Communication Library Reference

The GalilTools Communication Library (Galil class) provides methods for communication with a Galil motion controller over Ethernet, RS-232 or PCI buses. It consists of a native C++ Library and a similar COM interface which extends compatibility to Windows programming languages (e.g. VB, C#, etc). More...

Public Members

- ***Connecting and Disconnecting***
  - **Galil**(string address = "") **address property**
  - **~Galil**()
  - vector<string>**addresses**()
  - string **connection**()
  - int **timeout_ms**
- ***Basic Communication***
  - string **command**(string command = "MG TIME", string terminator = "\r", string ack = ":", bool trim = true)
  - double **commandValue**(string command = "MG TIME")
  - string **message**(int timeout = 500) **onMessage Event**
- ***Programs***
  - void **programDownload**(string program = "MG TIME\rEN")
  - void **programDownloadFile**(string file = "program.dmc")
  - string **programUpload**()
  - void **programUploadFile**(string file = "program.dmc")
- ***Arrays***
  - void **arrayDownload**(vector<double> array , string name = "array")
  - void **arrayDownloadFile**(string file = "arrays.csv")
  - vector<double> **arrayUpload**(string name = "array")
  - void **arrayUploadFile**(string file, string names = "")

- ***Advanced***

  - void **firmwareDownloadFile**(string file = "firmware.hex")
  - string **read**()
  - int **write**(string bytes = "\r")
  - int **interrupt**(int timeout_ms = 500) **onInterrupt Event**
  - static string **libraryVersion**()

- ***Data Record***

  - vector<string> **sources**()
  - void **recordsStart**(double period_ms = -1)
  - vector<char>**record**(string method = "QR") **onRecord Event**
  - double **sourceValue**(vector<char> record, string source = "TIME")
  - string **source**(string field = "Description", string source = "TIME")
  - void **setSource**(string field = "Description", string source="TIME", string to= "Sample counter")

Detailed Description

The GalilTools Communication Library (Galil class) provides methods for communication with a Galil motion controller over Ethernet, RS-232 or PCI buses. It consists of a native C++ Library and a similar COM interface which extends compatibility to Windows programming languages (e.g. VB, C#, etc).

A Galil object (usually referred to in sample code as "g") represents a connection to a Galil controller. For Ethernet controllers, which support more than one connection, multiple objects may be used to communicate with the controller.

The library is conceptually divided into six categories:

1. Connecting and Disconnecting - functions to establish and discontinue communication with a controller.
2. Basic Communication - The most heavily used functions for command-and-response and unsolicited messages.
3. Programs - Downloading and uploading embedded programs.
4. Arrays - Downloading and uploading array data.
5. Advanced - Lesser-used calls.
6. Data Record - Access to the data record in both synchronous and asynchronous modes.

C++ Library (Windows and Linux)

Both Full and Lite versions of GalilTools ship with a native C++ communication library. The Linux version (libGalil.so) is compatible with g++ and the Windows version (Galil1.dll) with Visual C++ 2008. Contact Galil if another version of the C++ library is required. See the getting started guide and the hello.cpp example in /lib.

COM (Windows)

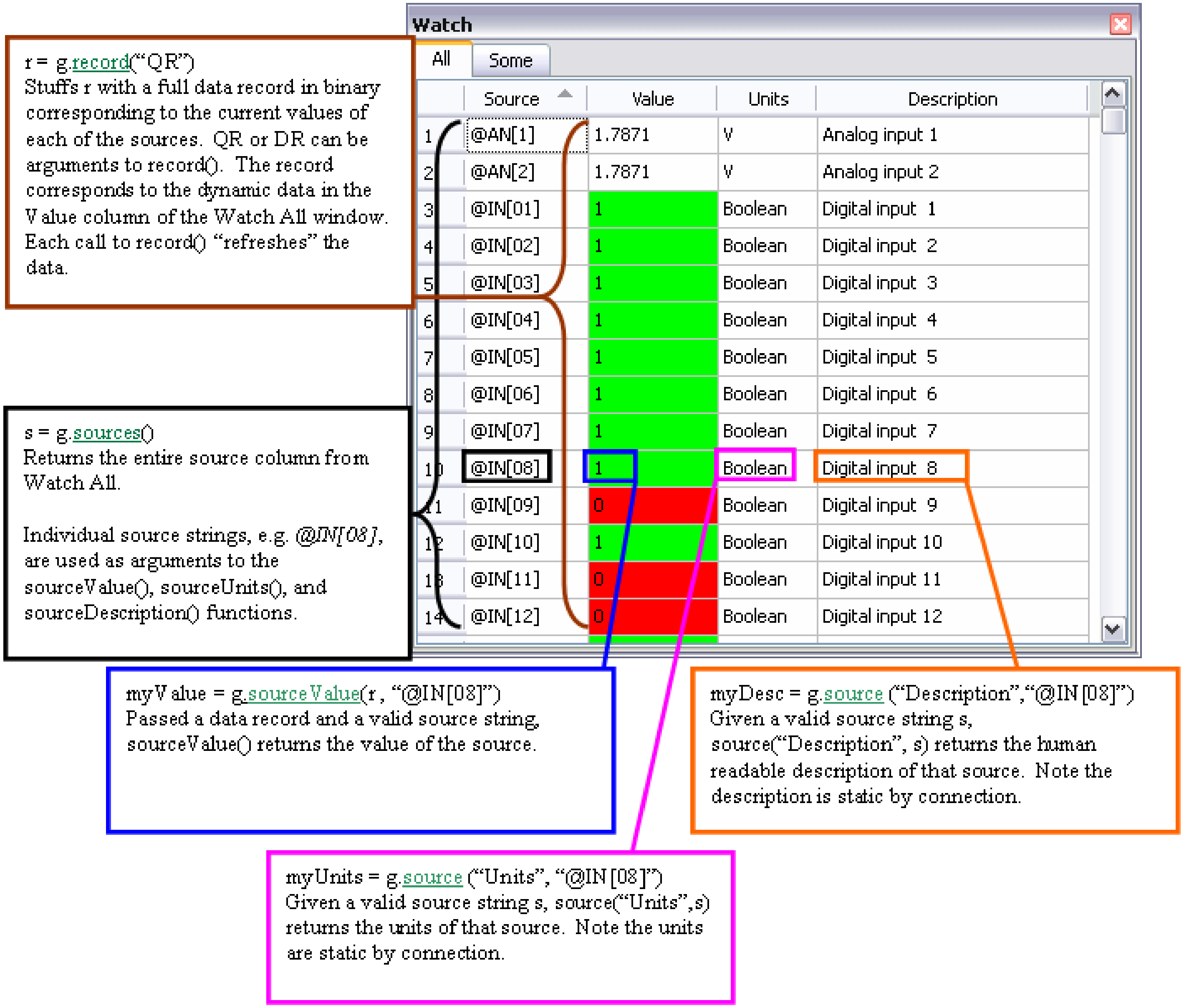
To further extend the language compatibility on Windows, a COM (Component Object Model) class built on top of the C++ library is also provided with Windows releases. This COM wrapper can be used in any language and IDE supporting COM (Visual Studio 2005, 2008, etc). The COM wrapper includes all of the functionality of the base C++ class. See the getting started guide and the hello.* examples in \lib for more info.

The Data Record

The data record is a Galil controller feature that is ideal for data collection and controller monitoring. It is a binary data structure generated by the controller's firmware either in a periodic and asynchronous manner (DR) or via an interrogated, synchronous command (QR). The transmitted data contains a variety of controller information: encoder positions, reference positions, profiler information, I/O, program information, user variables on the Accelera family, and much more.

