

QualityWindow

The continuous improvement software



QW DataHub

Administration Guide

May 2025

Table of Contents

Overview	5
Why Use QW DataHub?	5
Built for Flexibility	5
Standards and Protocols Supported	6
OPC – Classic and UA	6
DDE (Dynamic Data Exchange)	6
SQL (Structured Query Language)	6
Quality Window I/O Files	6
iFIX Gateway	7
Calc and Script	7
System Requirements	7
Supported Operating Systems	7
Minimum Hardware Requirements	7
Network Requirements	7
Installation	8
Installation Steps	8
Post-Installation	8
Configuring QW DataHub	8
Settings	9
TCP Server	9
General Settings	10
QW DataHub License	10
TCP Security	10
Saving Settings	11
Groups	11
Core Capabilities	12
Groups List Display	12
Script	13
Validating QW DataHub Connectivity	14
What You Should See After Installation	14
How to Perform a Connectivity Test	15
Troubleshooting connectivity	16

Supported Group Types.....	16
OPC UA Group.....	16
Configuration Fields	16
Manage Items	17
OPC Classic Group.....	18
Configuration Fields	18
Manage Items	18
DDE Group	19
Configuration Fields	19
Manage Items	19
FIX Group	20
Configuration Fields	20
Requirements	20
Manage Items	20
SQL Group	21
Configuration Fields	21
How to use this type.....	21
Requirements	22
Manage Items	24
FILE Group	24
Configuration Fields	24
Manage Items	25
CALC Group	25
Configuration Fields	25
Requirements	26
Manage Items	26
Script Structure	26
Key Differences from SCRIPT Groups	27
SCRIPT Group.....	27
Configuration Fields	27
Requirements	27
Manage Items	28
Script Structure	28

QW DataHub - Administration Guide

Access Scope	29
Best Practices	29
Key Differences from CALC Groups	29
Using QW DataHub	29
Data Hub Tab	29
Activity Log Tab	31
Integrating with QW DataHub	32
1. Using QW DataHub Tags in Quality Window Admin	32
2. Accessing QW DataHub from Scripts	33
Option 1: Using QWDataHubProxy (local COM object)	33
Option 2: Using QWComSrv for file-style access	33
3. Referencing Tags in QW Schedule	34
Appendix A: Troubleshooting	35
1. Tag Value Is Blank or Missing in Quality Window	35
2. Remote Connection Fails (TCP Access)	35
3. Script Fails to Retrieve Value	36
4. Scripted or Calculated Tag Shows "NOCHANGE"	36
Appendix B – Script Example	38
Full Example Code	38

Overview

QWDataHub is a real-time data integration and distribution engine designed specifically for manufacturing environments. It serves as a centralized data broker between a plant's various automation systems and Busitech's Quality Window software, making it easy to collect, manage, and share live process data across your facility.

At its core, QW DataHub connects to multiple data sources—like PLCs, OPC and DDE servers, SQL databases, and plant historian systems—and makes that information available to any Quality Window application in a clean, consistent, and meaningful way. This allows operators, engineers, and quality teams to make faster, more informed decisions based on current process conditions.

Why Use QW DataHub?

Manufacturing data often lives in fragmented systems that don't talk to each other well. QW DataHub simplifies this by:

- **Collecting live process data** from diverse industrial data sources
- **Providing alias-based access** to complex tag names so that users work with meaningful, human-readable identifiers
- **Bridging communication gaps**, such as network boundaries or protocol differences, using a built-in TCP gateway
- **Cushioning short-term outages** by caching the most recent good values of any given tag
- **Normalizing access** to both live and calculated data, including values generated by custom scripts or derived from databases

This data hub ensures that your Quality Window applications always have access to the most up-to-date plant information—without relying on rigid network configurations or cryptic tag structures.

Built for Flexibility

QWDataHub is highly configurable through its companion administration tool, **QW DataHub Admin**, which allows users to:

- Group and organize tags by data source
- Define update intervals for each connection
- Manage alias naming for readability
- Secure remote access via IP- or user-based filters
- Build calculated tags using script-based logic

Whether you're managing a single production line or coordinating data across multiple facilities, QW DataHub streamlines the flow of production data into the hands of people who need it.

Standards and Protocols Supported

QW DataHub is designed to integrate seamlessly with a wide range of industrial data sources, enabling smooth communication between manufacturing equipment, automation systems, and Quality Window applications. By supporting industry-standard protocols—both modern and legacy—it ensures manufacturers can unify data from diverse systems into a single, consistent interface.

OPC – Classic and UA

What it is: OPC is a family of standards that enables data exchange between control devices and software systems. QW DataHub supports both:

- **OPC Classic (OPC DA):** A COM/DCOM-based protocol, widely used in legacy Windows environments.
- **OPC UA (Unified Architecture):** A newer, platform-independent standard designed for secure, scalable, and firewall-friendly industrial communication.

Why it matters: With support for both Classic and UA, QW DataHub can connect to virtually any PLC, SCADA, or DCS system—regardless of age or vendor.

More info: [OPC Foundation](#)

DDE (Dynamic Data Exchange)

What it is: A Microsoft protocol used for sharing data between applications such as Excel and SCADA software.

Why it matters: DDE is still used in many older systems, including GE iFIX and legacy reporting tools. QW DataHub maintains compatibility with these systems for uninterrupted data flow.

More info: [Microsoft DDE Overview](#)

SQL (Structured Query Language)

What it is: The standard language for querying and manipulating relational databases.

Why it matters: QW DataHub can perform SQL queries on a timed basis and expose query results as tag values, making it easy to blend production data with information from MES, LIMS, or ERP systems.

More info: [W3Schools SQL Tutorial](#)

Quality Window I/O Files

What it is: File-based data used natively by Quality Window applications to store manually entered or automated values.

Why it matters: QW DataHub can read directly from these files, allowing Quality Window clients to access the most up-to-date values from other systems or previous operator entries.

More info: [Quality Window Knowledge Base](#)

iFIX Gateway

What it is: A SCADA platform from GE Digital, commonly used in industrial process control.

Why it matters: QW DataHub can connect to the iFIX COM API to pull tag values.

More info: [GE Digital iFIX](#)

Calc and Script

What it is: Virtual tags generated by logic or script-based calculations within QW DataHub.

Why it matters: These allow for real-time creation of derived values—such as conditional alerts, unit conversions, or statistical calculations—using data already available in the system.

More info: See [Calculated](#) and [Script](#) Group Types below from more details.

System Requirements

QW DataHub is a lightweight application and can run on most modern Windows environments. The following specifications represent the minimum requirements based on the supported operating systems.

Supported Operating Systems

- Windows 10 & 11
- Windows Server 2019,2022

Note: Older operating systems may function but are not officially supported.

Minimum Hardware Requirements

- 1.0 GHz dual-core processor
- 4 GB RAM
- 1 GB available disk space

Network Requirements

- Static IP address recommended for remote access
- One open TCP port (default: 6000)

- Firewall must allow inbound connections on the selected port
- Supports 20 concurrent clients by default (configurable)

Installation

Installing QW DataHub is a quick process and typically completed in just a few minutes. The application is distributed as a Windows installer package (.msi), available from Busitech upon purchase or through your maintenance agreement.

Installation Steps

1. **Run the installer**
Double-click the provided QWDataHub.msi file and follow the on-screen instructions.
2. **Accept the license agreement**
Review and accept the end-user license terms to proceed.
3. **Choose installation location**
The default path is usually acceptable, but you may specify a custom directory if needed.
4. **Finish installation**
Once complete, QW DataHub and QW DataHub Admin will be available in your Start menu.

Post-Installation

- **Launch QW DataHub Admin** to begin configuring your Groups and Tags.
- **Ensure your license key** is entered when prompted. The license begins with 108- and must include hyphens.
- **Confirm firewall settings** allow access to the selected TCP port if remote connections are required.

Note: QW DataHub runs as a background application. You can manage it via its system tray icon.

