# Table of Contents

## Welcome To FactoryTalk Transaction Manager
- What Is FactoryTalk Transaction Manager? ........................................ 11
- What Can FactoryTalk Transaction Manager Do For Me? .............. 12
- Understanding FactoryTalk Transaction Manager Concepts .......... 13
  - Transaction Control Manager Service .............................................. 13
  - FactoryTalk Transaction Manager Service and Control Connectors .............................................................................. 14
  - Control Connectors .............................................................................. 15
  - Enterprise Connectors ........................................................................ 16
  - Configuration Server ........................................................................ 16
  - Transactions ............................................................................................ 16
  - FactoryTalk .............................................................................................. 17
    - FactoryTalk Services Platform Components ................................ 17
- Intended Audience ...................................................................................... 19
- Where Can I Go for Help? ....................................................................... 19

## Getting Started
- Installing and Activating FactoryTalk Transaction Manager .......... 21
  - Distributed Installations ........................................................................ 21

## Exploring the FactoryTalk Transaction Manager User Interface
- Starting FactoryTalk Transaction Manager ........................................ 23
- Exploring the FactoryTalk Transaction Manager User Interface ... 23
  - Title Bar ................................................................................................... 25
  - Menu Bar .................................................................................................. 25
  - Toolbar ..................................................................................................... 26
  - Configuration Tree ............................................................................... 27
    - Configuration and Connector Status .............................................. 28
    - Configuration Server Status ........................................................... 29
  - Workspace ............................................................................................... 30
    - Transaction States ............................................................................ 30
  - Status Bar ................................................................................................. 31
  - Configuration Checklist ........................................................................ 31
    - Step 1: Defining and Naming a New Configuration .................... 33
    - Step 2: Defining Connectors ........................................................... 34
    - Step 3: Defining Data Points ............................................................ 35
Table of Contents

Step 4: Defining Data Objects ............................................................. 35
Step 5: Defining Transactions ............................................................. 36
Step 6: Verifying Transactions ............................................................. 37
Miscellaneous ............................................................................................... 37
Viewing Configuration Properties ..................................................... 38
Starting Configurations ........................................................................ 38
Stopping Configurations ....................................................................... 40
Starting and Stopping Connectors .................................................... 40
Monitoring Configurations ..................................................................... 40
Understanding FactoryTalk Transaction Manager External Files .. 41
Using the Service Console ......................................................................... 42

Chapter 4
Introducing FactoryTalk Transaction Manager Services .................. 45
Control Connectors ................................................................................... 46
  FactoryTalk Live Data ............................................................................. 46
  DDE ....................................................................................................... 47
  RSLinx Classic OPC ............................................................................... 47
  RSVView32 .......................................................................................... 48
  Generic OPC .......................................................................................... 48
Enterprise Database Connectors .......................................................... 49
  Microsoft OLE DB ................................................................................ 49
  ODBC ...................................................................................................... 50
  Oracle OCI .............................................................................................. 50
Enterprise Application Connectors ........................................................ 50
  Microsoft COM+ .................................................................................. 51
Time-series Data Compression .......................................................... 51
  FactoryTalk Metrics ........................................................................... 51
Enterprise Connector Options ................................................................ 51
FactoryTalk Transaction Manager Service ........................................... 52
Transaction Control Manager Service ................................................... 53
Configuration Server .................................................................................. 53

Chapter 5
Introducing Data Points ............................................................................ 55
FactoryTalk Live Data Data Points ........................................................ 57
  Selecting a Collection Mode ............................................................... 58
Transaction Timeout ................................................................. 82
Transaction Completion ............................................................ 82
Cached Transactions ................................................................. 82
Real-time Transactions ............................................................. 82
Bidirectional Transactions ......................................................... 84
Transactions With Bound Transaction Results ......................... 84
Database Triggers ....................................................................... 84
Using the Expression Editor ....................................................... 85
Logical and Mathematical Operators ........................................ 85
Time Functions .......................................................................... 86
Data Point Range and Advanced Functions ............................ 86
Parse Function .......................................................................... 87

