/update
-25536	+40000 ms/update
+32767	+32767 ms/update
+10	+10 ms/update

Let Visible

Description: Use Let Visible to set the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.
Visual Basic: object. Let Visible(ByVal bNewValue As Boolean)
Version: 1.0.0
Parameter: bNewValue………Boolean
Return Type: NONE
Remarks: This is a write-only property.

Sub Disconnect

Description: The Disconnect method closes a connection to a 6K controller

Visual Basic: object.Disconnect()

Version: 1.0.0

Parameter: NONE

Return Type: NONE

Remarks: As an out-of-process server, the 6K Communications Server does not shutdown until all client applications have disconnected from the server. In many cases, a proper disconnect does not take place if an unhandled error occurs in the client application and the program exits abnormally. This means that care must be exercised on the part of the client program to disconnect from the server on such occasions or when its services are no longer needed.

Sub Flush

Description: The Flush method removes all characters from the client's receive buffer. This method allows the programmer to clear the receive buffer prior to making a read.

Visual Basic: object.Flush()

Version: 1.0.0

Parameter: NONE

Return Type: NONE

Remarks: Use with caution. This method allows the programmer to clear the receive buffer, such that a subsequent Read call can yield a clean response. However, data arriving in the receive buffer is asynchronous to the application program and a thorough understanding of how the application program is structured is necessary to use this method correctly (for example, it would not be beneficial to Flush the buffer if only a partial response has been received).

Gem6K Ethernet Functions

AlarmStatus

Description: The AlarmStatus property returns the state of the controller's alarm status.

Visual Basic: object.**AlarmStatus**.(*bit* As Long) As Long

Version: **1.3.0** 6K/Gem6K Ethernet Only

Parameter: *bit*......... Long Integer.
Specifies the status bit of the alarm status to return. It can be a number between 0 and 32. Values between 1-32 represent the alarm bits as described in the table below (refer also to the `INTHW` command). Specifying a bit value of 0 returns the entire 32 bit alarms status as a long value; otherwise a value of 1 or 0 is returned to indicate the state of any single bit. When any single bit status is retrieved using the AlarmStatus property, that bit status is automatically cleared by the Communications Server. If a bit value of 0 is used then all alarm status bits are cleared.

Return Type: Long Integer.
This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: When the **6K/Gem6K** sends an alarm packet to the COM6SRVR, the FastStatus structure is automatically updated, regardless of state of FSEnabled.

Bit #	Function
1	Software (forced) Alarm #1
2	Software (forced) Alarm #2
3	Software (forced) Alarm #3
4	Software (forced) Alarm #4
5	Software (forced) Alarm #5
6	Software (forced) Alarm #6
7	Software (forced) Alarm #7
8	Software (forced) Alarm #8
9	Software (forced) Alarm #9
10	Software (forced) Alarm #10
11	Software (forced) Alarm #11
12	Software (forced) Alarm #12
13	Command Buffer Full
14	ENABLE input Activated
15	Program Complete
16	Drive Fault on any Axis
17	Reserved
18	Reserved
19	Limit Hit - hard or soft limit, on any axis

20	Stall Detected (stepper) or Position Error (servo) on any axis
21	Timer (TIMINT)
22	Ethernet fail (RESET or ER.22 occurred) (also invokes an error dialog)
23	Input - any of the inputs defined by INFNCi-I or LIMFNCi-I
24	Command Error
25	Motion Complete on Axis 1
26	Motion Complete on Axis 2 (Gem6K – Reserved)
27	Motion Complete on Axis 3 (Gem6K – Reserved)
28	Motion Complete on Axis 4 (Gem6K – Reserved)
29	Motion Complete on Axis 5 (Gem6K – Reserved)
30	Motion Complete on Axis 6 (Gem6K – Reserved)
31	Motion Complete on Axis 7 (Gem6K – Reserved)
32	Motion Complete on Axis 8 (Gem6K – Reserved)

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

AnalogInput

Description: The AnalogInput property returns the value (in counts) of the specified analog input.

Visual Basic: object.AnalogInput(*channel* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *channel*......... Long Integer.
Specifies the analog input channel (channel 1 or 2) value to return. This property uses only the first two analog inputs detected on an I/O brick connected to the 6K/Gem6K, regardless of the `ANIEN` (analog input enable) setting.

Return Type: Long Integer.
The method returns the specified analog input value in counts.

Remarks: Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

AxisStatus

Description: Use the AxisStatus property to retrieve the current axis status for the specified axis. This is the same as the legacy AxisStatus but if there are no Parameters supplied then the first axis is the default.

Visual Basic: object.AxisStatus(Optional *axis* As Long = 1) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *axis*......... Long Integer.
Specifies the axis about which the status pertains. The range for this value is 1-8. For Gem6K there is no need to supply an Axis Number.

Return Type: Long Integer.
The long integer value represents the current axis status for the specified axis. Refer to the TAS command description for a list of the status elements.
This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

DeviceConnect

Description: The Connect method opens a connection to a 6K,Gem6K, or Gemini controller over Ethernet or RS232

Visual Basic: object.DeviceConnect(Optional *iConnectInterface* As Long = 1, Optional *strNetAddress* As String = "1", Optional *bQuiet* As Boolean = False, Optional *lTimeOut* As Long = 15) As Long

Version: 1.3.0

Parameter: *iConnectInterface*………Long

Represents the Device you would like to connect to

6K = 1

GEMINI = 2

GEM6K = 3

strNetAddress………String.

Represents the RS232 port number or the IP Address

bQuiet………Boolean

Specifies whether the connection dialog will be shown. True will hide the connection dialog, false shows the connection dialog.

lTimeOut………Long

A constant that specifies a timeout period in mS for the Ethernet connection attempt. The range for lTimeout is (0 - 60000).

Return Type: Long Integer.

If the connection is successfully opened, the method returns a positive value representing the number of connected clients. If the connection is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: This is the new connect method. You can either connect over RS232 or Ethernet. By Default (i.e. DeviceConnect()) will connect to a 6K over RS232 on port 1.The Server can handle up to two RS232 connections. The RS232 server assumes 9600 Baud operation. The last two parameters are not needed for Serial Communication. The Server can handle a virtually unlimited number of Ethernet connections to different IP addresses (limited by computer resources). The 6K takes up to one minute for an Ethernet connection to truly expire and be available for a new connection.

For a RS232 Connection for a 6K ONLY

Background Commands: After a successful connection is made, a "PORT0:" commands are sent to the controller:

For an Ethernet Connection

Background Commands: After a successful connection is made, the following commands are sent to the controller:

```
!PORT0
!ERRLVL4
!EOT13,0,0
!EOL13,10,0
```

ECHO mode is initially disabled (`ECHO0`) by the 6K/Gem6K during Ethernet communications.

Examples: DeviceConnect(1, 2) ;Connects to a 6K over COM 2

```
DeviceConnect(3, "100.100.100.100", True)        ;Connects
to a Gem6K over
                          ;Ethernet and does this quietly
```

EncoderPos

Description: The EncoderPos property returns the current encoder position (TPE) in counts for the specified axis. This is the same as the legacy EncoderPos but if there are no Parameters supplied then the first axis is the default.

Visual Basic: object.EncoderPos(Optional *axis* As Long = 1) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *axis*………Long Integer.
Specifies the axis number of the encoder. The range for this value is 1-8. For Gem6K there is no need to supply an Axis Number.

Return Type: Long Integer.
The value represents the current encoder position (TPE) in counts for the specified axis.

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Inputs

Description: Use the Inputs property to check the current state of the inputs (TIN) on a specific brick.

Visual Basic: object.Inputs(*brick* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *brick*.........Long Integer.
Represents the number of the brick where the inputs reside. Range is 0-3. Brick 0 represents the onboard inputs. Bricks 1-3 represent expansion I/O bricks 1-3.