The Watch All window in GalilTools displays the entire data record for a particular controller and is an excellent way to conceptualize the data record operation.

Graphical depiction of the data record operation.



	Source	Value	Units	Description
1	@AN[1]	1.7871	V	Analog input 1
2	@AN[2]	1.7871	V	Analog input 2
3	@IN[01]	1	Boolean	Digital input 1
4	@IN[02]	1	Boolean	Digital input 2
5	@IN[03]	1	Boolean	Digital input 3
6	@IN[04]	1	Boolean	Digital input 4
7	@IN[05]	1	Boolean	Digital input 5
8	@IN[06]	1	Boolean	Digital input 6
9	@IN[07]	1	Boolean	Digital input 7
10	@IN[08]	1	Boolean	Digital input 8
11	@IN[09]	0	Boolean	Digital input 9
12	@IN[10]	1	Boolean	Digital input 10
13	@IN[11]	0	Boolean	Digital input 11
14	@IN[12]	0	Boolean	Digital input 12

r = g.record(“QR”)
Stuffs r with a full data record in binary corresponding to the current values of each of the sources. QR or DR can be arguments to record(). The record corresponds to the dynamic data in the Value column of the Watch All window. Each call to record() “refreshes” the data.

s = g.sources()
Returns the entire source column from Watch All.

Individual source strings, e.g. *@IN[08]*, are used as arguments to the sourceValue(), sourceUnits(), and sourceDescription() functions.

myValue = g.sourceValue(r, “@IN[08]”)
Passed a data record and a valid source string, sourceValue() returns the value of the source.

myDesc = g.source (“Description”,“@IN[08]”)
Given a valid source string s, source(“Description”, s) returns the human readable description of that source. Note the description is static by connection.

myUnits = g.source (“Units”, “@IN[08]”)
Given a valid source string s, source(“Units”,s) returns the units of that source. Note the units are static by connection.

As in GalilTools, the data record Units, Description, Scaling, and Offset can all be set with the setSource() function. See setSource() and Watch for more information.

QR vs. DR

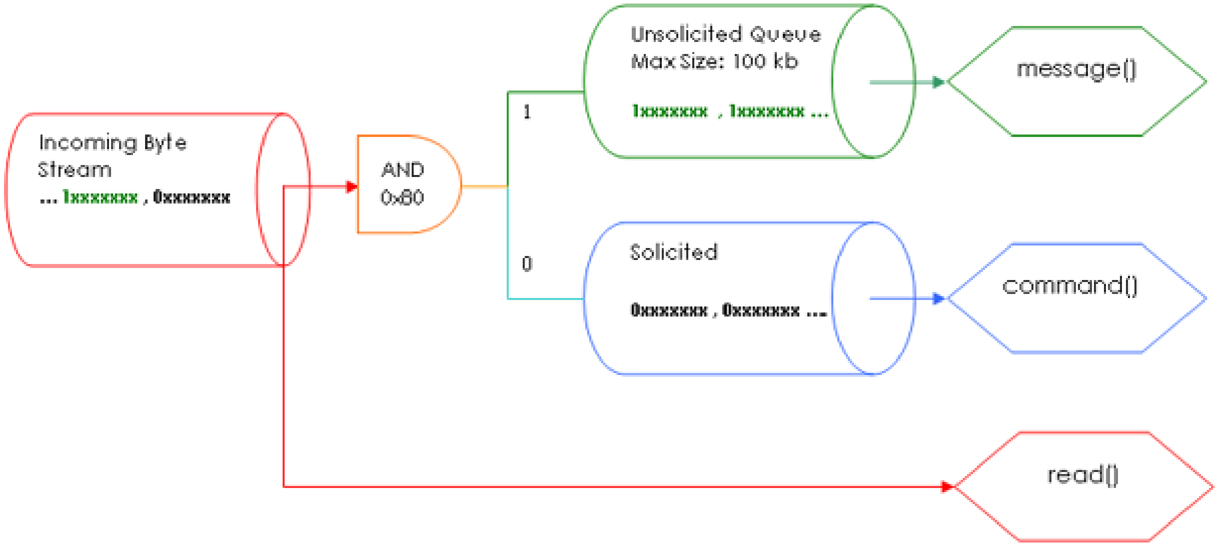
There are two methods for data record acquisition:

- ***QR***. By sending the command "QR" from the host, the controller responds back with a Data Record. This is a command-and-response acquisition. GalilTools Watch uses QR for low frequency, non-periodic data records.
- ***DR***. By using recordsStart(n), where n is a sample period in milliseconds, the controller is configured in asynchronous mode. Data will be sent by the controller periodically and without action from the host. The GalilTools Scope uses the DR method where possible for high-frequency, periodic data records.

The QR and DR details are abstracted by the data record API. Consult the controller command reference for more information regarding QR and DR.

Flow of data from controller.

The most significant bit in received bytes is used to determine unsolicited verses solicited data during a synchronous (solicited) transaction (See CW in controller command reference). If data is received by the library NOT during a synchronous transaction (e.g. command()) the data will automatically be put in the unsolicited queue regardless of the most significant bit.



Errors

The exceptions thrown by the Galil class were designed to allow for easy debugging by providing human-readable information. No error-code table needs to be referenced by the programmer as the error information is included in the error. **All thrown objects are strings**. String search functions or error codes can be used to determine the nature of an error.

"TC1" is sent to the controller automatically by the drivers in the event of a fault. This controller-generated error text is included in the string thrown by the drivers.

Example 1. Using string functions to programmatically handle thrown errors (below is a partial Try/Catch handler)

VB

```vb
Catch exception As System.Runtime.InteropServices.COMException
    Console.WriteLine(exception) 'print error message

    If exception.Message.Contains("COMMAND ERROR") Then
        Console.WriteLine("a command error occurred") 'special processing for command errors

    End If
End Try
```

For programmers wishing to use a traditional return code, the first 4 characters of every error message contains a unique error code.

The first character of the return code contains the broad category of error:	The next two digits contain a number specifying the function which generated the error. Note there is a private funtion at x01x that may throw in other functions.	The last digit (xxx0) is a serial number (0-9) to provide uniqueness among very similar errors.
1xxx TIMEOUT 2xxx COMMAND 3xxx MONITOR 4xxx FILE 5xxx OPEN 6xxx WRONG BUS 7xxx INVALID 8xxx WARNING 9xxx OFFLINE (COM only)	x00x Galil() x02x command() x03x message() x04x interrupt() x05x programUpload() x06x programDownload() x07x programUploadFile() x08x programDownloadFile() x09x arrayUpload() x10x arrayDownload() x11x arrayUploadFile() x12x arrayDownloadFile() x13x firmwareDownloadFile() x14x recordsStart() x15x record() x16x sourceValue() x17x source() x18x setSource() x19x sources() x20x write() x21x read() x22x commandValue() x23x connection() x24x timeout_ms	

Example 2. Using the first 4 characters of a thrown string as a return code.

C++

```cpp
catch(string e){
   int t = atoi( (e.substr(0,1)).c_str() ); // get int for type, first digit of code

   int f = atoi( (e.substr(1,2)).c_str() ); // int for function, middle two digits of code

   int u = atoi( (e.substr(3,1)).c_str() ); // int for unique serial, last digit of code

   //ints can now be used in switch statements to handle all variations of error
}
```

Member Documentation

top

C++ **Galil::Galil(string address = "")**

VB **address As String**

C# **string address**

Galil() is the constructor for the Galil C++ class and invoking it will create a new instance and open the specified connection. There is one argument to the constructor class which indicates the connection to open. Set the *address* property when using the COM class on Windows.