Chapter 8
Introducing Online Edits ........................................................... 89
Understanding Online Edit Concepts ......................................... 90
Online Edits Workflow .............................................................. 91
Configurations That Use Online Edits ........................................ 91
Assembling Pending Edits ......................................................... 93
Canceling Pending Edits ........................................................... 94
Pending Edit Alerts .................................................................... 94
Occurrence Conditions of Pending Edit Alerts ....................... 94
Creating a Configuration That Uses Online Edits ..................... 96
Editing Data Points In a Running Configuration That Uses Online Edits ................................................................. 96
Adding New Data Points ............................................................ 96
Editing Existing Data Points ....................................................... 97
Saving Data Point Pending Edits ............................................... 97
Assembling Data Point Pending Edits ........................................ 98
Canceling Data Point Pending Edits .......................................... 98
Editing Transactions In a Running Configuration That Uses Online Edits ................................................................. 98
Saving Transaction Pending Edits .............................................. 100
Assembling Transaction Pending Edits ..................................... 100
Viewing Transaction Differences ............................................ 101
Canceling Transaction Pending Edits ........................................ 102
Chapter 9
Remote User Interface ........................................................................................................ 103
Configuring the Remote User Interface........................................................................ 104
Distributed Configurations ................................................................................................. 105
Licensing Required For a Distributed Configuration................................................ 105
Establishing Microsoft Windows Privileges ............................................................... 105
Creating a Distributed Configuration ........................................................................... 105
Using UNC Paths ........................................................................................................ 106
    Changing the Transaction Cache File Path ..................................................... 106
    Changing the Error Log File Path .............................................................. 107
Data Point Buffering ........................................................................................................ 107
    Buffering In the Controller ........................................................................ 107
    Buffering In the FactoryTalk Transaction Manager Control Connector .......... 107
    Buffering In the Transaction Control Manager Service ......................... 108
    Buffering In the FactoryTalk Transaction Manager Service ..................... 109
    Buffering In Cached Transaction Files ....................................................... 109
Increasing Performance .................................................................................................... 110
    Control System ............................................................................................ 110
    Database ........................................................................................................ 110
    FactoryTalk Transaction Manager ................................................................ 111
    Hardware and Operating Environment ....................................................... 112

Chapter 10
External Trigger Sample Application ................................................................................ 115
Application Contents ........................................................................................................ 115
Running the Sample Application ..................................................................................... 116

Chapter 11
Introducing FactoryTalk Transaction Manager and Microsoft COM+ Objects ............ 119
Creating the Remote Component .................................................................................... 120
Creating the Client Application ....................................................................................... 121
Installing the Remote Component .................................................................................. 121
Setting Up the Remote Client ......................................................................................... 123
Creating the Microsoft COM+ Setup Program .............................................................. 124
<table>
<thead>
<tr>
<th>Securing FactoryTalk Transaction Manager Using FactoryTalk Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moving the Client Sample Application.............................................. 125</td>
</tr>
<tr>
<td>Including the COM+ Enterprise Application Connector In a FactoryTalk Transaction Manager Configuration ............ 125</td>
</tr>
<tr>
<td>Defining the COM+ Enterprise Application Connector .................. 125</td>
</tr>
<tr>
<td>Defining the COM+ Data Object....................................................... 125</td>
</tr>
<tr>
<td>Code Sample A (ComSampleVB).......................................................... 126</td>
</tr>
<tr>
<td>Code Sample B (ClientSampleVB)...................................................... 127</td>
</tr>
</tbody>
</table>

Chapter 12

Considerations When Using FactoryTalk Transaction Manager With FactoryTalk Security ..................................................... 131
Using FactoryTalk Transaction Manager With Single Sign-on (SSO).................................................................................. 134
FactoryTalk Security Permissions To Perform FactoryTalk Transaction Manager Tasks .......................................................... 136
Writing Product-Specific Security Privileges From a Previous Release To a File ................................................................. 137
Mapping Old Product-Specific Security Privileges To New FactoryTalk Security Permissions ..................................................... 139

Chapter 13

Assemble Pending Edits........................................................................ 141
Audit Trail .............................................................................................. 141
Binding................................................................................................... 141
Bound Value .......................................................................................... 141
Configuration........................................................................................ 141
Configuration Server............................................................................. 141
Configuration That Uses Online Edits .................................................. 142
Control Connector.................................................................................. 142
Control System...................................................................................... 142
Data Object............................................................................................ 142
Data Point ............................................................................................... 142
Database View......................................................................................... 142
DDE Control Connector......................................................................... 142
DSN......................................................................................................... 143
Enterprise Connector............................................................................ 143
Expression Editor.................................................................................. 143

Glossary

Assemble Pending Edits........................................................................ 141
Audit Trail .............................................................................................. 141
Binding................................................................................................... 141
Bound Value .......................................................................................... 141
Configuration........................................................................................ 141
Configuration Server............................................................................. 141
Configuration That Uses Online Edits .................................................. 142
Control Connector.................................................................................. 142
Control System...................................................................................... 142
Data Object............................................................................................ 142
Data Point ............................................................................................... 142
Database View......................................................................................... 142
DDE Control Connector......................................................................... 142
DSN......................................................................................................... 143
Enterprise Connector............................................................................ 143
Expression Editor.................................................................................. 143
FactoryTalk Metrics Enterprise Application Connector ............... 143
FactoryTalk Security ........................................................................... 143
FactoryTalk Transaction Manager User Interface ......................... 143
Generic OPC Control Connector ....................................................... 143
Local FactoryTalk Transaction Manager Computer ....................... 143
Microsoft COM+ Enterprise Application Connector .................... 143
Microsoft OLE DB Enterprise Database Connector ....................... 144
OCI ............................................................................................................. 144
ODBC......................................................................................................... 144
ODBC Enterprise Database Connector ............................................. 144
OLE DB...................................................................................................... 144
Oracle OCI Enterprise Database Connector .................................... 144
Pending Edits ......................................................................................... 144
Pending Edit Alerts ............................................................................... 144
Remote FactoryTalk Transaction Manager Computer .................. 145
RSLinx Classic OPC Control Connector .......................................... 145
RSView32 Control Connector ............................................................. 145
SQL ............................................................................................................. 145
Table ............................................................................................................ 145
Tag ............................................................................................................. 145
Time-series Data Compression Enterprise Database Connector . 145
Transaction................................................................................................ 145
Transaction Control Manager Service ................................................ 145
FactoryTalk Transaction Manager Service ........................................ 146
Transaction Result Binding ................................................................... 146
Transaction Result Code ........................................................................ 146
UNC............................................................................................................ 146
Unidirectional Transactions................................................................. 146
Welcome To FactoryTalk Transaction Manager

