

**QWDataHub** is a centralized repository of current plant data. It was created to overcome several obstacles in the realm of automated data collection:

- 1- Capturing live data from DDE and OPC I/O Servers efficiently
- 2- Being able to sustain short outages in I/O servers by caching data
- 3- Being able to cross network boundaries by using a TCP gateway
- 4- Being able to reference meaningful Tag Aliases instead of cryptic PLC tags
- 5- Being able to bridge OPC, DDE, SQL, QW IO files, iFix, and Scripted tags

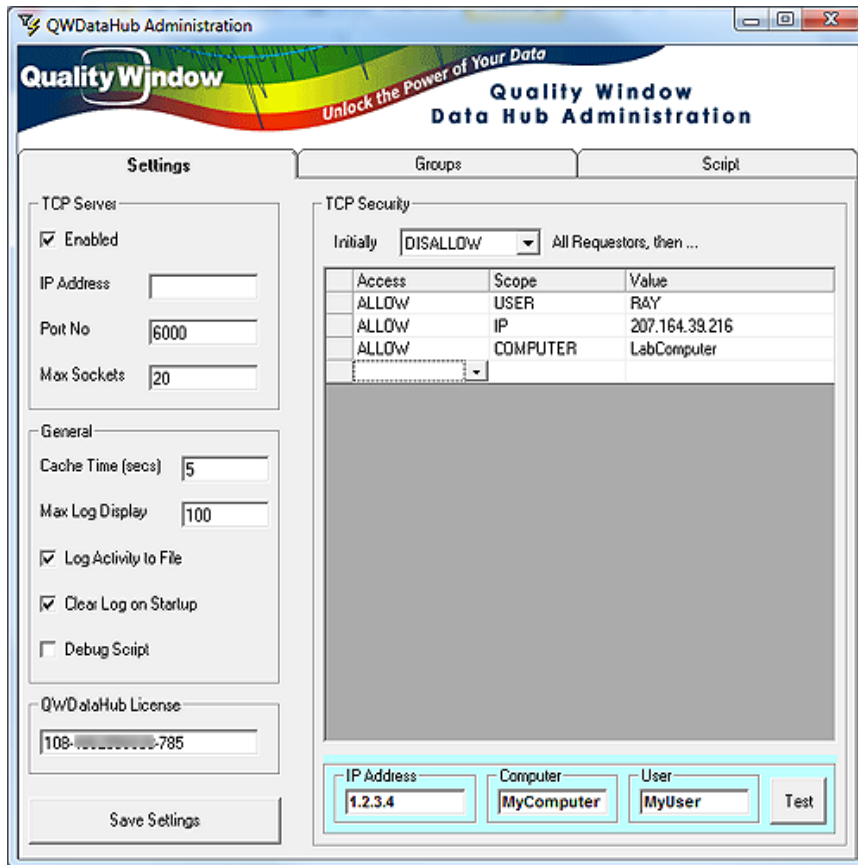
QWDataHub is managed and configured by a separate program called QWDataHubAdmin, which allows you to create a list of I/O Servers you wish to work with along with many tags you wish to share data with. The list is first broken down into manageable Groups, where a group can be an association with a given OPC Server, DDE Server, SQL database, iFix Gateway, List of QW I/O files, or calculated tags derived via scripting. Once a group is defined, a list of actual Tag Names is defined, along with a Tag Alias name, as well as an optional Tag Description.

Once you have all your Groups and Tags defined, launching QWDataHub will first attempt to connect to each of the servers and start gathering the current values of each of their tags, which in turn will be accessible to any Quality Window application that needs the values.