C++ Type	Argument	Example	Description
string	address	"GALILPCI1"	Connection string for controller (see examples of valid strings below). If "" (null), display the Connections dialog to allow the user to choose a controller.

Example 1. null string

With a null string passed to the Galil constructor, the Connections dialog will be presented for the user to choose the connection to open.

VB

```vb
Dim g As New Galil.Galil 'Dimension g variable for instantiated object
```

```
g.address = "" 'Specify null address, Connections dialog will display
```

C++

```
Galil g("");   //prompt user for connection
```

C++

```
Galil g;   //uses default "" from header
```

Example 2. Valid addresses (Note: strings in quotes are valid for all programming languages)

C++

```
Galil g("COM1 19200");          //RS-232 port 1, 19200 baud, Windows (DMC-21x3/40x0)

Galil g("/dev/ttyS0 19200");    //RS-232 port 1, 19200 baud, Linux (DMC-21x3/40x0)
Galil g("192.168.1.105");       //IP Address (TCP) (DMC-21x3/40x0/RIO)
Galil g("RIO47100-13");         //DNS name for supported controller (DHCP support) (RIO/DMC-4000)

Galil g("GALILPCI1");           //Galil PCI (DMC-18x6/18x2), Windows (DMC-18x2/18x6)
Galil g("/dev/galilpci0");      //Galil PCI (DMC-18x6/18x2), Linux (DMC-18x2/18x6)
Galil g("OFFLINE");             //Offline connection.
```

Note: The connection behaviour can be modified by appending address line switches to the connection address string. See the connection address options table in the connections chapter for a complete list.

Example 3. Path to Connection file

If the address ends in the substring ".con", it refers to a con file as saved by the Connections dialog.

C++

```
Galil g("connection.con");                                              //Connect to string in con file

Galil g("c:/documents and settings/John Doe/Galil/myconnection.con");   //.con file at different path than executable
```

C#

```
Galil.Galil g = new Galil.Galil(); //Dimension g variable for instantiated object

g.address = "myController.con"; //connect to contents of con file
```

A .con file is simply a text file containing a valid address.

A connection can be changed simply by modifying the object's address.

Example 4. Changing connection of Galil object

VB

```
For i As Integer = 1 To 5 '5 iterations

    g.address = "" 'display connection chooser to determine connection
    Label1.Text = g.connection 'display connection on Form label
Next i
```

Upon invoking the constructor, several values are interrogated: controller revision information, presence of analog inputs, serial number, connected handle, etc., and the following state changes are made:

Command Sent	Purpose	Consequences
CFI	Set the current connected handle as the handle used for unsolicited messages. (Ethernet Only)	CFI will "steal" the unsolicited handle away from any other connection currently using it.
EO0	Disable Echo on RS-232 connections.	Commands typed will not be echoed back by the controller. This is important so that echoed commands are not interpreted as command responses.
CW1	Mark Unsolicited Bytes.	All unsolicited messages will have the most significant bit (extended ASCII) set. This allows the library to differentiate between solicited (zero in the highest bit) and unsolicited (highest bit 1) bytes. See message() for more info.

For the COM version, if it is desirable to set the address property frequently, the user may benefit from creating a new instance of the Galil object for each iteration. When doing so, note that the original connection is not released until the original object is destroyed. For example in dot net languages, after assigning a new instance to the variable, the user can invoke the garbage collector (GC.Collect()) to destroy the old instance of Galil and thereby disconnect from the controller.

```
Throws:

        5000 OPEN ERROR.  User cancelled connection from dialog in Galil::Galil()
        4000 FILE ERROR.  Galil::Galil() failed to open file...
        5001 OPEN ERROR.  OFFLINE specified to Galil::Galil()
        5002 OPEN ERROR.  Galil::Galil() failed to open PCI device...
        5003 OPEN ERROR.  Galil::Galil() failed to open RS-232 port...
        5004 OPEN ERROR.  Galil::Galil() failed to open Ethernet host...
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
```

top

C++

Galil::~Galil()

~Galil is the destructor for a Galil object and is automatically invoked when a Galil object's scope is ended (or when delete is called in C++). When the destructor runs, the connection to the controller is closed.

Example 1. Destructor gets called when scope of object ends.

C++

```
{// scope for g
        Galil g("192.168.1.100"); //Constructor -- Connect to controller
}//destructor invoked on g -- Connection is closed
```

Example 2. Disconnecting from a controller (handling OFFLINE error).

VB

```
Try 'main Try Catch block
    g = New Galil.Galil 'create a new Galil object

    g.address = "10.0.6.134" 'specify address
    MsgBox("Connected to " + g.connection) 'display connection information

    Try 'nested Try catch block
      g.address = "OFFLINE" 'OFFLINE causes the destructor to run, disconecting from the controller
    Catch ex As Exception 'OFFLINE throws an error so catch and ignore it

    End Try
    MsgBox("Now Disconnected") 'code keeps running
Catch ex As Exception 'catch for the main Try.  This would catch normal errors, such as connection timeout
End Try
```

If the DR interval is 0 upon instantiation and recordsStart() is called during the extent of a Galil object, the destructor will issue a DR0 to terminate the data record automatic dispatch. If the DR interval was nonzero at connection or if recordsStart() was not called, the destructor does not change the DR state.

TCP connections will be closed, however UDP resources, being connectionless, are not released. If the user desires to close UDP resources, the IH command may be used prior to destructor. See the controller's command reference under IH.

`Throws: none`

top

C++

vector<string> Galil::addresses()

Returns a list of available addresses to connect to (e.g. "1.2.3.4"). Each item in the list may be fed to the constructor Galil(). This is a dynamically generated list with the same contents as the Available Tab in the GalilTools connections dialog. Each time this function is called is like clicking refresh in the Available Tab

C++ Type	Argument	Example	Description
vector<string>	return	"192.168.1.100 ; 192.168.1.101 ; 192.168.1.102"	A list of all available controllers by address

Although not necesarily addresses of controllers, available RS-232 ports will also be listed.

Example 1. Printing available addresses

VB

```
Dim g As New Galil.Galil 'note, no connection needed to invoke addresses()
For Each address As String In g.addresses()
    TextBox1.AppendText(address + vbCrLf)
Next
```

Throws:none

top

C++

string Galil::connection()

Returns a string containing connection-related information. This information includes the address, controller firmware revision information, serial number, and Ethernet handle (where applicable). The following are return examples.

GALILPCI1, DMC1886 Rev 1.0b, 36
COM2 19200, DMC2182 Rev 1.0p, 202
192.168.1.2, DMC4020 Rev 1.0a, 189, IHD

C++ Type	Argument	Example	Description
string	return	"GALILPCI1, DMC1886 Rev 1.0b, 36"	Controller and connection information

Example 1. Printing the connection information

VB
```
MsgBox(g.connection, MsgBoxStyle.DefaultButton1, "Galil Controller Info") 'display message box with connection info
```
C#
```
Console.WriteLine(g.connection()); //write connection information to the console
```
C++
```
cout << g.connection() << endl;  //print connection info to screen
```

```
Throws:
        9230 UNINITIALIZED OBJECT ERROR.  Galil::connection() called without Galil::address set
```

---

top

C++ int Galil::timeout_ms

Specifies the general purpose timeout in milliseconds for Galil(), command(), commandValue(), programUpload(), programUploadFile(), arrayUpload(), arrayUploadFile(), and record(). It defaults to 500ms.