In this chapter you will learn about the following:

- What Is FactoryTalk Transaction Manager? (page 11)
- What Can FactoryTalk Transaction Manager Do For Me? (page 12)
- Understanding FactoryTalk Transaction Manager Concepts (page 13)
- FactoryTalk (page 17)
- Intended Audience (page 19)
- Where Can I Go for Help? (page 19)

What Is FactoryTalk Transaction Manager?

FactoryTalk Transaction Manager (previously known as RSSql) is an industrial transaction software engine that shares data between your shop floor systems and your enterprise applications (for example, corporate databases) or COM+. FactoryTalk Transaction Manager can interact with the following shop floor systems:

- Human Machine Interfaces (HMI).
- Programmable Logic Controllers (PLC).
- ControlLogix Controllers.
- Distributed Control Systems (DCS).
What Can FactoryTalk Transaction Manager Do For Me?

FactoryTalk Transaction Manager helps you manage your manufacturing processes by integrating the data in your control systems with enterprise applications.

The following are examples of FactoryTalk Transaction Manager applications:

- **Automate data logging**
  FactoryTalk Transaction Manager can move large amounts of data in a fast and robust manner. In addition, the software has built-in fault tolerance and the ability to optimize reading and writing of both control and enterprise data. You can use the software to automate the following types of processes:
  - Monitoring performance of control systems such as machine usage.
  - Tracking product information such as Work in Progress status and raw material availability.
  - Updating real-time process information such as temperature, pressure, and alarm states.

- **Control the plant floor using business rules and quality**
  FactoryTalk Transaction Manager provides the interface for a repository of business rules. A business rule can be any logic required to run your plant such as product specifications or quality parameters. By placing business rules in a database or COM+ server in a central location, the rules are easier to manage within an enterprise system. Additionally, the software can assure quality data to meet the requirements of today’s advanced manufacturing companies.

- **Manage recipes**
  FactoryTalk Transaction Manager moves data (for example, recipe information) from a database to an HMI or control system.
Understanding FactoryTalk Transaction Manager Concepts

FactoryTalk Transaction Manager consists of several design-time and run-time components. This guide describes the following components in detail:

- Transaction Control Manager service.
- FactoryTalk Transaction Manager service.
- Control connectors.
- Enterprise connectors.
- Configuration Server.
- Transactions.

Transaction Control Manager Service

The Transaction Control Manager is a service that controls and executes FactoryTalk Transaction Manager transactions contained in a configuration, but with the additional functionality of the FactoryTalk Live Data control connector embedded in it. In a configuration enabled for editing, the Transaction Control Manager replaces the separate FactoryTalk Transaction Manager and control connector services.

The Transaction Control Manager service can connect to Rockwell Software products and all OPC servers; therefore, the use of this service is the preferred method for all new FactoryTalk Transaction Manager configurations. For more information on how the Transaction Control Manager service functions, see the following figure.
In a configuration that uses online edits, the Transaction Control Manager service performs the duties of the FactoryTalk Transaction Manager service. For more information on online edits, see "Understanding Online Edits (page 89)".

The FactoryTalk Transaction Manager service is used to control and execute FactoryTalk Transaction Manager transactions contained in configurations created prior to CPR 7 or when you have a business reason to not run the Transaction Control Manager service. For more information on how the FactoryTalk Transaction Manager functions, see the following figure.
Control Connectors

The FactoryTalk Transaction Manager service communicates with the industrial control system device via a control connector. A control connector is a Microsoft Windows service that collects data from a controller and sends it to the FactoryTalk Transaction Manager service in the FactoryTalk Transaction Manager. You can use the following types of control connectors:

- FactoryTalk Live Data.
- DDE.
- RSLinx Classic OPC.
- RSView32.
- Generic OPC.