Configuring QW DataHub

QW DataHub is configured using **QW DataHub Admin**, which includes three main tabs:

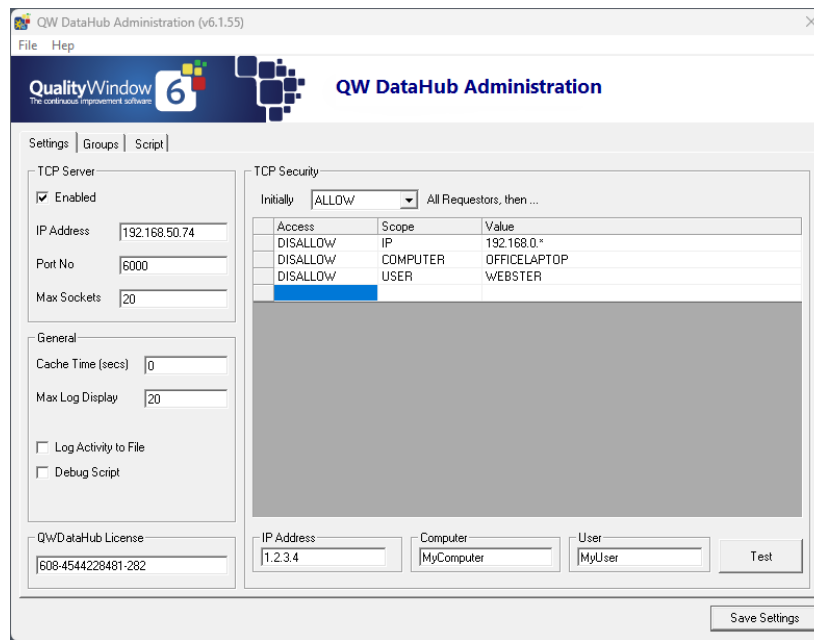
- **Settings** – Controls system-level behavior such as TCP access, logging, caching, and license entry.
- **Groups** – Defines the data sources (OPC, DDE, SQL, etc.) and the tags you want to collect and share.
- **Script** – Allows you to create calculated or logic-based tags using built-in scripting.

QW DataHub - Administration Guide

Each tab plays a key role in how QW DataHub collects, processes, and shares live data across your plant systems.

Settings

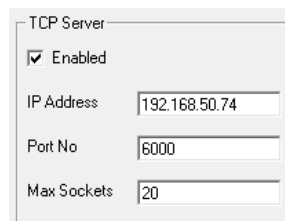
The **Settings** tab in QW DataHub Admin controls the core behavior of QW DataHub, including TCP communication, logging, security, and licensing. This is typically the first area to configure after installation.



QW DataHub Admin Settings Tab

TCP Server

These settings control how QW DataHub communicates with remote clients, such as Quality Window applications running on other machines.



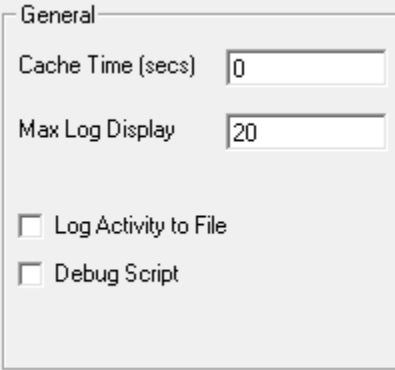
QW DataHub TCP Settings

- **Enabled:** Check this box to allow QW DataHub to accept incoming TCP connections.
- **IP Address:** Enter the IP address of the local machine you want QW DataHub to listen on. Leave blank to use the default adapter.
- **Port No:** Specifies the port used for TCP communication (default: 6000). Ensure this port is open on your firewall.

- **Max Sockets:** Sets the maximum number of concurrent client connections. The default is 20, but this can be increased for high-use environments.

General Settings

These control basic behavior such as caching, logging, and debugging.



QW DataHub General Settings

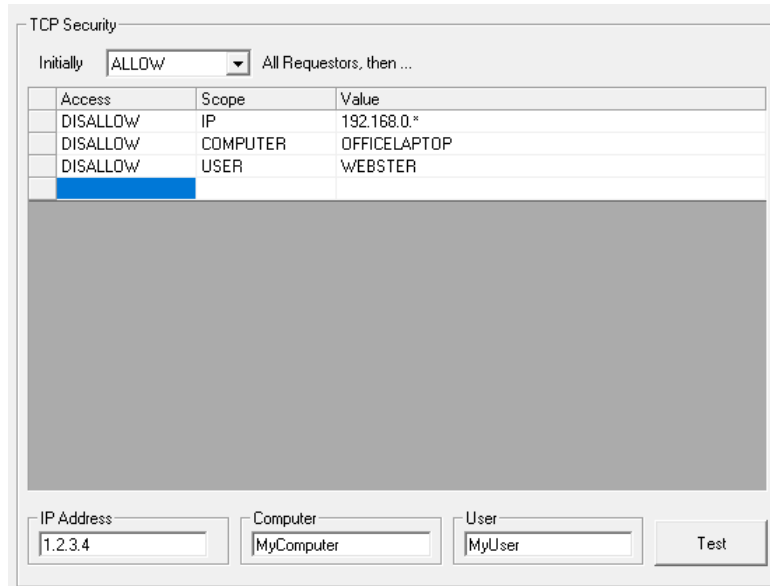
- **Cache Time (secs):** Specifies how long to keep the last known value for a tag if a data source is temporarily unavailable.
 - 0 = no caching
 - -1 = cache indefinitely
 - Example: A value of 5 allows QW DataHub to serve the last known value for up to 5 seconds if the live source is unreachable.
- **Max Log Display:** Controls how many log entries are shown in the in-app log viewer. A higher number may reduce performance.
- **Log Activity to File:** When enabled, all activity is written to QWDataHub.LOG.
- **Debug Script:** When enabled, any scripting errors will generate pop-up messages. Recommended only while developing or testing scripts.

QW DataHub License

Enter your license key here to activate QW DataHub. The license format begins with 608- and must include the hyphens.

TCP Security

This section controls which computers or users are allowed to connect to QW DataHub when TCP access is enabled.



Access	Scope	Value
DISALLOW	IP	192.168.0.*
DISALLOW	COMPUTER	OFFICELAPTOP
DISALLOW	USER	WEBSTER

IP Address: 1.2.3.4 Computer: MyComputer User: MyUser Test

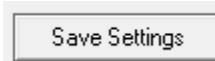
QW DataHub Security Settings

- **Initially [ALLOW / DISALLOW]:** Choose whether to allow all connections by default, or block all unless explicitly allowed.
- **Access Rules Table:**
 - **Access:** Choose ALLOW or DISALLOW.
 - **Scope:** Apply the rule by IP, COMPUTER, or USER.
 - **Value:** The specific address, hostname, or username to include in the rule. Wildcards (e.g., 192.168.0.*) are supported.

At the bottom of this section, you can test rules using the **IP Address**, **Computer**, and **User** fields and clicking **Test**. The system will show whether access would be granted based on the current rule set.

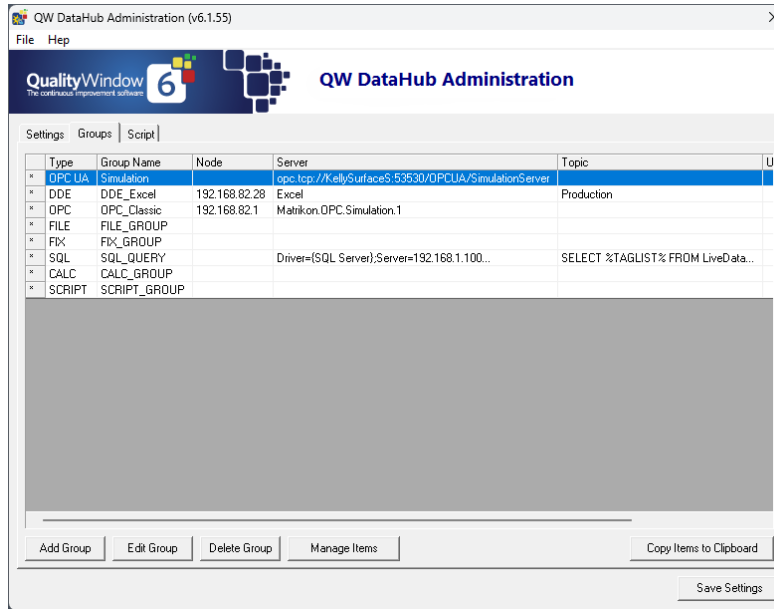
Saving Settings

Click **Save Settings** to apply all changes. Note that settings do not take effect until QW DataHub is restarted.