Return Type: Long Integer.
The value represents the current state of the inputs (`TIN`) for the specified brick.
This function could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

IsBitSet

Description: Use the IsBitSet to test the validity of a bit.

Visual Basic: object.IsBitSet(*TestValue* As Long, *TestBit* As Long) As Boolean.

Version: 1.3.0

Parameter: *TestValue*.........Long Integer
Represents the value you are testing. If the bit you are testing is a 1, the Boolean returns TRUE.
TestBit.........Long Integer
Represents the bit to test.

Return Type: Boolean

Remarks: The IsBitSet uses the 6K language convention, where the bit pattern is represented as 1 to 32, from left to right when represented as a string. To get the string value, use the LongTo 6Kbinary.

Longto6Kbinary

Description: Use the LongTo6KBinary function to convert a Long Integer to binary.

Visual Basic: object.LongTo6KBinary(*InValue* As Long) As String

Version: 1.3.0

Parameter: *InValue*.........As Long

Return Type: String

Remarks: Using a Long Integer is more efficient for testing bits, and manipulating and storing data. However, it is more convenient to show the data using the 6K language convention. Programmers familiar with the 6K language might want to see, for instance, the Axis Status represented as a string of 32 bits (1s and 0s). For example, when using the AxisStatus() method, the bit pattern 0000_0000_0000_1000_0000_0001_0000_0000 returns the value 8392704.

MotorPos

Description: The MotorPos property returns the current commanded position (TPC) for the specified axis.

Visual Basic: object.MotorPos(Optional *axis* As Long = 1) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *axis*......... Long Integer.
Specifies the axis number (range is 1-8).

Return Type: Long Integer.
The value represents the current commanded position (TPC) in counts for the specified axis.

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.
Defaults to axis one.

MotorVel

Description: The MotorVel property returns the current commanded motor velocity (TVEL) for the specified axis.

Visual Basic: object.MotorVel(Optional *axis* As Long =1) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *axis*......... Long Integer.
Specifies the axis number (range is 1-8).

Return Type: Long Integer.
The value represents the current commanded velocity (TVEL) in counts for the specified axis.

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.
Defaults to axis one.

Outputs

Description: The Outputs property returns the state of the outputs (TOUT) on the specified brick.

Visual Basic: object.Outputs(*brick* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *brick*......... Long Integer.
Represents the number of the brick where the outputs reside. Range is 0-3. Brick 0 represents the onboard outputs. Bricks 1-3 represent expansion I/O bricks 1-3.

Return Type: Long Integer.
The value represents the state of the outputs (`TOUT`) on the specified brick. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Ping

Description: Pings a Gem6K or 6K at the specified IP address.

Visual Basic: object.Ping(Optional *strIPAddr* As String = "", Optional *lTimeOut* As Long = 15000) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strIPAddr* ………String

Represents the IP address of the Device if NOT connected already.

LtimeOut………Long

Timeout period in mS for Ping. The range for lTimeout is (0 - 30000)

Return Type: Long.

The method returns a positive value if the operation is successful; otherwise, it returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: Allows you to ping a device with out being connected. If no IP address is given then the Current IP Address given during the Connect method is used.

ReadResponse

Description: The Read method retrieves command responses from the controller.

Visual Basic: object.ReadResponse() As String

Version: 1.3.0

Parameter: NONE

Return Type: String.

The read method does not wait for incoming responses from the controller. It returns immediately with a string containing the controller's response at the time of the request. If no response is available, this method returns an empty string. The Read method response is limited to 256 characters. If the response is longer than 256 characters, the excess characters will remain in the Com6srvr buffer. Multiple reads are necessary for long responses.

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

RequestFastStatusUpdate

Description: The RequestFastStatusUpdate method allows the Com6srvr to request a fast status update as needed, without having to enable the fast status "Streaming Mode" (FSEnabled) or set an update interval (FSUpdateRate).

Visual Basic: object.RequestFastStatusUpdate As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
If the RequestFastStatusUpdate call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: This method is one of two "On Demand" fast status update options. The other option is for the 6K to execute the `NTSFS` command. Using an On Demand update technique is more efficient for interactive PC applications than the Streaming Mode, and reduces network traffic.

SendBinaryVariable

Description: The SendBinaryVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendBinaryVariable(*varnum* As Long, *value* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer.
Specifies the one variable to be sent. Only one bit can be set in the Varnum.
value.........Variant.
Specifies the value of the variable to be sent. The Value specifies the actual variable being sent.

Return Type: Long Integer.
If the SendBinaryVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the Varnum or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: For more information about sending variable packets, see *Communications Server (COM6SRVR) - Programming Notes* in Help. The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

SendFile

Description: The SendFile method is used to download program files to the controller. (Refer also to SendFileBlocking and SetSendFileDelay.)

Visual Basic: object.SendFile(Optional *filename* As String = "") As Long

Version: 1.3.0

Parameter: *filename*.........String.
Represents the name of the program file (containing 6K programs/code) to be downloaded. If the filename is an empty string, then the user will be prompted for the filename.

Return Type: Long Integer.
If successful, the method returns a positive value; if unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFile method strips comments from the downloaded 6K code. That is, all text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.
Background Commands: At the beginning of a file download operation, these commands are sent to the controller:

```
  !PORT0
  !ERRLVL0
```

After the download process is completed, these commands are sent to the controller:

```
  !PORT0
  !ERRLVL4
  !EOT13,0,0
  !EOL13,10,0
```

Note: If the download process is canceled, an "END" command is sent to the controller.

SendFileBlocking

Description: This method, like the SendFile method, is used to download program files to the controller. It differs from SendFile in that it blocks the return of the method call until the 6K acknowledges that the file has been downloaded. A dialog informs the user to wait for the 6K to acknowledge.

Visual Basic: object.SendFileBlocking(Optional *filename* As String = "") As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *filename*.........String.
Represents the name of the program file (containing 6K programs/code) to be downloaded. If the filename is an empty string, then the user will be prompted for the filename.

Return Type: Long Integer.
The method returns a positive value if the operation is successful; otherwise, it returns an error code. The dialog has a Cancel button (a software specified Timeout is not provided). If the user clicks the Cancel button, the method returns error code -15 SendFileBlocking cancelled out.
Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileBlocking method strips comments from the downloaded 6K code. That is, all text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.
Background Commands: (same as SendFile)
Note: If the download process is canceled, an "END" command is sent to the controller and the method returns error code -15 SendFileBlocking cancelled out.

SendFileQuiet

Description: The SendFileQuiet method is used to download program files to the controller while suppressing the download status dialog message.

Visual Basic: object.SendFileQuiet(Optional *filename* As String = "") As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strFileName*.........String
Represents the name of the program file (containing 6K or Gem6K program/code) to be downloaded. As default the filename is an empty string so the user will be prompted for the filename.

Return Type: Long Integer.
The method returns a positive value if the operation is successful; otherwise, it returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileQuiet method strips comments from the download code. That is, all text between the comment delimiter (semi - colon) and the command delimiter (carriage return or line feed) is removed.
Note: The SendFileQuiet method should be called when motion is *not* in progress and programs are *not* running.
Background Commands: At the beginning of a file download operation, these commands are sent to the controller:
!PORT0
!ECHO0
!ERRLVL0
!EOT1,0,0
!EOL10,0,0
!TDIR
After the download process is completed, these commands are sent to the controller:
!PORT0
!EOT13,0,0
!EOL13,10,0
!ERRLVL4
!ECHO1
Note: If the download process is canceled, an "END" command is sent to the controller.

