FactoryTalk Historian SE Live Data Interface User Guide

Version 9.01.00

Rockwell Automation

User Manual

Original Instructions

Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.

WARNING: Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.

ATTENTION: Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss. Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

IMPORTANT: Identifies information that is critical for successful application and understanding of the product.

These labels may also be on or inside the equipment to provide specific precautions.

SHOCK HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.

BURN HAZARD: Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.

ARC FLASH HAZARD: Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

The following icon may appear in the text of this document.

Tip: Identifies information that is useful and can help to make a process easier to do or easier to understand.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

Contents

Introduction

The FactoryTalk® Historian system uses the PI System as a back end to store tag data. The FactoryTalk Historian Live Data Interface allows FactoryTalk Historian to connect with Rockwell Automation® data sources. FactoryTalk Historian Live Data Interface provides buffering capability before the data is permanently stored in the Historian archive.

FactoryTalk Historian Live Data Interface is integrated with the FactoryTalk Historian Live Data service provided by FactoryTalk® Services Platform. It is a FactoryTalk Historian Live Data client that relies on FactoryTalk Historian Live Data service to talk to FactoryTalk data servers such as FactoryTalk® View Site Edition (SE), FactoryTalk® Linx, and OPC servers that are part of a FactoryTalk application.

Tip:

The FactoryTalk Historian documentation uses the terms tag and point to mean the same thing. An example of a tag name is:

`FTLDEnt:rc:RSLinxC.secondArray000.`

FactoryTalk Historian Live Data Interface is configured using the FactoryTalk® Administration Console or Interface Configuration Utility. See Configure FactoryTalk Historian Live Data Interface on page 9 for more information.

Installation

The steps to install the FactoryTalk Historian Live Data Interface are explained in the *FactoryTalk Historian SE Installation and Configuration Guide*, available in **Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE**.

Features

The table lists the features of FactoryTalk Historian Live Data Interface.

Item	Description
PI SDK	The PI SDK and PI API are bundled together and must be installed on each PI Interface node. FactoryTalk Historian Live Data Interface does not specifically make PI SDK calls. It uses the PI API calls to support longer instrument tag fields and multiple-character point sources. The PI SDK cannot be used if the interface is set up to use Disconnected Startup because it is based on API calls only.
Source of timestamps	FactoryTalk Historian Live Data Interface can accept timestamps from the FactoryTalk Live Data server, or it can provide timestamps from the FactoryTalk Historian server. The command-line parameter controls timestamps. For more information, see Startup command file on page 26.
UniInt-based	UniInt stands for Universal Interface. UniInt is not a separate product or file. It is an OSIsoft-developed framework used by developers, and it is integrated into many interfaces, including this interface. The purpose of UniInt is to keep a consistent feature set and behavior across as many Rockwell Automation interfaces as possible. It also allows for the rapid

Item	Description
	development of new interfaces. In any UniInt-based interface, the interface uses some of the UniInt-supplied configuration parameters and some interface-specific parameters.
Platforms	FactoryTalk Historian Live Data Interface is designed to run on the Microsoft Windows operating systems. Due to its dependency on FactoryTalk Services Platform, the FactoryTalk Historian Live Data Interface is not supported on non-Windows platforms. To see a list of operating systems supported, refer to the *FactoryTalk Historian SE Release Notes*.

User documentation

The user documentation is available in the following location:

C:\Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE

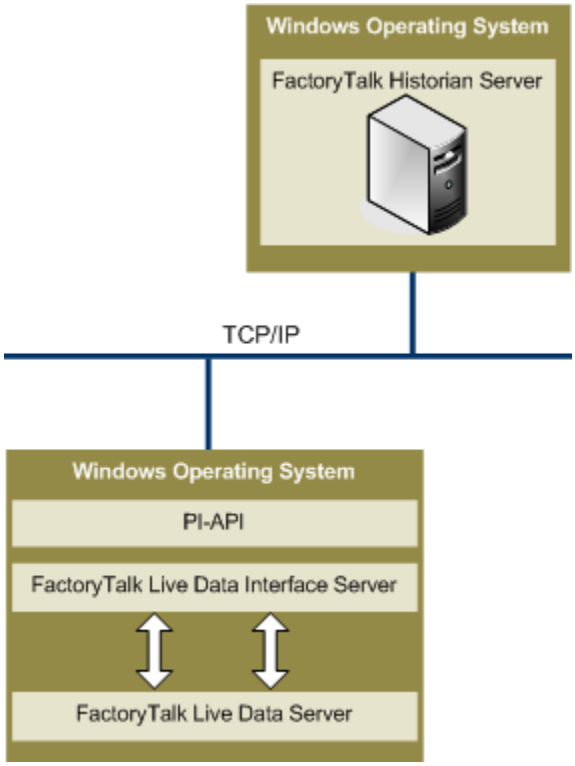
The location contains the following subfolders and user documents:

- Asset Framework
- FTHistorianConfig
- Historian Server
- Release Notes
- Third-party Licenses
- FactoryTalk Historian SE Installation and Configuration Guide.pdf (hse-in025_-en-e.pdf)
- FactoryTalk Historian SE Live Data Interface User Guide.pdf (hseld-um024_-en-e.pdf)
- AVEVA-PI-Server-2018-SP3-Patch-4-High-Availability-Administration-Guide-EN.pdf
- AVEVA-PI-Server-2018-SP3-Patch-4-Installation-and-Upgrade-Guide-EN.pdf

Principles of operation

FactoryTalk Historian Live Data Interface is a FactoryTalk Historian Live Data client that enables process data to be passed between a FactoryTalk Live Data server (for example, FactoryTalk Linx) and a FactoryTalk Historian server. Each instance of FactoryTalk Historian Live Data Interface can provide data to a single FactoryTalk Historian SE server or collective. Multiple instances of the interface may be configured, if necessary, when API buffering is used.

The following figure shows the basic workflow of the FactoryTalk Historian Live Data Interface.



Processing loops

At startup, FactoryTalk Historian Live Data Interface tries to establish a connection to both the FactoryTalk Historian Live Data server and the FactoryTalk Historian server.

Once the startup is complete, the Interface enters the processing loop, which includes:

- Servicing scheduled input points. Each Scan Class is processed in turn.
- Servicing output points as events arrive.
- Servicing triggered input points as events arrive.

The Historian Point Database is checked every 2 minutes for points that are added, edited, and deleted. When point updates are detected, the points are loaded (or reloaded) by the interface as appropriate. The 2-minute update interval can be adjusted with the `/updateinterval` command-line parameter discussed in the *PI-Universal-Interface-(UniInt)-Framework-User-Guide.pdf*.

Tip: The document is available in **Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE\Historian Server**.

The interface processes a maximum of 25 point updates at a time. If more than 25 points are added, edited, or deleted at one time, the interface will process the first 25 points, wait 30 seconds (or the length of time specified by the `/updateinterval` parameter, whichever is the lower), process the next 25 points, and so on. After all points have been processed, the interface will resume checking for updates every 2 minutes (or the length of time specified by the `/updateinterval` parameter).

All tag edits are performed in the following way: old versions of edited tags are deleted from the interface, new versions are added. Therefore, it is more efficient to stop and then start the interface if a large number of tags are edited.

FactoryTalk Diagnostics

FactoryTalk Historian Live Data Interface sends messages about its operation to FactoryTalk Diagnostics.

FactoryTalk Diagnostics provides the following information about FactoryTalk Historian Live Data Interface:

- Informational messages on the interface startup and shutdown.
- The scan rate of each scan class.
- The count of points loaded by the interface.
- Error messages for points rejected by the interface because they were configured incorrectly.
- Error messages for points rejected by the FactoryTalk Live Data server or error messages sent from the FactoryTalk Live Data server.