Note that QWDataHub runs as a tray application, so closing the window only hides the display. Right-clicking on the QWDataHub tray icon brings up a menu that allows you to shut down the program. The following are the menu items on the tray icon:

- Show QWDataHub** - Display QWDataHub Screen
- Clear Log File** - Clears the current log file
- QWDataHub Administration** - Opens the Administration program
- QWDataHub Documentation** - Display Help information
- Shutdown QWDataHub** - Terminate the Data Hub

The following describes the administration process using **QWDataHubAdmin**:



## TCP Server Settings

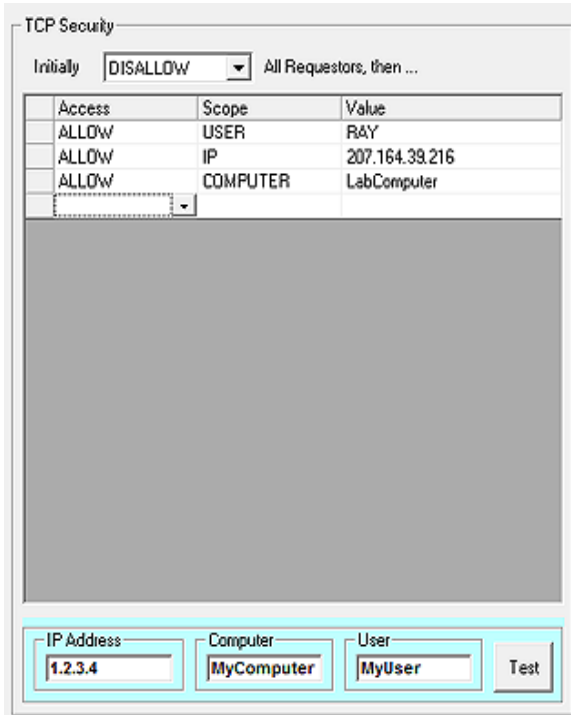
- Enabled:** Setting TCP Server to Enabled allows outside computers to communicate with QWDataHub via a TCP Gateway. This can be useful if your I/O servers are connected to QWDataHub, and you have Quality Window clients that need to collect information from QWDataHub remotely.
- IP Address:** If you have multiple network adapters on the computer running QWDataHub, you can specify a specific IP address to listen on. If left blank, QWDataHub will use the IP address of the first network adapter it finds. In most cases, you can leave this blank.
- Port No:** This is the specific TCP Port number that QWDataHub will listen on. Choose a port that is not used and make sure that firewalls have this port unblocked. The default is port 6000.
- Max Sockets:** This is the maximum number of TCP sockets to reserve for QWDataHub. It is directly related to the maximum number of concurrent users that can communicate with QWDataHub. The default is 20 sockets.

## General Settings

- Cache Time (secs):** When an I/O Server is having communication problems, or tags have a Bad quality, QWDataHub can be configured to overcome these temporary outages by defining a Cache Time. If you set the Cache Time to 5 seconds, then any clients attempting to read the value of an offline tag will get the last known good value. If the tag does not return a good value in the allotted Cache Time, then a null value will be returned. Setting the Cache Time to 0 disables the cache, and a setting of -1 sets the cache to infinity.
- Max Log Display:** QWDataHub has a mini Log display screen to show external activity. Max Log Display sets the maximum number of activity records to display in the grid. Setting this value too high can have drastic effects on performance. Note that this has no effect on the physical log file being written.
- Log Activity to File:** When enabled, each activity record will be written to a file called QWDataHub.LOG in the same folder as the executable.
- Clear Log/ Startup:** When enabled, the activity log file will be deleted every time QWDataHub is started. If not enabled, you should monitor the size of the log file so as not to waste too much hard drive space.
- Debug Script:** When enabled, if any scripting errors occur, a message will be displayed to the user. Turn this option ON if you've just created or modified a script so that you can determine the cause of error, otherwise turn this option OFF if you plan to run QWDataHub unattended and are confident that your script is working fine.
- QWDataHub License** Each running copy of QWDataHub requires a license number to operate. When you purchased QWDataHub you would have received this license number by email. The license number begins with '108-'. Enter your license number here making sure to include the hyphens.

## TCP Security

The TCP Gateway in QWDataHub allows you to define who should or should not be able to access the data. This can be based on their IP address, their logged-on User Name, or their Netbios computer name.