SendFileQuietBlocking

Description: This method, like the SendFileQuiet method, is used to download program files to the controller while suppressing the download dialog. It differs from SendFileQuiet in that it blocks the return of the method call until the Gem6K acknowledges that the file has been downloaded. A dialog informs the user to wait for the Gem6K to acknowledge. The dialog has a CANCEL button (a software specified Timeout is no provided). If the user clicks the CANCEL button, the method returns the error -15.

Visual Basic: object.SendFileBlocking(Optional *filename* As String = "") As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strFileName*.............String

Represents the name of the program file (containing 6K/Gem6K programs/code) to be downloaded. By default the filename is an empty string; the user will be prompted for the filename unless one is supplied.

Return Type: Long Integer.

The method returns a positive value if the operation is successful; otherwise, it returns an error code (error code -15 is returned when the user clicks the CANCEL button). Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileQuietBlocking method strips comments from the downloads 6K/Gem6K code. That is, al text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.

Background Commands: (same as SendFile)

Note: If the download process is canceled, an "END" command is sent to the controller and the error code (-15) is returned.

SendIntegerVariable

Description: The SendIntegerVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendIntegerVariable(*varnum* As Long, *value* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer.
Specifies the one variable to be sent. Only one bit can be set in the Varnum.
value.........Variant.
Specifies the value of the variable to be sent. The Value specifies the actual variable being sent.

Return Type: Long Integer.
If the SendIntegerVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the Varnum or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: For more information about sending variable packets, see *Communications Server (COM6SRVR) - Programming Notes* in Help. The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

SendRealVariable

Description: The SendRealVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendRealVariable(*varnum* As Long, *value* As Double) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *varnum*..........Long Integer.
Specifies the one variable to be sent. Only one bit can be set in the Varnum.
value..........Variant.
Specifies the value of the variable to be sent. The Varnum specifies the actual variable being sent.

Return Type: Long Integer.
If the SendRealVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the Varnum or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors. Please see "Possible Error Codes from Com6srvr."

Remarks: For more information about sending variable packets, see *Communications Server (COM6SRVR) - Programming Notes* in Help. The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

SendVariable

Description: The SendVariable method sends one variable from the variable packet to the 6K controller.

Visual Basic: object.SendVariable(VariableMask As Long, NewValue As Variant) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *VariableMask*.........Long Integer.

Specifies the one variable to be sent. Constants are defined for the mask bits (mask bits for Visual Basic are provided below). Only one bit can be set in the VariableMask.

NewValue..........Variant.

Specifies the value of the variable to be sent. The actual variable being sent is specified by the VariableMask. Because the SendVariable Method can be used to send integer, real or binary variables, the data type can either be a long integer or a double floating point value. Using a Variant parameter allows the flexibility of sending any integer type, while allowing the Com6srvr to cast the Variant into the appropriate data type. Please see "Possible Error Codes from Com6srvr."

Return Type: Long Integer.

If the SendVariable call is successful, the method returns the number of bytes sent. If the call is unsuccessful, the method returns an error code. Error codes are returned if more than one bit is set in the VariableMask or if the Variant data type is incompatible. Error codes are also returned if there are Ethernet communications errors.

Remarks: The data range of real variables in the 6K and the number of significant figures available in a double data type in the PC programming language may cause some rounding errors. The 6K can store data with greater significance, but with a smaller range of values (refer to the VAR command).

Variable Packet Mask Bits for Visual Basic

```
Public Const VARI1 As Long = 1
Public Const VARI2 As Long = 2
Public Const VARI3 As Long = 4
Public Const VARI4 As Long = 8
Public Const VARI5 As Long = 16
Public Const VARI6 As Long = 32
Public Const VARI7 As Long = 64
Public Const VARI8 As Long = 128
Public Const VARI9 As Long = 256
Public Const VARI10 As Long = 512
Public Const VARI11 As Long = 1024
Public Const VARI12 As Long = 2048
Public Const VAR1 As Long = 4096
Public Const VAR2 As Long = 8192
Public Const VAR3 As Long = 16384
Public Const VAR4 As Long = 32768
Public Const VAR5 As Long = 65536
Public Const VAR6 As Long = 131072
Public Const VAR7 As Long = 262144
Public Const VAR8 As Long = 524288
Public Const VAR9 As Long = 1048576
Public Const VAR10 As Long = 2097152
Public Const VAR11 As Long = 4194304
Public Const VAR12 As Long = 8388608
Public Const VARB1 As Long = 16777216
Public Const VARB2 As Long = 33554432
Public Const VARB3 As Long = 67108864
Public Const VARB4 As Long = 134217728
Public Const VARB5 As Long = 268435456
Public Const VARB6 As Long = 536870912
Public Const VARB7 As Long = 1073741824
Public Const VARB8 As Long = &H80000000
```

SetSendFileDelay

Description: SetSendFileDelay allows you to specify the delay for each character when making a call to the SendFile method. (See Remarks below for detail)

Visual Basic: object.SetSendFileDelay(*delay* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *delay*.........Long Integer.
The parameter specifies the delay for each character transmitted. Valid range is 0-100 (milliseconds).
0 = no delay.

Return Type: Long Integer.
If the specified delay is valid, the method returns zero; if the specified delay is out of range, the method returns error code -13 Specified timeout is out of range. Please see "Possible Error Codes from Com6srvr."

Remarks: When making a call to the SendFile Ethernet method, a 2-ms per character delay is inserted to allow the commands to be transmitted and processed through the TCP/IP stack and the 6K internal buffers. In some cases the delay is not necessary, because the TCP/IP stack takes care of flow-control. In other cases, it might be desirable to allow a longer delay, such as when sending data over a very busy network. This method provides a means to control the delay, allowing a delay of 0-100 ms per character.

SetWatchdog

Description: The SetWatchdog method enables Ethernet watchdog hand-shaking between the Com6srvr and the 6K Controller.

Visual Basic: object.SetWatchdog(*Timeout* As Long, *Ticker* as Long) as Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *Timeout*.........Timeout period in seconds (see guidelines below)

Ticker.........Number of "heartbeat" packets to send during the timeout period

Return Type: Long Integer.

Returns the number of bytes sent to the watchdog if successful, or a negative error code (usually error code -11, which indicates that an invalid configuration was specified) if not successful. Please see "Possible Error Codes from Com6srvr."

Remarks: The Ethernet watchdog allows the Com6srvr and 6K Controller to gracefully recover when communication between the 6K and Com6srvr is lost. Such situations might arise from the loss of power to the 6K or to the PC while an Ethernet connection was active. Because the Com6srvr and the 6K establish a "virtual" software based connection, both parties agree to be "connected". By enabling the Watchdog, a heartbeat packet is sent periodically by the Com6srvr. The 6K detects the heartbeat and echoes it back to the Com6srvr. When the watchdog is enabled, the Com6srvr sends the heartbeat and expects an echo. Similarly, the 6K controller is expecting the heartbeat from the Com6srvr. If the Com6srvr does not detect the echoed heartbeat (within the constraints set by the Timeout and Ticker parameters), the watchdog is considered timed out. If the 6K does not receive the heartbeat (within the same Timeout and Ticker constraints), the 6K considers the watchdog timed out.

Loss or delay of a single echoed heartbeat could happen quite frequently on a busy network connection. Therefore, we provide a method whereby a number of re-tries are attempted over a specific timeout period. If all re-tries fail within the timeout period, then the watchdog is considered to have timed out. This functionality is provided by the Timeout and Ticker parameters. The constraints for these parameters are as follows:

- To enable the watchdog, set Timeout > 0 > Ticker.
- To disable the watchdog, set Timeout = 0 and set Ticker = 0.
- The Timeout/Ticker ratio must be ≤ 65.

RECOMMENDATION: Set Timeout = 100 and Ticker = 5, which provides a heartbeat once every twenty seconds (100 seconds / 5 tries = 20 seconds/attempt). If none of the 5 heartbeats are acknowledged in 100 seconds, the watchdog times out.