Because FactoryTalk Historian Live Data Interface is based on the PI-UniInt framework, a few messages are sent to the local PI Message Log on the interface node by the PI-UniInt. The standard OSIsoft interface routines or the PI API produce the error messages.

Tip:

- The FactoryTalk Diagnostics Setup and Viewer are available from the **Tools** menu of FactoryTalk Administration Console.
- For details on reading the message logs, see Rockwell Automation Knowledgebase Document ID: 1129979-How to read new UniInt Interface message logs?.
  One of the ways of reading the message logs is to use the **PIGetMsg** utility. The utility is available in the following locations:
  C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\adm

To learn more about viewing error messages and accessing the log file, refer to the help files for FactoryTalk Diagnostics Setup and FactoryTalk Diagnostics Viewer.

Configure FactoryTalk Historian Live Data Interface

To configure a FactoryTalk Historian Live Data Interface, we recommend that you use FactoryTalk Administration Console.

However, you must use the Interface Configuration Utility (ICU) to configure FactoryTalk Historian Live Data Interface in the following cases:

- If you are configuring redundant FactoryTalk Historian Live Data Interfaces.
  For more information, refer to the Rockwell Automation Knowledgebase Document ID: QA13134 - FactoryTalk Historian SE: Redundant Interface Configuration Guide.
- If you are configuring the buffer subsystem on FactoryTalk Historian Live Data Interface.
  For more information, refer to *Enable buffering* in the *FactoryTalk Historian SE Installation and Configuration Guide*.

  Tip: Enabling the buffer subsystem is a recommended step during the installation of the FactoryTalk Historian Live Data Interface. To take advantage of this feature, you must install FactoryTalk Historian Live Data Interface on a separate machine from the FactoryTalk Historian SE.

- If you are configuring the **Disconnected Startup** option.
  For more information, refer to (Unint) Disconnected Startup in *Interface Configuration Utility Help*.

  Tip:
  - To open the Interface Configuration Utility (ICU) online help, run Interface Configuration Utility, and select **Help** > **Contents** and Index from the main menu.
  - If you create FactoryTalk Historian Live Data Interface using the Interface Configuration Utility, it will not appear in the list of interfaces in the FactoryTalk Administration Console. As a result, you will not be able to configure the interface from the FactoryTalk Administration Console.

Configure FactoryTalk Historian Live Data Interface using the FactoryTalk Administration Console

Refer to *Configure FactoryTalk Historian Live Data Interface* in the *FactoryTalk Historian SE Installation and Configuration Guide* for information on configuring FactoryTalk Historian Live Data Interface using the FactoryTalk Administration Console. This section also includes steps on how to enable buffering on the remote interface.

You can configure FactoryTalk Historian Live Data Interface on your local computer (the computer on which the FactoryTalk Historian SE server is installed). However, we recommend that you always install FactoryTalk Historian Live Data Interface on the computer that has the data server installed. For more information, refer to *Verify the FactoryTalk Historian Live Data Local Interface* in the *FactoryTalk Historian SE Installation and Configuration Guide*.

Configure FactoryTalk Historian Live Data Interface using the Interface Configuration Utility

If you configure an interface in the Interface Configuration Utility (ICU), the batch file of the interface (**FTLDInt.bat**) will be maintained by the ICU, and all configuration changes will be kept in that file.

Tip: For more information on the Interface Configuration Utility, refer to the *PI-Interface-Configuration-Utility-(PI-ICU)-1.5.1-User-Guide.pdf*, available in **Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE\Historian Server**.

To configure FactoryTalk Historian Live Data Interface with the ICU

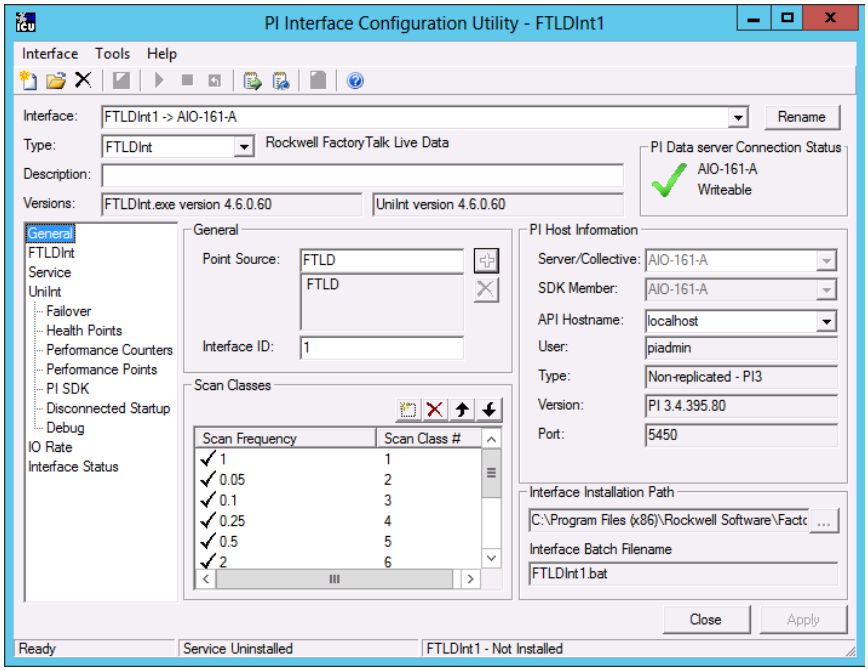
1. Run the Interface Configuration Utility.

   Tip: To open the Interface Configuration Utility, search for Interface Configuration Utility using the search feature in the **Start** menu or the **Start** screen, depending on the version of Microsoft Windows that you use.

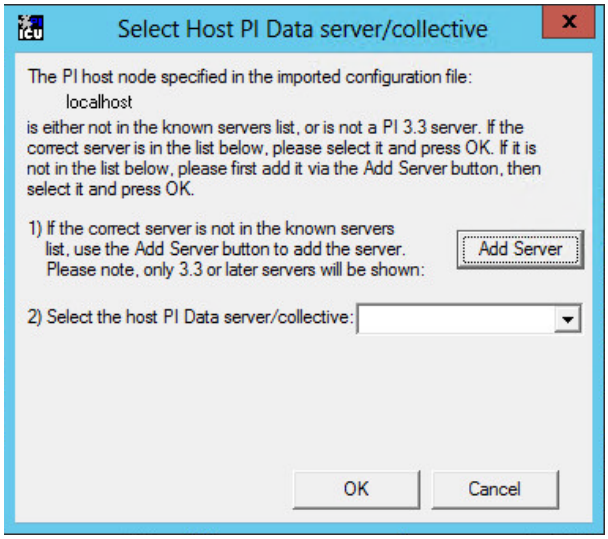
2. On the **Interface** menu, click .

   The **Open Interface Configuration File** dialog box appears.

3. If you're not already there, navigate to **[Drive letter]:\Program Files (x86)\Rockwell Software\FactoryTalk Historian\PIPC\Interfaces\LDInterface**.
4. Select the **FTLDInt_FTLD.bat.bak** file and click **Open**. The FactoryTalk Historian Live Data Interface configuration is displayed on the **General** page of the ICU.