Access	Scope	Value
ALLOW	USER	RAY
ALLOW	IP	207.164.39.216
ALLOW	COMPUTER	LabComputer

IP Address: 1.2.3.4    Computer: MyComputer    User: MyUser    Test

You must first define the Initial Setting as to whether you want to Initially Allow everyone and then start adding exceptions to disallow – or Initially Disallow everyone, and then start adding the exceptions to allow. Note that the statements are processed top down, and every statement is executed – the result at the end of processing determines whether a particular user will have access or not. Note that you can enter a '0' in any one of the segments of an IP address to act as a wildcard.

The bottom portion of the screen (in blue) is a testing area. Enter a user's IP Address, Computer Name and User Name and then click on the **Test** button to determine, based on the table definition above, whether or not that user would be allowed access.

## Save Settings

The Save Settings button will save the contents of Settings, Groups and Script screens. These new settings will not take effect until QWDataHub is restarted. Note that Settings and Groups are saved to a file called QWDataHub.INI and the Script is saved to a file called QWDataHub.QWX

## Managing Groups and Items

QWDataHub allows you to organize items into what's called Groups. A group is a list of related items that stem from the same IO Server, and that would share the same Update Rate or Topic Name.



**Add Group**                      Add a new Group to the list

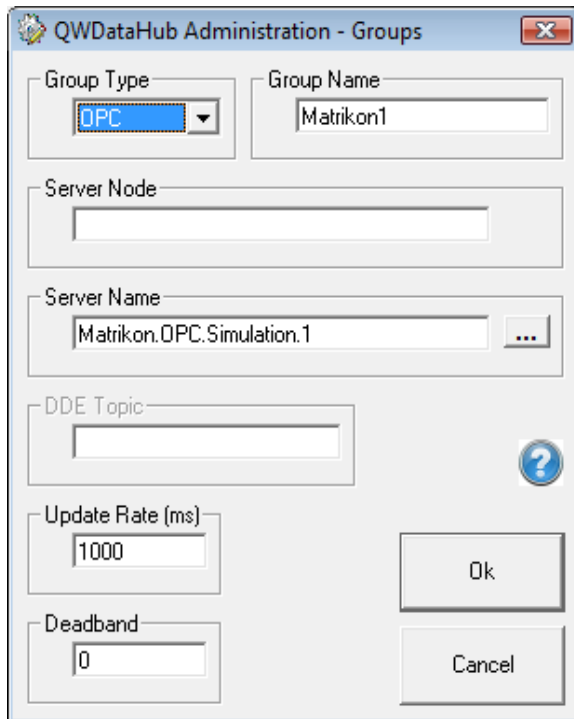
**Edit Group**                      Edit an existing Group

**Delete Group**                    Delete an existing Group

**Manage Items**                    Manage Items within a Group

**Copy Items to Clipboard** - Copies all Groups and Items to the Clipboard You can later Paste in Excel.

Click on the **Add Group** button to create a new group.