Control connectors can be used to reference data points or memory locations within your control or shop floor system, such as "tags" from an RSView32 application.
Enterprise Connectors

The Transaction Control Manager service and the FactoryTalk Transaction Manager service communicate with enterprise systems, such as databases, via an enterprise connector service. An enterprise connector is a Microsoft Windows service that transfers data between the Transaction Control Manager service or the FactoryTalk Transaction Manager service and a database. You can use the following types of enterprise connectors:

- Open Database Connectivity (ODBC).
- Oracle Call Interface (OCI).
- Microsoft OLE DB.
- Microsoft COM+.
- Time-series Data Compression.
- FactoryTalk Metrics connector.

Enterprise connectors can be used to create data objects that reference elements in your enterprise system, such as tables or stored procedures.

Configuration Server

The Configuration Server is a service that runs continuously to provide a single interface to the configuration files (with the .dat file extension) which make up the FactoryTalk Transaction Manager configuration. The Configuration Server simplifies access to the configuration files by filtering all changes to the files and by communicating with other FactoryTalk Transaction Manager services. A collection of all changes that affect a configuration is recorded in an audit trail (via either FactoryTalk Diagnostics or the Configuration Server *.log file). For more information, refer to the FactoryTalk Transaction Manager online help.

Transactions

FactoryTalk Transaction Manager creates transactions, or discrete operations that transfer data from the control system to or from the enterprise system. For example, a single transaction can download data from an Oracle database via a stored procedure, to tags in a
ControlLogix processor. Alternatively, a transaction can send multiple data points from a distributed control system to a Microsoft SQL Server database to be logged for reporting. Transactions can be modified at runtime. For more information, see "Understanding Online Edits (page 89)".

The transaction model organizes the task of data management. FactoryTalk Transaction Manager’s flexibility provides many options for customizing all aspects of a transaction. The software can manage many transactions at once, allowing for sophisticated manufacturing data collection and control applications. By using the software, you can also monitor, modify, and enable or disable individual transactions, making the development and implementation of an application easier.

**FactoryTalk**

FactoryTalk is a manufacturing information platform that integrates plant-wide control systems and connects the enterprise with the production facility.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrate</td>
<td>FactoryTalk eliminates both functionality gaps and overlaps by providing common services (such as diagnostics and access to real-time data) and by sharing plant resources (such as tags and graphic displays) throughout a production facility.</td>
</tr>
<tr>
<td>Communicate</td>
<td>FactoryTalk transforms plant-floor data into useful information, and delivers it to the people who need it, e.g. maintenance engineers or enterprise planners.</td>
</tr>
<tr>
<td>Collaborate</td>
<td>FactoryTalk allows defining plant-floor resources once, and then allows simultaneous access to those resources across system boundaries.</td>
</tr>
</tbody>
</table>

**FactoryTalk Services Platform Components**

With each coordinated release, additional Rockwell Software products build on the FactoryTalk platform and integrate more of the FactoryTalk components. All of the FactoryTalk components are installed together as a platform, integrated into each FactoryTalk-enabled product installation process:
### Item Description

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryTalk Directory</td>
<td>FactoryTalk-enabled products use the FactoryTalk Directory to share a common address book, which automatically finds and provides access to plant-floor resources, such as data tags and graphic displays. Unlike a single database, FactoryTalk Directory provides searchable references to resources stored anywhere across an automation system, offering the benefits of central data storage without the risk of a single point of failure. Changes made to the automation system automatically update across all participants in a FactoryTalk-enabled application.</td>
</tr>
<tr>
<td>FactoryTalk Live Data</td>
<td>FactoryTalk Live Data manages connections between FactoryTalk-enabled products and data servers. It notifies clients when a connection is lost, automatically reconnects clients, and combines data from multiple controllers and servers into a single group with a single data server connection. This results in faster real-time data transfer and more reliable, efficient connections to data servers. It also assists in redundancy support for data servers by automatically handling detection and failovers for all FactoryTalk-enabled products.</td>
</tr>
<tr>
<td>FactoryTalk Administration Console</td>
<td>The FactoryTalk Administration Console is a stand-alone tool that allows you to configure and manage FactoryTalk-enabled applications.</td>
</tr>
<tr>
<td>FactoryTalk Audit and FactoryTalk Diagnostics</td>
<td>FactoryTalk Audit and FactoryTalk Diagnostics provides the ability to log errors, warnings, and other status messages generated throughout a FactoryTalk-enabled system to either local logs or a central location.</td>
</tr>
<tr>
<td>FactoryTalk Security</td>
<td>FactoryTalk Security is intended to improve the security of your automation system by limiting access to the users with a legitimate need. FactoryTalk Security authenticates user identities and authorizes user requests to access a FactoryTalk-enabled system. These security services are fully integrated with the FactoryTalk Directory and are included as part of the FactoryTalk Services Platform that is installed with many Rockwell Software products. FactoryTalk Security replaces the product-specific privilege configuration that was available in previous releases of FactoryTalk Transaction Manager. For more information about using FactoryTalk Transaction Manager with FactoryTalk Security, refer to &quot;Securing FactoryTalk Transaction Manager using FactoryTalk Security (page 131)&quot;.</td>
</tr>
</tbody>
</table>
**Intended Audience**

For this guide it is assumed that you are a control engineer or a database administrator, and that you are familiar with the following:

- Using personal computers.
- Microsoft Windows operating systems.
- DDE or OPC servers (for example, RSLinx Classic or RSView32).
- Configuration of database connections such as ODBC, OCI, or Microsoft OLE DB.
- Microsoft COM+.
- RSLinx Enterprise.
- FactoryTalk View SE.