If you are performing these steps on a computer on which no FactoryTalk Historian server is installed, the following dialog box may appear:



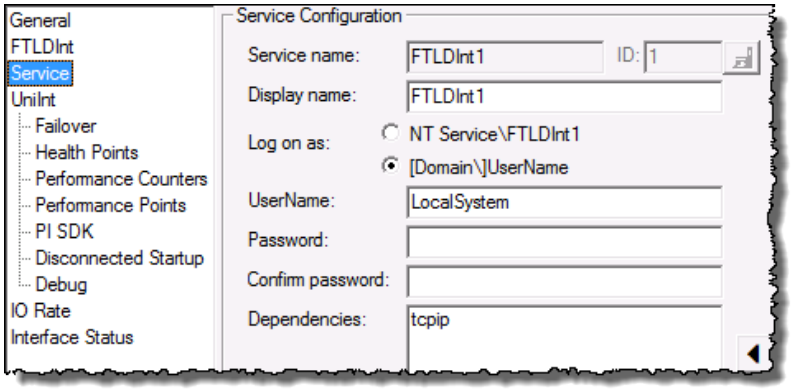
5. (Optional.) Add your FactoryTalk Historian server, set the host PI Data server, and then click **OK**.
6. If not already chosen, from the **Type** list, select **FTLDInt**.
7. (Optional.) If you are adding another interface, change the value in the **Interface ID** box.
8. If you want to use a custom point source value for your interface, edit the **Point Source** box.

   For details, see Use multiple FactoryTalk Live Data point source values on page 21.
9. If necessary, edit the **Scan Frequency** and **Scan Class** values.

Item	Description
Scan Frequency	Indicates the frequency at which the interface reads values from the FactoryTalk Live Data server.
Scan Class #	Associated with the **Scan Frequency** value. If there is no interface defined, the Scan Class # will be displayed without a time period.

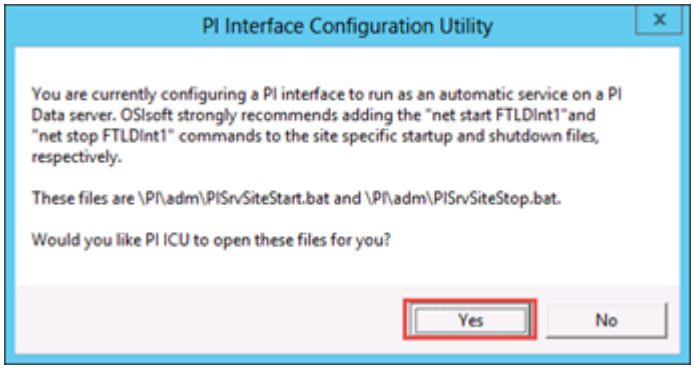
10. Click **Apply**.
11. In the left pane, click **Service**.
12. Under **Service Configuration**, in the **Display name** box, change the name from PI-FTLDInt<Number> to `FTLDInt<Number>`.
13. For the **Log on as** options, select **[Domain\]UserName**.
14. In the **UserName** box, enter `LocalSystem`.

You should get the following configuration:



15. Click **Create**.

    The following message may appear:



16. Read the message, click **Yes**, and then add the commands to the open files, as recommended in the message.

In this example, the **API Hostname** is set to `localhost`, which means that the interface will be configured to communicate with the local FactoryTalk Historian SE server.

If you want the interface to communicate with a remote FactoryTalk Historian server, you must:

1. Add the remote server to the Connection Manager on page 12.
2. Define the remote server as the API host on page 13.

Tip: This step is required to achieve a connection to the correct remote server.

Once you have added the server to the Connection Manager and defined it as the API host, you can select the server from the **API Hostname** list on the **General** page in the ICU.

There are additional parameters available for configuring the interface:

- /MultiCOM
- /uiDll
- /FTDirectory
- /FTContext

To view them, click **FTLDInt** in the left pane of the dialog box.

See Command-line parameters on page 26 for more information.

Add remote servers to the Connection Manager

Perform these steps if you didn't add a FactoryTalk Historian server in Configure FactoryTalk Historian Live Data Interface using the Interface Configuration Utility on page 33.

To add a remote server to the Connection Manager

1. In the ICU, select **Interface** > **SDK Connections**.
2. In the **Connection Manager** dialog box, select **Server** > **Add Server**.

3. In the **Network Node** text box, enter the name of the remote server, and click **OK**.
4. In the **Connection Manager** dialog box, click **Save**.

Define remote server as API host

Perform these steps if you didn't set a FactoryTalk Historian server as the API host in Configure FactoryTalk Historian Live Data Interface using the Interface Configuration Utility on page 9.

To use a remote server as the API host

1. In the ICU, select **Interface** > **SDK Connections**.
2. On the **Tools** menu, click **Options**.
3. From the **Default Server** list, select the remote server you want to use as the API host, and then click **OK**.
4. In the **Connection Manager** dialog box, click **Save**.

Point source

The FactoryTalk Historian Live Data Interface uses FactoryTalk Live Data as a point source. A point source is a unique, single- or multi-character string that is used to identify a Historian point as a point that belongs to a particular interface.

For example, the string Boiler1 may be used to identify points that belong to the MyInt Interface. To implement it, the Point Source attribute would be set to Boiler1 for every Historian point that is configured for the MyInt interface. Then, if */ps=Boiler1* is used on the startup command line of the MyInt interface, the interface will search the Historian Point Database upon startup for every Historian point that is configured with a point source of Boiler1.

Before an interface loads a point, it also examines additional Historian point attributes to determine whether a particular point is valid for the interface. For additional information, see Command-line parameters on page 26.

PI 3 server node: reserved point sources

Several subsystems and applications that are shipped with the PI System are associated with default point source characters:

Subsystem	Point source character
Totalizer	T
Alarm	G and @
Random	R
RampSoak	9
Performance Equations	C

Do not use these point source characters or change the default point source characters for these applications.

If a point source character is not explicitly defined when creating a PI point, the point is assigned a default point source character of *Lab* (PI 3). Therefore, do not use Lab as a point source character for an interface to avoid confusion.

IMPORTANT: Do not use a point source character that is already associated with another interface program. However, it is acceptable to use the same point source for multiple instances of an interface.

Point configuration

A FactoryTalk Historian point is the basic building block for controlling data flow to and from the FactoryTalk Historian SE server. A single point is configured for each measurement value that must be archived.

Point attributes

Historian points (tags) have approximately 50 attributes. These attributes define how data is to be collected and stored for the point. The proper configuration of these attributes is the key to optimizing the FactoryTalk Historian server for both data storage efficiency and quick retrieval. Each FactoryTalk Historian interface handles specific point attributes differently.

A tag is a label or name for a Historian point.

Tip:

The FactoryTalk Historian documentation uses the terms tag and point to mean the same thing. An example of a tag name is:

```
FTLDEnt:rc:RSLinxC.secondArray000
```

The information presented in this chapter is necessary to define FactoryTalk Historian Live Data points for use with a FactoryTalk Live Data server. Failing to configure FactoryTalk Historian data points correctly will result in poor or no communication between the interface and the FactoryTalk Live Data server. For more information on errors that may occur, see Error and informational messages on page 35.

Length

The Tag field allows a maximum of 1023 characters.

Point source

A point source is a unique single or multiple-character string that is used to identify the PI point as a point that belongs to a particular interface. The point source for FactoryTalk Historian Live Data Interface is `FTLD`. For additional information, see the `/ps` command-line parameter description in Command-line parameters on page 26.

Point type

Typically, device point types do not need to correspond to Historian point types. For example, integer values from a device can be sent to floating point or digital Historian tags. Similarly, a floating point value from the device can be sent to integer or digital Historian tags, although the values will be truncated.

PI 3 server nodes

The following point types are supported on PI 3 servers:

- float16
- float32
- float 64

- int16
- int32
- digital
- string

Tip: For more information on the individual point types, refer to the *PI-Data-Archive-2018-SP3-Patch-3-Reference-Guide-EN.pdf*, available in **Program Files (x86)\Common Files\Rockwell \Help\FactoryTalk Historian SE\Historian Server.**

Location1

Location1 indicates to which copy of the interface the point belongs. The value of this attribute must match the `/id` startup parameter. The default value for the FactoryTalk Historian Live Data Local Interface is `1`.

