

**Preface**

**Copyright notice**

© 2018 Rockwell Automation Technologies, Inc. All rights reserved. Printed in USA.
This document and any accompanying Rockwell Software products are copyrighted by Rockwell Automation, Inc. Any reproduction and/or distribution without prior written consent from Rockwell Automation, Inc. is strictly prohibited. Please refer to the license agreement for details.

**End User License Agreement (EULA)**

You can view the Rockwell Automation End-User License Agreement ("EULA") by opening the EULA.rtf file located in the License folder on your product installation DVD.

**Other licenses**

The third-party licenses are available in the Third-party Licenses folder on your hard drive.

This product includes software developed by The Apache Software Foundation (http://www.apache.org/).
Please read the LICENSE files present in the Third-party Licenses folder on your hard drive.
The names "log4net" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.

The Third-party Licenses folder is located in the Common Files\Rockwell\Help\<Name of FactoryTalk Historian suite>\ folder in the Program Files (x86) directory on your hard drive.

**Trademark notices**

Any Rockwell Automation software or hardware not mentioned here is also a trademark, registered or otherwise, of Rockwell Automation, Inc.

**Other trademarks**

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.
ControlNet is a registered trademark of ControlNet International.
DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA)
OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.
Oracle, SQL*Net, and SQL*Plus are registered trademarks of Oracle Corporation. All other trademarks are the property of their respective holders and are hereby acknowledged.

**Warranty**

This product is warranted in accordance with the product license. The product’s performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product’s implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

**Environmental compliance**


**Contact Rockwell**

Customer Support Telephone — 1.440.646.3434

Online Support — http://www.rockwellautomation.com/support/
# Table of Contents

## Introduction

- Installation .................................................................................. 7
- Features ....................................................................................... 8
- User documentation .................................................................... 9

## Principles of operation

- Processing loops ....................................................................... 12
- FactoryTalk Diagnostics ........................................................... 13

## Configure FactoryTalk Historian Live Data Interface

- Configure FTLD Interface using the FactoryTalk Administration Console ................................................................. 16
- Configure FTLD Interface using the Interface Configuration Utility (ICU) ............................................................................. 16
  - Add remote servers to the Connection Manager .................. 22
  - Define remote servers as API hosts ................................... 22

## Point source

- PI 3 server node: reserved point sources .................................. 23

## Point configuration

- Point attributes .......................................................................... 25
  - Length ................................................................................... 26
  - Point source .......................................................................... 26
  - Point type .............................................................................. 26
  - PI 3 server nodes .................................................................. 26
  - Location1 .............................................................................. 27
  - Location2 .............................................................................. 27
  - Location3 .............................................................................. 27
  - Location4 .............................................................................. 27
    - Advised and Polled tags ................................................... 28
      - Output tags ....................................................................... 28
      - Location5 .......................................................................... 28
      - InstrumentTag ................................................................... 28
        - Length ........................................................................ 28
      - ExDesc ........................................................................... 29
        - Length ........................................................................ 29
      - SourceTag ........................................................................ 29
      - Compression testing ....................................................... 29
Table of Contents

<table>
<thead>
<tr>
<th>The I/O Rates tag configuration</th>
<th>Chapter 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception reporting..................</td>
<td>31</td>
</tr>
<tr>
<td>Output points..........................</td>
<td>32</td>
</tr>
<tr>
<td>Trigger Method 1 (recommended).....</td>
<td>32</td>
</tr>
<tr>
<td>Trigger method 2........................</td>
<td>32</td>
</tr>
<tr>
<td>Sample tag configurations...........</td>
<td>33</td>
</tr>
<tr>
<td>Scan classes................................</td>
<td>33</td>
</tr>
<tr>
<td>Polled tags................................</td>
<td>33</td>
</tr>
<tr>
<td>Advised tags............................</td>
<td>34</td>
</tr>
<tr>
<td>Event tags................................</td>
<td>34</td>
</tr>
<tr>
<td>Use multiple FTLD point source values</td>
<td>35</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitor the I/O Rates tag on the interface node...........</td>
</tr>
<tr>
<td>Configure the I/O Rates tag with ICU .........................</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Command-line parameters........................................</td>
</tr>
<tr>
<td>Sample FTLDInt.bat file...........................................</td>
</tr>
<tr>
<td>Setting file.......................................................</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 10</td>
</tr>
<tr>
<td>Chapter 11</td>
</tr>
<tr>
<td>Chapter 12</td>
</tr>
<tr>
<td>Appendix A</td>
</tr>
<tr>
<td>The I/O Rates tag configuration</td>
</tr>
<tr>
<td>---------------------------------</td>
</tr>
<tr>
<td>Performance Point configuration</td>
</tr>
<tr>
<td>Startup command file</td>
</tr>
<tr>
<td>Interface node clock</td>
</tr>
<tr>
<td>Chapter 10</td>
</tr>
<tr>
<td>Use the Interface Configuration Utility</td>
</tr>
<tr>
<td>Use the Administrative Tools</td>
</tr>
<tr>
<td>Chapter 12</td>
</tr>
<tr>
<td>System errors and PI errors</td>
</tr>
</tbody>
</table>