Groups

The **Groups** tab in QW DataHub Admin is where you define data connections and organize the tags that QW DataHub will manage. Each group represents a specific data source (like an OPC UA server, SQL database, or tag calculation script), and each group contains one or more tags identified by a meaningful alias. See Supported Group Types section for further details on the supported group types and how use them.



QW DataHub Groups Tab

Core Capabilities

The Groups tab provides several tools to help you manage your data connections:

- **Add Group:** Create a new group by choosing a group type (e.g., OPC UA, SQL) and configuring its connection settings.
- **Edit Group:** Open an existing group to modify its settings or update the list of tags.
- **Delete Group:** Remove an existing group from the configuration.
- **Manage Items:** Open the item editor to define or update the individual tags within a group.
- **Copy Items to Clipboard:** Export all groups and their tags to the clipboard. This is useful for pasting into Excel for review or bulk editing.

A	B	C	D	E	F	G	H	I	J	
1	Type	Group Name	Node	Server	Topic Name	UpdateRate	Deadband	Tag Alias	Tag Name	Description
2	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Counter	ns=3:=1001	
3	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Random	ns=3:=1002	
4	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Sawtooth	ns=3:=1003	
5	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Inusoid	ns=3:=1004	
6	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Square	ns=3:=1005	
7	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Triangle	ns=3:=1006	
8	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	Simulation Constant	ns=3:=1007	
9	OPC UA	Simulation		opc.tcp://KellySurfaceS:53530/OPCUA/SimulationServer		1000	0	QWSimulation CurrentBatchNumber	ns=3:=1009	
10	DDE	DDE_Excel	192.168.82.28	Excel	Production	1000	0	SPEC5 USL	R2C6	Upper Spec Limit
11	OPC	OPC_Classic	192.168.82.1	Matrxion OPC Simulation 1		1000	0	OPC_Classic @ClientCount	@ClientCount	
12	FILE	FILE_GROUP				1000	0	Identifiers BatchNumber	C.MyIdentifiers@BatchNo.io	Current Batch number
13	FIX	FIX_GROUP				1000	0	Tag1	FIXTag1	
14	SQL	SQL_QUERY		Driver={SQL Server};Server=192.168.1.100;Database=Prod	SELECT %TAGLIST% FROM LiveData	1000	0	Current Speed	Speed	
15	SQL	SQL_QUERY		Driver={SQL Server};Server=192.168.1.100;Database=Prod	SELECT %TAGLIST% FROM LiveData	1000	0	Current Temperature	Temperature	
16	CALC	CALC_GROUP				1000	0	Calculations ProductionWeight		
17	CALC	CALC_GROUP				1000	0	Calculations ProductionRate		
18	SCRIPT	SCRIPT_GROUP				1000	0	Script Speed		
19	SCRIPT	SCRIPT_GROUP				1000	0	Script Temperature		
20	SCRIPT	SCRIPT_GROUP				1000	0	Script Team		

Example output of Copy items to Clipboard in Excel

Groups List Display

The main Groups list provides a summary of all currently defined groups, showing:

- **Group Name** – The user-defined name of the group

QW DataHub - Administration Guide

- **Group Type** – Indicates the type of connection (e.g., OPC UA, SQL, FILE)
- **Update Rate** – How often the group is polled (in milliseconds)
- **Enabled** – Whether the group is active and polling

Groups can be enabled or disabled individually, allowing you to stage or troubleshoot configurations without deleting them.

Script

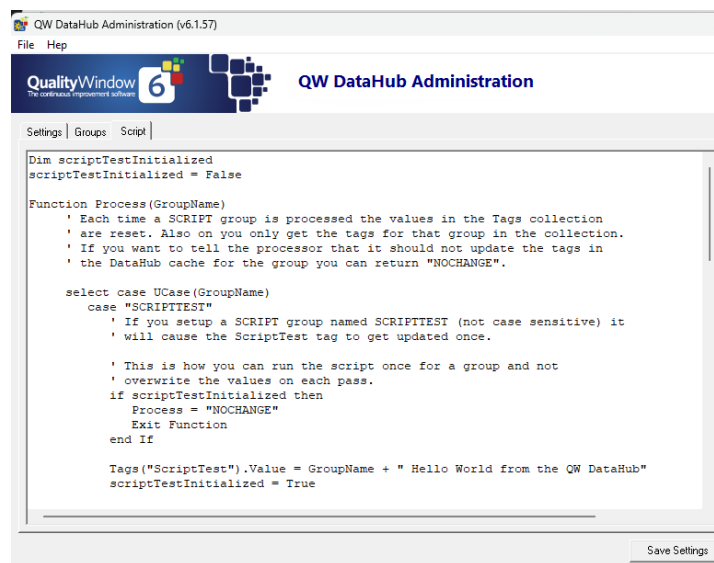
The **Script** tab in QW DataHub Admin is where you define scripting logic for **SCRIPT** and **CALC** group types. It contains shared VBScript code that QW DataHub runs to assign or calculate tag values.

Two main functions are supported:

- `Process(Group Name)` is used by **SCRIPT** groups and runs each time the group updates. It can assign values to any tag within that group.
- `Value(Tag Alias)` is used by **CALC** groups and runs once per tag. It can access tags across all groups to calculate values dynamically.

This tab enables logic such as:

- Derived values (e.g., unit conversions)
- Condition-based outputs (e.g., alarm flags)
- One-time or stateful operations
- Simulated tag values



```

Dim scriptTestInitialized
scriptTestInitialized = False

Function Process(Group Name)
    ' Each time a SCRIPT group is processed the values in the Tags collection
    ' are reset. Also on you only get the tags for that group in the collection.
    ' If you want to tell the processor that it should not update the tags in
    ' the DataHub cache for the group you can return "NOCHANGE".

    select case UCase(Group Name)
        case "SCRIPTTEST"
            ' If you setup a SCRIPT group named SCRIPTTEST (not case sensitive) it
            ' will cause the ScriptTest tag to get updated once.

            ' This is how you can run the script once for a group and not
            ' overwrite the values on each pass.
            if scriptTestInitialized then
                Process = "NOCHANGE"
                Exit Function
            end If

            Tags("ScriptTest").Value = Group Name + " Hello World from the QW DataHub"
            scriptTestInitialized = True
    end select
End Function
    
```

QW DataHub Admin Script Tab

By default, a sample script is included to serve as a working example. It demonstrates both types of script groups in action and can be deleted or modified as needed.

To use these examples, make sure the corresponding groups (CalcExample, ScriptExample, etc.) are added and enabled in the Groups tab, as they are disabled by default.

Validating QW DataHub Connectivity

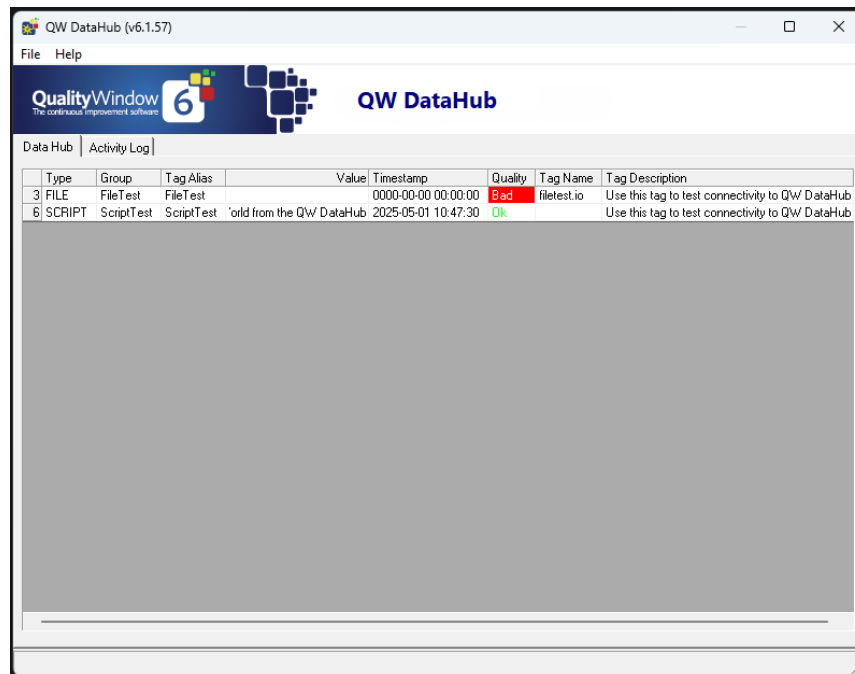
After installing and configuring the QW DataHub License and settings, it's important to confirm that external systems can successfully connect and interact with the QW DataHub instance. To support this, fresh installs include two preconfigured test tags: one **ScriptTest** tag and one **FileTest** tag.