Location2

Location2 is not used for FactoryTalk Historian Live Data Interface.

Location3

Location3 is used to define a data collection mode.

Item	Description
0 - Polled or Event	Data is collected based on the scan rate and saved in the buffer. When the Historian server requests data, the data stored in the buffer is sent to the server. In this method, the values being sent to the Historian server are from a buffer, so they may not represent the exact (current) values in the controllers.
1 - Advised (default)	Data is collected only when a value changes in the controller. It is not based on the scan rate. It is the default data collection method and is the most efficient because data is sent to the Historian server only when the value changes.
2 - Output	Data is written back to the Live Data server such as FactoryTalk Linx, the HMI server such as FactoryTalk View, or the OPC server such as RSLinx Classic. Use this method when you want to write data points back to the data servers.

Location4

Location4 defines the scan class for the Historian point. The scan class determines the frequency at which input points are scanned for new values. For more information, see the description of the `/f` parameter in Command-line parameters on page 26.

The updates from the FactoryTalk Live Data server come in groups: at startup, the interface defines a group on the Live Data server and adds all points within the given scan class to the group. The FactoryTalk Live Data server is queried for all points within a group at the same time; therefore, some consideration should be given to the creation of scan classes. Having more than one scan class with the same scan period is allowed, and using different offsets on those scan classes may improve performance.

Advised and Polled tags

Advised tags and Polled tags use Location4 to specify the requested update rate for the group.

Output tags

Location4 is ignored for Output tags.

Tip: Advised, Polled, and Output data collection methods are explained in Location3 on page 16.

Location5

If `Location5=1` and `Location3=0`, it will force an asynchronous read from the data server. It should only be used for event-triggered points due to performance concerns.

InstrumentTag

The InstrumentTag contains the ItemID of the tag. The format of this field depends on the FactoryTalk Live Data server:

Type of the tag	InstrumentTag syntax
Device tags	<application name>/<area name>:<data server name>:[<shortcut name>]<tag name>
HMI tags	<application name>/<area name>:<HMI server Name>:<folder name>\<tag name>

The field must match the point defined on the FactoryTalk Live Data server, including punctuation, spaces, and case.

The InstrumentTag field allows a maximum of 1023 characters.

FactoryTallk Historian Live Data Interface gets the $Global scope from the `/FTDirectory` parameter and / from the `/FTContext` parameter. For additional information on these parameters, see Command-line parameters on page 26.

ExDesc

The ExDesc (Extended Descriptor) is a string attribute. Typically, this attribute is used to implement Trigger Input points.

For example: If a PI point has the ExDesc attribute `"EVENT='Tag1' Anychange,"` it means trigger on any change as long as the Tag1 value of the current event is different from the value of the previous event.

The ExDesc field allows a maximum of 1023 characters.

SourceTag

An output point is associated with a trigger point by setting the SourceTag attribute of the output point equal to the tag name of the trigger point.

For more information, see Output points on page 19.

Compression testing

For each data point, you can set the following attributes to configure its compression testing specification:

Item	Description
Compression Deviation (CompDev)	Specifies in engineering units how much a value may differ from the previous value before it is considered to be a significant value. As a rule of thumb, set CompDev to the precision of the data source or hardware (instrument). Set it a little "loose" to err on the side of collecting, rather than losing data. After collecting data for a while, go back and check the data for your most important tags, and then adjust CompDev, if necessary. **Note:** Setting the CompDev attribute value too low causes too little data compression, and wastes space in the archive. Setting the value too high causes the loss of useful data. For most flows, pressures, and levels, use a deviation specification of 1% or 2% of span. For temperatures, the deviation should usually be 1 degree or 2 degrees.
Compression Minimum (CompMin)	A point is archived if the elapsed time since the previous time that the point was saved is greater than or equal to the minimum time, and the value has changed by more than the deviation. For data points associated with interfaces that send exception reports, set CompMin to 0.
Compression Maximum (CompMax)	A point is archived if the elapsed time since the previous time that the point was saved is greater than the maximum time. The recommended maximum time specification is one work shift (for example, 8 hours). Duplicate values will be archived if the elapsed time exceeds CompMax. You typically set CompMax to the same value for all points in the system.

Tip: For information on compression testing, refer to *Exception Reporting and Compression Testing* in *PI-Data-Archive-2018-SP3-Patch-3-Reference-Guide-EN.pdf*, available in **Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE\Historian Server**.

Exception reporting

For each data point, you can set the following three attributes to configure its exception reporting specification:

Item	Description
Exception Deviation (ExcDev):	Specifies in engineering units how much a point's value must change before the interface considers it a significant value and sends it to the server. As a general rule, you should set the exception slightly smaller than the precision of the instrument system.
Exception Minimum (ExcMin):	Specifies a limit on how frequently the interface can report values to the server. For example, if you want the interface to wait full ten minutes before reporting a new value to the server, then you would set the ExcMin attribute to ten minutes. ExcMin is typically set to zero.
Exception Maximum (ExcMax):	Specifies a limit on how long the interface can go without reporting a value to the Historian server. After the ExcMax time period, the interface sends the next new value to the server, regardless of whether the new value is different from the last reported value.

Tip: For information on exception reporting, refer to *Exception Reporting and Compression Testing* in *PI-Data-Archive-2018-SP3-Patch-3-Reference-Guide-EN.pdf*, available in **Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE\Historian Server**.

Output points

Output points control the flow of data from the Historian server to any destination that is external to the server, such as the FactoryTalk Live Data server. FactoryTalk Historian Live Data Interface uses `Location3=2` to indicate an output point.

Outputs are triggered for UniInt-based interfaces. That is, outputs are not scheduled to occur periodically. There are two mechanisms for triggering an output, as described in the sections that follow.

Trigger Method 1 (recommended)

For Trigger Method 1, a separate trigger point must be configured. The output point must have the same point source as the interface. The trigger point can be associated with any point source, including the point source of the interface. Also, the point type of the trigger point does not need to be the same as the point type of the output point.

The output point is associated with the trigger point by setting the SourceTag attribute of the output point equal to the tag name of the trigger point. An output is triggered when a new value is sent to the snapshot of the trigger point. The new value does not need to be different than the previous value that was sent to the snapshot to trigger an output, but the timestamp of the new value must be more recent than the previous value.

If no error is indicated, then the value that was sent to the trigger point is also written to the output point. If the output is unsuccessful, then an appropriate digital state that is indicative of the failure is written to the output point. If an error is not indicated, the output still may not have succeeded because the interface may not be able to tell with certainty that an output has failed.

Trigger Method 2

For Trigger Method 2, a separate trigger point is not configured. To trigger an output, write a new value to the snapshot of the output point itself. The new value does not need to be different than the previous value to trigger an output, but the timestamp of the new value must be more recent than the previous value.

Trigger Method 2 may be easier to configure than Trigger Method 1, but Trigger Method 2 has a significant disadvantage. If the output is unsuccessful, there is no tag to receive a digital state that is indicative of the failure, which is important for troubleshooting.

Sample tag configurations

See the following sample tag configurations:

- Scan classes on page 20
- Polled tags on page 20
- Advised tags on page 20
- Event tags on page 20

Scan classes

Scan classes are defined in the startup file. Each /F= parameter defines a scan class, which is numbered in order.

For example, if the .bat file reads

```
/F=2 /F=1:00 /F=1:30:00 /F=00:00:05,00:00:01
```

then these scan classes have been defined:

- Scan Class 1 has a scan period of 2 seconds.
- Scan Class 2 has a scan period of 60 seconds.
- Scan Class 3 has a scan period of 5400 seconds (90 minutes).
- Scan Class 4 has a scan period of 5 seconds, with an offset of 1 second.