Technical support and resources

Appendix A

Before you call or write for help
Find the version and build numbers
View computer platform information

Appendix A
Introduction

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

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

**NOTE** 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.

FTLD Interface is configured using the FactoryTalk Administration Console or Interface Configuration Utility. See "Configure FactoryTalk Historian Live Data Interface (page 15)" 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.

The document is available in the Common Files\Rockwell\Help\FactoryTalk Historian SE <version> Live Data Interface\ folder in your Program Files (x86) directory.
The features of FTLD Interface include:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI SDK</td>
<td>The PI SDK and the PI API are bundled together and must be installed on each PI Interface node. FTLD 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 will be set up to use Disconnected Startup because it is based on API calls only.</td>
</tr>
<tr>
<td>Source of timestamps</td>
<td>FTLD Interface can accept timestamps from the FactoryTalk Live Data server, or it can provide timestamps from the FactoryTalk Historian server. This is controlled by a command-line parameter. See &quot;Startup command file (page 43)&quot; for more information on using the command line to control timestamps.</td>
</tr>
<tr>
<td>UniInt-based</td>
<td>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 interfaces as possible. It also allows for a rapid development of new interfaces. In any UniInt-based interface, the interface uses some of the UniInt-supplied configuration parameters and some interface-specific parameters.</td>
</tr>
</tbody>
</table>
FTLD 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 folder in your **Program Files (x86)** directory:

**Common Files\Rockwell\Help\FactoryTalk Historian SE <version> Live Data Interface\**

The location contains the following documents:

- Buffering-User-Guide_EN.pdf
- FTHistorianConfig.chm
- FTHistorianSERN.chm
- FT Historian SE Installation Assistant.pdf
- FT Historian SE Installation and Configuration Guide.pdf
- FT Historian SE Live Data Interface User Guide.pdf (this document)
- High-Availability-Administrator-Guide_EN.pdf
- PI-Interface-Configuration-Utility_1.4.16.79.pdf
- UniInt-Interface-User-Guide.pdf
Chapter 2

Principles of operation

FactoryTalk Live Data (FTLD) Interface is a FactoryTalk 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 FTLD Interface can provide data to a single FactoryTalk Historian SE server or collective. Multiple instances of the interface may be configured, if required, if API buffering is used.

The figure below shows the basic workflow of the FactoryTalk Historian Live Data Interface.
**Processing loops**

At startup, FTLD Interface tries to establish a connection to both the FactoryTalk 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 `UniInt-Interface-User-Guide.pdf`.

**TIP**

The document is available in the `Common Files\Rockwell\Help\FactoryTalk Historian SE <version>\Live Data Interface\` folder in your `Program Files (x86)` directory.

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 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.
FTLD Interface sends messages about its operation to FactoryTalk Diagnostics.

FactoryTalk Diagnostics provides the following information about FTLD 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 FTLD server, or error messages sent from the FTLD server.

Because FTLD 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 error messages are produced by the standard OSIsoft interface routines or by the PI API.
NOTES

- The FactoryTalk Diagnostics Setup and Viewer are available from the Tools menu of FactoryTalk Administration Console.

- For details on reading the message logs, see "How to read new UniInt 4.5.0.x and later Interface message logs (https://techsupport.osisoft.com/Troubleshooting/KB/KB00401)".