WHEN A WATCHDOG TIMEOUT OCCURS:

- In the 6K: When the 6K detects a watchdog timeout, it attempts to send an alarm packet to the Com6srvr (AlarmStatus bit #22). It then closes the Ethernet connection and reports "disconnected" in the TNT report. If the user has enabled error-checking bit #22 (ERROR.22-1), the 6K will execute a GOSUB branch to the ERRORP program. Within the ERRORP program, the watchdog timeout can be cleared by disabling ERROR bit #22 (ERROR.22-0).

- In the Com6srvr: When the COM6SRVR detects a watchdog timeout, the IsWatchdogTimedOut returns TRUE. (If the COM6SRVR receives the alarm packet from the 6K, it will also display an alert dialog to the user.) A client application can poll the IsWatchdogTimedOut. When a timeout is detected by the Com6srvr, the Client application should "disconnect" the Com6srvr. You can use Sub Disconnect. After the Com6srvr has been disconnected, creating a new Com6srvr object and "connecting" Ethernet will clear the watchdog timeouts. All client applications for that particular 6K Ethernet connection should be disconnected.

Var

Description:	The Var property returns the value of the specified real variable (VAR). Variables VAR1 through VAR12 may be reported.
Visual Basic:	object.Var(*lNum* As Long) As Long
Version:	1.3.0 6K/Gem6K Ethernet Only
Parameter:	*lNum*.........Long Integer. Represents number of the real variable (VAR*lNum*). Range is 1-12.
Return Type:	Double The value represents the value of the specified real variable (VAR). The initial value is zero until an Extended Fast Status packet arrives.
Remarks:	This is a read only property Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

VarB

Description: The VarB property returns the value of the specified binary variable (VARB).

Visual Basic: object.VarB(*varnum* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer. Represents number of the binary variable (VARB*varnum*). Range is 1-10.

Return Type: Long Integer.

The value represents the value of the specified binary variable (VARB).

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

VarI

Description: The VarI property returns the value of the specified integer variable (VARI).

Visual Basic: object.VarI(*varnum* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *varnum*.........Long Integer. Represents number of the binary variable (VARI*varnum*). Range is 1-10.

Return Type: Long Integer.
The value represents the value of the specified integer variable (`VARI`).

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

WriteCmd

Description: The WriteCmd method is used to send commands to the controller.

Visual Basic: object.WriteCmd(*sCmd* As String) As Long

Version: 1.3.0

Parameter: *sCmd*.........String.
A string of commands to be sent. Multiple commands can be sent, but each command should be separated with a valid 6K command delimiter (colon, carriage return, or line feed). The command string should be limited to 256 characters or less. Larger command strings may cause an overflow in the 6K's command buffer.

Return Type: Long Integer.
This method returns a positive value corresponding to the number of bytes sent, or an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

WriteCmdBlocking

Description: The WriteCmdBlocking method, an alternative to the WriteCmd method, is used to send commands to the controller. The primary difference from the WriteCmd method is that WriteCmdBlocking does not return from the method call until the commands have been executed in the controller within a specified time.

Visual Basic: object.WriteCmdBlocking(*sCmd* As String, *nTimeout* As Long) As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *sCmd*………String.

A string of commands to be sent. Multiple commands can be sent, but each command should be separated with a valid 6K command delimiter (colon, carriage return, or line feed). The command string should be limited to 256 characters or less. Larger command strings may cause an overflow in the 6K's command buffer.

Ntimeout………Long Integer.

Specifies the time period to wait for acknowledgement from the 6K. The time is specified in seconds. A value of zero specifies an infinite period.

Return Type: Long Integer.

This method returns a positive value corresponding to the number of bytes sent, or an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: The WriteCmdBlocking method can be used as an alternative to the WriteCmd method. The discussion below outlines the benefits of the WriteCmdBlocking method, as compared to the WriteCmd method.

The WriteCmd method sends commands to the 6K by placing commands into the TCP/IP send buffer. The WriteCmd method then returns when the last 6K command has been placed into the TCP/IP buffer. If the 6K is busy performing motion or processing commands from the buffer, the WriteCmd method is very likely to return before the recently written commands are executed within the 6K.

The WriteCmdBlocking method sends commands to the 6K by placing commands into the TCP/IP send buffer, just like the WriteCmd method. However, WriteCmdBlocking does not return from the method call until the command has been executed within the 6K controller. To prevent a permanently blocked call, a timeout parameter has been added to allow the function to return in the event that the 6K controller does not execute the commands within a specified time.

The WriteCmdBlocking functions by requesting an acknowledgement from the 6K that the commands have been executed. A timeout can occur for several reasons: loss of power to the 6K, the Ethernet cable is disconnected, processing of commands took longer than expected, an error occurred within the applications (such as, a KILL occurred or the 6K program has jumped to the error routine).

Note: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

Gem6K Ethernet Properties

Get ActualTorque

Description: The ActualTorque property returns the current actual torque (TTRQA).

Visual Basic: object.Get ActualTorque As Long

Version: 1.3.0 Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the current actual torque (TTRQA). A value of -32,768 corresponds to 100% of torque or force in the negative direction based on the DMTSCL command. A value of +32,767 corresponds to 100% of torque or force in the positive direction based on DMTSCL command.

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get ActualVelocity

Description: The ActualVelocity property returns the current actual velocity (TVELA).

Visual Basic: object.Get ActualVelocity As Long

Version: 1.3.0 Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The Value represents the current actual velocity (TVELA) in counts.

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get CommandCount

Description: Use the Get CommandCount property to ascertain how many 6K commands have been executed (outside of defined programs) since the 6K controller was powered up.

Visual Basic: object.Get CommandCount As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the number of 6K commands that have been executed outside of defined programs, since the 6K controller was powered up.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

This property allows users to track when commands being sent to the controller (via the communications ports) have been executed. The value is reset to zero each time power is cycled on the 6K. The return value is affected by any background commands sent in conjunction with the Connect, GetFile, and SendFile methods.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get CommandedTorque

Description: The CommandedTorque property returns the current commanded Torque (TTRQ)

Visual Basic: object.Get CommandedTorque As Long

Version: 1.3.0 Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the current commanded torque (TTRQ). A value of -32,768 corresponds to 100% of torque or force in the negative direction based on the DMTSCL command. A value of +32,767 corresponds to 100% of torque or force in the positive direction based on the DMTSCL command.

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get ConfigurationStatus

Description: Use the Configurations property to retrieve the current configuration status (TCS).

Visual Basic: object.Get ConfigurationStatus As Long

Version: 1.3.0 Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the current configuration status. Refer tot he TCS command description for a list of the status elements.

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Counter

Description: The Get Counter property returns the current Time Frame Counter value.

Visual Basic: object.Get Counter As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The values represent the current Time Frame Counter value.
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
The Time Frame Counter is a free-running timer in the controller. The Counter is updated at the System Update Rate (2 milliseconds).
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get ExtendedAxisStatus

Description: Use the ExtendedAxisStatus property to retrieve the current extended axis status (TASX).

Visual Basic: object.Get ExtendedAxisStatus As Long

Version: 1.3.0 Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The long integer value represents the current extended axis status. Refer tot he TASX command description for a list of the status elements.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get ErrorStatus

Description: The Get ErrorStatus property returns the current error status (TER) of task 0 only.

Visual Basic: object.Get ErrorStatus As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The values represent the current error status (TER) of task 0 only.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get FsEnabled

Description: The Get FSEnabled property sets or returns the state of FastStatus polling.

Visual Basic: object.Get FSEnabled As Boolean

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE.

Return Type: Boolean.

Remarks: This is a read-only property.