**Where Can I Go for Help?**

Consult the following resources for additional information about the product:

- **Release Notes**
  The release notes contain current information about the product, including hardware and software requirements, new features, known and fixed anomalies.

- **RSBizWare Administration Guide**
  The administration guide helps the RSBizWare administrator install and configure the software as well as understand the architecture of the RSBizWare suite and its components.

- **Online help**
  The online help provides general information and step-by-step procedures for working with the product.

- **Rockwell Automation Support Center**
  The support center provides a variety of services, such as trainings, webinars, and online support that will improve your experience using the RSBizWare suite.
Chapter 1  Welcome To FactoryTalk Transaction Manager
Chapter 2

Getting Started

In this chapter you will learn about the following:

- Installing and Activating FactoryTalk Transaction Manager (page 21)
- Distributed Installations (page 21)

Installing and Activating FactoryTalk Transaction Manager

FactoryTalk Transaction Manager is distributed as a part of the RSBizWare suite.

For details on hardware, software requirements, and activation options, refer to the FactoryTalk Transaction Manager Release Notes and RSBizWare Release Notes.

For details on installing and activating FactoryTalk Transaction Manager, refer to the RSBizWare Administration Guide.

All the documents listed above are available on the RSBizWare installation DVD.

Distributed Installations

You may want to run FactoryTalk Transaction Manager or its services on more than one computer. To run the software in a distributed mode, it must be installed on all computers that are referenced in a configuration. For more information, see "Distributed Configurations (page 105)".
Exploring the FactoryTalk Transaction Manager User Interface

In this chapter you will learn about the following:

- Starting FactoryTalk Transaction Manager (page 23)
- Exploring the FactoryTalk Transaction Manager User Interface (page 23)
- Configuration Checklist (page 31)
- Miscellaneous (page 37)
- Understanding FactoryTalk Transaction Manager External Files (page 41)
- Using the Service Console (page 42)

**Starting FactoryTalk Transaction Manager**

To start FactoryTalk Transaction Manager go to Start > Programs > Rockwell Software > FactoryTalk Transaction Manager > FactoryTalk Transaction Manager.

**Exploring the FactoryTalk Transaction Manager User Interface**

When you start FactoryTalk Transaction Manager for the first time (and if it is the first Rockwell Software product run on your computer), you are automatically logged on to FactoryTalk Security using your Windows-linked user account information. The system graphic appears in the right pane (or workspace) of the FactoryTalk Transaction Manager user interface.
For more information on FactoryTalk Security, see "Securing FactoryTalk Transaction Manager with FactoryTalk Security (page 131)". To view the procedures for configuring and using FactoryTalk Transaction Manager, on the menu, click Help > Quick Start.

The FactoryTalk Transaction Manager user interface includes the following elements:

- Title bar (page 25)
- Menu bar (page 25)
- Toolbar (page 26)
- Configuration tree (page 27)
- Workspace (page 30)
- Status bar (page 31)
Title Bar

The title bar displays the name of the configuration or configuration server, depending on what is selected in the configuration tree.

Menu Bar

You can access many features from the FactoryTalk Transaction Manager menu bar.

NOTE

For a description of the available shortcuts, see "Toolbar (page 26)".

Review the following for additional information.

<table>
<thead>
<tr>
<th>Use this menu:</th>
<th>To do the following:</th>
</tr>
</thead>
</table>
| Security      | • Log on or log off of FactoryTalk Security.  
                 • View your FactoryTalk Security permissions on the current configuration server.  
                 For more information on FactoryTalk Security, see "Securing FactoryTalk Transaction Manager with FactoryTalk Security (page 131)". |
| Configuration | • Create a new configuration.  
                 • Set or change properties in a configuration.  
                 • Access the Configuration Checklist.  
                 • Delete, backup, or restore a configuration.  
                 • Stop or start a configuration. |
| Define        | Define:  
                 • Connectors.  
                 • Data objects.  
                 • Data points.  
                 • Transactions.  
                 • Options for error logging.  
                 • Scheduled events. |
## Chapter 3  Exploring the FactoryTalk Transaction Manager User Interface