Polled tags

Polled tags are read once every scan period. To set up a polled tag, set `Location1` to match the `/ID` parameter, `Location3=0`, and `Location4=scanclass#`.

For example:

Tag	InstrumentTag	Loc1	Loc2	Loc3	Loc4	Loc5
FiveSec.PV	ItemID1	1	0	0	1	0
OneMin.PV	ItemID2	1	0	0	2	0
NinetyMin.PV	ItemID3	1	0	0	3	0

Advised tags

For Advised tags, the interface asks the FactoryTalk Live Data server to send data when it changes, and how often it should read the device to see if there is a new value.

For example:

Tag	InstrumentTag	Loc1	Loc2	Loc3	Loc4	Loc5
AdvFiveSecs.PV	ItemID1	1	0	1	1	0
AdvOneMin.PV	ItemID2	1	0	1	2	0
AdvNinetyMins .PV	ItemID3	1	0	1	3	0

Event tags

Event tags are read when the triggering event occurs. An event happens when the FactoryTalk Historian snapshot receives a value for the trigger tag. It may have the same timestamp and quality and value as the last event so the snapshot value for that trigger may seem the same, but the act of receiving a value for the trigger tag causes the interface to receive a notification that the trigger has been updated.

To configure triggered input tags, specify the name of the trigger tag in the ExDesc field using the following format:

```
EVENT='triggertagname' event_condition
```

where `triggertagname` is enclosed in single quotes and, if specified, the `event_condition` immediately follows the `triggertagname`. If the `event_condition` is not specified, then it defaults to `Anychange`.

The update rate for event item groups is also related to the scan class, so the server will be asked to update its cache once every scan period for every event tag defined. This is probably faster or slower than necessary. You must set the Loc4 attribute to make event tags work well. The Location 5 attribute should have the value 1 for Event tags.

Typical example:

Tag	ExDesc	Instrument Tag	Loc1	Loc2	Loc3	Loc4	Loc5
PM1_Temp.PV	EVENT='PM1_Trigger	ItemID1	1	0	0	2	1
PM1_Rate.PV	EVENT='PM1_Trigger	ItemID2	1	0	0	3	1

In this case, PM1_Trigger are tags that are updated by this interface, by another interface, or by manual entry. When a PM1_Trigger gets a new event in the PI snapshot, the interface will send data for both PM1_Temp.PV and PM1_Rate.PV to the PI server.

Use multiple FactoryTalk Historian Live Data point source values

You can edit each FactoryTalk Historian Live Data Interface point source and make it a unique number so that only the data points that match that unique point source get sent from the FactoryTalk Historian server.

To use custom point source values

1. Open FactoryTalk Administration Console.
2. In the Explorer, go to **Network > System > Connections > Historical Data > Production Historian**.
3. Right-click **Production Historian**, and then click **Properties**.
4. Click the **Point Sources** tab.
5. Under **Interface Type Allocation**, enter `FTLDx`, where `x` is a point source value ranging from 1 to 99.
   For example, FTLD4
6. Click **Apply**.
7. Restart the FactoryTalk Historian SE server to synchronize the licenses:
   a. Using Windows Search, find and run **Stop FactoryTalk Historian SE**.
   b. When the server is stopped, find and run **Start FactoryTalk Historian SE**.

   Tip: If you don't want to restart the server, you may continue with the next steps without the restart. The licenses will be synchronized in up to 20 minutes.

8. Open the Interface Configuration Utility.
9. In the **Interface** list, select your FactoryTalk Historian Live Data interface.
10. Under **General**, change **Point Source** from FTLD to the FTLDx that you have configured in FactoryTalk Administration Console (step 5).
    For example, FTLD4.
11. Click **Apply**.
12. Restart the interface ( ).

13. (optional) If you are using redundant interfaces, both the primary and secondary interface should use the same point source that you have configured (for example, FTLD4).
14. Edit the data points so that their point source matches your custom point source value (for example, FTLD4).

The I/O Rates tag configuration

The I/O Rates tag measures the throughput of a FactoryTalk Historian Live Data Interface. In particular, the value of an I/O Rate point represents a 10-minute average of the total number of values per minute that FactoryTalk Historian Live Data Interface sends to the FactoryTalk Historian server. Because values are averaged over a 10-minute interval, the first calculated value is not written to the Historian server earlier than 10 minutes after the interface has started. You can configure one I/O Rates tag for each copy of FactoryTalk Historian Live Data Interface that is in use.

Tip: The Historian system documentation often uses the terms Event Counter Tag and I/O Rate Point synonymously.

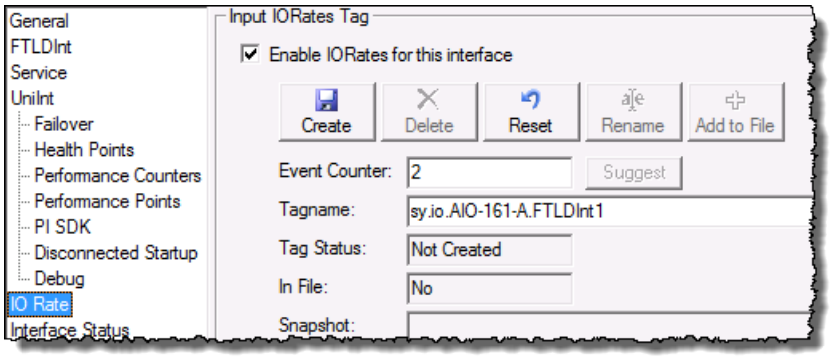
Monitor the I/O Rates tag on the interface node

For Windows nodes, the 10-minute rate averages (in events/minutes) can be monitored with a client application such as FactoryTalk Historian Vision.

Configure the I/O Rates tag with ICU

The Interface Configuration Utility (ICU) provides a user interface for creating and managing the I/O Rates tag.

To access the I/O Rates tag data in ICU, select **IO Rate** from the left pane of the ICU dialog box.



ICU currently allows for one I/O Rates tag to be configured for each copy of the interface that is in use. Some interfaces allow for multiple I/O Rates tags.

The **Input IORates Tag** section contains the following elements:

Item	Description
Enable IORates for this interface	Select the checkbox to enable I/O Rates for the selected interface. Clear the checkbox to disable I/O Rates for the selected interface.
Create	Click it to create the suggested I/O Rates tag with the tag name indicated in the **Tagname** text box.
Delete	Click it to delete the I/O Rates tag listed in the **Tagname** text box.
Reset	Click it to reset the I/O Rates **Event Counter** and **Tag** settings.
Rename	Click it to change the name of the I/O Rates tag using the **Rename IORates Tag** dialog box.

Item	Description
Add to File	Click it to add the tag to the **IORates.dat** file with the event counter listed in the **Event Counter** text box.
Event Counter	The Event Counter correlates a tag specified in the **iorates.dat** file with this copy of the interface. The command-line equivalent is `/ec=x`, where *x* is the same number that is assigned to a tag name in the **iorates.dat** file.
Suggest	• In the ICU dialog box, click it if you want the system to suggest the next available Event Counter. The button is active if the Event Counter text box is empty or contains an illegal value. • In the **Rename IORates Tag** dialog box, click it if you want the system to suggest a tag name.
Tagname	Type the name of the I/O Rates tag or click the **Tag Search** icon to find the tag using the **Tag Search** dialog box.
	Tag Search. Click it to find the I/O Rates tag for the interface using the **Tag Search** dialog box.
Tag Status	Indicates whether the I/O Rates tag exists in the Historian server. The text box may have the following values: • **Created** The tag exists on the Historian server. • **Not Created** The tag does not yet exist on the Historian server. • **Deleted** The tag has been deleted from the Historian server. • **Unknown** The ICU is not able to access the Historian server.
In File	Indicates whether the I/O Rates tag listed in the tag name and the event counter is in the **IORates.dat** file. It may have the **Yes** or **No** values.
Snapshot	Holds the snapshot value of the I/O Rates tag, if the I/O Rates tag exists in the FactoryTalk Historian server. The value of the text box is updated when you click **IO Rate** in the left pane of the ICU dialog box, and when the interface is first started. You can refresh it manually by clicking the **Refresh snapshot** icon.
	Refresh snapshot. Click it to refresh the snapshot.