**Group Type:** QWDataHub supports several group types, each of which have their own list of settings.

**OPC:** Connect to an existing OPC server

**DDE:** Connect to an existing DDE server

**SQL:** Connect to an SQL database

**FILE:** Access Quality Window IO files

**FIX:** Connect to an existing iFix I/O server

**CALC:** Define calculated Tags based on other tags

**SCRIPT:** Define calculated Tags using Scripting

**Group Name:** Enter a unique, descriptive name for the group

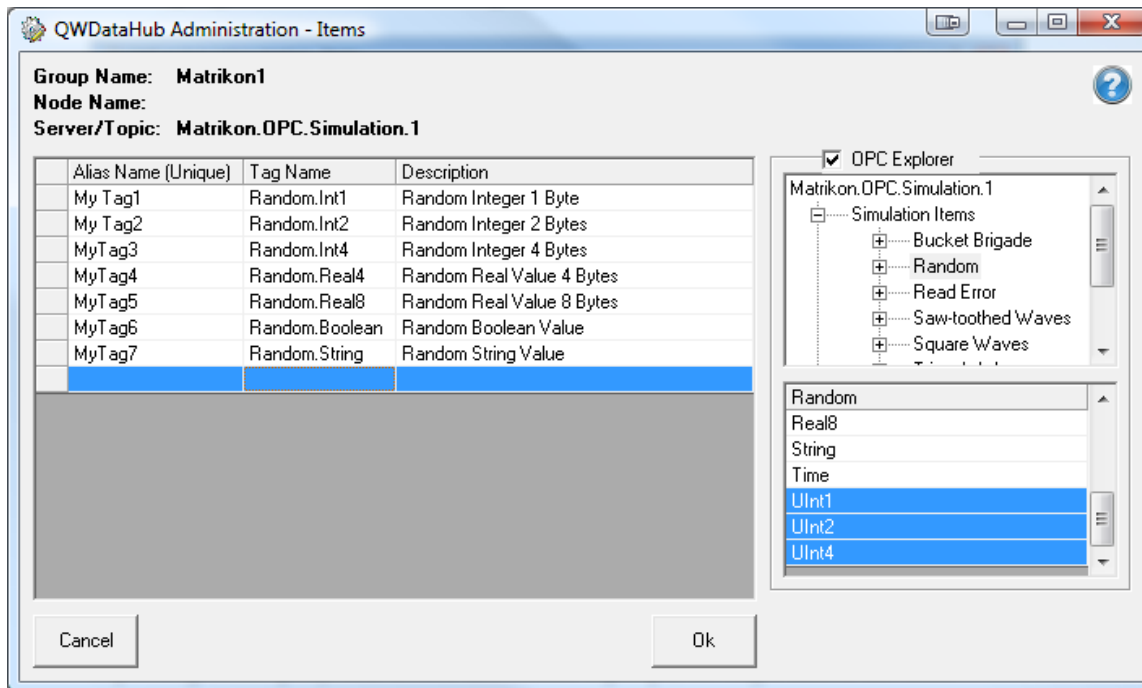
**Server Node:** If the OPC or DDE I/O server is located on a remote computer, enter the Node name of the server here. If the server is local to this computer, leave this entry blank.

**Server Name:** Enter the OPC or DDE server name here. Typically, this is either the executable name or the class name of the server. For OPC Servers, click on the '...' button to browse all existing OPC servers on the Node specified.

- DDE Topic:** DDE servers typically have their tags grouped by what's called a Topic. Enter the DDE Topic Name here.
- Update Rate:** Each group type supports an Update Rate expressed in milliseconds. For OPC and DDE, this is typically the maximum elapsed time to check in with the I/O server for changes. For types FILE, SQL and FIX, this is the time interval to either collect the SQL, iFix data, or to check the QW IO files for changes. For types CALC and SCRIPT, this is the interval at which the script will be called. Setting this value too low could result in degradation of the server, and setting it too high could allow collection of stale data. Note that for a given server, you could split the items into 2 groups where tags that may be used later as triggers could have a faster refresh rate, and tags that do not change that often could be on a slower refresh rate.
- Deadband:** Some OPC servers support what's called a Deadband. This deadband is expressed as a percentage, meaning only pick up data that have changed a certain percentage. The valid range is 0-100%, where 0 means to send back any value that has changed, basically disabling deadband.
- Enabled:** You can set one or more groups to be disabled, meaning they will not be scanned for values. This can be handy if you are adding large groups of tags and want to start testing one group at a time.

Once you have your Group defined, press **Ok** to save the group. You are now ready to add items (Tags) to your group by clicking on the **Manage Items** button:

## Managing Items within a Group:



For each item in the group, enter a meaningful **Tag Alias Name**, an external **Tag Name**, along with an optional **Description** of the Tag. Note that for OPC, DDE and FIX groups, this is the PLC Tag Name. For FILE groups, this is the full path of the Quality Window IO file. For SQL groups, this is either the Database Field Names or Field Values depending on the query. For CALC and SCRIPT groups, the Tag Name is not used.

For OPC groups, a sidebar will appear showing all available Tags for the server selected. Check the **OPC Explorer** box to display the list. The folder tree is displayed at the top, and by clicking any folder, available tags are displayed below. (it may take some time if you have a lot of tags defined in a single folder). You can either double click a Tag Name to select it and add it to your list of tags, or highlight a group of them, right-click and then choose Select Item(s) and they will appear in your list of tags.