Example 1. Specifying the timeout for a connection

VB
```
g.timeout = 2000 'set timeout to two seconds
```

```
Throws (COM only):
        9240 UNINITIALIZED OBJECT ERROR.  Galil::timeout_ms set without Galil::address set
        9241 UNINITIALIZED OBJECT ERROR.  Galil::timeout_ms read without Galil::address set
```

---

top

C++ string Galil::command(string command = "MG TIME", string terminator = "\r", string ack = ":", bool trim = true)

Sends a command to the Galil controller and returns the response from the controller if ack is received within timeout_ms. If an unexpected response is returned (usually a "?") or the timeout is exceeded, the function will throw an error.

C++			

Type	Argument	Example	Description
string	return	"123"	The function returns a string containing the response from the controller.
string	command	"TPA"	A valid Galil command to send to the controller.
string	terminator	"\r"	A string to append to the end of command. In almost all cases a carriage return should be used. NOTE: Do not append a \r in the "command" string.
string	ack	":"	The expected last character acknowledgemnet in a good response. In almost all cases a colon is expected.
bool	trim	true	If true, the leading space and the trailing carriage return, line feed and colon will be stripped from the response. False is useful for terminal applications.

Example 1. Basic command() use:

VB

```
Button1.Text = g.command("MG@IN[1]") 'update Button1 text with the state of controller digital input 1
```

C#

```
button1.Text = g.command("TI","\r",":",true); //update button1 text with input byte 0 (Tell Input, TI). Note, C# requires all command() args present
```

Example 2. Using the extra arguments (advanced). Specify a different terminator to allow program download with overwrite (See DL in controller command reference).

C++

```
g.programDownload("MG\"Start\"\r#HERE\rMG\"Overwrite This\"\rEN\r"); //download code

g.command("DL#HERE\rMG\"Overwritten\"\rEN\r","\\"); //overwrite the program at label #HERE.  Note the backslash terminator.
```

Example 3. Using the extra arguments (advanced). Specify a different terminator to allow array download with overwrite (See QD in controller command reference).

VB

```
Dim arrayData = New Integer() {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} 'create array

g.arrayDownload(arrayData, "A") 'download the array
g.command("QD A[],8,9" + vbCr + "100" + vbCr + "200", "\", ":", True) 'overwrite the last two elements of array.  Note the backslash terminator.
```

Example 4. Binary command() use (advanced)

In almost all applications, use of binary commands is not necessary; however, for those applications requiring it, the command function can be used.

C++
```
g.command(string("\xA0\x00\x00\x01",4),""); //binary for BGA.  Note no carriage return in terminator.
```

```
Throws:
        7020 INVALID COMMAND ERROR.  DL, UL, ED, and QD are not allowed from Galil::command()
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9020 UNINITIALIZED OBJECT ERROR.  Galil::command() called without Galil::address set
```

top

C++ double Galil::commandValue(string command = "MG TIME")

Convenience function for interrogating a single integer ("MG_TPA"), real number ("MG@AN[1]"), or boolean ("MG_BGA") value from the controller.

C++ Type	Argument	Example	Description
double	return	2.3524	Response from controller converted to floating point
string	command	"MG@AN[1]"	The command to send to the controller. NOTE: Do not append a \r in the "command" string.

Example 1. Interrogating Position

C#
```
double positionA = g.commandValue("TPA");
```

Note: For commands that don't return a number (e.g. XQ, ST, BG), a zero will be returned.

```
Throws:
        7020 INVALID COMMAND ERROR.  DL, UL, ED, and QD are not allowed from Galil::command()
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9220 UNINITIALIZED OBJECT ERROR.  Galil::commandValue() called without Galil::address set
```

top

C++	**string Galil::message(int timeout_ms = 500)**
VB Event	**Sub onMessage(ByVal message As String)**
C# Event	**void onMessage(string message)**

Returns unsolicited messages (MGs) sent from the controller. Traffic transmitted by the controller to the host can be categorized into two primary groups: solicited and unsolicited. Solicited bytes are responses from commands sent by the host. For example, a solicited message is an encoder position string returned from *g.command("TPX")*. Unsolicited messages are transmitted asynchronously and could be output from MG commands from a running program aboard the controller, asynchronous error messages, trace (TR1) output, or responses from commands in a controller-side DMC program (TP, RP, etc).

For Ethernet connections, the library will automatically open two Ethernet handles on the controller. One TCP handle for command-and-response traffic, and one UDP handle for asynchrounous data, including messages (MGs).

C++: If the host-side message queue is empty, message() will wait up to timeout_ms for a message. If none occur within timeout_ms, a timeout error will be thrown. If there are already messages in the queue when message() is called, the function will immediately return the contents. If a zero timeout is specified, no errors will be thrown; message() will simply return the waiting queue (even if it is empty, ""). A -1 timeout will cause message() to block until a message is received.

COM: There are no timeouts for the onMessage event.

C++ Type	Argument	Example	Description
string	return	"Hello\rMy TP Value-123.0000"	Returned contents of the unsolicited messages buffer
int	timeout_ms	500	The time in milliseconds to wait for the existence of data in the message queue

Example 1. Printing unsolicited messages

VB

```
Private Sub g_onMessage(ByVal message As String) Handles g.onMessage 'event runs when message received

   messagesTextBox.AppendText(message) 'print to textbox
End Sub
```

C#

```
//in initialization code block

g.onMessage += new Galil.Events_onMessageEventHandler(g_onMessage); //hook up to the onMessage Event

//event subroutine
void g_onMessage(string message) { //handler for the onMessage event
   messagesTextBox.AppendText(message); }
```

C++

```
cout << g.message(0); //Print current message buffer to console. No waiting or throws.
```

```
Throws:

        1030 TIMEOUT ERROR.  Galil::message() took longer than ... ms to read MG
```

top

C++ void Galil::programDownload(program = "MG TIME\rEN")

Downloads a program from a host buffer to the Galil controller. For very long programs which are larger than the controller's program space (e.g. 80 characters by 2000 lines on DMC-40x0), the function will attempt to compress the program before downloading. This is done by removing white space and concatenating multiple commands onto each line of code. A warning (8060) will be thrown signaling that compression was required and a try-catch block included in every call to this function is recommended to handle this possible warning condition.

If line zero of the buffer contains *REM DISABLE COMPRESSION*, code will not be compressed and downloading otherwise compressable code will throw an error and fail.

See write() for examples of downloading code with insertion and while motion is profiling.

C++ Type	Argument	Example	Description

string	program	"MG TIME\rEN"	String containing the program to download with each line separated by a carriage return.

Example 1. Download a program to controller from buffer

C++
```
g.programDownload("#A\rMGTIME\rWT500\rJP#A\rEN"); //Download a program to controller
```

Example 2. Download program from textbox and then execute

VB
```
g.programDownload(ProgramTextBox.Text) 'Download the contents of the textbox
g.command("XQ") 'Send execute command
```

```
Throws:
        7060 INVALID CHARACTER ERROR.  Galil::programDownload() can't download program with backslash \ character.  Use {^92} in MG commands
        7061 INVALID LENGTH ERROR.  Galil::programDownload() can't compress line "..." of ... columns or more
        7062 INVALID LENGTH ERROR.  Galil::programDownload() can't download compressed program with more than ... lines by ... columns.
           Contact Galil for special  memory expansion firmware:   www.galilmc.com/products/accessories/upgd_options.html#expanded_memory
        7063 INVALID LENGTH ERROR.  Galil::programDownload() can't download program with more than ... lines by ... columns because compression is disabled.
        8060 COMPRESSION WARNING.  Galil::programDownload() modified program to fit in ... lines by ... columns (check LS)
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9060 UNINITIALIZED OBJECT ERROR.  Galil::programDownload() called without Galil::address set
```

top

C++ void Galil::programDownloadFile(string file = "program.dmc")

Downloads a program file to the Galil controller. The program text is read from a file located on the host file system. For very long programs which are larger than the controller's program space (e.g. 80 characters by 2000 lines on DMC-40x0), the function will attempt to compress the program before downloading. This is done by removing white space and concatenating multiple commands onto each line of code. A warning (8060) will be thrown signaling that compression was required and a try-catch block included in every call to this function is recommended to handle this possible warning condition.