Performance Point configuration

Performance Point tags document how long it takes to complete a scan. Due to the architecture of this interface, the performance point tags are not valid - the server's response is asynchronous, so the time to scan bears no relation to the amount of time it may take to get the data from the server.

Startup command file

In Windows, command file names have a **.bat** extension. The Windows continuation character (^) allows for the use of multiple lines for the startup command. The maximum length of each line is 1024 characters (1 kilobyte). The number of parameters is unlimited, and the maximum length of each parameter is 1024 characters.

Command-line parameters should begin with a "/" character. For example, `/ps=M`.

The Interface Configuration Utility (ICU) provides a tool for configuring the FactoryTalk Historian Live Data Interface startup command file.

Command-line parameters

The following table lists command-line parameters and their descriptions.

For the complete list of the parameters, see *PI-Universal-Interface-(UniInt)-Framework-User-Guide.pdf*, available in **Program Files (x86)\Common Files\Rockwell\Help\FactoryTalk Historian SE\Historian Server**.

IMPORTANT: We recommend that you always use the Interface Configuration Utility to modify the startup file. If you manually change the startup file and then open it using the Interface Configuration Utility, the utility will rewrite all the startup parameters in the file.

Required parameters:

Parameter	Description
/ps=Source	The `/ps` parameter specifies the point source for the interface. The Source value is case-insensitive. The length of the Source value is limited to 100 characters by the PI-UniInt. The value can contain any character except `'*'` and `'?'`. The point source that is assigned with the `/ps` parameter corresponds to the Point Source attribute of the individual Historian point. The interface will attempt to load only those Historian points that have the appropriate point source. Strategies for assigning a point source character vary depending on the interpretation of the `/id` parameter by a particular interface. See the `/id` parameter description for more information.
/id=x	Example: `/id=1` The `/id` parameter specifies the interface identifier. The interface identifier is a string that is no longer than 9 characters. FactoryTalk Historian Live Data Interfaces also use the `/id` parameter to identify a particular interface copy number that corresponds to an integer value that is assigned to one of the Location code point attributes, most frequently Location1. For these interfaces, you should use only numeric characters in the identifier.
/host=host:port	The `/host` parameter specifies the PI Home node. Host is the IP address of the PI Server node or the domain name of the PI Server node. Port is the port number for TCP/IP communication. The port is always 5450.
/uiDLL=FTLDIntCtl.dll /uiDLL="c:\Program Files\PIPC\Interfaces\PIUniint\FTLDIntCtl.dll"	The /uiDLL parameter is used to specify the DLL file name of FactoryTalk Historian Live Data Interface. The DLL file is installed in the \Windows\System32\ or Windows\SysWOW64\ directory. If the DLL file is located elsewhere, the full path to the DLL must be provided in the /uiDLL parameter.

Parameter	Description
/FTDirectory=$Global	The `/FTDirectory` parameter specifies the FactoryTalk directory. It can only be set to `$Global`.
/FTContext="/"	This parameter is reserved. Set to "`/`".

/f=SS
or
/f=SS,SS
or
/f=HH:MM:SS
or
/f=HH:MM:SS,hh:mm:ss
	Required for reading scan-based inputs.
	The `/f` parameter defines the time period between scans in terms of hours (HH), minutes (MM), and seconds (SS). The scans can be scheduled to occur at discrete moments in time with an optional time offset specified in terms of hours (hh), minutes (mm), and seconds (ss). If HH and MM are omitted, then the time period that is specified is assumed to be in seconds.
	Each instance of the `/f` parameter on the command line defines a scan class for the interface. The first occurrence of the `/f` parameter on the command line defines the first scan class of the interface; the second occurrence defines the second scan class, and so on. Historian points are associated with a particular scan class via the Location4 point attribute. For example, all points that have Location4 set to 1 will receive input values at the frequency defined by the first scan class. Similarly, all points that have Location4 set to 2 will receive input values at the frequency specified by the second scan class, and so on.
	Two scan classes are defined in the following example:
	`/f=00:01:00,00:00:05 /f=00:00:07`, or, equivalently:
	`/f=60,5 /f=7`
	The first scan class has a scanning frequency of 1 minute with an offset of 5 seconds, and the second scan class has a scanning frequency of 7 seconds with no offset.
	When no offset is specified, the scan class will be scheduled for immediate execution. That is, the interface will not wait for a well-defined moment in time before scanning when no offset is specified.
	One can also specify subsecond scan classes on the command line such as:
	`/f=0.5 /f=0.1`

Optional parameters:

Parameter	Description
/maxstoptime=stoptime	Default = 120 seconds. When an interface receives a signal from the operating system to shut down, it must perform a number of cleanup functions. If for some reason the execution of these functions takes longer than the stop time, the interface will shut down without finishing these functions.
/MultiCom	Use the `/MultiCom` parameter if you want the PI-UniInt executable to select the Multithreaded Threading model (`COINIT_MULTITHREADED`) for the call to `CoInitializeEx()`. Otherwise, the PI-UniInt will use the Apartment Threaded model (`COINIT_APARTMENTTHREADED`).
/perf=interval	Default = 8 hours When the percentage of scans that an UniInt-based interface performs on time drops below 95%, UniInt will write the performance summaries for each scan class into the PIPC.log file. For example, if `/perf=0.025`, UniInt will write performance summaries every 90 seconds if the percentage of on-time scans is below 95%. The minimum time between summaries is 60 seconds. Setting `/perf=0` disables summaries. If the `/perf` parameter is omitted, then by default, every 8 hours, UniInt checks whether summaries are needed.

Parameter	Description
	If the inputs for the interface are unsolicited, then performance summaries should be disabled by setting `/perf=0`, because performance summaries are meaningless for unsolicited input points.
/PISDK=#	Default = 0. The `/pisdk` parameter can be used to enable or disable the PI SDK. • Use `/pisdk=1` to enable the PI SDK. • Use `/pisdk=0` to disable the PI SDK.
/q	Default = no queuing. When the `/q` parameter is present, Snapshots and exceptions are queued before they are sent to the PI Server node. The maximum queue size is close to 4000 bytes. The queue is flushed between scans if it is not filled. For an interface collecting unsolicited data, the queue is flushed four times a second if it is not filled.
/sio	Default = send initial outputs. The `/sio` parameter stands for suppressing initial outputs. The parameter applies only to interfaces, such as the FactoryTalk Live Data, that support outputs. If the `/sio` parameter is not specified, FactoryTalk Historian Live Data Interface will behave in the following manner: when FactoryTalk Historian Live Data Interface is started, it determines the current Snapshot value of each output tag. Next, FactoryTalk Historian Live Data Interface writes this value to each output tag. In addition, whenever an individual output tag is edited while FactoryTalk Historian Live Data Interface is running, it will write the current Snapshot value to the edited output tag. This behavior is suppressed if the `/sio` parameter is specified on the command line. That is, outputs will not be written when FactoryTalk Historian Live Data Interface starts or when an output tag is edited. In other words, when the `/sio` parameter is specified, outputs will only be written when they are explicitly triggered.