You can move items around in the list by clicking on the grey square to the left of the Alias Name, and drag and drop to the new location. Right-clicking on an item presents a popup menu allowing you to:

- Cut**                      Cut the selected item(s) to the clipboard
- Copy**                     Copy the selected item(s) to the clipboard
- Paste**                    Paste one or more items from the clipboard
- Delete**                  Delete the selected item(s)
- Insert**                    Insert a new blank row before the current item



**Duplicate** Duplicate the current item

**Select All Items** Select all items in the list

Note that Alias Names cannot contain the characters: ‘,’ or ‘~’.

Tag Names and Description cannot contain the character: ‘~’.

Tip: You can create an alias naming convention to logically group your items:

Line 1.Winder.Motor Speed

Line1.Winder.Temperature

Line 2.Winder.Motor Speed

Line 2.Winder Temperature

Although you can name your aliases to whatever you like, by creating a naming convention, it will be much easier for you to manage and for your users to identify.

Note that you can also use MS Excel to manage your items. In QWDataHubAdmin, right-click on a Tag Alias and then Select All Items. Then right-click again and choose Copy. Then go to Excel and choose Paste, to paste in all items. You can then use Excel to manipulate items, and then select them, right-click Copy, and then go back to QWDataHubAdmin and Paste them back in.

## Setting up SQL Groups

QWDataHub allows you to create SQL groups, whereby a database query will be performed on a timed basis, and the resulting Values would be available as Tags in QWDataHub. Note that the results of queries should be a single record of values, as QWDataHub can only save a single value for Tag. If more than one record is returned from the query, then only the last one in the recordset will be used. For each query, a properly formatted Connection String must be entered, an SQL statement, as well as a list of individual Tag Names.

There are 2 types of queries that can be performed:

- 1- A query where each field name in the result becomes a Tag and its value
- 2- A query where a single field returns multiple Tag names and Values

### 1- Traditional Query:

```
SELECT [Weight1], [Weight2], [Weight3] FROM [LIVE]
```

Assuming the table [LIVE] only contains current values, the result from this query would return a single record with 3 fields and their corresponding values. If you created 3 Tag Names for the group, called Weight1, Weight2, Weight3 then after the query executes, those 3 tags/values would be exposed in QWDataHub, since the tag names match the field names in the recordset. Doing a `SELECT * FROM [LIVE]` would have given you the same result, since it would only do a match from the tag names defined. You can also include the special shortcut `%TAGLIST%` into the query, such as:

```
SELECT %TAGLIST% FROM [LIVE]
```

At run time, `%TAGLIST%` would be replaced with a list of the Tag names you have defined for the group. This not only simplifies the query, but makes it easier to maintain.

### 2- Transposed Query:

```
TRANSPOSE [TagName],[Value] FROM [LIVE]
```

```
WHERE [TagName] IN ('Weight1','Weight2','Weight3') ORDER BY [TagName]
```

Assuming [LIVE] only contains current values, the result from this query would return 3 records – one for each TagName/Value pair. Again, you can use the shortcut `%TAGLIST%` instead of listing all of the individual tags in the query.

```
TRANSPOSE [TagName],[Value] FROM [LIVE]
```

```
WHERE [TagName] IN (%TAGLIST%) ORDER BY [TagName]
```

By using the keyword TRANSPOSE instead of SELECT in the query, QWDataHub treats this query differently. It will transpose the records, making each tag name a field name, so that the result looks like:

Weight1	Weight2	Weight3
123	456	789

After transposing, any Field Name that matches a corresponding Tag Name in the group's list, will get assigned its corresponding value.