If line zero of the file contains *REM DISABLE COMPRESSION*, code will not be compressed and downloading otherwise compressable code will throw an error and fail.

See write() for examples of downloading code with insertion and while motion is profiling.

C++ Type	Argument	Example	Description
string	program	"galilProgram.dmc"	String containing path and filename of dmc file. If only the filename is specified, the executable's directory will be used as the path.

Example 1. Display file chooser and download selected file

VB

```
Dim chooseFile As New OpenFileDialog()  'dimension file chooser object

chooseFile.Filter = "DMC Files (*.dmc)|*.dmc" 'set filter to dmc files
chooseFile.ShowDialog() 'display chooser to user
g.programDownloadFile(chooseFile.FileName) 'download selected path (note, no error checking included)
```

Example 2. Download a program to controller from file

C++

```
g.programDownloadFile("myProgram.dmc"); //Download a program to controller
```

```
Throws:
        4080 FILE ERROR.  Galil::programDownloadFile() failed to open file...
        7060 INVALID CHARACTER ERROR.  Galil::programDownload() can't download program with backslash \ character.  Use {^92} in MG commands
        7061 INVALID LENGTH ERROR.  Galil::programDownload() can't compress line "..." of ... columns or more
        7062 INVALID LENGTH ERROR.  Galil::programDownload() can't download compressed program with more than ... lines by ... columns.
          Contact Galil for special  memory expansion firmware:   www.galilmc.com/products/accessories/upgd_options.html#expanded_memory
        7063 INVALID LENGTH ERROR.  Galil::programDownload() can't download program with more than ... lines by ... columns because compression is disabled.
        8060 COMPRESSION WARNING.  Galil::programDownload() modified program to fit in ... lines by ... columns (check LS)
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9080 UNINITIALIZED OBJECT ERROR.  Galil::programDownloadFile() called without Galil::address set
```

top

C++ string Galil::programUpload()

Uploads the controller's program to a buffer on the host.

C++ Type	Argument	Example	Description
string	return	"#A\r\nMGTIME\r\nWT500\r\nJP#A\r\nEN"	Uploaded controller program with lines separated by carriage return, new line.

Example 1. Upload program and print to screen

C#

```
OutputTextBox.AppendText(g.programUpload()); //Append controller program buffer to form textbox (VB syntax similar)
```

C++

```
string buffer = g.programUpload();  // declare string and stuff with uploaded program
cout << buffer;         // print buffer to console
```

```
Throws:
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9050 UNINITIALIZED OBJECT ERROR.  Galil::programUpload() called without Galil::address set
```

top

C++

void Galil::programUploadFile(string file = "program.dmc")

Accepts a string argument containing the path to a local file where the uploaded program is to be saved. If no path is specified, the executable's directory is used. If a file with the same name and path exists, it will be overwritten. If no matching file exists, a new file will be created.

C++ Type	Argument	Example	Description
string	file	"uploadedProgram.dmc"	Desired filename where the controller program will be saved

Example 1. Upload program and save to file

C++

```
g.programUploadFile("Galil.dmc");  // upload program and save to file
```

Example 2. Display save file chooser and save controller program buffer to location user specifies

VB

```
Dim saveFile As New SaveFileDialog() 'dimension file chooser object
saveFile.Filter = "DMC Files (*.dmc)|*.dmc" 'set filter to dmc files

saveFile.ShowDialog() 'display chooser to user
g.programUploadFile(saveFile.FileName) 'save controller buffer to file
```

```
Throws:
        4070 FILE ERROR.  Galil::programUploadFile() failed to open file...
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9070 UNINITIALIZED OBJECT ERROR.  Galil::programUploadFile() called without Galil::address set
```

top

C++

void Galil::arrayDownload(vector<double> array , string name = "array")

Downloads a single, one dimensional array to the controller. If the array name already exists on the controller, the existing array will be deallocated and replaced with the new array (this allows for smaller or larger arrays than the existing array to be downloaded).

Do not execute this API while a Record Array Mode is running on the controller (RC,RD,RA). _RC can be interrogated to ensure the Record Array Mode is idle before downloading to the array table.

C++ Type	Argument	Example	Description
vector<double>	array	(1,2,3,4)	Data on host for download.
string	name	"A"	Name of array on controller where data will be saved. Array does not need to preexist.

Example 1. Downloading an array of numbers to the controlller

VB

```
Dim myArray() As Integer = {1, 2, 3, 4, 5} 'create an array of information to download to the controller

g.arrayDownload(myArray, "array") 'Create an array on the controller named "array" and load the information from myarray into it
```

```
vector<double> arraydata;  //declare vector to hold array
```

C++

```
for(int i = 0; i < 1000 ; i ++)
        arraydata.push_back(3.1416);  //fill array with 1000 values of pi

g.arrayDownload(arraydata,"A"); //download array to controller array "A"
```

```
Throws:
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        7101 INVALID ARRAY DOWNLOAD ERROR. Galil::arrayDownload() does not modify ... while RC is running.
        9100 UNINITIALIZED OBJECT ERROR.  Galil::arrayDownload() called without Galil::address set
```

top

C++ void Galil::arrayDownloadFile(string file = "arrays.csv")

Allows a group of arrays to be downloaded to the controller from a csv file.

If there are existing array names on the controller which match array names in the specified csv file, the controller arrays are overwritten. If there are existing array names on the controller not in the csv file, the non-matching arrays are untouched. This means that the csv download function will append to the array table where possible, and overwrite where necessary.

Do not execute this API while a Record Array Mode is running on the controller (RC,RD,RA). _RC can be interrogated to ensure the Record Array Mode is idle before downloading to the array table.

C++ Type	Argument	Example	Description
string	file	"dataTable.csv"	Path to csv file on host containing one or more arrays to write to the controller's array table. Note the array names are the first row of the csv.

Example 1. Downloading multiple arrays from a csv file to the controller.

VB

```
g.arrayDownloadFile("arrays.csv") 'download the contents of arrays.csv to the controller array table
Label2.Text = g.command("LA") 'write the names and sizes of the controller's arrays (LA) to a form label
```

C++
```
g.arrayDownloadFile("array.csv"); //download the contents of array.csv to the controller array table
```

```
Throws:

       4112 FILE ERROR.  Galil::arrayDownloadFile() failed to open file...
       7121 INVALID ARRAY DOWNLOAD ERROR. Galil::arrayDownloadFile() does not modify ... while RC is running.
       1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
       1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
       2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
       3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
       9120 UNINITIALIZED OBJECT ERROR.  Galil::arrayDownloadFile() called without Galil::address set
```

top

C++ vector<double> Galil::arrayUpload(string name = "array")

Uploads a particular array from the controller to a host buffer.

C++ Type	Argument	Example	Description
vector<double>	return	(1.234 , 2.456 , ...)	Contents of the uploaded array
string	name	"array"	Name of controller-side array to be uploaded. Array names can be interrogated with the "LA" (List Arrays) command.