<table>
<thead>
<tr>
<th>Use this menu:</th>
<th>To do the following:</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>• View:</td>
</tr>
<tr>
<td></td>
<td>• Information about the currently defined transaction.</td>
</tr>
<tr>
<td></td>
<td>• Diagnostic information about the configuration that is currently running.</td>
</tr>
<tr>
<td></td>
<td>• Error log files.</td>
</tr>
<tr>
<td></td>
<td>• The system graphic.</td>
</tr>
<tr>
<td></td>
<td>• Switch between large and small icons.</td>
</tr>
<tr>
<td></td>
<td>• Obtain a status update.</td>
</tr>
<tr>
<td></td>
<td>• Enable or disable the toolbar or status bar.</td>
</tr>
<tr>
<td>Tools</td>
<td>• Create a configuration report.</td>
</tr>
<tr>
<td></td>
<td>• Verify the selected configuration.</td>
</tr>
<tr>
<td></td>
<td>• Use wizards to create data logging, duplicate data points, or duplicate transactions.</td>
</tr>
<tr>
<td></td>
<td>• Set options for messages and the Log Viewer.</td>
</tr>
<tr>
<td>Help</td>
<td>• Open the FactoryTalk Transaction Manager Online Help system to get assistance with using FactoryTalk Transaction Manager. You can use the index of the online help system to search for a topic for which you want to view information.</td>
</tr>
<tr>
<td></td>
<td>• Access:</td>
</tr>
<tr>
<td></td>
<td>• The FactoryTalk Transaction Manager Release Notes.</td>
</tr>
<tr>
<td></td>
<td>• The Quick Start Guide for FactoryTalk Transaction Manager.</td>
</tr>
<tr>
<td></td>
<td>• The product manuals.</td>
</tr>
<tr>
<td></td>
<td>• Error code resources.</td>
</tr>
<tr>
<td></td>
<td>• View information about:</td>
</tr>
<tr>
<td></td>
<td>• Support and training.</td>
</tr>
<tr>
<td></td>
<td>• Copy protection.</td>
</tr>
<tr>
<td></td>
<td>• License type and software version.</td>
</tr>
</tbody>
</table>

In addition, you can use this option to obtain FactoryTalk Transaction Manager license and version information.

### Toolbar

The toolbar contains shortcuts to several commonly used FactoryTalk Transaction Manager functions. Each button on the toolbar is a graphical representation of a command that is also available from the FactoryTalk Transaction Manager menu bar.
<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>☐</td>
<td>Creates a new FactoryTalk Transaction Manager configuration.</td>
</tr>
<tr>
<td>❋</td>
<td>Displays the About FactoryTalk Transaction Manager dialog box.</td>
</tr>
<tr>
<td>📚</td>
<td>Displays the online help for any design element (click this button and place the cursor on any design element).</td>
</tr>
<tr>
<td>⏯</td>
<td>Starts the configuration or connector currently selected in the Configuration tree. If the edit enabled configuration has pending edits, a message appears prompting the user to ignore the pending edits and start the configuration anyway.</td>
</tr>
<tr>
<td>⏮</td>
<td>Stops the configuration or connector currently selected in the Configuration tree.</td>
</tr>
<tr>
<td>📦</td>
<td>Assembles all data point and transaction pending edits in the edit enabled configuration. For more information, see &quot;Understanding Online Edits (page 89)&quot;.</td>
</tr>
<tr>
<td>🔄</td>
<td>Opens the Pending Edit Alerts dialog box. For more information, see &quot;Understanding Online Edits (page 89)&quot;.</td>
</tr>
<tr>
<td>🕵</td>
<td>Verifies the selected configuration. You can verify multiple transactions and choose to save the results to a text file.</td>
</tr>
<tr>
<td>📱</td>
<td>Displays information about the current transactions in the right pane.</td>
</tr>
<tr>
<td>📈</td>
<td>Displays diagnostic information about the configuration that is currently running in the right pane.</td>
</tr>
<tr>
<td>📋</td>
<td>Displays error log information in the right pane.</td>
</tr>
<tr>
<td>📦</td>
<td>Displays the system graphic in the right pane.</td>
</tr>
</tbody>
</table>

**Configuration Tree**

The FactoryTalk Transaction Manager user interface is divided into two panes. The left pane is known as the **Configuration tree**. It displays all the items of the current configurations in the form of a tree.
The top level in the item tree is the Configuration Server, with the name of the computer on which it is running. The second level is the configuration name. The third level displays the Transaction Control Manager service (if the configuration uses online edits) or the FactoryTalk Transaction Manager service. The Transaction Control Manager service or the FactoryTalk Transaction Manager service are followed by the control connectors and enterprise connectors defined for the configuration. For more information, see "Understanding Online Edits (page 89)".

**Configuration and Connector Status**

The traffic lights in the Configuration tree represent the status of FactoryTalk Transaction Manager connectors and configurations, turning from red to green when services are started.

Review the following for additional information.