Note that if you expect more than one time stamped record from the query, you may have to use the following syntax:

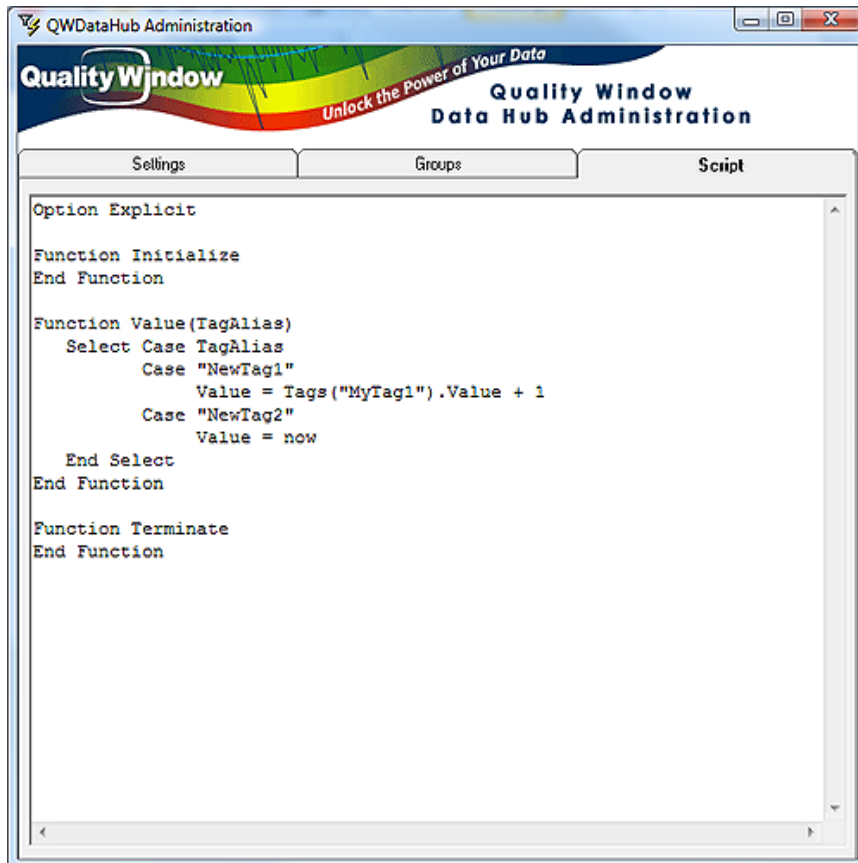
```
TRANSPOSE [Timestamp],[TagName],[Value] FROM [LIVE]  
  
WHERE [TagName] IN (%TAGLIST%) ORDER BY [Timestamp],[TagName]
```

This will ensure that you get the latest values of each of the tags.

You may have to consult your database administrator for the proper syntax for the Connection String, or you can use QWSQLWizard included with QW 5.0 to derive it.

## Calculated Tags in QWDataHub

The Script screen allows you define scripting code for items defined as CALC or SCRIPT. If you do not have a Group Type defined as CALC or SCRIPT, then you can ignore this screen.



Basically, how calculated tags works is as follows:

When QWDataHub first starts up, it calls an entry point in your script called **Initialize**.

When QWDataHub shuts down, it calls an entry point called **Terminate**.

For each item in your CALC group, it calls a function called **Value** passing the TagAlias name to be processed. This function expects you to do anything you like and end up returning a single value to set that particular tag to. During this function, you have full access to all other tags within QWDataHub, making it easy to create calculations based on other tag values. Ideas include converting units of a tag, conditional tag values (if line speed is below 100 then clear a value), if a tag value has changed, buffer some related tag values, etc...

**Note:** that you should test the tag values prior to using them in calculations

```
If Tags("MyTag1").Value <> "" Then ...  
    If IsNumeric(Tags("MyTag1").Value) Then ...
```

Within scripting, you have access to a class object called Tags() that have the following properties:

Tags(TagAlias).Value  
Tags(TagAlias).TagName  
Tags(TagAlias).TagAlias  
Tags(TagAlias).Description

You can also access the Tags class as a collection:

Tags.Count  
Tags(l).TagAlias

Note that the Tags class should be treated as Read-only.