Example 1. Uploading array data from controller

C++
```
vector<double> arrayUp;  //create vector for array
arrayUp = g.arrayUpload("A"); //upload array from controller
```

Example 2. Uploading and printing array data from controller

C#
```
Array arrayUp = (Array)g.arrayUpload("a");  //upload controller array "a" to local host array.

for (int i = 0; i < arrayUp.Length; i++)

    OutputTextBox.AppendText(i.ToString() + ": " + arrayUp.GetValue(i) + "\r\n");  //print the uploaded array
```

```
Throws:
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9090 UNINITIALIZED OBJECT ERROR.  Galil::arrayUpload() called without Galil::address set
```

top

C++ void Galil::arrayUploadFile(string file, string names = "")

Uploads the entire controller array table or a subset thereof and saves it as a csv file specified by the user.

C++ Type	Argument	Example	Description
string	file	"c:\\Documents and Settings\\JohnDoe\\array\\array.csv"	The path to the file to which the array data should be saved
string	names	"array1 posX A B"	A space delimited string containing the array names that should be uploaded. A null string, "", uploads all arrays.

Example 1. Uploading a controller array subset to a csv file

C++

```
g.arrayUploadFile("arraysUp.csv", "A B C"); //upload arrays A, B, and C from controller to arraysUp.csv
```

Example 2. Uploading the entire array table to a user-specified path and filename, and then opening that file in Microsoft Excel

VB

```
Dim saveFile As New SaveFileDialog() 'dimension file chooser object

saveFile.Filter = "CSV Files (*.csv)|*.csv" 'set filter to csv files
saveFile.ShowDialog() 'display chooser to user
g.arrayUploadFile(saveFile.FileName) 'save controller array table to csv file

'Shell() is used to execute external applications in the Windows shell.  The path to Excel is version-specific
Shell("C:\Program Files\Microsoft Office\Office10\EXCEL.EXE """ + saveFile.FileName + """", AppWinStyle.MaximizedFocus, False)
```

```
Throws:
        4110 FILE ERROR.  Galil::arrayUploadFile() failed to open file...
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9110 UNINITIALIZED OBJECT ERROR.  Galil::arrayUploadFile() called without Galil::address set
```

top

C++

void Galil::firmwareDownloadFile(string file = "firmware.hex")

Allows controller firmware to be upgraded. Note that Galil distributes firmware files in a zip archive which must be decompressed prior to loading. Firmware files end in a .hex extension. Ensure the correct firmware version/model before downloading (loading an incompatible firmware will require a reloading of the correct firmware).

DMC-21x3: this function is only allowed via RS-232.

Upon invoking firmwareDownloadFile() a GUI progress bar dialog will display.

C++ Type	Argument	Example	Description
string	file	"c:\\Documents and Settings\\JohnDoe\\Galil Firmware\\d400rcur.hex"	The path to the firmware hex file to download. If no path is specified, the executable directory is used.

Example 1. Ask user for firmware file path and update controller firmware

VB

```
Dim chooseFile As New OpenFileDialog()  'dimension file chooser object

chooseFile.Filter = "Hex Files (*.hex)|*.hex" 'set filter to hex files
chooseFile.ShowDialog() 'display chooser to user
g.firmwareDownloadFile(chooseFile.FileName) 'download firmware
```

C++

```
string filename; //create string for filename
cout << "Enter firmware path to download:" << endl; //prompt user
```

```
getline(cin,filename); //get input from the console (hex file expected) -- getline requires #include <fstream>
g.firmwareDownloadFile(filename); //update controller firmware
```

```
Throws:
        4130 FILE ERROR.  Galil::firmwareDownloadFile() failed to open file...
        6130 WRONG BUS ERROR.  Galil::firmwareDownloadFile() isn't allowed via Ethernet.  Use RS-232
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of > response...
        9130 UNINITIALIZED OBJECT ERROR.  Galil::firmwareDownloadFile() called without Galil::address set
```

top

C++ string Galil::read()

Allows for direct read access to the input buffer from the controller. This function is not intended for normal use, as the other APIs provide a higher level of functionality. This function should be used in cases of firmware specials which break the standard Galil communication standards, or for users who wish to implement their own lower-level data-parsing algorithms. See Detailed Description for a graphical overview of how the input buffer is parsed.

C++ Type	Argument	Example	Description
string	return	"123\r\n:"	Actual bytes read

Example 1. Read generic data from controller

C++

```
string r;
g.command("CW2");       //controller won't set MSB for MGs. Note Galil software usually expects CW1

while (1){
   r = g.read();       //read any available data

   if (r.size() != 0) //if non-empty

      cout << r;       //print
}//while
```

VB

```
Dim r As String

g.command("CW2") 'controller won't set MSB for MGs. Note Galil software usually expects CW1
While (Not Console.KeyAvailable) 'no key pressed

    r = g.read() 'read any available data
    If (r.Length <> 0) Then 'if non-empty

        Console.WriteLine(r) 'print
    End If
End While
```

Note: The status of the data adjustment bit (CW) will influence how unsolicited messages are returned from the controller.

```
Throws:
        9210 UNINITIALIZED OBJECT ERROR.  Galil::read() called without Galil::address set
```

top

C++

int Galil::write(string bytes = "\r")

Allows for direct writes to the controller. This function is not intended for normal use, rather for special applications where command() can not be used. Note: command() can be used for the majority of even special applications.

C++ Type	Argument	Example	Description
int	return	18	The actual number of bytes written (which can be less than the length of the "bytes" argument below)
string	bytes	"MGTIME\rTPX\rMGTIME\r"	Buffer to write to the controller. Length must be less than or equal to 511 bytes.

Example 1. write/read transaction

```
string buffer = "";  //buffer to hold read data
g.write("\x12\x16\r"); //write <ctrl>R<ctrl>V to the controller
```

C++

```
while ( buffer == "") buffer = g.read(); //while read returns nothing keep reading
cout << "read() returned " << buffer.size() << " bytes.  It said " << buffer << endl;  //print response
```

VB

```
Dim buffer As String = "" 'buffer to hold read data
g.write(Chr(18) + Chr(22) + vbCr) 'write <ctrl>R<ctrl>V to controller (Chr takes ascii decimal, not hex)
While (buffer.Length = 0) 'while read returns nothing
    buffer = g.read() 'keep reading
End While
Console.WriteLine("Read() reaturned " + buffer.Length.ToString + " bytes.  It said " + buffer + vbCrLf) 'print response
```

Example 2. Program download with insertion, or while motion is profiling

C++

```
/*
  programDownload() and programDownloadFile() do not support
  download with insertion or download while motion is profiling
  Assume file 'dmcFile' is open and
  'lines' contains the number of program lines
  in the file (less REM lines)
  'lineCap' contains the max number of lines downloadable
  Program line capacity:
   DMC-40x0, 18x6, 41x3 = 2000
   RIO-47xx0 = 200
   RIO-47xx2 = 400
   Others = 1000
  'charsCap' contains the number of characters per line
   RIO-xxxx = 40
   Others = 80

  'location' is a std::string containing the insertion point
   ""        download at beginning of program buffer
   #Label    Begins download at line following #Label (more lines checking required)
   #         Begins download at end of program in RAM (more lines checking required)
*/
if (lines <= lineCap)
{
        g.write("DL"+location+"\r"); //start the download

        while (!dmcFile.eof())
        {
            getline(dmcFile,line);
            if (line.substr(0,3) != "REM")//filter out REM lines
            {
                if (line.size() > charsCap)
                {
                cout << "Line too long: " << line << endl;
                g.write("\\");//close write session on controller
                return 1;
                }
              g.write(line + "\r");
            }
        }//while
        g.write("\\");//close write session on controller
}
else
 cout << "Too many lines: " << lines << endl;
```

```
Throws:
        9200 UNINITIALIZED OBJECT ERROR.  Galil::write() called without Galil::address set
```

top

C++	**int Galil::interrupt(int timeout_ms = 500)**
VB Event	**Sub onInterrupt(ByVal status As Integer)**
C# Event	**void onInterrupt(int status)**

Provides access to the PCI and Ethernet interrupt status byte for both the EI and UI conditions. DMC-18x2/6 and DMC-40x0 Rev 1.0b only.