/TA=<Any positive value>	Use the Trend Advise (`/TA`) parameter for advice tags to send the value from the preceding scan if the new value's timestamp is greater than the preceding scan value's timestamp plus the number of scan periods multiplied by the **TA** parameter.
	Example: `/TA=1.0`
	The event from the preceding scan is resent using the following algorithm:
	`If ((Tnew - Tprev) > (/TA * R)`
	`{`
	`send previous value at Tnew - R before sending new value at Tnew`
	`}`
	Where:
	`Tnew` = new timestamp (the timestamp of the received event)
	`Tprev` = previous timestamp (the timestamp of the event from the preceding scan)
	`R` = scan rate
	Tip: The same functionality is provided by the OPC DA Interface. For details, see the OPC DA user documentation.

/TSofPIorFT	The `/TSofPIorFT` parameter specifies which timestamp will be used to determine when to send data to the Historian server: • `/TSofPIorFT=0` means using server timestamp • `/TSofPIorFT=1` means using FactoryTalk Live Data server timestamp

Parameter	Description
	Notes: By default, there is no `/TSofPIorFT` parameter in the .bat file. In this case, the Historian server timestamp will be used. An unsolicited point will always use FactoryTalk Live Data Server timestamps.
/US	With the Update Snapshot (`/US`) parameter enabled, if the current snapshot is a system digital state (such as **I/O timeout** and **Shutdown**) and the interface reads in a new value that is older than the snapshot, the interface sends the new value one second after the snapshot timestamp of the system digital state. This check is omitted if the current snapshot is a good value. This is useful for setpoints that rarely change. **Tip:** The same functionality is provided by the OPC DA Interface. For details, see the OPC DA user documentation.

Sample FTLDInt.bat file

The following is a sample startup command file that comes with the installation:

```
".\FTLDInt.exe" /MultiCOM /PS=FTLD /ID=1 /host=ENK2012R2TEST:5450 /pisdk=0 /
maxstoptime=120 /sio /perf=8 /uiDll=FTLDIntCtl.dll /q /FTDirectory=$Global /
FTContext="/" /f=1 /f=0.05 /f=0.1 /f=0.25 /f=0.5 /f=2 /f=5 /f=10 /f=60 /f=120
```

Setting file

For more flexibility, FactoryTalk Historian Live Data Interface can use an INI format file to configure interface information. This format file does not install with FactoryTalk Historian Live Data Interface, but you can create the file manually. It should be named **FTLDInt.ini** and it must be created on the same path as the **FTLDInt.exe** file and the **FTLDIntCtl.dll** file.

The following is a sample setting file:

```
[FTLDIntSetting]

OnceMaxUnsolEvents=4

ScanClassToUpdateRate=1

PIOrFTLDTimestamp=0

FTLDResponseInterval=5000
```

Review the following for additional information.

Item	Description
OnceMaxUnsolEvents	The OnceMaxUnsolEvents key specifies the maximum number of unsolicited events that FactoryTalk Historian Live Data Interface can process in each loop for each Advised tag. The default value for OnceMaxUnsolEvents key is 4. That is, the interface can process 4 events for each Advised tag when the interface collects data.
ScanClassToUpdateRate	For polled points, FactoryTalk Historian Live Data Interface calculates the update rate from its scan time. The update rate is used while adding an item to the FactoryTalk Live Data server. The algorithm is `"Update Rate = Scan Time / ScanClassToUpdateRate"`

Item	Description
	For example, if a point has a 1-second scan period and the ScanClassToUpdateRate is 2, FactoryTalk Historian Live Data Interface will use 500 ms as the update rate.
PIOrFTLDTimestamp	The `PIOrFTLDTimestamp` key specifies which timestamp will be used to determine when to send data to the FactoryTalk Historian server. `PIOrFTLDTimestamp=0` means the FactoryTalk Historian server timestamp is used, and 1 means the FactoryTalk Live Data server timestamp is used. This setting is similar to the `/TSofPIorFT` command-line parameter. See Command-line parameters on page 26 for more information.
FTLDResponseInterval	The `FTLDResponseInterval` key specifies the time interval during which FactoryTalk Historian Live Data Interface checks for data points that were marked 'bad_quality' by the FactoryTalk Live Data service. The FactoryTalk Live Data service will mark a data point 'bad_quality' if the data point links to an incorrect item (data point) in the FactoryTalk Diagnostics, or if the FactoryTalk Live Data service was not able to receive a data point from FactoryTalk Historian Live Data Interface because several other data points were being added to the FactoryTalk Historian server at that time. The period can be set in milliseconds (ms). The default is 5000 ms.

Interface node clock

Make sure that the time and time zone settings on the computer are correct. Check the settings in the **Date and Time** program of Control Panel. If the locale where the interface node resides observes Daylight Saving Time, select **Automatically adjust clock for Daylight Saving Time** in **Time Zone Settings**.

Make sure that the TZ environment variable is not defined on the computer. To check it, enter `set` in the Command Prompt window. All the environment variables currently defined in Windows will be listed. If the TZ environment variable is defined, remove it using the **System** item of Control Panel.

Tip:

It is possible for computer nodes to startup with different clock times before synchronizing to the server time. If synchronization is left to the Microsoft defaults, it can take several minutes for the system to synchronize all node clocks in the network. To minimize clock synchronization time, you can modify the registry on the computer nodes in the FactoryTalk Historian SE system to adjust local clocks to the server time. To do this, set the MaxAllowedPhaseOffset entry to 1 on every node in the FactoryTalk Historian SE system. The entry resides in the following subkey:

```
My Computer\HKEY-_LOCAL_MACHINE\System\ControlSet001\services\W32Time
\Config.
```

IMPORTANT: We recommend that only advanced users modify the registry. Refer to the system documentation for more information on the Windows registry.

Security

The trust database or security mappings are required to establish connections between the FactoryTalk Historian SE server and any remote computer that should be able to communicate with the server so that FactoryTalk Historian Live Data Interface is allowed to write data to the FactoryTalk Historian server.

If FactoryTalk Historian Live Data Interface cannot write data to the FactoryTalk Historian server because it has insufficient privileges, a `10401` error will be reported in the PI Message Log.

Tip: It is highly recommended to use security mapping. For details, refer to *Securing the Historian server* in the *FactoryTalk Historian SE Installation and Configuration Guide.*

For details on reading the message log, see Rockwell Automation Knowledgebase Document ID: 1129979-How to read new UniInt Interface message logs? One of the ways of reading the message logs is to use the **PIGetMsg** utility. The utility is available in the following locations:

C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\adm

Start and stop the interface

Once you have installed FactoryTalk Historian Live Data Interface as a service, you can start and stop it in two ways:

- Using the Interface Configuration Utility (ICU) on page 33
- Using the Administrative Tools program of Control Panel on page 34

The FactoryTalk Historian Live Data Interface service may stop immediately after the startup for a variety of reasons. One of the typical reasons is that the service is not able to find the command-line parameters in the associated **.bat** file. To avoid it, make sure that the root names of the **.bat** file and the **.exe** file are the same, and the files are in the same directory. Usually, they are stored in **[Drive letter]:\Program Files (x86)\Rockwell Software\FactoryTalk Historian\PIPC\Interfaces\LDInterface**.

Further troubleshooting of the service may require consulting the PI Message Log, the **pipc.log** file, Windows Event Viewer, or other sources of log messages. See Error and informational messages on page 35 for additional information.

Tip:

For details on reading the message log, see Rockwell Automation Knowledgebase Document ID: 1129979-How to read new UniInt Interface message logs?.

One of the ways of reading the message logs is to use the **PIGetMsg** utility. The utility is available in the following locations:

C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\adm

Use the Interface Configuration Utility

Perform these steps to use the ICU to start and stop FactoryTalk Historian Live Data Interface.