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 (x86)\Rockwell Software\FactoryTalk Historian\PIPC\adm (on a 64-bit operating system)

- C:\Program Files\Rockwell Software\FactoryTalk Historian\PIPC\adm (on a 32-bit and 64-bit operating system)

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 (FTLD), we recommend that you use FactoryTalk Administration Console.

However, you must use the Interface Configuration Utility (ICU) to configure FTLD Interface in the following cases:

- If you are configuring redundant FTLD Interfaces.
  For more information, refer to the Rockwell Automation KB 59932 (https://rockwellautomation.custhelp.com/app/answers/detail/a_id/59932) article.

- If you are configuring the buffer subsystem on FTLD Interface.
  For more information, refer to "Enable buffering" in the FactoryTalk Historian SE Installation and Configuration Guide.

  **NOTE** 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 FTLD Interface on a separate machine than the FactoryTalk Historian SE.

- If you are configuring the **Disconnected Startup** option.
  For more information, refer to "(Unint) Disconnected Startup" in the Interface Configuration Utility Help.
Chapter 3  Configure FactoryTalk Historian Live Data Interface

NOTE
To open the Interface Configuration Utility (ICU) online help, run Interface Configuration Utility, and select Help > Contents and Index from the main menu.

NOTE
If you create FTLD 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 FTLD 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 FTLD Interface using the FactoryTalk Administration Console. This section also includes steps on how to enable buffering on the remote interface.

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

The document is available in the Common Files\Rockwell\Help\FactoryTalk Historian SE <version>\Live Data Interface\ folder in your Program Files (x86) directory.

NOTE
For more information on the Interface Configuration Utility, refer to the PI-Interface-Configuration-Utility_1.4.16.79.pdf available in the Common Files\Rockwell\Help\FactoryTalk Historian SE <version>\Live Data Interface\ folder in your Program Files (x86) directory.
If you configure an interface in the 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.

**To configure FTLD 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.

   The **Interface Configuration Utility** dialog box appears.

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

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 FTLD 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 FTLD point source values (page 35).

9. If necessary, edit the **Scan Frequency** and **Scan Class** values.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scan Frequency</td>
<td>Indicates the frequency at which the interface reads values from the FactoryTalk Live Data server.</td>
</tr>
<tr>
<td>Scan Class #</td>
<td>Associated with the <strong>Scan Frequency</strong> value. If there is no interface defined, the Scan Class # will be displayed without a time period.</td>
</tr>
</tbody>
</table>

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, type **LocalSystem**.

   You should get the following configuration:
15. Click **Create**.

The following message may appear:

![PI Interface Configuration Utility](image)

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

Note that 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 need to:

1. Add the remote server to the Connection Manager (page 22).
2. Define the remote server as the API host (page 22).
NOTE This step is required to ensure 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 (page 43)" for more information.
Add remote servers to the Connection Manager

Perform these steps if you didn't add a FactoryTalk Historian server in "Configure FTLD Interface using the Interface Configuration Utility (ICU) (page 16)".

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.
   The Add Server dialog box appears.
3. In the Network Node text box, type the name of the remote server, and click OK.
4. In the Connection Manager dialog box, click Save.

Define remote servers as API hosts

Perform these steps if you didn't set a FactoryTalk Historian server as the API host in "Configure FTLD Interface using the Interface Configuration Utility (ICU) (page 16)".

To use a remote server as the API host:

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

Point source

The FactoryTalk Historian Live Data Interface uses FTLD 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 this, 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 start-up 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 (page 43)".

### 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:

<table>
<thead>
<tr>
<th>Subsystem</th>
<th>Point source character</th>
</tr>
</thead>
<tbody>
<tr>
<td>Totalizer</td>
<td>T</td>
</tr>
<tr>
<td>Alarm</td>
<td>G and @</td>
</tr>
<tr>
<td>Random</td>
<td>R</td>
</tr>
<tr>
<td>RampSoak</td>
<td>9</td>
</tr>
<tr>
<td>Performance Equations</td>
<td>C</td>
</tr>
</tbody>
</table>

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.
Chapter 5

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 needs to 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.