C++: When invoked, this function will sleep until an interrupt is received (or timeout expires) and returns the controller's status byte. A timeout of zero will check for an interrupt condition but will not throw an error. Instead, it will return the status byte or return zero if there isn't a status byte waiting. A timeout of -1 will block until an interrupt occurs.

COM: The onInterrupt event does not have a timeout.

C++ Type	Argument	Example	Description
int	return	0xf0	The returned status byte from an interrupt event. See chapter 4 in PCI controller manual for a reference table. See EI in the DMC-40x0 command reference for further Ethernet

			information.
int	timeout_ms	100	time in milliseconds to wait for interrupt to occur

Example 1. Returning status byte from an interrupt.

VB

```
Private Sub g_onInterrupt(ByVal status As Integer) Handles g.onInterrupt 'subroutine runs when Interrupt event occurs

   interruptsTextBox.AppendText(status.ToString + vbCrLf) 'print status byte to text box
End Sub
```

C#

```
//in initialization code block
g.onInterrupt += new Galil.Events_onInterruptEventHandler(g_onInterrupt); //hook up to the onInterrupt Event

//event subroutine
void g_onInterrupt(int status){ //handler for the Interrupt event
    interruptsTextBox.AppendText(status.ToString()+ "\r\n"); }
```

C++

```
int status = g.interrupt(100); //sleep for interrupt 100ms
```

```
Throws:
        1040 TIMEOUT ERROR.  Galil::interrupt() took longer than ... ms to read status byte
        6040 WRONG BUS ERROR.  Galil::interrupt() not supported.  Use Galil::message()
```

top

C++

static string Galil::libraryVersion()

Returns a string containing the library version. This is a static function and does not require a Galil object to be instantiated (C++) or address property to be set (Com). The string returned will resemble: *1.0.0.0 Jan 30 2008 14:32:55 libGalil.so*

C++ Type	Argument	Example	Description
string	return	"0.1.0.4 Aug 20 2008 18:28:39 GalilClass0.dll 1.1.0.4 Aug 20 2008 18:28:19 Galil1.dll"	The library version. COM version (line 1) is also included when using it.

Example 1. Printing the library version

C++

```
cout << Galil::libraryVersion() << endl; //print library version information to console
```

VB

```
statusTextBox.AppendText(g.libraryVersion) 'Printing library version info to text box
```

`Throws:none`

top

C++

vector<string> Galil::sources()

Returns an array of strings corresponding to the names of all available data record sources. A source (e.g. _TPA) is used as a key into sourceValue() to return a source's value from a given record. It is also the key into sourceUnits() and sourceDescription() which return static but useful information about a source's unit of measure and description.

The returned sources array is static for a particular controller and so therefore need only be retrieved once per connection.

The array is returned in alphabetic order.

See the detailed description for more information.

C++ Type	Argument	Example	Description
vector<string>	return	("@AN[1]","@AN[2]","@IN[01]","@IN[02]",...,"_TVB","_ZAA","_ZAB")	An array of strings containing source names in the controller data record.

Example 1. Retrieving sources and assigning a source name to a string variable.

VB

```
Dim s As Object = g.sources() 'create an object to hold all available sources

Dim firstSource As String = s(0) 'create a string to hold the first source name
```

```
MsgBox("The first available source is " + firstSource) 'print the first source name in the data record
```

C#

```
Array s = (Array)g.sources();     //get all sources available on this controller

String firstSource = s.GetValue(0).ToString(); //get the first source name for example
label1.Text = label1.Text + "The first available source is : " + firstSource + "\r\n";
```

C++

```
const vector<string> s = g.sources();//create a vector variable to hold all available sources

const string firstSource = s[0];//create a string to hold the name of the first source
cout << "The first available source is " << firstSource << endl;
```

Example 2. Printing all available sources

VB

```
Dim s As Object = g.sources() 'create an object to hold all available sources

For Each element As String In s  'step through sources and print every source name
    watchTextBox.AppendText(element + vbCrLf)

Next
```

C++

```
vector<string> s = g.sources();  //create a vector to hold all available sources

for (int i = 0; i < s.size() ; i++)  //step through sources and print every source name

   cout << s[i] << endl;
```

```
Throws:

        9190 UNINITIALIZED OBJECT ERROR.  Galil::sources() called without Galil::address set
```

---

top

C++ void Galil::recordsStart(double period_ms = -1)

Issues a DRn command to the controller to begin the asynchronous DR method of data record acquisition. The actual argument sent to the controller depends on the controller, but the fastest supported DR rate is configured by default. If period_ms > 0, the library will attempt to set DR to period_ms milliseconds based upon the controller's servo update rate (TM). If period_ms is unattainable, recordsStart() will set DR to an approximate value, throw an error, or set DR to zero (depending on the value of period_ms).

The standard 18x2 firmware does not support the DR data record method (use "QR" in record()). For stand-alone controllers, DR is valid when connected over Ethernet, but not over RS-232.

A UDP handle will be opened on Ethernet controllers to support the DR data.

Consult the controller command reference under DR for more information.

C++ Type	Argument	Example	Description
double	period_ms	32	The desired period in true milliseconds of the asynchronous data dispatch or -1 for as fast as possible. 0 disables the data record dispatch.

Example 1. Starting the data record with DR and checking the current DR value.

VB

```
g.recordsStart() 'start the records via DR mode

MsgBox("Started records in DR mode.  DR argument is " + g.command("MG_DR")) 'check sample rate
```

C#

```
g.recordsStart(-1); //start data records in DR mode
label1.text = g.command("MG_DR");  //check sampling rate
```

C++

```
g.recordsStart(); //start the records via DR mode
cout << "Started records in DR mode.  DR argument is " << g.command("MG_DR") << endl;  //check current sample rate
```

```
Throws:
        6140 WRONG BUS ERROR.  Galil::recordsStart() isn't allowed via RS-232.  Use Ethernet
        5140 OPEN ERROR.  Galil::recordsStart() failed to open UDP handle on port...
        1010 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to write
        1011 TIMEOUT ERROR.  Galil::command("...") took longer than ... ms to read : response...
        2010 COMMAND ERROR.  Galil::command("...") got ? instead of : response...
        3010 MONITOR ERROR.  Galil::command("...") got > instead of : response.  Got...
        9140 UNINITIALIZED OBJECT ERROR.  Galil::recordsStart() called without Galil::address set
```

top

C++ (QR & DR)

vector<char> Galil::record(string method = "QR")

VB (QR)	**Function record(Optional ByVal method As String = "QR") As Object**
VB Event (DR)	**Sub onRecord(ByVal record As Object)**
C# (QR)	**public virtual object record(string method)**
C# Event (DR)	**void onRecord(object record)**

Returns the controller's data record. If the QR method is specified, this function will perform a synchronous record acquisition by sending QR and waiting for the response (somewhat like command()). If the method is DR, this function will wait for the asynchronous transmission from the DR mode. Note, recordsStart() must be called prior to record() if the DR method is to be used. If the record is not returned within timeout_ms, the function will throw a timeout error.

COM: the onRecord event will occur when a DR record is received. The record() function is also available for the QR method.

C++ Type	Argument	Example	Description
vector<char>	return	*binary data*	record() will return this vector filled with the data record bytes.
string	method	"QR"	A string (either "QR" or "DR") specifying the data record acquisition method to be used.