<table>
<thead>
<tr>
<th>If you see this icon</th>
<th>A configuration is:</th>
<th>A connector is:</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="running" /></td>
<td>Running properly. All connectors in that configuration are running properly. All transactions start based on the triggering rules that you defined.</td>
<td>Running properly.</td>
</tr>
<tr>
<td><img src="image" alt="caution" /></td>
<td>Experiencing problems with one or more connectors that are not functioning properly.</td>
<td>This state does not apply to a connector.</td>
</tr>
</tbody>
</table>
If you see this icon: | A configuration is: | A connector is: |
---|---|---|
| | Stopped. All of the connectors are not running. Additionally, the Transaction Control Manager service or the FactoryTalk Transaction Manager service is not running. | Not running properly. |

### Configuration Server Status

The icons that are displayed in the Configuration tree represent the status of the configuration servers. The following table shows the status icons, state, and a description of that state:

<table>
<thead>
<tr>
<th>Icon</th>
<th>Status</th>
<th>Description:</th>
</tr>
</thead>
</table>
| | Running | The configuration server host computer:  
• Resides in the FactoryTalk Administration Console.  
• Is working properly (the service is running and FactoryTalk Transaction Manager can connect to it).  
• Has been configured in the FactoryTalk Administration Console so that the current user has permissions (page 136) to communicate with it. |
| | Unknown | The configuration server host computer:  
• May or may not reside in the FactoryTalk Administration Console.  
• Cannot be connected to or queried in FactoryTalk Transaction Manager.  
• Cannot provide any information to FactoryTalk Transaction Manager so that the software can determine why it is not working properly. |
| | No privilege | The configuration server host computer:  
• Resides in the FactoryTalk Administration Console.  
• Is working properly.  
• User does not have permissions (page 136) to communicate with the Configuration Server. |
Workspace

The right pane of the FactoryTalk Transaction Manager user interface is known as the workspace. Depending on the view option that you have selected, it displays:

- The Transaction Definition View.
- The Transaction Monitor View.
- The Error Log Files View.
- The FactoryTalk Transaction Manager system graphic.

The following is a sample Error Log File View.

<table>
<thead>
<tr>
<th>Date / Time</th>
<th>Source</th>
<th>Level</th>
<th>Transaction ID</th>
<th>Occurrence ID</th>
<th>Message Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/26/2003 09:46...</td>
<td>TRX</td>
<td>ERR</td>
<td>000010</td>
<td>000010</td>
<td>D00r thread transaction TRADS_Unsched_Cond</td>
</tr>
<tr>
<td>01/26/2003 09:46...</td>
<td>TRX</td>
<td>ERR</td>
<td>000010</td>
<td>000010</td>
<td>D00r thread transaction TRADS_Unsched_Cond</td>
</tr>
</tbody>
</table>

Transaction States

Transactions are displayed in the Transaction Definition view. The states of the transaction definitions are described in the table below:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current</td>
<td>The current definition of a transaction (with or without pending edits).</td>
</tr>
<tr>
<td>Edit Pending</td>
<td>Changes that are made to the current transaction definition, but not assembled.</td>
</tr>
<tr>
<td>Add Pending</td>
<td>A new transaction (created online), but not assembled.</td>
</tr>
</tbody>
</table>
The following is a sample Transaction Definition View.

<table>
<thead>
<tr>
<th>Transaction Name</th>
<th>State</th>
<th>Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>10SEC_LOG_CLOCKsEC6</td>
<td>Current</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>10SEC_LOG_WALLCLOCK4</td>
<td>Current</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>13SEC_FLOAT0</td>
<td>Current</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>18secBadPoint</td>
<td>Current</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>19secInput1onlyLogged</td>
<td>Current</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>float0_insert</td>
<td>Edit Pending</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>float1_insert</td>
<td>Add Pending</td>
<td>FactoryTalk Tr</td>
</tr>
<tr>
<td>INPUT1_BOOLINDEX</td>
<td>Current</td>
<td>FactoryTalk Tr</td>
</tr>
</tbody>
</table>

**Status Bar**

The status bar at the bottom of the main FactoryTalk Transaction Manager user interface displays the following information:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rows Selected</td>
<td>The number of rows selected before you saved, assembled or canceled pending edits.</td>
</tr>
<tr>
<td>Transactions Selected</td>
<td>The number of transactions selected.</td>
</tr>
<tr>
<td>Passed</td>
<td>The number of successful operations.</td>
</tr>
<tr>
<td>Disregarded</td>
<td>FactoryTalk Transaction Manager does not allow the action to be performed.</td>
</tr>
<tr>
<td>Failed</td>
<td>Operations that did not take place because of an internal error.</td>
</tr>
</tbody>
</table>

**Configuration Checklist**

A FactoryTalk Transaction Manager configuration consists of a set of transactions that use control and enterprise connector elements required to perform the transactions. You may create many configurations, but the Transaction Control Manager service or the FactoryTalk Transaction Manager service can run only one configuration at a time. Therefore, all the transactions required to implement an application must be contained in a single configuration.
You can create a configuration using the Configuration Checklist, which lists the required steps and displays the progress of your configuration. A green check mark indicates a completed step, while a yellow check mark indicates a partially completed step.