The table above lists the items in the FastStatus structure. If the FSEnabled property is set to TRUE, then FastStatus information is automatically retrieved from the controller on a continual basis. Be aware that enabling automatic FastStatus polling provides fresh data from the controller on a continual basis, but this will impair the controller's processing time and create a high volume of traffic over the Ethernet network interface. If you intend to enable automatic FastStatus polling, be sure to first set the FSUpdateRate property accordingly. If the FSEnabled property is set to FALSE, automatic FastStatus polling is turned off (but the FastStatus structure will retain the values from the last update).

Get FsUpdateRate

Description: The Get FSUpdateRate property is used to set the millisecond interval on which the controller automatically updates its FastStatus information.

Visual Basic: object.Get FSUpdateRate As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE.

Return Type: Long Integer.

Remarks: This is a read-only property.

This property should be set before the FSEnabled property is set to TRUE. Setting a larger value for this property means that information will be update less frequently, thereby consuming less of the controller's processing resources. A small value will provide for more frequent updates, but consume more processing time. Valid values for this property are from 10 to 65536.

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

VB Users: COM6SRVR interprets the FSUpdateRate as an unsigned 16-bit integer value. Visual Basic does not support use of unsigned data types. Therefore, you have to pass a signed 16-bit integer and allow the COM6SRVR to interpret it as unsigned. Thus, to allow slower update intervals than 32767 ms, a VB programmer would pass a negative value (see examples below):

Value Passed	**Result**
-1	65535 ms/update
-32768	+32768 ms/update
-30000	+35536 ms/update
-25536	+40000 ms/update
+32767	+32767 ms/update
+10	+10 ms/update

Get IPAddress

Description: The Get IPAddress property returns the controller's IP Address (NTADDR).

Visual Basic: object.Get IPAddress As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the controller's IP Address (`NTADDR`).
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Get IsConnected

Description: The Get IsConnected returns the connection state.
Visual Basic: object. Get IsConnected() As Boolean
Version: 1.3.0
Parameter: NONE
Return Type: Boolean.
Remarks: This is a read-only property.

Get IsWatchDogTimedOut

Description: The Get IsWatchdogTimedOut method interrogates the current status of the Ethernet Watchdog. The Ethernet Watchdog is a handshake established between the Com6srvr and the 6K to monitor that the Ethernet connection is still active and "connected".

Visual Basic: object.Get IsWatchdogTimedOut As Boolean

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Boolean.

A TRUE indicates that the Ethernet connection has been lost (possible causes: the 6K was reset, or the Ethernet connection was broken). The property is cleared when a new Ethernet connection is established.

Remarks: This is a read-only property.

For further information, refer to the SetWatchdog method.

Get LastCom6OcxError

Description:	The Get LastCom6OcxError gets the Com6Ocx Error code
Visual Basic:	object.Get LastCom6OcxError() As Com6OcxError
Version:	1.3.0
Parameter:	NONE
Return Type:	Com6OcxError (see table below)
Remarks:	This is a read-only property.

The property returns a number (0 through -20) that refers to the Com6OcxError code.

The Com6srvrNet.ocx performs additional range checking because some parameters pass to the Com6srvrNet.Ocx as Long data and are then converted to Integer or Short data before being passed to Com6srvr. If parameters are determined to be out of range, the function stores an error code within the OCX. To display the last error, use the command LastCom6OcxError.

Note: By calling the last error, you also clear the internal error code in the OCX.

Com6srvrNet.ocx Error Codes

Error Code	**Description**
0	No errors are present. This error clears on successful connect or by calling LastCom6OcxError
-1	Specified axis out of range. Used Axis one.
-2	Specified brick out of range. Used Brick zero (on board)
-3	Specified analog channel out of range. Used Channel one
-4	Specified variable number out of range. Used VarI/B1
-5	Specified alarm bit out of range. Used zero.
-6	Specified TestBit in the IsBitSet function is out of range. Returns FALSE.
-7	Unable to perform operation because not connected.
-8	Unable to create Com6srvr object.
-9	Connect attempted failed.
-10	Called Property/Method while not connected.
-11	Invalid Internet Protocol Address or Invalid Parameter.
-12	Specified delay is out of range.
-13	Specified timeout is out of range.
-14	WriteBlocking timed out.
-15	SendFileBlocking cancelled out.
-16	Invalid Device for Function or Property
-17	Already connected to device
-18	Not Connected to fast status
-19	Ethernet Function call only
-20	RS232 Function call only

Get Limits

Description: The Get Limits property returns the current limit status (TLIM).

Visual Basic: object.Get Limits As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the current limit status (`TLIM`).
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get SettlingTime

Description: The SettlingTime property returns the time it took for the last move to settle within into the target zone (TSTLT).

Visual Basic: object.Get SettlingTime As Long

Version: 1.3.0 Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the number of milliseconds it took for the last move to settle within the target zone.
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get SystemStatus

Description: The Get SystemStatus property returns the system status (TSS) for task 0 only.

Visual Basic: object.Get SystemStatus As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the system status (TSS) for task 0 only.
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Timer

Description: The Get Timer property returns the current Timer value (TTIM) for task 0 only.

Visual Basic: object.Get Timer As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.
The value represents the current Timer value (`TTIM`) for task 0 only.
This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.
Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Triggers

Description: The Get Triggers property returns the Trigger Interrupt Status (TTRIG).

Visual Basic: object.Get Triggers As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: NONE

Return Type: Long Integer.

The value represents the current state of the Trigger Interrupt Status (`TTRIG`).

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get UserStatus

Description: The Get UserStatus property returns the current state of the user status register (TUS).

Visual Basic: object.Get UserStatus As Long

Parameter: NONE

Return Type: Long Integer.

The value represents the current state of the user status register (TUS).

This Property could return an error code from the COM6SRVR. Please see "Possible Error Codes from Com6srvr."

Remarks: This is a read-only property.

Requires FastStatus to be enabled, or use of RequestFastStatusUpdate or NTSFS command, or generation of an Alarm in the Gem6K.

Get Visible

Description: Use Get Visible to get the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.
Visual Basic: object. Get Visible() As Boolean
Version: 1.3.0
Parameter: NONE
Return Type: Boolean
Remarks: This is a read-only property.
When the Boolean value is 1, the visible property tests as TRUE.

Let FsEnabled

Description: The FSEnabled property sets or returns the state of FastStatus polling.
Visual Basic: object.Let FSEnabled (By Val *bNew Value* As Boolean)
Version: 1.3.0 6K/Gem6K Ethernet Only
Parameter: *bNew Value* ………As Boolean.
Return Type: NONE.
Remarks: This is a write-only property.

The table above lists the items in the FastStatus structure. If the FSEnabled property is set to TRUE, then FastStatus information is automatically retrieved from the controller on a continual basis. Be aware that enabling automatic FastStatus polling provides fresh data from the controller on a continual basis, but this will impair the controller's processing time and create a high volume of traffic over the Ethernet network interface. If you intend to enable automatic FastStatus polling, be sure to first set the FSUpdateRate property accordingly. If the FSEnabled property is set to FALSE, automatic FastStatus polling is turned off (but the FastStatus structure will retain the values from the last update).

Let FsUpdateRate

Description: The FSUpdateRate property is used to set the millisecond interval on which the controller automatically updates its FastStatus information.

Visual Basic: object.Let FSUpdateRate (ByVal *lNewValue* As Long)

Parameter: *lNewValue*………Long Integer.

Return Type: NONE

Remarks: This is a write-only property.

This property should be set before the FSEnabled property is set to TRUE. Setting a larger value for this property means that information will be update less frequently, thereby consuming less of the controller's processing resources. A small value will provide for more frequent updates, but consume more processing time. Valid values for this property are from 10 to 65536.