For CALC Tags, if you return "NOCHANGE" as the value of a tag, then the value currently in memory will be retained. This can be handy if an outside process is writing values to this tag and you want to maintain the value. This can be shown as follows:

Option Explicit

Function Value(TagAlias)

Value = ""

Select Case TagAlias

Case "NewTag1"

If Tags("MyTag1").Value <> "" Then

If IsNumeric(Tags("MyTag1").Value) Then

Value = Tags("MyTag1").Value + 1

End If

End If

Case "NewTag2"

If Tags("MyTag2").Value <> "" Then

If IsNumeric(Tags("MyTag2").Value) Then

If Tags("MyTag2").Value < 100 Then

Value = 0

Else

Value = Tags("MyTag2").Value

End If

End If

End If

Case Else

Value = "NOCHANGE"

End Select

End Function

## SCRIPT Tags in QWDataHub

The Script screen also allows you to define SCRIPT Tags, where the execution of the script will determine the results of the tags in that group. SCRIPT tags and CALC tags are almost identical in that they both return values to their tags, but the differences are that CALC tags are executed inline and have access to ALL other tags, and SCRIPT tags are executed on a separate thread and only have access to the tags in its own group.

There is no real limit as to what you can do in a SCRIPT group in terms of the scripting code, since it runs in its own thread, it will not degrade the performance of QWDataHub. However, the script you write for CALC items should be very lean and efficient as it could have an impact on performance. The script you use for both CALC and SCRIPT groups are contained in the same file.

Each time the Update Rate of the SCRIPT group has been reached, QWDataHub calls the function Process with the name of the Group as a parameter:

```
Dim LastValue
Function Process(Group Name)
  Select Case Group Name
    Case "MyGroup"
      LastValue = LastValue + 1
      Tags("sTag1").Value = LastValue
      Tags("sTag2").Value = Now
    End Select
  End Function
```

As an example, your script could go out and query one or more databases, and return some of the results back to the tags defined in your script group. You can have more than one SCRIPT group, each with different Update Rates.

## Accessing QWDataHub externally

QWDataHub is simply a data repository and requires external programs to communicate with it to actually collect and store data.

In Quality Window, any Variable within an application can have an InputFile specified, so that its value can be derived when the Add key is pressed. InputFiles can have the following format:

```
HUB:TagAliasName          ie HUB:MyTag1
or
TCP:IPAddress:PortNo!TagAliasName  ie TCP:207.164.39.216:6000!MyTag1
```

You can also use a DNS name in place of the IP address, and port 6000 is the default, so you need not have to specify it.

TCP:MyDataHub!MyTag1

QWSchedule can also reference tags in the same format as above.

You can also access QWDataHub directly via scripting as follows:

```
Function GetValue(TagAlias)
    Dim oHub, Result
    On Error Resume Next
    Result = ""
    Set oHub = GetObject("QWDataHubProxy.Application")
    If Not oHub Is Nothing Then
        If oHub.IsConnected = True Then
            Result = oHub.Send("GetValue", TagAlias)
        End If
        Set oHub = Nothing
    End If
    GetValue = Result
End Function
```

Or

```
Dim oComSrv, Value
Set oComSrv = CreateObject("QWComSrv.QWComSrvClass")
oComSrv.Silent = True
Value = oComSrv.ReadInputFile("HUB:MyTag1")
Value = oComSrv.ReadInputFile("TCP:1.2.3.4:6000!MyTag1")
oComSrv.WriteOutputFile "HUB:MyTag1", Value
Set oComSrv = Nothing
```

## System Requirements for QWDataHub:

- Runs under Windows 7,10 Windows Server 2003,2008,2012,2016
- Minimum of 1GB RAM  
(2GB or more recommended)
- Static IP address accessible to all QWDataHub clients (if using TCP Gateway)
- Requires 1 TCP Port (default port set to 6000 on install)  
Can be changed as required  
  
Selected port must be allowed in Firewall settings
- Default of 20 concurrent users (sockets)  
Number of sockets can be increased/decreased as required
- Less than 100MB of disk space required by server for operation
- Requires access to any I/O servers defined

**Note: Clients require licensed copy of QW 5.0 (Build 800 or higher) to communicate with QWDataHub.**