What You Should See After Installation

When you open QW DataHub after a fresh install, you'll notice the following:

- A **SCRIPT** tag (**ScriptTest**) showing a default value like “Hello from QW DataHub” with a quality status of **Ok**
- A **FILE** tag (**FileTest**) showing **Bad** quality with a blank or null value

This setup is intentional. The **FileTest** tag is designed to simulate a remote write and allows you to validate TCP connectivity and port access from a separate machine.

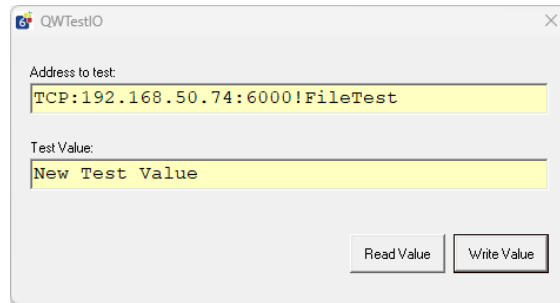


Type	Group	Tag Alias	Value	Timestamp	Quality	Tag Name	Tag Description
3	FILE	FileTest		0000-00-00 00:00:00	Bad	filetest.io	Use this tag to test connectivity to QW DataHub
6	SCRIPT	ScriptTest	orld from the QW DataHub	2025-05-01 10:47:30	Ok		Use this tag to test connectivity to QW DataHub

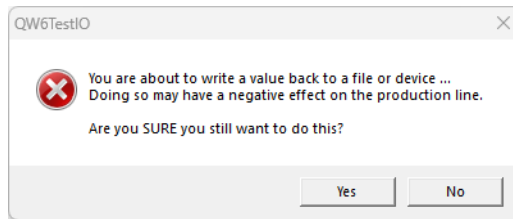
QW DataHub new install defaults

How to Perform a Connectivity Test

1. **Open QW Admin on a separate machine** from the one where QW DataHub is installed.
2. **Launch the IO Test Utility** (typically named QWTestIO.exe or QW6TestIO.exe).
3. **Enter the address using TCP format:** `TCP:{IP Address}:{Port}!FileTest`
 - Example: `TCP:192.168.50.74:6000!FileTest`
4. In the **Test Value** box, type a test string (e.g., New Test Value).
5. Click **Write Value**.



A warning will appear reminding you that writing to a tag may affect production systems. Click **Yes** to proceed if you are in a test environment.



6. Return to the QW DataHub interface on the host machine. If the test was successful:
 - The **FileTest** tag will update to show the written value
 - The **Quality** column will change from **Bad** to **Ok**
 - The **Timestamp** will update to the current time

Type	Group	Tag Alias	Value	Timestamp	Quality	Tag Name	Tag Description
3 FILE	FileTest	FileTest	New Test Value	2025-05-01 11:14:22	Ok	filetest.io	Use this tag to test connectivity to QW DataHub
6 SCRIPT	ScriptTest	ScriptTest	World from the QW DataHub	2025-05-01 10:47:30	Ok		Use this tag to test connectivity to QW DataHub

Example Successful Connectivity on FileTest

7. You can also test reading by using the **Read Value** button in the IO Test Utility to confirm the written value is correctly retrieved.

Troubleshooting connectivity

If writing to the tag fails or the quality remains **Bad**, check the following:

- The TCP address is correct (host IP and port number match the QW DataHub settings)
- QW DataHub’s **TCP Server** is enabled and listening on the expected port
- There is **no firewall blocking** the TCP port (ensure an inbound rule exists)
- The client machine is **not blocked by TCP Security settings** in QW DataHub Admin

For local testing from the same machine, you can use the simplified format:

HUB:FileTest

This test confirms both tag connectivity and TCP-level communication, making it a key validation step in any new QW DataHub deployment.

Supported Group Types

Each group type has its own configuration screen and tag setup process. Below is a description of each supported group type, its purpose, configuration requirements, and how tags (items) are managed.

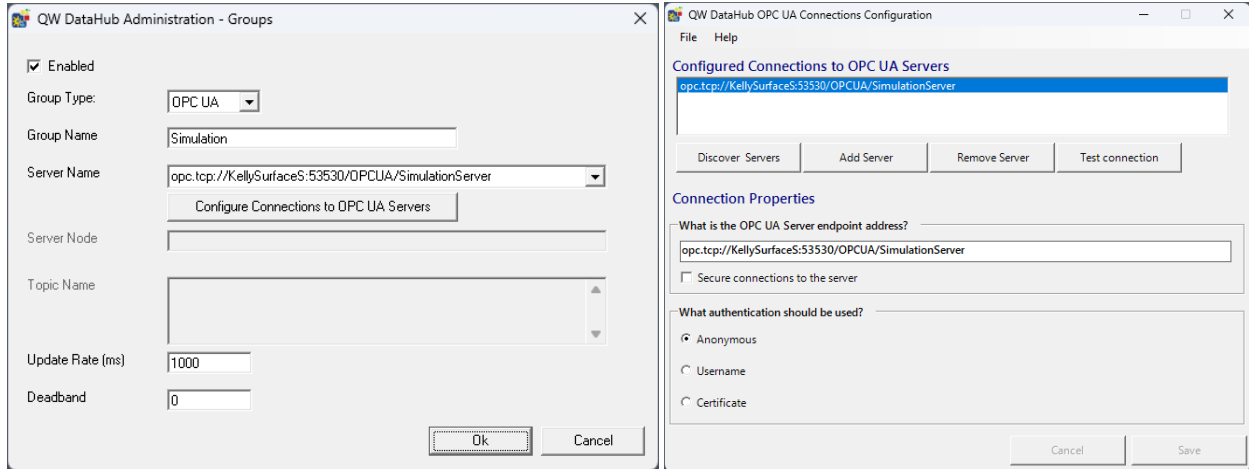
OPC UA Group

Connects to an OPC UA (Unified Architecture) server. Ideal for modern, secure, cross-platform industrial data access to OPC UA servers like PLCs, gateways, or simulators.

Configuration Fields

Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Server Name	Select from configured OPC UA endpoints
Configure Connections	Add or edit OPC UA server definitions, authentication, and encryption
Update Rate (ms)	How often the OPC UA server is polled in milliseconds
Deadband (%)	Ignores minor value changes below this percentage

QW DataHub - Administration Guide

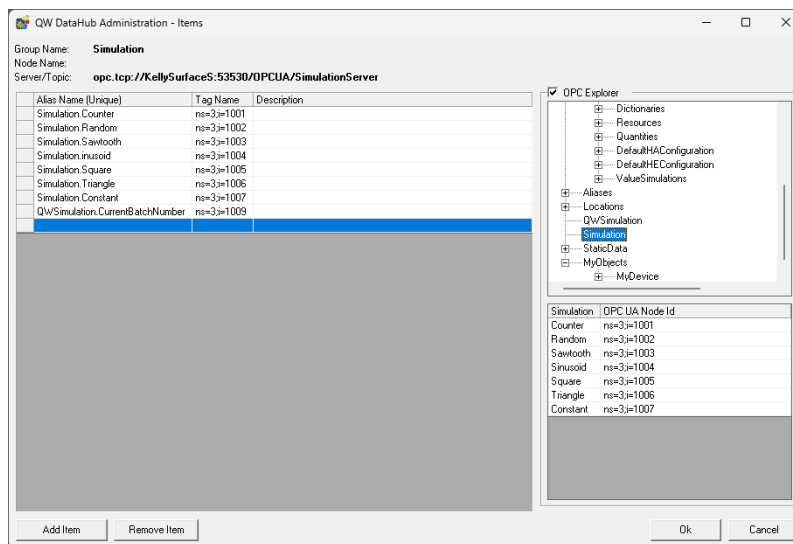


OPC UA Group Settings (left). OPC UA Connections Configuration (right)

Manage Items

Browse the OPC UA tag hierarchy using the built-in Explorer

Item Property	Description
Tag Alias	User friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window
Tag Name	The node ID provided by the OPC UA explorer for this OPC UA node. <i>Example: ns=3;i=1007</i>
Description	Optional Description for this Item definition