NOTE

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 Live Data (FTLD) points for use with a FactoryTalk Live Data server. Failing to correctly configure FactoryTalk Historian data points will result in poor or no communication between the interface and the FactoryTalk Live Data server. See "Error and informational messages (page 59)" for more information on errors that may occur.
Chapter 5  Point configuration

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 (page 43)".

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

**NOTE**

For more information on the individual point types, refer to the **PI-Data-Archive-2017-R2-Reference-Guide_EN.pdf**, available in the **Common Files\Rockwell\Help\FactoryTalk Historian SE <version> Live Data Interface** folder in your **Program Files (x86)** directory.
**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 FTLD Local Interface is 1.

**Location2**

Location2 is not used for FTLD Interface.

**Location3**

Location3 is used to define a data collection mode:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - Polled or Event</td>
<td>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.</td>
</tr>
<tr>
<td>1 - Advised (default)</td>
<td>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.</td>
</tr>
<tr>
<td>2 - Output</td>
<td>Data is written back to the Live Data server such as FactoryTalk Linx, HMI server such as FactoryTalk View, or OPC server such as RSLinx Classic. Use this method when you want to write data points back to the data servers.</td>
</tr>
</tbody>
</table>

**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 (page 43)".

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.

**NOTE**

Advised, Polled, and Output data collection methods are explained in "Location3 (page 27)".

**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:

<table>
<thead>
<tr>
<th>Type of the tag</th>
<th>InstrumentTag syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device tags</td>
<td><code>&lt;application name&gt;/&lt;area name&gt;:&lt;data server name&gt;[:&lt;shortcut name&gt;]&lt;tag name&gt;</code></td>
</tr>
<tr>
<td>HMI tags</td>
<td><code>&lt;application name&gt;/&lt;area name&gt;:&lt;HMI server Name&gt;:&lt;folder name&gt;&lt;tag name&gt;</code></td>
</tr>
</tbody>
</table>

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

**Length**

The InstrumentTag field allows a maximum of 1023 characters.
FTLD Interface gets the $Global scope from the /FTDirectory parameter and / from the /FTContext parameter. For additional information on these parameters, see "Command-line parameters (page 43)".

**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.

**Length**

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 (page 32)".

**Compression testing**

For each data point, you can set the following attributes to configure its compression testing specification:
### Chapter 5  Point configuration

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression Deviation (CompDev)</td>
<td>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. <strong>Note:</strong> Setting the CompDev attribute value too low causes too little data compression, and wastes space in the archive. Setting the value too high causes 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 or 2 degrees.</td>
</tr>
<tr>
<td>Compression Minimum (CompMin)</td>
<td>A point is archived if the elapsed time since the previous time 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.</td>
</tr>
<tr>
<td>Compression Maximum (CompMax)</td>
<td>A point is archived if the elapsed time since the previous time 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.</td>
</tr>
</tbody>
</table>
### Exception reporting

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

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exception Deviation (ExcDev):</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Exception Minimum (ExcMin):</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Exception Maximum (ExcMax):</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>

**NOTE**

For information on compression testing, refer to "Exception Reporting and Compression Testing" in the PI-Data-Archive-2017-R2-Reference-Guide_EN.pdf, available in the Common Files\Rockwell\Help\FactoryTalk Historian SE <version> Live Data Interface\ folder in your Program Files (x86) directory.
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 FTLD server. FTLD Interface uses Location3=2 to indicate an output point.

Outputs are triggered for UniInt-based interfaces. That is, outputs are not scheduled to occur on a periodic basis. 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 usually 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 very important for troubleshooting.

See the following sample tag configurations:

- Scan classes (page 33)
- Polled tags (page 33)
- Advised tags (page 34)
- Event tags (page 34)

**Sample tag configurations**

**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:

<table>
<thead>
<tr>
<th>Tag</th>
<th>InstrumentTag</th>
<th>Loc 1</th>
<th>Loc 2</th>
<th>Loc 3</th>
<th>Loc 4</th>
<th>Loc 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FiveSec.PV</td>
<td>ItemID1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>OneMin.PV</td>
<td>ItemID2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>NinetyMin.PV</td>
<td>ItemID3</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

### Advised tags

For Advised tags, the interface asks the FTLD 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:

<table>
<thead>
<tr>
<th>Tag</th>
<th>InstrumentTag</th>
<th>Loc 1</th>
<th>Loc 2</th>
<th>Loc 3</th>
<th>Loc 4</th>
<th>Loc 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>AdvFiveSecs.PV</td>
<td>ItemID1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>AdvOneMin.PV</td>
<td>ItemID2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>AdvNinetyMins.PV</td>
<td>ItemID3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
</tbody>
</table>

### 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 Any change.

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:

<table>
<thead>
<tr>
<th>Tag</th>
<th>ExDesc</th>
<th>Instrument Tag</th>
<th>Loc1</th>
<th>Loc2</th>
<th>Loc3</th>
<th>Loc4</th>
<th>Loc5</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM1_Temp.PV</td>
<td>EVENT='PM1_Trigger'</td>
<td>ItemID1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>PM1_Rate.PV</td>
<td>EVENT='PM1_Trigger'</td>
<td>ItemID2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

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 FTLD point source values

You can edit each FTLD 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**, type `FTLDx`, where `x` is a point source value ranging from 1 to 99.
   
   Example: FTLD4

6. Click **Apply**.

7. Restart the FactoryTalk Historian SE server to synchronize the licenses:
   
   1. Using Windows Search, find and run **Stop FactoryTalk Historian SE**.
   
   2. 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 FTLD interface.

10. **Under General**, change **Point Source** from FTLD to the `FTLDx` that you have configured in FactoryTalk Administration Console (step 5).
   
   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 an FTLD 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 FTLD 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 FTLD Interface that is in use.

**NOTE** 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 PI ProcessBook.

**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:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable IORates for this</td>
<td>Select the check box to enable I/O Rates for the selected interface. Clear the check box to disable I/O Rates for the selected interface.</td>
</tr>
<tr>
<td>interface</td>
<td></td>
</tr>
<tr>
<td>Create</td>
<td>Click it to create the suggested I/O Rates tag with the tag name indicated in the Tagname text box.</td>
</tr>
<tr>
<td>Delete</td>
<td>Click it to delete the I/O Rates tag listed in the Tagname text box.</td>
</tr>
<tr>
<td>Reset</td>
<td>Click it to reset the I/O Rates Event Counter and Tag settings.</td>
</tr>
<tr>
<td>Rename</td>
<td>Click it to change the name of the I/O Rates tag using the Rename IORates Tag dialog box.</td>
</tr>
<tr>
<td>Add to File</td>
<td>Click it to add the tag to the IORates.dat file with the event counter listed in the Event Counter text box.</td>
</tr>
<tr>
<td>Event Counter</td>
<td>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.</td>
</tr>
</tbody>
</table>
### The I/O Rates tag configuration

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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 TagSearch 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 just 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. |
### Section 6: The I/O Rates tag configuration

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Image" /></td>
<td><strong>Refresh snapshot.</strong> Click it to refresh the snapshot.</td>
</tr>
</tbody>
</table>
Chapter 7

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.
Chapter 8

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 FTLD Interface startup command file.

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

For the complete list of the parameters, see UniInt-Interface-User-Guide.pdf. The document is available in the Common Files\Rockwell\Help\FactoryTalk Historian SE <version> Live Data Interface\ folder in your Program Files (x86) directory.