VB Users: COM6SRVR interprets the FSUpdateRate as an unsigned 16-bit integer value. Visual Basic does not support use of unsigned data types. Therefore, you have to pass a signed 16-bit integer and allow the COM6SRVR to interpret it as unsigned. Thus, to allow slower update intervals than 32767 ms, a VB programmer would pass a negative value (see examples below):

Value Passed	Result
-1	65535 ms/update
-32768	+32768 ms/update
-30000	+35536 ms/update
-25536	+40000 ms/update
+32767	+32767 ms/update
+10	+10 ms/update

Let Visible

Description: Use Let Visible to set the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.
Visual Basic: object. Let Visible(ByVal bNewValue As Boolean)
Version: 1.3.0
Parameter: *bNewValue*………Boolean
Return Type: NONE
Remarks This is a write-only property.

Sub Disconnect

Description: The Disconnect method closes a connection to a 6K controller

Visual Basic: object.Disconnect()

Version: 1.3.0

Parameter: NONE

Return Type: NONE

Remarks: As an out-of-process server, the 6K Communications Server does not shutdown until all client applications have disconnected from the server. In many cases, a proper disconnect does not take place if an unhandled error occurs in the client application and the program exits abnormally. This means that care must be exercised on the part of the client program to disconnect from the server on such occasions or when its services are no longer needed.

Sub Flush

Description: The Flush method removes all characters from the client's receive buffer. This method allows the programmer to clear the receive buffer prior to making a read.

Visual Basic: object.Flush()

Version: 1.3.0

Parameter: NONE

Return Type: NONE

Remarks: Use with caution. This method allows the programmer to clear the receive buffer, such that a subsequent Read call can yield a clean response. However, data arriving in the receive buffer is asynchronous to the application program and a thorough understanding of how the application program is structured is necessary to use this method correctly (for example, it would not be beneficial to Flush the buffer if only a partial response has been received).

6K & Gem6K RS-232 Functions

DeviceConnect

Description: The Connect method opens a connection to a 6K,Gem6K, or Gemini controller over Ethernet or RS232

Visual Basic: object.DeviceConnect(Optional *iConnectInterface* As Long = 1, Optional *strNetAddress* As String = "1", Optional *bQuiet* As Boolean = False, Optional *lTimeOut* As Long = 15) As Long

Version: 1.3.0

Parameter: *iConnectInterface*………Long
Represents the Device you would like to connect to
6K = 1
GEMINI = 2
GEM6K = 3
strNetAddress………String.
Represents the RS232 port number or the IP Address
bQuiet………Boolean
Specifies whether the connection dialog will be shown. True will hide the connection dialog, false shows the connection dialog.
lTimeOut………Long
A constant that specifies a timeout period in mS for the Ethernet connection attempt. The range for lTimeout is (0 - 60000).

Return Type: Long Integer.
If the connection is successfully opened, the method returns a positive value representing the number of connected clients. If the connection is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: This is the new connect method. You can either connect over RS232 or Ethernet. By Default (i.e. DeviceConnect()) will connect to a 6K over RS232 on port 1.The Server can handle up to two RS232 connections. The RS232 server assumes 9600 Baud operation. The last two parameters are not needed for Serial Communication. The Server can handle a virtually unlimited number of Ethernet connections to different IP addresses (limited by computer resources). The 6K takes up to one minute for an Ethernet connection to truly expire and be available for a new connection.
For a RS232 Connection for a 6K ONLY
Background Commands: After a successful connection is made, a "PORT0:" commands are sent to the controller:
For an Ethernet Connection

Background Commands: After a successful connection is made, the following commands are sent to the controller:

```
  !PORT0
  !ERRLVL4
  !EOT13,0,0
  !EOL13,10,0
```

ECHO mode is initially disabled (`ECHO0`) by the 6K/Gem6K during Ethernet communications.

Examples:

```
DeviceConnect(1, 2)      ;Connects to a 6K over COM 2
DeviceConnect(3, "100.100.100.100", True)
;Connects to a Gem6K over Ethernet and does this quietly
```

IsBitSet

Description: Use the IsBitSet to test the validity of a bit.

Visual Basic: object.IsBitSet(*TestValue* As Long, *TestBit* As Long) As Boolean.

Version: 1.3.0

Parameter: *TestValue*.........Long Integer

Represents the value you are testing. If the bit you are testing is a 1, the Boolean returns TRUE.

TestBit.........Long Integer

Represents the bit to test.

Return Type: Boolean

Remarks: The IsBitSet uses the 6K language convention, where the bit pattern is represented as 1 to 32, from left to right when represented as a string. To get the string value, use the LongTo 6Kbinary.

Longto6Kbinary

Description: Use the LongTo6KBinary function to convert a Long Integer to binary.

Visual Basic: object.LongTo6KBinary(*InValue* As Long) As String

Version: 1.3.0

Parameter: *InValue*.........As Long

Return Type: String

Remarks: Using a Long Integer is more efficient for testing bits, and manipulating and storing data. However, it is more convenient to show the data using the 6K language convention. Programmers familiar with the 6K language might want to see, for instance, the Axis Status represented as a string of 32 bits (1s and 0s). For example, when using the AxisStatus() method, the bit pattern 0000_0000_0000_1000_0000_0001_0000_0000 returns the value 8392704.

ReadResponse

Description: The Read method retrieves command responses from the controller.

Visual Basic: object.ReadResponse() As String

Version: 1.3.0

Parameter: NONE

Return Type: String.

The read method does not wait for incoming responses from the controller. It returns immediately with a string containing the controller's response at the time of the request. If no response is available, this method returns an empty string. The Read method response is limited to 256 characters. If the response is longer than 256 characters, the excess characters will remain in the Com6srvr buffer. Multiple reads are necessary for long responses.

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

SendFile

Description: The SendFile method is used to download program files to the controller.

Visual Basic: object.SendFile(Optional *filename* As String = "") As Long

Version: 1.3.0

Parameter: *filename*.........String.
Represents the name of the program file (containing 6K programs/code) to be downloaded. If the filename is an empty string, then the user will be prompted for the filename.

Return Type: Long Integer.
If successful, the method returns a positive value; if unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFile method strips comments from the downloaded 6K code. That is, all text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.
Background Commands: At the beginning of a file download operation, these commands are sent to the controller:

```
  !PORT0
  !ERRLVL0
```

After the download process is completed, these commands are sent to the controller:

```
  !PORT0
  !ERRLVL4
  !EOT13,0,0
  !EOL13,10,0
```

Note: If the download process is canceled, an "END" command is sent to the controller.

SendFileQuiet

Description: The SendFileQuiet method is used to download program files to the controller while suppressing the download status dialog message.

Visual Basic: object.SendFileQuiet(Optional *filename* As String = "") As Long

Version: 1.3.0 6K/Gem6K Ethernet Only

Parameter: *strFileName*.........String

Represents the name of the program file (containing 6K or Gem6K program/code) to be downloaded. As default the filename is an empty string so the user will be prompted for the filename.

Return Type: Long Integer.

The method returns a positive value if the operation is successful; otherwise, it returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFileQuiet method strips comments from the download code. That is, all text between the comment delimiter (semi - colon) and the command delimiter (carriage return or line feed) is removed.

Note: The SendFileQuiet method should be called when motion is *not* in progress and programs are *not* running.

Background Commands: At the beginning of a file download operation, these commands are sent to the controller:

```
!PORT0
!ECHO0
!ERRLVL0
!EOT1,0,0
!EOL10,0,0
!TDIR
```

After the download process is completed, these commands are sent to the controller:

```
!PORT0
!EOT13,0,0
!EOL13,10,0
!ERRLVL4
!ECHO1
```

Note: If the download process is canceled, an "END" command is sent to the controller.

SetBaudRate

Description: The SetBaudRate method sets the baudrate transmission for the Com6srvr RS232 interface.