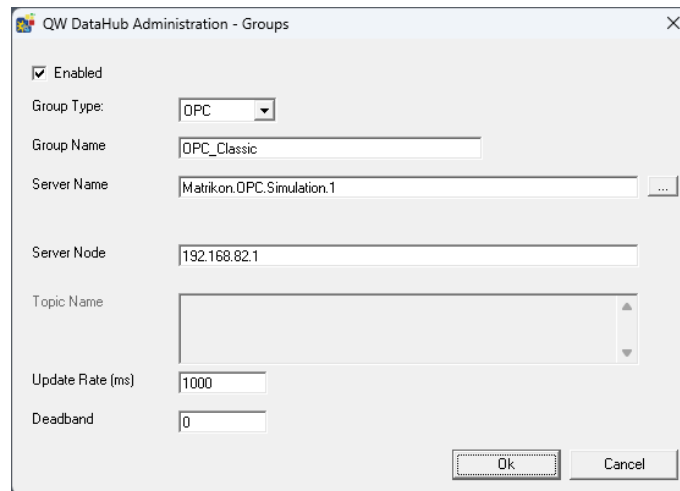
OPC UA Manage Items View

OPC Classic Group

Connects to OPC DA (Classic) servers using COM/DCOM. Best suited for legacy control systems and Windows-based automation platforms such as Matrikon, Kepware, or legacy DCS systems.

Configuration Fields

Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Server Name	Select from registered local OPC servers
Server Node	Remote computer name (if applicable)
Update Rate (ms)	How often the OPC server Classic is polled in milliseconds
Deadband (%)	Ignores minor value changes below this percentage



OPC Classic Group Editor

Manage Items

Use the OPC Explorer to select available tags

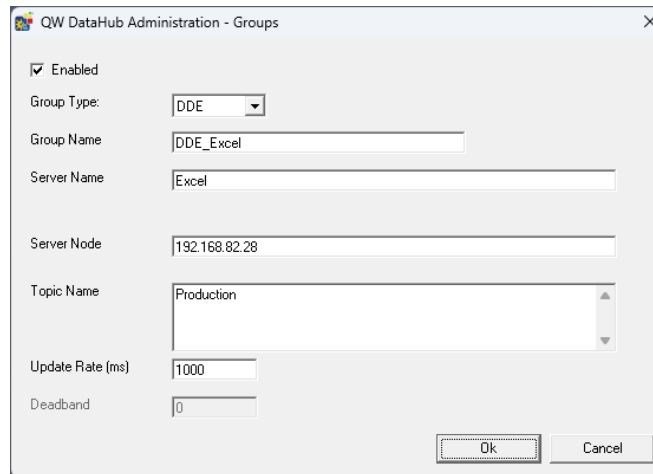
Item Property	Description
Tag Alias	User friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window
Tag Name	Provided automatically when you use the OPC Explorer to select available tags. The exact tag name on the OPC Server
Description	Optional Description for this Item definition

DDE Group

Connects to standard DDE-compatible applications such as Excel or legacy HMI software, DDE based systems like LabView or custom tools.

Configuration Fields

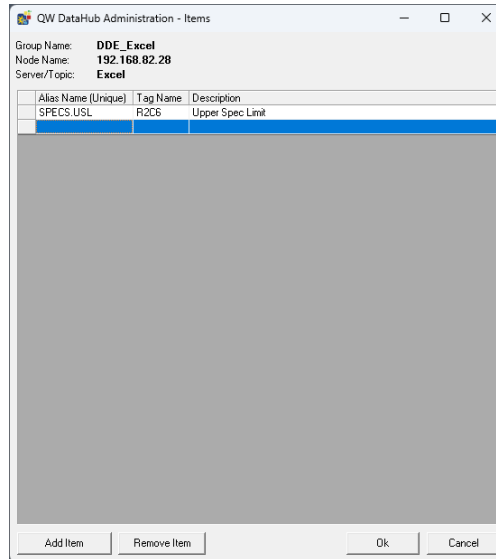
Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Server Name	DDE server name (e.g., Excel)
Server Node	Remote computer name (if applicable)
Topic Name	Context identifier (e.g., Excel workbook name)
Update Rate (ms)	How often the DDE server is polled in milliseconds
Deadband (%)	Ignores minor value changes below this percentage



DDE Group Editor

Manage Items

Item Property	Description
Tag Alias	The Unique Identifier or User friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window
Tag Name	Provide cell index in Excel, or tag details from other DDE source. Eg: <i>R1C1</i> for Excel cell A1
Description	Optional Description for this Item definition



DDE Group Items Management

FIX Group

The FIX group type connects directly to a locally installed GE iFIX system using its **native COM automation interface** (FixData API). This method is ideal for high-performance, locally hosted environments where minimal configuration and real-time access are critical. This group type is best used when working within a single-node iFIX system or when high-performance access to live data is required without network dependencies.

Configuration Fields

Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Update Rate (ms)	How often the tags are updated in milliseconds

Requirements

- ✓ QW DataHub must be installed on the same computer as GE iFIX.
- ✓ GE iFIX must be running and properly licensed.
- ✓ The iFix Automation interface COM interface must be registered and accessible.
- ✓ No remote access is supported with this group type.

Manage Items

Item Property	Description
---------------	-------------

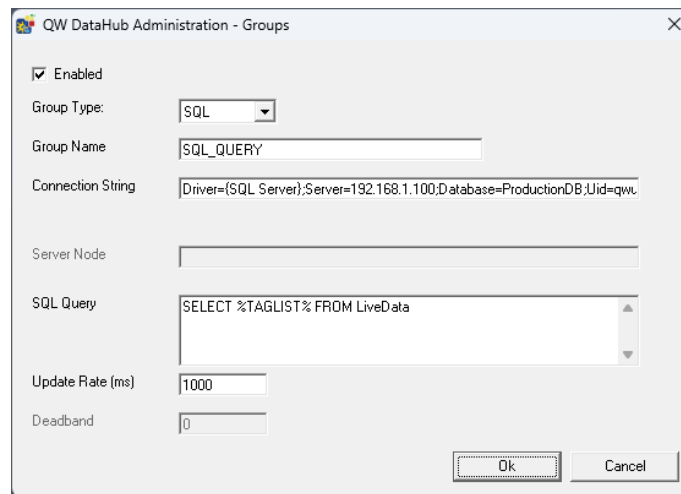
Tag Alias	User friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window
Tag Name	The exact iFIX tag ID (e.g., Pump01.Speed)
Description	Optional Description for this Item definition

SQL Group

Connects to a SQL database to run a query at regular intervals and return values as tags. This group type is best used to access live data from lab systems, MES, ERP, or any structured database.

Configuration Fields

Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Connection String	Full ODBC/OLE DB Connection String
SQL Query	Query to run on server to return column names. Supports %TAGLIST% for dynamic queries based on the defined tags.
Update Rate (ms)	How often the query is executed to update tags



SQL Group Editor

How to use this type

The SQL group type in QW DataHub allows you to retrieve live values from a database on a timed interval. These values are mapped to tag aliases and made available to Quality Window.

Requirements

- A valid connection string
- A properly formatted SQL query
- A list of tag aliases that match fields or tag names in the result set

Note: QW DataHub stores only one value per tag. If your query returns multiple records, only the last record will be used.

There are two supported query modes: **Traditional** and **Transposed**.

Traditional Query Mode

This mode maps each field in the result row directly to a tag alias.

Use this when your query returns a single record with multiple columns.

Example Query:

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

Example Result:

Weight1	Weight2	Weight3
123	456	789

If your SQL group contains tags named Weight1, Weight2, and **Weight3**, those values will be assigned accordingly.

You can also simplify the query using **%TAGLIST%**, which is replaced at runtime with the list of tag names defined in the group:

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

Important: If your query returns more than one row, only the last row will be used.

Transposed Query Mode

This mode is used when your result set includes tag names as values in a field, rather than as column headers.

Use this when your database stores tag/value pairs as rows.

Example Query:

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

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

Or using the dynamic tag list:

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

Example Result:

TagName	Value
Weight1	123
Weight2	456
Weight3	789

QW DataHub will convert this into:

Weight1	Weight2	Weight3
123	456	789

This mode is ideal when you're working with wide historian tables, flat tag/value logs, or flexible schema systems.