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:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( /ps=Source )</td>
<td>The ( /ps ) parameter specifies the point source for the interface. The Source value is not case sensitive. 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.</td>
</tr>
<tr>
<td>( /id=x )</td>
<td>Example: ( /id=1 )                                                                                           The ( /id ) parameter specifies the interface identifier. The interface identifier is a string that is no longer than 9 characters. FTLD Interfaces also use the ( /id ) parameter to identify a particular interface copy number which 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.</td>
</tr>
<tr>
<td>( /host=host:port )</td>
<td>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.</td>
</tr>
<tr>
<td>( /uiDLL=FTLDIntCtl.dll )</td>
<td>The ( /uiDLL ) parameter is used to specify the DLL file name of FTLD Interface. The DLL file is usually 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.</td>
</tr>
<tr>
<td>( /FTDirectory=$Global )</td>
<td>The ( /FTDirectory ) parameter specifies the FactoryTalk directory. It can only be set to ( $Global ).</td>
</tr>
<tr>
<td>( /FTContext=&quot;/&quot; )</td>
<td>This parameter is reserved. Set to “/”.</td>
</tr>
</tbody>
</table>
Parameter | Description
---|---
/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 sub-second scan classes on the command line such as:
/f=0.5 /f=0.1
**Optional parameters:**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/maxstoptime=stoptime</td>
<td>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.</td>
</tr>
<tr>
<td>/MultiCom</td>
<td>• 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).</td>
</tr>
<tr>
<td>/perf=interval</td>
<td>Default = 8 hours. When the percentage of scans that a 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. 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.</td>
</tr>
<tr>
<td>/PISDK=#</td>
<td>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.</td>
</tr>
<tr>
<td>/q</td>
<td>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.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/sio</td>
<td>Default = send initial outputs. The /sio parameter stands for suppress initial outputs. The parameter applies only to interfaces, such as the FTLD, that support outputs. If the /sio parameter is not specified, FTLD Interface will behave in the following manner: when FTLD Interface is started, it determines the current Snapshot value of each output tag. Next, FTLD Interface writes this value to each output tag. In addition, whenever an individual output tag is edited while FTLD 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 FTLD 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.</td>
</tr>
<tr>
<td>/TA=&lt;Any positive value&gt;</td>
<td>Use the Trend Advise (/TA) parameter for advise 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 ((T_{\text{new}} - T_{\text{prev}}) &gt; (/\text{TA} \times R)) { send previous value at (T_{\text{new}} - R) before sending new value at (T_{\text{new}}) } Where: (T_{\text{new}} = ) new timestamp (the timestamp of the received event) (T_{\text{prev}} = ) 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.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| /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 FTLD server timestamp  
  **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 FTLD Server timestamps. |
| /US                | With the Update Snapshot (/US) parameter enabled, if the current snapshot is a system digital state (such as I/O timeout, Shutdown, and so forth) 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, FTLD Interface can use an INI format file to configure interface information. This format file does not install with FTLD Interface, but you can create the file manually. It should be named `FTLDInt.ini` and it must be created in the same path as the `FTLDInt.exe` file and the `FTLDIntCtl.dll` file.

The following is a sample setting file:

```ini
[FTLDIntSetting]
OnceMaxUnsolEvents=4  
ScanClassToUpdateRate=1  
PIOrFTLDTimestamp=0  
FTLDResponseInterval=5000
```
Review the following for additional information.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OnceMaxUnsolEvents</td>
<td>The OnceMaxUnsolEvents key specifies the maximum number of unsolicited events that FTLD 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.</td>
</tr>
<tr>
<td>ScanClassToUpdateRate</td>
<td>For polled points, FTLD Interface calculates the update rate from its scan time. The update rate is used while adding an item to the FTLD server. The algorithm is [\text{Update Rate} = \frac{\text{Scan Time}}{\text{ScanClassToUpdateRate}}] For example, if a point has a 1-second scan period and the ScanClassToUpdateRate is 2, FTLD Interface will use 500 ms as the update rate.</td>
</tr>
<tr>
<td>PIOrFTLDTimestamp</td>
<td>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 FTLD server timestamp is used. This setting is similar to the /TSofPIorFT command-line parameter. See &quot;Command-line parameters (page 43)&quot; for more information.</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FTLDResponseInterval</td>
<td>The <code>FTLDResponseInterval</code> key specifies the time interval during which FTLD Interface checks for data points that were marked 'bad_quality' by the FTLD service. The FTLD 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 FTLD service was not able to receive a data point from FTLD 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.</td>
</tr>
</tbody>
</table>
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, check **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, type `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.

---

**NOTE**

It is possible for computer nodes to boot up 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.
Chapter 10

Security

The trust database must be configured so that FTLD Interface is allowed to write data to the FactoryTalk Historian server.

If FTLD 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.
• For details on the trust configuration, please refer to the FactoryTalk Historian SE server documentation, available in the Common Files\Rockwell\Help\FactoryTalk Historian SE <version> Live Data Interface\ folder in your Program Files (x86) directory.