Visual Basic: object.**SetBaudRate**(*lBaudRate* As Long) As Long

Version: **1.3.0** 6K/Gem6K over RS232 Only

Parameter: *lBaudRate*.........Long Integer
The value representing the baudrate that Com6srvr will transmit at.

Return Type: The returned Long represents the baudrate that was set.

Remarks: In order to communicate with a 6K or Gem6K, you must set the baudrate in the Com6srvr and the baudrate in the controller to the same value. Use the 6K command BAUD to set the controller baudrate before changing the Com6srvr baudrate.

WriteCmd

Description: The WriteCmd method is used to send commands to the controller.

Visual Basic: object.WriteCmd(*sCmd* As String) As Long

Version: 1.3.0

Parameter: *sCmd*.........String.
A string of commands to be sent. Multiple commands can be sent, but each command should be separated with a valid 6K command delimiter (colon, carriage return, or line feed). The command string should be limited to 256 characters or less. Larger command strings may cause an overflow in the 6K's command buffer.

Return Type: Long Integer.
This method returns a positive value corresponding to the number of bytes sent, or an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

6K & Gem6K RS-232 Properties

Get IsConnected

Description:	The Get IsConnected returns the connection state.
Visual Basic:	object. Get IsConnected() As Boolean
Version:	1.3.0
Parameter:	NONE.
Return Type:	Boolean.
Remarks:	This is a read-only property.

Get LastCom6OcxError

Description: The Get LastCom6OcxError gets the Com6Ocx Error code

Visual Basic: object.Get LastCom6OcxError() As Com6OcxError

Version: 1.3.0

Parameter: NONE

Return Type: Com6OcxError (see table below)

Remarks: This is a read-only property.

The property returns a number (0 through -20) that refers to the Com6OcxError code.

The Com6srvrNet.ocx performs additional range checking because some parameters pass to the Com6srvrNet.Ocx as Long data and are then converted to Integer or Short data before being passed to Com6srvr. If parameters are determined to be out of range, the function stores an error code within the OCX. To display the last error, use the command LastCom6OcxError.

Note: By calling the last error, you also clear the internal error code in the OCX.

Com6srvrNet.ocx Error Codes

Error Code	**Description**
0	No errors are present. This error clears on successful connect or by calling LastCom6OcxError
-1	Specified axis out of range. Used Axis one.
-2	Specified brick out of range. Used Brick zero (on board)
-3	Specified analog channel out of range. Used Channel one
-4	Specified variable number out of range. Used VarI/B1
-5	Specified alarm bit out of range. Used zero.
-6	Specified TestBit in the IsBitSet function is out of range. Returns FALSE.
-7	Unable to perform operation because not connected.
-8	Unable to create Com6srvr object.
-9	Connect attempted failed.
-10	Called Property/Method while not connected.
-11	Invalid Internet Protocol Address or Invalid Parameter.
-12	Specified delay is out of range.
-13	Specified timeout is out of range.
-14	WriteBlocking timed out.
-15	SendFileBlocking cancelled out.
-16	Invalid Device for Function or Property
-17	Already connected to device
-18	Not Connected to fast status
-19	Ethernet Function call only
-20	RS232 Function call only

Get Visible

Description: Use Get Visible to get the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.

Visual Basic: object. Get Visible() As Boolean

Version: 1.3.0

Parameter: NONE

Return Type: Boolean

Remarks: This is a read-only property.
When the Boolean value is 1, the visible property tests as TRUE.

Let Visible

Description: Use Let Visible to set the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.
Visual Basic: object. Let Visible(ByVal bNewValue As Boolean)
Version: 1.3.0
Parameter: bNewValue………Boolean
Return Type: NONE
Remarks: This is a write-only property.

Sub Disconnect

Description: The Disconnect method closes a connection to a 6K controller

Visual Basic: object.Disconnect()

Version: 1.3.0

Parameter: NONE

Return Type: NONE

Remarks: As an out-of-process server, the 6K Communications Server does not shutdown until all client applications have disconnected from the server. In many cases, a proper disconnect does not take place if an unhandled error occurs in the client application and the program exits abnormally. This means that care must be exercised on the part of the client program to disconnect from the server on such occasions or when its services are no longer needed.

Sub Flush

Description: The Flush method removes all characters from the client's receive buffer. This method allows the programmer to clear the receive buffer prior to making a read.

Visual Basic: object.Flush()

Version: 1.3.0

Parameter: NONE

Return Type: NONE

Remarks: Use with caution. This method allows the programmer to clear the receive buffer, such that a subsequent Read call can yield a clean response. However, data arriving in the receive buffer is asynchronous to the application program and a thorough understanding of how the application program is structured is necessary to use this method correctly (for example, it would not be beneficial to Flush the buffer if only a partial response has been received).

Gemini RS-232 Functions

DeviceConnect

Description: The Connect method opens a connection to a 6K,Gem6K, or Gemini controller over Ethernet or RS232

Visual Basic: object.DeviceConnect(Optional *iConnectInterface* As Long = 1, Optional *strNetAddress* As String = "1", Optional *bQuiet* As Boolean = False, Optional *lTimeOut* As Long = 15) As Long

Version: 1.3.0

Parameter: *iConnectInterface*………Long
Represents the Device you would like to connect to
6K = 1
GEMINI = 2
GEM6K = 3
strNetAddress………String.
Represents the RS232 port number or the IP Address
bQuiet………Boolean
Specifies whether the connection dialog will be shown. True will hide the connection dialog, false shows the connection dialog.
lTimeOut………Long
A constant that specifies a timeout period in mS for the Ethernet connection attempt. The range for lTimeout is (0 - 60000).

Return Type: Long Integer.
If the connection is successfully opened, the method returns a positive value representing the number of connected clients. If the connection is unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: This is the new connect method. You can either connect over RS232 or Ethernet. By Default (i.e. DeviceConnect()) will connect to a 6K over RS232 on port 1.The Server can handle up to two RS232 connections. The RS232 server assumes 9600 Baud operation. The last two parameters are not needed for Serial Communication. The Server can handle a virtually unlimited number of Ethernet connections to different IP addresses (limited by computer resources). The 6K takes up to one minute for an Ethernet connection to truly expire and be available for a new connection.
For a RS232 Connection for a 6K ONLY
Background Commands: After a successful connection is made, a "PORT0:" commands are sent to the controller:
For an Ethernet Connection

Background Commands: After a successful connection is made, the following commands are sent to the controller:

```
  !PORT0
  !ERRLVL4
  !EOT13,0,0
  !EOL13,10,0
```

ECHO mode is initially disabled (`ECHO0`) by the 6K/Gem6K during Ethernet communications.

Examples: DeviceConnect(1, 2)   ;Connects to a 6K over COM 2
DeviceConnect(3, "100.100.100.100", True)
;Connects to a Gem6K over Ethernet and does this quietly

IsBitSet

Description:	Use the IsBitSet to test the validity of a bit.
Visual Basic:	object.IsBitSet(*TestValue* As Long, *TestBit* As Long) As Boolean.
Version:	1.3.0
Parameter:	*TestValue*.........Long Integer Represents the value you are testing. If the bit you are testing is a 1, the Boolean returns TRUE. *TestBit*.........Long Integer Represents the bit to test.
Return Type:	Boolean
Remarks:	The IsBitSet uses the 6K language convention, where the bit pattern is represented as 1 to 32, from left to right when represented as a string. To get the string value, use the LongTo 6Kbinary.

Longto6Kbinary

Description: Use the LongTo6KBinary function to convert a Long Integer to binary.

Visual Basic: object.LongTo6KBinary(*InValue* As Long) As String

Version: 1.3.0

Parameter: *InValue*.........As Long

Return Type: String

Remarks: Using a Long Integer is more efficient for testing bits, and manipulating and storing data. However, it is more convenient to show the data using the 6K language convention. Programmers familiar with the 6K language might want to see, for instance, the Axis Status represented as a string of 32 bits (1s and 0s). For example, when using the AxisStatus() method, the bit pattern 0000_0000_0000_1000_0000_0001_0000_0000 returns the value 8392704.