Timestamped Transposed Queries

If your table contains historical entries, you can include a timestamp to ensure QW DataHub uses only the latest values for each tag.

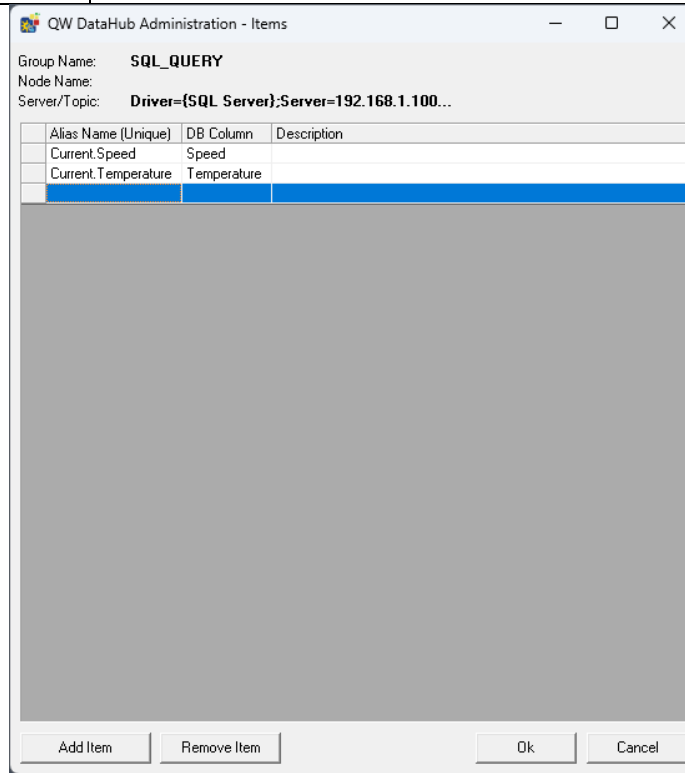
Example Query:

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

This format is especially useful for time-based sampling systems where multiple values may exist for a tag.

Manage Items

Item Property	Description
Tag Alias	The Unique Identifier or User friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window.
Tag Name	Name of Column in Query result for this value.
Description	Optional Description for this Item definition.



SQL Group Item Management example

FILE Group

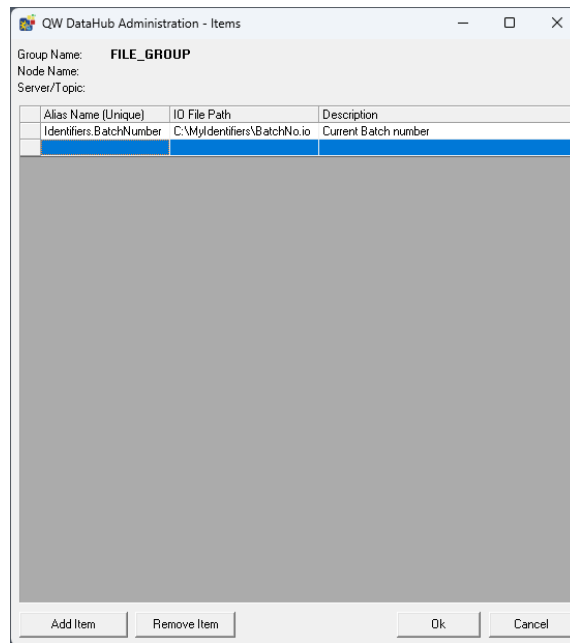
Monitors Quality Window I/O files written by external systems or interfaces. This group type is best used to access data from structured files updated by external equipment, scripts, or middleware.

Configuration Fields

Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Update Rate (ms)	How often the file content is refreshed in milliseconds

Manage Items

Item Property	Description
Tag Alias	The Unique Identifier or User friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window.
Tag Name	Fully qualified path to QW I/O File. <i>Tip: Avoid using network mapped drives and instead use UNC path for network resources</i>
Description	Optional Description for this Item definition.



File Group Item management example

CALC Group

The CALC group type allows you to create **calculated tags** based on values from other tags in QW DataHub. Each tag is updated individually using a shared script, and its value is determined by a function that returns a result based on its alias. This type of best for performing lightweight, inline calculations—such as unit conversions, simple conditions, or data manipulation—based on values from other tags.

Configuration Fields

Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group

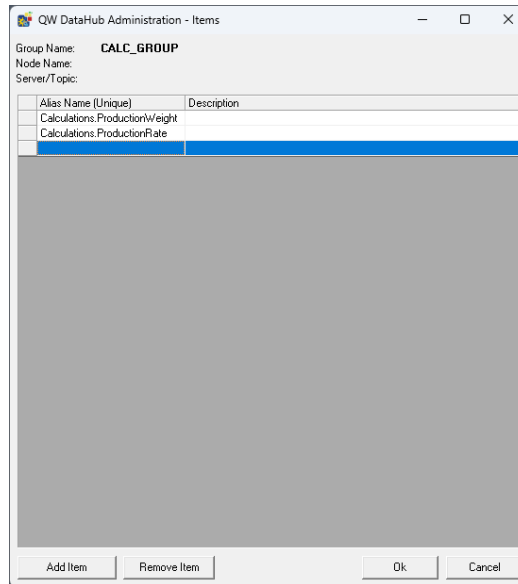
Update Rate (ms)	How often the calculation is re-calculated and refreshed in milliseconds
-------------------------	--

Requirements

- ✓ Tags must be defined in the group manually
- ✓ A script must be written using the `Value(TagAlias)` function

Manage Items

Item Property	Description
Tag Alias	The Unique Identifier or User-friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window.
Description	Optional Description for this Item definition.



Calc Group Item Management example

Script Structure

The CALC group uses a shared function named `Value(TagAlias)` that is called once per tag. You define logic for each tag inside this function.

Example:

```
Function Value(TagAlias)
  Select Case TagAlias
    Case "Speed_kph"
      Value = Tags("Speed_mph").Value * 1.60934
    Case "IsAlarm"
```

```

If Tags("Pressure").Value > 100 Then
    Value = 1
Else
    Value = 0
End If
Case Else
    Value = "NOCHANGE"
End Select
End Function
    
```

In this model:

- All other tags in QW DataHub are accessible using `Tags("TagAlias")`
- Return `"NOCHANGE"` to skip updating the value
- Scripts should be efficient—CALC groups run inline and affect performance

Key Differences from SCRIPT Groups

- CALC evaluates each tag **independently** using `Value(TagAlias)`
- SCRIPT runs group-level logic on a **separate thread** using `Process(GroupName)`
- CALC has **access to all tags** in QW DataHub, whereas SCRIPT only accesses its own group

CALC groups are ideal for quick, formula-based tags where each tag's logic can be isolated and calculated on demand.

SCRIPT Group

The SCRIPT group type allows you to generate values for multiple tags using custom VBScript logic that runs on a scheduled interval in its own dedicated thread. This type is ideal for advanced or external logic where tag values depend on shared state, database lookups, counters, or external input.

Configuration Fields

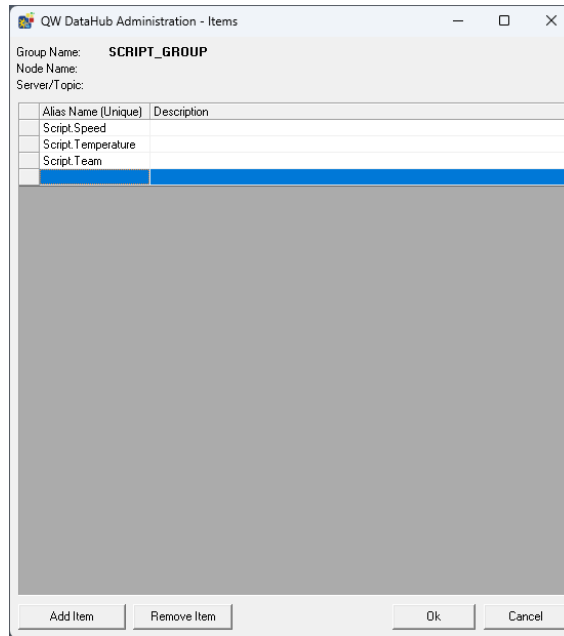
Group Property	Description
Enabled	Activates or disables the group
Group Name	A unique name for the group
Update Rate (ms)	How often the script is executed in milliseconds

Requirements

- ✓ All tag aliases in the group must be manually defined
- ✓ Script logic must be written using the `Process(GroupName)` function
- ✓ Tag values must be assigned directly in the script

Manage Items

Item Property	Description
Tag Alias	The Unique Identifier or User-friendly name to access this DataHub data item from scripts, QW Schedule or Quality Window.
Description	Optional Description for this Item definition.