• For details on reading the message logs, see "How to read new UniInt 4.5.0.x and later Interface message logs (https://techsupport.osisoft.com/Troubleshooting/KB/KB00401)". 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 (x86)\Rockwell Software\FactoryTalk Historian\PIPC\adm\ (on a 64-bit operating system)

  • C:\Program Files\Rockwell Software\FactoryTalk Historian\PIPC\adm\ (on a 32-bit and 64-bit operating system)
Chapter 11

Start and stop the interface

Once you have installed FTLD Interface as a service, you can start and stop it in two ways:

- Using the Interface Configuration Utility (ICU) (page 56).
- Using the Administrative Tools program of Control Panel (page 57).

The FTLD Interface service may terminate immediately after the startup for a variety of reasons. One of 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 name of the .bat file and the .exe file are the same, and that the files are located 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 (page 59)" for additional information.
Chapter 11  Start and stop the interface

NOTE

• For details on reading the message logs, see "How to read new UnivInt 4.5.0.x and later Interface message logs (https://techsupport.osisoft.com/Troubleshooting/KB/KB00401)."

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 (x86)\Rockwell Software\FactoryTalk Historian\PIPC\adm\ (on a 64-bit operating system)

• C:\Program Files\Rockwell Software\FactoryTalk Historian\PIPC\adm\ (on a 32-bit and 64-bit operating system)

Use the Interface Configuration Utility

To start FTLD 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.

   The Interface Configuration Utility dialog box appears.

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 FTLD Interface:

1. 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 (page 57) program of Control Panel.
Use the Administrative Tools

To start FTLD 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.

   The Services dialog box appears.

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 FTLD Interface, In the Services dialog box, right-click FTLD<X> and select Stop.
Chapter 12

Error and informational messages

FTLD 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.

NOTE • For details on reading the message logs, see "How to read new UniInt 4.5.0.x and later Interface message logs (https://techsupport.osisoft.com/Troubleshooting/KB/KB00401)".
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 (x86)\Rockwell Software\FactoryTalk Historian\PIPC\adm (on a 64-bit operating system)
• C:\Program Files\Rockwell Software\FactoryTalk Historian\PIPC\adm (on a 32-bit and 64-bit operating system)