Example 1. Getting a record() using QR.

VB

```
Dim r As Object  = g.record() 'get a record from the controller using QR (default) method
```

C#

```
Object r = g.record("QR"); //get a record using QR
```

C++

```
vector<char> r = g.record();//get a record from the controller using QR method
```

Example 2. Getting a record() using DR.

```
Public Class Form1
    Dim WithEvents g As Galil.Galil 'dimension Galil object WithEvents enabled

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

VB

```vb
        Try
            g = New Galil.Galil
            g.address = "192.168.1.11" 'connect to Ethernet controller
            Me.Text = g.connection() 'print connection information to form titlebar

            g.recordsStart(100) 'start records at 100 ms period
        Catch ex As System.Runtime.InteropServices.COMException
            TextBox1().AppendText(ex.Message)

        End Try
    End Sub
    Private Sub g_onRecord(ByVal record As Object) Handles g.onRecord 'event to handle record

        TextBox1.AppendText(g.sourceValue(record, "TIME").ToString + vbCrLf) 'print TIME source for each record

    End Sub
End Class
```

C#

```csharp
Galil.Galil g; //galil object variable

private void Form1_Load(object sender, EventArgs e)
{
  try

  {
    g = new Galil.Galil(); //create a new object

    g.address = "192.168.1.11"; //connect to controller
    this.Text = g.connection(); //print connection in form titlebar

    g.onRecord += new Galil.Events_onRecordEventHandler(g_onRecord); //hook up to event

    g.recordsStart(100); //start records at 100 ms period
  }catch(System.Runtime.InteropServices.COMException ex){ //catch any errors

    textBox1.AppendText(ex.Message);
  }//try
}//Form1_Load
void g_onRecord(object record) //event runs when record received

{
  textBox1.AppendText(g.sourceValue(record, "TIME").ToString() + "\r\n"); //print time

}
```

C++

```
g.recordsStart(); //start the asynchronous DR mode
vector <char> record = g.record("DR"); //get a record via DR
```

```
Throws:
       6150 WRONG BUS ERROR.  Galil::record("DR") not supported on RS-232.  Use Ethernet or Galil::record("QR")
       1150 TIMEOUT ERROR.  Galil::record("DR") took longer than ... ms to read ... bytes
       1151 TIMEOUT ERROR.  Galil::record("DR") took longer than ... ms to read ... bytes
       1152 TIMEOUT ERROR.  Galil::record("QR") took longer than ... ms to write
       1153 TIMEOUT ERROR.  Galil::record("QR") took longer than ... ms to read ... bytes.  Got ... bytes
       9150 UNINITIALIZED OBJECT ERROR.  Galil::record() called without Galil::address set
```

top

C++ double Galil::sourceValue(vector<char> record, string source = "TIME")

Maps a data record source name to its value. Given a data record returned from record() and a valid source name as a key, this function returns the appropriate value.

If an invalid source name is specified to sourceValue(), the value of TIME is returned.

C++ Type	Argument	Example	Description
double	return	6.4306	The value returned corresponding to source
vector< char>	record	*binary data*	A vector (as returned from record()) containing one valid controller data record
string	source	"@AN[1]"	A valid string as found in the sources() array specifying the desired data

Example 1. Returning the TIME value from a data record buffer.

VB

```
Dim r As Object = g.record("QR") 'Get a record.  In this example, the QR method is used

Label3.Text = g.sourceValue(r, "TIME").ToString 'Print the current TIME value into a form label
```

Example 2. Display the value of the first source of the data record.

C#

```
label1.Text = label1.Text + firstSource + " value is: " + g.sourceValue(r, firstSource) + "\r\n";
```

C++

```
cout << firstSource << " value is: " << g.sourceValue(r , firstSource) << endl; //display value
```

```
Throws:
        9160 UNINITIALIZED OBJECT ERROR.  Galil::sourceValue() called without Galil::address set
```

top

C++ string Galil::source(string field = "Description", string source = "TIME")

Provides access to source related fields. By specifing a valid source field name and source, the field's value is returned.

C++ Type	Argument	Example	Description
string	return	"Sample counter"	The value returned corresponding to the specified source's specified field
string	field	*"Description"*	The field of interest for a specified source. Valid fields are "Description", "Units", "Scale", & "Offset"
string	source	"TIME"	A valid source as found in the sources() array specifying the source in which to lookup field

Example 1. Printing all fields of a source.

VB

```
TextBox1.AppendText(g.source("Description", "@AN[1]") + vbCrLf)
TextBox1.AppendText(g.source("Units", "@AN[1]") + vbCrLf)
TextBox1.AppendText(g.source("Scale", "@AN[1]") + vbCrLf)
TextBox1.AppendText(g.source("Offset", "@AN[1]") + vbCrLf)
'Prints:
'Analog input 1
'V
'3276.8
'0
```

```
Throws:
        9170 OFFLINE ERROR.  Galil::source() called without Galil::address set
```

top

C++ void Galil::setSource(string field = "Description", string source="TIME", string to= "Sample counter")

Used to set the various fields of a data record source to custom values instead of Galil factory defaults.

C++ Type	Argument	Example	Description
string	field	*"Description"*	The field to set for a specified source. Valid fields are "Description", "Units", "Scale", & "Offset" This argument is overloaded to take a file path to a GalilTools project file. This allows loading data record configurations created from the GalilTools Watch-All tab. Leave the rest of the arguments blank.
string	source	"TIME"	A valid source as found in the sources() array specifying the source in which to change field
string	to	"16 bit upcounter"	The value to assign to the specified field in the specified source

Changing the Scale and Offset fields will affect the values returned from sourceValue(). These fields are used for pre-processing the raw controller data record before being passed up to the API. Set scale=1 and offset=0 to receive the raw controller data record values from sourceValue(). See the Watch chapter for more information.

Example 1. Modifying fields of a source.

```
'print default values
TextBox1.AppendText("@AN[1]=" + g.sourceValue(g.record("QR"), "@AN[1]").ToString)
TextBox1.AppendText(" " + g.source("Units", "@AN[1]"))
TextBox1.AppendText(" (" + g.source("Description", "@AN[1]"))
TextBox1.AppendText(" with scale " + g.source("Scale", "@AN[1]"))
TextBox1.AppendText(" and offset " + g.source("Offset", "@AN[1]") + ")" + vbCrLf)
'change fields
g.setSource("Description", "@AN[1]", "RAW ADC Counts")
g.setSource("Units", "@AN[1]", "counts(signed 16bit)")
```

VB

```vb
g.setSource("Scale", "@AN[1]", "1")
'print new values
TextBox1.AppendText("@AN[1]=" + g.sourceValue(g.record("QR"), "@AN[1]").ToString)
TextBox1.AppendText(" " + g.source("Units", "@AN[1]"))
TextBox1.AppendText(" (" + g.source("Description", "@AN[1]"))
TextBox1.AppendText(" with scale " + g.source("Scale", "@AN[1]"))
TextBox1.AppendText(" and offset " + g.source("Offset", "@AN[1]") + ")")
'prints, depending on voltage on analog input one
'@AN[1]=-9.248046875 V (Analog input 1 with scale 3276.8 and offset 0)
'@AN[1]=-30256 counts(signed 16bit) (RAW ADC Counts with scale 1 and offset 0)
```

Example 2. Loading a GalilTools project file.

VB

```vb
g.setSource("C:\Documents and Settings\user\Galil\default.project")
```

```
Throws:
        9150 OFFLINE ERROR.  Galil::record() called without Galil::address set
```

Contents