To start FactoryTalk Historian Live Data Interface

1. Run the Interface Configuration Utility.

   Tip: To open the Interface Configuration Utility, search for Interface Configuration Utility using the search feature in the **Start** menu or the **Start** screen, depending on the version of Microsoft Windows that you use.

2. From the **Interface** list, select the interface you want to start.
3. On the toolbar, click  .
4. Wait until the status on the status bar at the bottom of the dialog box changes to **Running**.

To stop FactoryTalk Historian Live Data Interface

- In the ICU, click  on the toolbar. The service status on the status bar changes to **Stopped**.

You may additionally check the status of the service in the Administrative Tools on page 34 program of Control Panel.

Use the Administrative Tools

Perform these steps to use the Administrative Tools to start and stop FactoryTalk Historian Live Data Interface.

To start FactoryTalk Historian Live Data Interface

1. Open Services.

   Tip: To open Services, search for Services using the search feature in the **Start** menu or the **Start** screen, depending on the version of Microsoft Windows that you use.

2. Right-click **FTLD<X>** (where `X` is the interface number) and select **Start**.
3. Wait until the status of the service changes to `Started`.

To stop FactoryTalk Historian Live Data Interface

- In the **Services** dialog box, right-click **FTLD<X>** and select **Stop.**

Error and informational messages

FactoryTalk Historian Live Data Interface is based on the PI-UniInt framework, therefore a few error messages are sent to the PIPC log by PI-UniInt. Other error messages are sent to the FactoryTalk Diagnostics system. When troubleshooting, we recommend that you check the PI Message Log, FactoryTalk Diagnostics, and the PIPC log.

Tip:

For details on reading the message log, see Rockwell Automation Knowledgebase Document ID: 1129979-How to read new UniInt Interface message logs?.

One of the ways of reading the message logs is to use the **PIGetMsg** utility. The utility is available in the following locations:

C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\adm

The following is the list of error messages sent to FactoryTalk Diagnostics.

Severity	Message text
Error	Failed to disconnect from FactoryTalk.
Error	Failed to initialize COM library.
Error	Failed to initialize COM security.
Error	Missing or invalid interface ID parameter.
Error	Failed to initialize FactoryTalk Diagnostics.
Error	Failed to launch the FactoryTalk Historian Live Data interface due to the lack of the parameter /FTDirectory.
Error	Failed to launch the FactoryTalk Historian Live Data interface due to the lack of the parameter /FTContext.
Error	Failed to connect to FactoryTalk Directory scope %s.
Error	System error: Insufficient memory.
Warning	Failed to remove FactoryTalk Live Data item <%s>.
Warning	PI point <%s> is refused because of failure to add FactoryTalk Live Data item <%s>.
Warning	PI point <%s> is refused because of the invalid attribute of Location [3].
Warning	PI point <%s> is refused because of the invalid attribute of InstrumentTag.
Warning	PI point <%s> is refused because of the invalid attribute of Location [4].
Warning	The value quality of PI point <%s (PointID: %d)> with FactoryTalk Live Data item <%s> is bad.
Warning	Failed to convert PIEvent to FactoryTalk Live Data variant type of data.
Warning	Failed to write value to FactoryTalk Live Data item <%s>.
Warning	Monitor disconnected from FactoryTalk service.
Warning	Monitor reconnected to FactoryTalk service successfully.
Warning	The value quality of PI point <%s (PointID: %d)> with FactoryTalk Live Data item <%s> is bad because the FactoryTalk Live Data service does not respond in %d ms.
Information	PI point <%s> has been removed from the FactoryTalk Historian Live Data interface (InterfaceID: %d).

Severity	Message text
Information	PI point <%s> is edited in the FactoryTalk Historian Live Data interface (InterfaceID: %d).
Information	PI point <%s> has been added to the FactoryTalk Historian Live Data interface (InterfaceID: %d).
Information	Connected to FactoryTalk Directory scope %s successfully.
Information	FactoryTalk Historian Live Data interface(ID:%d) has Scan Class %d = %s.
Information	Disconnected from FactoryTalk Directory %s successfully.

System errors and PI errors

System errors are associated with positive error numbers. Errors related to PI are associated with negative error numbers.

Error descriptions

The descriptions of system and PI errors can be obtained with the `pidiag` utility. It is a command-line utility (**pidiag.exe**) in the **C:\Program Files\Rockwell Software\FactoryTalk Historian\Server\adm** directory on the computer on which the FactoryTalk Historian server is installed.

The following is the syntax to retrieve error descriptions of the messages:

```
\PI\adm\pidiag -e error_num
```

Technical support and resources

Rockwell Automation provides 24/7 dedicated technical support internationally.

You can read complete information about technical support options, and access all of the following resources at the Rockwell Automation Support Web site (https://rockwellautomation.custhelp.com/).

Before you call or write for help

When you contact Rockwell Automation Technical Support, please provide:

- Product name, version, and/or build numbers.
- Computer platform (CPU type, operating system, and version number).
- The time that the difficulty started.
- The message logs at that time. Consult your product documentation on the location of the message log files.

Find the version and build numbers

To find version and build numbers for each Historian Server subsystem (which vary depending on installed upgrades, updates, or patches), use either of the following methods:

To check the numbers with System Management Tools (SMT)

1. Search for **System Management Tools** in Windows Search, and then open it.
2. Under **Collectives and Servers**, select the name of the server you want to check.
3. Under **System Management Tools**, select **Operation** > **PI Version**.
   The **Version in Memory** and **Version on Disk** columns display information on versions of all the server subsystems.

If you do not have System Management Tools installed, open a command prompt, change to the **pi\adm** directory, and enter *piversion -v*. To see individual version numbers for each subsystem, change to the **pi\bin** directory and enter the subsystem name followed by the option *-v* (for example, *piarchss.exe -v*).

View computer platform information

To view platform specifications, press **Windows + R** to open the **Run** dialog box, and then type `msinfo32.exe`.

Legal Notices

Rockwell Automation publishes legal notices, such as privacy policies, license agreements, trademark disclosures, and other terms and conditions on the Legal Notices page of the Rockwell Automation website.

Software and Cloud Services Agreement

Review and accept the Rockwell Automation Software and Cloud Services Agreement here.

Open Source Software Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses.

You can view a full list of all open source software used in this product and their corresponding licenses by opening the oss_license.txt file located in your product's OPENSOURCE folder on your hard drive. This file is divided into these sections:

- Components
  Includes the name of the open source component, its version number, and the type of license.
- Copyright Text
  Includes the name of the open source component, its version number, and the copyright declaration.
- Licenses
  Includes the name of the license, the list of open source components citing the license, and the terms of the license.

The default location of this file is:

C:\Program Files (x86)\Common Files\Rockwell\Help\<*product name*>\Release Notes\OPENSOURCE\oss_licenses.txt.

You may obtain Corresponding Source code for open source packages included in this product from their respective project web site(s). Alternatively, you may obtain complete Corresponding Source code by contacting Rockwell Automation via the **Contact** form on the Rockwell Automation website: http://www.rockwellautomation.com/global/about-us/contact/contact.page. Please include "Open Source" as part of the request text.

Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	rok.auto/support
Knowledgebase	Access Knowledgebase articles.	rok.auto/knowledgebase
Local Technical Support Phone Numbers	Locate the telephone number for your country.	rok.auto/phonesupport
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	rok.auto/literature
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	rok.auto/pcdc

Documentation feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at rok.auto/docfeedback.

Waste Electrical and Electronic Equipment (WEEE)

At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at rok.auto/pec.

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.

rockwellautomation.com — expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444
EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640
ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846

HSELD-UM024H-EN-E - March 2024
Supersedes Publication HSELD-UM024G-EN-E - November 2023