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

<table>
<thead>
<tr>
<th>Severity</th>
<th>Message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>Failed to disconnect from FactoryTalk.</td>
</tr>
<tr>
<td>Error</td>
<td>Failed to initialize COM library.</td>
</tr>
<tr>
<td>Error</td>
<td>Failed to initialize COM security.</td>
</tr>
<tr>
<td>Error</td>
<td>Missing or invalid interface ID parameter.</td>
</tr>
</tbody>
</table>
### Severity | Message text
--- | ---
**Error** | Failed to initialize FactoryTalk Diagnostics.
**Error** | Failed to launch the FTLD interface due to the lack of the parameter /FTDirectory.
**Error** | Failed to launch the FTLD 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 FTLD 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 FTLD service does not respond in %d ms.
**Information** | PI point <%s> has been removed from the FTLD interface (InterfaceID: %d).
**Information** | PI point <%s> is edited in the FTLD interface (InterfaceID: %d).
<table>
<thead>
<tr>
<th>Severity</th>
<th>Message text</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information</td>
<td>PI point &lt;%s&gt; has been added to the FTLD interface (InterfaceID: %d).</td>
</tr>
<tr>
<td>Information</td>
<td>Connected to FactoryTalk Directory scope %s successfully.</td>
</tr>
<tr>
<td>Information</td>
<td>FTLD interface(ID:%d) has Scan Class %d = %s.</td>
</tr>
<tr>
<td>Information</td>
<td>Disconnected from FactoryTalk Directory %s successfully.</td>
</tr>
</tbody>
</table>
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`) located 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
```
Appendix A

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 (http://www.rockwellautomation.com/support/).

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 log(s) 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 `piadm` directory, and type `piversion -v`. To see individual version numbers for each subsystem, change to the `pi\bin` directory and type 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`. 
A
Add remote servers to the Connection Manager 22
Advised and Polled tags 28
Advised tags 34

B
Before you call or write for help 63

C
Command-line parameters 43
Compression testing 29
Configure FactoryTalk Historian Live Data Interface 15
Configure FTLD Interface using the FactoryTalk Administration Console 16
Configure FTLD Interface using the Interface Configuration Utility (ICU) 16
Configure the I/O Rates tag with ICU 37

D
Define remote servers as API hosts 22
Documentation Feedback 67

E
Error and informational messages 59
Error descriptions 62
Event tags 34
Exception reporting 31
ExDesc 29

F
FactoryTalk Diagnostics 13
Features 8
Find the version and build numbers 63

I
Installation 7
Installation Assistance 67
InstrumentTag 28
Interface node clock 51
Introduction 7

L
Length 26, 28, 29
Location1 27
Location2 27
Location3 27
Location4 27
Location5 28

M
Monitor the I/O Rates tag on the interface node 37

N
New Product Satisfaction Return 67

O
Output points 32
Output tags 28
<table>
<thead>
<tr>
<th>P</th>
<th>SourceTag 29</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performance Point configuration</td>
<td>Start and stop the interface 55</td>
</tr>
<tr>
<td>PI 3 server node</td>
<td>Startup command file 43</td>
</tr>
<tr>
<td>reserved point sources 23</td>
<td>System errors and PI errors 62</td>
</tr>
<tr>
<td>PI 3 server nodes 26</td>
<td></td>
</tr>
<tr>
<td>Point attributes 25</td>
<td></td>
</tr>
<tr>
<td>Point configuration 25</td>
<td></td>
</tr>
<tr>
<td>Point source 23, 26</td>
<td></td>
</tr>
<tr>
<td>Point type 26</td>
<td></td>
</tr>
<tr>
<td>Polled tags 33</td>
<td></td>
</tr>
<tr>
<td>Preface 2</td>
<td></td>
</tr>
<tr>
<td>Principles of operation 11</td>
<td></td>
</tr>
<tr>
<td>Processing loops 12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Rockwell Automation Support 67</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample FTLDInt.bat file 48</td>
<td></td>
</tr>
<tr>
<td>Sample tag configurations 33</td>
<td></td>
</tr>
<tr>
<td>Scan classes 33</td>
<td></td>
</tr>
<tr>
<td>Security 53</td>
<td></td>
</tr>
<tr>
<td>Setting file 48</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Technical support and resources 63</td>
<td></td>
</tr>
<tr>
<td>The I/O Rates tag configuration 37</td>
<td></td>
</tr>
<tr>
<td>Trigger Method 1 (recommended) 32</td>
<td></td>
</tr>
<tr>
<td>Trigger method 2 32</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>U</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Use multiple FTLD point source values 35</td>
<td></td>
</tr>
<tr>
<td>Use the Administrative Tools 57</td>
<td></td>
</tr>
<tr>
<td>Use the Interface Configuration Utility 56</td>
<td></td>
</tr>
<tr>
<td>User documentation 9</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>V</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>View computer platform information 64</td>
<td></td>
</tr>
</tbody>
</table>
Rockwell Automation Support

Rockwell Automation provides technical information on the Web to assist you in using its products. At http://www.rockwellautomation.com/support/ you can find technical and application notes, sample code, and links to software service packs. You can also visit our Support Center at https://rockwellautomation.custhelp.com/ for software updates, support chats and forums, technical information, FAQs, and to sign up for product notification updates.

In addition, we offer multiple support programs for installation, configuration, and troubleshooting. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/services/online-phone.

Installation Assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

<table>
<thead>
<tr>
<th>United States or Canada</th>
<th>1.440.646.3434</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside United States or Canada</td>
<td>Use the Worldwide Locator at <a href="http://www.rockwellautomation.com/rockwellautomation/support/overview.page">http://www.rockwellautomation.com/rockwellautomation/support/overview.page</a>, or contact your local Rockwell Automation representative.</td>
</tr>
</tbody>
</table>

New Product Satisfaction Return

Rockwell Automation tests all of its products to help ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

<table>
<thead>
<tr>
<th>United States</th>
<th>Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outside United States</td>
<td>Please contact your local Rockwell Automation representative for the return procedure.</td>
</tr>
</tbody>
</table>

Documentation Feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete this form, publication RA-DU002, available at http://www.rockwellautomation.com/literature/.