Script Group Item management example

Script Structure

The SCRIPT group uses a function called `Process (GroupName)` that runs once per update cycle. Inside this function, you assign values to tags using their alias names.

Example:

```
Dim counter
Function Process(GroupName)
    counter = counter + 1
    Tags("CycleCount").Value = counter
    Tags("Timestamp").Value = Now
End Function
```

You can use shared variables, conditionals, time stamps, or call external COM objects as needed. Each tag's value is updated only when explicitly assigned within the script.

Access Scope

- Tags in this group are accessible via `Tags("Alias")`
- Tags **outside** this group are **not accessible**—SCRIPT groups are isolated
- The Tags collection provides `.Value`, `.TagAlias`, `.TagName`, and `.Description`

Best Practices

- Keep logic tight and efficient; long-running scripts can delay updates
- Use persistent variables for counters, state tracking, or history
- Use `NOCHANGE` only if building logic to retain current values
- Avoid accessing external systems unless the response time is reliable

Key Differences from CALC Groups

- SCRIPT evaluates **all tags in the group together** in a single pass
- CALC evaluates **each tag independently**
- SCRIPT logic runs **on a dedicated thread**
- SCRIPT groups cannot access tags outside their own group
- Ideal for logic that requires **shared context** or **state across tags**

SCRIPT groups are powerful when you're generating related values in bulk, applying conditional logic across multiple tags, or simulating data feeds with stateful behavior.

Using QW DataHub

The QW DataHub interface provides real-time visibility into the current status of all tags, their values, and incoming client requests. It is typically launched automatically when QW DataHub starts and can also be opened manually via the system tray icon or Start menu.

The interface consists of two main tabs: **Data Hub** and **Activity Log**, plus a menu for administrative actions.

Data Hub Tab

The **Data Hub** tab displays a live view of all active tags being managed by QW DataHub.

QW DataHub - Administration Guide

Type	Group	Tag Alias	Value	Timestamp	Quality	Tag Name	Tag
1	OPC UA	Simulation.Counter	25	2025-04-16 12:56:34	Ok	ns=3;#1001	
2	OPC UA	Simulation.Random	0.03866129	2025-04-16 12:56:34	Ok	ns=3;#1002	
3	OPC UA	Simulation.Sawtooth	1.6	2025-04-16 12:56:34	Ok	ns=3;#1003	
4	OPC UA	Simulation.inusoid	1.48629	2025-04-16 12:56:34	Ok	ns=3;#1004	
5	OPC UA	Simulation.Square	2	2025-04-16 12:56:33	Ok	ns=3;#1005	
6	OPC UA	Simulation.Triangle	1.066667	2025-04-16 12:56:34	Ok	ns=3;#1006	
7	OPC UA	Simulation.Constant		2025-04-16 12:55:11	Ok	ns=3;#1007	
8	OPC UA	QWSimulation.CurrentBatchNumber	90005	2025-04-16 12:55:11	Ok	ns=3;#1009	
9	DDE	DDE_Excel	SPECS_USL	0000-00-00 00:00:00	Bad	R2C6	Upd
10	OPC	OPC_Classic	@ClientCount	0000-00-00 00:00:00	Bad	@ClientCount	
11	FILE	FILE_GROUP	Identifiers.BatchNumber	0000-00-00 00:00:00	Bad	BatchNo.io	Cur
12	FIX	FIX_GROUP	Tag1	0000-00-00 00:00:00	Bad	FIXTag1	
13	SQL	SQL_QUERY	Current.Speed	0000-00-00 00:00:00	Bad	Speed	
14	SQL	SQL_QUERY	Current.Temperature	0000-00-00 00:00:00	Bad	Temperature	
15	CALC	CALC_GROUP	Calculations.ProductionWeight	0000-00-00 00:00:00			
16	CALC	CALC_GROUP	Calculations.ProductionRate	0000-00-00 00:00:00			
17	SCRIPT	SCRIPT_GROUP	Script.Speed	0000-00-00 00:00:00			
18	SCRIPT	SCRIPT_GROUP	Script.Temperature	0000-00-00 00:00:00			
19	SCRIPT	SCRIPT_GROUP	Script.Team	0000-00-00 00:00:00			

SQL_QUERY Finished processing.

Example of QW DataHub main view

Columns shown:

- **Type:** The group type (e.g., OPC UA, SQL, CALC)
- **Group:** The group name this tag belongs to
- **Tag Alias:** The alias name configured in the group
- **Value:** The most recent value retrieved or calculated
- **Timestamp:** The time the value was last updated
- **Quality:** Shows Ok for valid, current values and Bad if the tag is unavailable or failed to update
- **Tag Name:** The source tag name (from the data source)
- **Tag Description:** Any optional notes you added to the tag configuration

Reading the Grid:

- Tags with green "Ok" status are updating successfully.
- Tags with red "Bad" status indicate a problem, such as a failed connection, invalid tag name, or script error.

This tab is useful for monitoring system health at a glance and verifying that your group configurations are working as intended.

Activity Log Tab

The **Activity Log** tab shows incoming tag requests from connected clients, such as Quality Window applications, QWSchedule, or scripts.

The screenshot shows the 'Activity Log' tab in the QW DataHub application. The window title is 'QW DataHub (v6.1.55) - Listening on 192.168.50.74 Port 6000'. The table contains the following data:

Timestamp	IP Address	Requestor	Command	Tag Name	Value
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:57		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005
2025-04-16 12:57:58		Local	GetValue	QW.Simulation.CurrentBatchNumber	90005

At the bottom of the window, a status bar indicates 'SQL_QUERY Finished processing.'

Example of QW DataHub Activity Log

Columns shown:

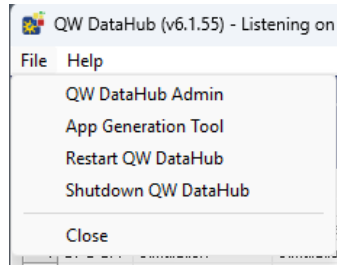
- **Timestamp:** When the request was received
- **IP Address:** The source of the request
- **Requestor:** Usually "Local" or a machine name
- **Command:** Type of request, such as GetValue
- **Tag Name:** The tag being requested
- **Value:** The value returned for that tag at the time of the request

This view is especially useful for:

- Confirming that a client is actively querying QW DataHub
- Verifying which tags are being used
- Diagnosing network or access permission issues

Menu Options

The **File menu** and the **system tray context menu** provide quick access to key actions:



QW DataHub file menu options

- **QW DataHub Admin:** Opens the configuration tool for managing groups, tags, TCP access, and scripts.
- **App Generation Tool:** Opens a companion tool for building Quality Window applications based on tag data.
- **Restart QW DataHub:** Restarts the DataHub engine and reloads all groups and scripts.
- **Shutdown QW DataHub:** Closes the application completely.
- **Close:** Hides the window but leaves QW DataHub running in the background.

You can access these options from the menu bar or by right-clicking the QW DataHub icon in the system tray.

Integrating with QW DataHub

Once your groups and tags are configured in QW DataHub, the values are available for use across the Quality Window platform. You can reference live tag values from QW DataHub in three main ways:

1. Using QW DataHub Tags in Quality Window Admin

Within **Quality Window Admin**, you can connect any variable in your application to a QW DataHub tag by specifying it as an **Input File** or **Output File** properties.

Format:

HUB:TagAlias

Example:

HUB:Line1.Temperature

This configuration will fetch the current value of the Line1.Temperature tag from QW DataHub each time the record is initialized (typically when the Add key is pressed).