ReadResponse

Description: The Read method retrieves command responses from the controller.

Visual Basic: object.ReadResponse() As String

Version: 1.3.0

Parameter: NONE

Return Type: String.

The read method does not wait for incoming responses from the controller. It returns immediately with a string containing the controller's response at the time of the request. If no response is available, this method returns an empty string.

The Read method response is limited to 256 characters. If the response is longer than 256 characters, the excess characters will remain in the Com6srvr buffer. Multiple reads are necessary for long responses.

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

SendFile

Description: The SendFile method is used to download program files to the controller.

Visual Basic: object.SendFile(Optional *filename* As String = "") As Long

Version: 1.3.0

Parameter: *filename*.........String.
Represents the name of the program file (containing 6K programs/code) to be downloaded. If the filename is an empty string, then the user will be prompted for the filename.

Return Type: Long Integer.
If successful, the method returns a positive value; if unsuccessful, the method returns an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: To speed up downloads, the SendFile method strips comments from the downloaded 6K code. That is, all text between the comment delimiter (semi-colon) and the command delimiter (carriage return or line feed) is removed.
Background Commands: At the beginning of a file download operation, these commands are sent to the controller:

```
  !PORT0
  !ERRLVL0
```

After the download process is completed, these commands are sent to the controller:

```
  !PORT0
  !ERRLVL4
  !EOT13,0,0
  !EOL13,10,0
```

Note: If the download process is canceled, an "END" command is sent to the controller.

WriteCmd

Description: The WriteCmd method is used to send commands to the controller.

Visual Basic: object.WriteCmd(*sCmd* As String) As Long

Version: 1.3.0

Parameter: *sCmd*.........String.

A string of commands to be sent. Multiple commands can be sent, but each command should be separated with a valid 6K command delimiter (colon, carriage return, or line feed). The command string should be limited to 256 characters or less. Larger command strings may cause an overflow in the 6K's command buffer.

Return Type: Long Integer.

This method returns a positive value corresponding to the number of bytes sent, or an error code. Please see "Possible Error Codes from Com6srvr."

Remarks: You should disable Timer events in VB5 and VBScript when reading and writing to the COM6SRVR (see Microsoft Support Online Article ID176399).

Gemini RS-232 Properties

Get IsASCII

Description: The Get IsASCII returns the mode of communication
Visual Basic: object. Get IsASCII() As Boolean
Version: 1.3.0
Parameter: NONE.
Return Type: Boolean.
Remarks: This is a read-only property.

Get IsConnected

Description: The Get IsConnected returns the connection state.
Visual Basic: object. Get IsConnected() As Boolean
Version: 1.3.0
Parameter: NONE.
Return Type: Boolean.
Remarks: This is a read-only property.

Get LastCom6OcxError

Description: The Get LastCom6OcxError gets the Com6Ocx Error code
Visual Basic: object.Get LastCom6OcxError() As Com6OcxError
Version: 1.3.0
Parameter: NONE
Return Type: Com6OcxError (see table below)
Remarks: This is a read-only property.

The property returns a number (0 through -20) that refers to the Com6OcxError code.

The Com6srvrNet.ocx performs additional range checking because some parameters pass to the Com6srvrNet.Ocx as Long data and are then converted to Integer or Short data before being passed to Com6srvr. If parameters are determined to be out of range, the function stores an error code within the OCX. To display the last error, use the command LastCom6OcxError.

Note: By calling the last error, you also clear the internal error code in the OCX.

Com6srvrNet.ocx Error Codes

Error Code	**Description**
0	No errors are present. This error clears on successful connect or by calling LastCom6OcxError
-1	Specified axis out of range. Used Axis one.
-2	Specified brick out of range. Used Brick zero (on board)
-3	Specified analog channel out of range. Used Channel one
-4	Specified variable number out of range. Used VarI/B1
-5	Specified alarm bit out of range. Used zero.
-6	Specified TestBit in the IsBitSet function is out of range. Returns FALSE.
-7	Unable to perform operation because not connected.
-8	Unable to create Com6srvr object.
-9	Connect attempted failed.
-10	Called Property/Method while not connected.
-11	Invalid Internet Protocol Address or Invalid Parameter.
-12	Specified delay is out of range.
-13	Specified timeout is out of range.
-14	WriteBlocking timed out.
-15	SendFileBlocking cancelled out.
-16	Invalid Device for Function or Property
-17	Already connected to device
-18	Not Connected to fast status
-19	Ethernet Function call only
-20	RS232 Function call only

Get Visible

Description:	Use Get Visible to get the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.
Visual Basic:	object. Get Visible() As Boolean
Version:	1.3.0
Parameter:	NONE
Return Type:	Boolean
Remarks:	This is a read-only property. When the Boolean value is 1, the visible property tests as TRUE.

Let Visible

Description: Use Let Visible to set the Boolean value that determines whether the Com6srvrNet.ocx displays visible on a form.
Visual Basic: object. Let Visible(ByVal bNewValue As Boolean)
Version: 1.3.0
Parameter: *bNewValue*………Boolean
Return Type: NONE
Remarks: This is a write-only property.

Sub Disconnect

Description:	The Disconnect method closes a connection to a 6K controller
Visual Basic:	object.Disconnect()
Version:	1.3.0
Parameter:	NONE
Return Type:	NONE
Remarks:	As an out-of-process server, the 6K Communications Server does not shutdown until all client applications have disconnected from the server. In many cases, a proper disconnect does not take place if an unhandled error occurs in the client application and the program exits abnormally. This means that care must be exercised on the part of the client program to disconnect from the server on such occasions or when its services are no longer needed.

Sub Flush

Description: The Flush method removes all characters from the client's receive buffer. This method allows the programmer to clear the receive buffer prior to making a read.

Visual Basic: object.Flush()

Version: 1.3.0

Parameter: NONE

Return Type: NONE

Remarks: Use with caution. This method allows the programmer to clear the receive buffer, such that a subsequent Read call can yield a clean response. However, data arriving in the receive buffer is asynchronous to the application program and a thorough understanding of how the application program is structured is necessary to use this method correctly (for example, it would not be beneficial to Flush the buffer if only a partial response has been received).

Possible Error Codes from Com6srvr

Since the Com6srvrNet.ocx is a wrapper for the Com6srvr.exe, in certain error conditions, Com6srvr.exe will return an error code that is not a function of the Com6rvrNet.ocx. These errors are not handled by the Com6srvrNet.ocx and are just passed back as the return value for the function or property call.

-1 Bad Ethernet connection due to socket error
-2 Ethernet connection was shut down
-3 Connection attempt failed
-4 Maximum number of Ethernet connections exceeded
-5 Ethernet or RS232 connection not yet established
-6 No filename specified
-7 Unable to locate specified file
-8 Unable to open specified file
-9 Unable to ping Ethernet connection
-10 Unable to create Ethernet socket
-11 Invalid parameter passed to function
-12 Unable to create or connect Ethernet watchdog socket
-13 Unable to create or connect Ethernet fast status socket
-14 Unable to create or connect Ethernet alarm socket
-15 Unable to create or connect Ethernet command socket
-16 Unable to create client ring buffer for Ethernet command socket
-17 SetWatchdog returns this error when Windows runs out of timers.
-18 Unable to write due to XOFF condition (Gemini server only)
-19 WriteBlocking timed out
-20 SendFileBlocking was canceled

The following error codes were added in Version 1.3.0

-21 The Fast Status is Not Enabled
-22 Invalid Device for Function or Property
-23 Function or Property only Valid for Ethernet
-24 Function or Property only Valid for RS232