Variable Details	
Variable Number	3
Variable Name	BatchNumber
Field Type	Text
Length	10
Required	No
Protected	No
Sampling Plan	0
Convert to Capital Letters	No
Input File	HUB:QWSimulation.CurrentBatchNumber
Output File	
Units of Measure	
Reference Fields	(Collection)
Help File	

Example of use of hub in QW Admin Variable Definition

For Remote Access

If the Quality Window application is on a different computer than QW DataHub, use the TCP format:

```
TCP:<IP Address>:<Port>!TagAlias
```

Example:

```
TCP:192.168.1.10:6000!Line1.Temperature
```

Or using a hostname and the default port (6000 can be omitted):

```
TCP:MyDataHub!Line1.Temperature
```

2. Accessing QW DataHub from Scripts

Within any **Quality Window script**, you can retrieve a QW DataHub tag value programmatically.

Option 1: Using QWDataHubProxy (local COM object)

```
Function GetValue(TagAlias)
  Dim oHub, Result
  On Error Resume Next
  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
```

Option 2: Using QWComSrv for file-style access

```
Dim oComSrv, Value
Set oComSrv = CreateObject("QWComSrv.QWComSrvClass")
oComSrv.Silent = True
Value = oComSrv.ReadInputFile("HUB:MyTag1")
```

```
Value = oComSrv.ReadInputFile("TCP:192.168.1.10:6000!MyTag1")
oComSrv.WriteOutputFile "HUB:MyTag1", Value
Set oComSrv = Nothing
```

Use HUB: for local access and TCP: for remote access. This method supports both reading and writing values.

3. Referencing Tags in QW Schedule

QWSchedule can also reference QW DataHub tags using the same format as Quality Window:

Local Access:

```
HUB:TagAlias
```

Remote Access:

```
TCP:192.168.1.10:6000!Line1.Temperature
```

Or using a hostname and the default port (6000 can be omitted):

```
TCP:MyDataHub!Line1.Temperature
```

You can configure this in the schedule definition where a field or action requires a data source. The integration allows schedules to act on live tag values, such as triggering alerts or batch records based on current process conditions.

See our [Automating Data Collection guide](#) for more details on leveraging QW DataHub and QW Schedule to automatically add records to Quality Window Applications for analysis.

Appendix A: Troubleshooting

This section provides guidance for resolving the most common connectivity, configuration, and runtime issues encountered when using QW DataHub in live environments.

1. Tag Value Is Blank or Missing in Quality Window

Possible Causes:

- The **Tag Alias** is misspelled or not defined in QW DataHub
- The group containing the tag is **disabled**
- The data source is **offline**, returning a bad quality value
- **Cache Time** is set too low or disabled, and the tag is temporarily unavailable
- Input File in QW Admin is incorrectly formatted

Actions:

- Open **QW DataHub Admin > Groups > Manage Items** and verify that the tag exists and is enabled
- Check if the data source is connected and updating in QW DataHub
- Confirm Input File entry in QW Admin uses the format:

```
HUB:TagAlias
```

- Review the **QW DataHub Log** file for read errors or value timeouts

2. Remote Connection Fails (TCP Access)

Possible Causes:

- TCP port is blocked by Windows Firewall
- QW DataHub's **TCP Server** is not enabled
- IP address or hostname is incorrect
- Access is restricted by **TCP Security rules**
- QW DataHub is not running on the target machine

Actions:

- In **QW DataHub Admin > Settings**, verify:
 - TCP Server is **Enabled**
 - Port number (default: 6000) matches what the client is using

QW DataHub - Administration Guide

- Ensure the port is open in the firewall (inbound TCP)
- On the remote client, test connection using:

```
TCP:<IPAddress>:<Port>!TagAlias
```

- In **TCP Security**, verify your client is allowed access (IP/User/Computer Name rules)

3. Script Fails to Retrieve Value

Possible Causes:

- QWDataHubProxy or QWComSrv is not registered properly
- COM security settings prevent access
- Tag name passed to script is incorrect or blank
- QW DataHub is not running or is in error state

Actions:

- Test the COM object manually in VBScript:

```
Set oHub = GetObject(, "QWDataHubProxy.Application")
MsgBox oHub.Send("GetValue", "YourTagAlias")
```

- Verify QW DataHub is running (check system tray icon)
- Confirm that the tag alias exists and is enabled
- Ensure script host has necessary Windows permissions to instantiate COM objects

4. Scripted or Calculated Tag Shows "NOCHANGE"

Possible Causes:

- Your script explicitly returns "NOCHANGE"
- Input tag used in the calculation is missing or not numeric
- Script logic contains an unhandled condition

Actions:

- Review your CALC or SCRIPT logic for the relevant alias
- Add debug code:

```
If Tags("MyInput").Value = "" Then MsgBox "Input missing"
```

- Temporarily enable **Debug Script** in QW DataHub settings to display script errors
- Ensure all referenced tags exist and are updating

5. Tags from SQL Group Are Not Updating

Possible Causes:

- SQL query is invalid or poorly formed
- SQL connection string is incorrect or not authenticated
- `%TAGLIST%` is not resolving properly
- Query returns multiple records in standard mode

Actions:

- Test SQL connection string in a database client (e.g., Management Studio)
- Replace `%TAGLIST%` manually to test query logic
- For standard queries, ensure only one record is returned
- Use **TRANSPOSE** if your data is structured as rows of tag/value pairs

Appendix B – Script Example

The default script provides a working example of how to use both **SCRIPT** and **CALC** group types.

- **SCRIPTTEST**: Demonstrates a one-time initialization. It sets the ScriptTest tag to a greeting message only once after startup.
- **SCRIPTEXAMPLE**: Simulates static process data by setting the Pressure and Speed_mph tags on each update.
- **CALC Group**: Uses Value(TagAlias) to calculate derived tags:
 - Speed_kph is calculated by converting Speed_mph to kilometers per hour.
 - IsAlarm is set to 1 if Pressure exceeds 100, otherwise 0.

These examples serve as templates for building your own logic and can be safely modified or removed.

Full Example Code

```
Dim scriptTestInitialized
scriptTestInitialized = False

Function Process(GroupName)
    ' Each time a SCRIPT group is processed the values in the Tags collection
    ' are reset. Also on you only get the tags for that group in the collection.
    ' If you want to tell the processor that it should not update the tags in
    ' the DataHub cache for the group you can return "NOCHANGE".

    select case UCase(GroupName)
        case "SCRIPTTEST"
            ' If you setup a SCRIPT group named SCRIPTTEST (not case sensitive) it
            ' will cause the ScriptTest tag to get updated once.

            ' This is how you can run the script once for a group and not
            ' overwrite the values on each pass.
            if scriptTestInitialized then
                Process = "NOCHANGE"
            Exit Function

```

```
end If

Tags("ScriptTest").Value = GroupName + " Hello World from the QW DataHub"
scriptTestInitialized = True

case "SCRIPTEXAMPLE"
' If any other SCRIPT group is defined they will cause the
' following tags to be set.
Tags("Pressure").Value = 125
Tags("Speed_mph").Value = 60

end select

End Function

Function Value(TagAlias)
' In CALC group you have access to all the currently set Tags in the DataHub
cache across all groups.
' You can see in this examples how we make sure the tags exist and have an
appropriate value
' to perform calculations.
' If you do not want to update the value in the cache for the TagAlias you can
set the
' Value to NOCHANGE.

Select Case TagAlias
Case "Speed_kph"
' The Speed_kph tag is calculated based on the values in the Speed_mph
tag.

' Make sure the tag called Speed_mph exists
If Tags("Speed_mph") Is Nothing Then
Value = "NOCHANGE"
```

```
        Exit Function
    End If
    ' Make sure the tag called Speed_mph tag has been initialized with
    ' a numeric value.
    If Not IsNumeric(Tags("Speed_mph").Value) Then
        Value = "NOCHANGE"
        Exit Function
    End If

    Value = Tags("Speed_mph").Value * 1.60934
Case "IsAlarm"
    ' The IsAlarm tag is set based on the values in the Pressure tag.

    ' Make sure the tag called Pressure exists
    If Tags("Pressure") Is Nothing Then
        Value = "NOCHANGE"
        Exit Function
    End If
    ' Make sure the tag called Pressure tag has been initialized with
    ' a numeric value.
    If Not IsNumeric(Tags("Pressure").Value) Then
        Value = "NOCHANGE"
        Exit Function
    End If

    If Tags("Pressure").Value > 100 Then
        Value = 1
    Else
        Value = 0
    End If
Case Else
```

QW DataHub - Administration Guide

```
Value = "NOCHANGE"  
End Select  
End Function
```