To display the Configuration Checklist for an already existing configuration:

1. In the Configuration tree, select the configuration name for which you want to display the Configuration Checklist.
2. On the menu, click Configuration > Checklist. The Configuration Checklist dialog box appears.

To create a new Configuration Checklist:

On the menu, click Configuration > New. The Configuration Checklist dialog box appears.

This chapter is not intended to provide step-by-step directions for creating a configuration, but to provide an overview to help you understand configurations. The following sequence mimics the Configuration Checklist. It is recommended, at least initially, that you follow this sequence when creating configurations:

- Define and name a new configuration. (page 33)
- Define control and enterprise connectors. (page 34)
- Define the data points that will be used in transactions. (page 35)
- Define the data objects that will be used in transactions. (page 35)
- Define transactions, which transfer data between data points, in the control system and data objects. (page 36)
- Verify transactions. (page 37)
Step 1: Defining andNaming a New Configuration

To define a new configuration:

1. In the Configuration Name text field, type the name of the new configuration.
2. Under Define Configuration, click Step 1. The FactoryTalk Transaction Manager Configuration dialog box appears.
3. Enter a name for the configuration and the directory path in which the configuration (*.dat) files will reside. This directory becomes the default location for the other files used in the configuration.

   **NOTE** Each configuration must have a unique name and directory path.

4. (Optional). If you want to create a configuration that uses online edits, select the Enable Online Edits check box.
5. Select the check boxes next to the control and enterprise connector services which will be used in the configuration.

   **NOTE** For a configuration that uses online edits, only the FactoryTalk Live Data control connector can be used. It is selected by default. You may additionally select any enterprise connector. For more information, see “Understanding Online Edits (page 89)”.

6. Click Apply. A message appears informing you the information was successfully saved.
7. Click OK, and then Close. A check mark appears next to Step 1 to indicate it is complete.
Step 2: Defining Connectors

To define a connector:

1. Under Define Connectors, from the drop-down list select the connector service to define.

2. Click Step 2. The Connector Definition dialog box appears. This dialog box allows you to configure communication and security settings for the Microsoft Windows services.

3. Click Apply. A message appears informing you the information was successfully saved.

4. Click OK, and then click Close. A check mark appears next to Step 2 to indicate it is complete.

You can create multiple control or enterprise connectors in a configuration, but you can create only one connector of each type on each host computer. A FactoryTalk Transaction Manager Professional activation is required to distribute connectors.

Only one instance of the Transaction Control Manager service or the FactoryTalk Transaction Manager service can be used by a configuration.

The connector service uses the Microsoft Windows user name and password assigned to the connector when it runs on the host computer.

In general, control connectors must run on the computer that contains the data server to which they will communicate. This is not the case with some OPC servers that support DCOM or OPC used in a FactoryTalk architecture.

After this step is complete and you exit the Configuration Checklist, the new configuration appears in the Configuration tree. Each connector service that is used by the configuration appears in the item tree. Individual control and enterprise connectors appear under the connector services. The Configuration tree displays all configurations recognized by each registered configuration server.
Step 3: Defining Data Points

To define data points:

1. Under Define Data Points, from the drop-down list select the control connector for which you want to add or edit points.

   NOTE
   If you have multiple control connectors, you must configure them individually.

2. Click Step 3. The Select An Application dialog box appears.

3. From the list of applications, select the application for which you want to define the data points, and then click OK. The (FactoryTalk Transaction Manager) FactoryTalk Data Point dialog box appears.

4. Under Select Tag(s), in the Folders list, browse to the directory containing the tags you want to use.

5. From the tags list, select the tags you want to use in a configuration, and then click Add Selected tag(s). The selected tags appear in a numbered list at the bottom of the dialog box.

   TIP
   You can select multiple tags by holding Ctrl and clicking on the tags.

6. Click Apply, and then click Close. A check mark appears next to Step 3 to indicate it is complete.

For more information about data points, see "Defining Data Points (page 55)". In a configuration that uses online edits, you can perform Step 3 on the Configuration Checklist while the configuration is running. For information about distributing control connectors and remote browsing capabilities, see "Exploring Advanced Topics (page 103)".

Step 4: Defining Data Objects

To define a data object:

1. Under Define Data Objects, from the drop-down list select the enterprise connector for which you want to define the data object.
2. Click Step 4. The **Data Object Definition** dialog box appears. Use this dialog box to configure connections to the database and create a data object.

   **NOTE** Depending on the type of the enterprise connector, the items available for configuration in the dialog box may differ. For more information about enterprise connectors, see "Defining Data Objects (page 67)".

3. Select the table, view, or connection (or appropriate enterprise object for your type of connector) to configure the data object to use. Depending on the connector type, different properties are displayed, such as whether to insert or update the rows in the table you select. For information about distributing enterprise connectors, see "Exploring Advanced Topics (page 103)".

4. Click **Apply**, and then click **Close**. A check mark appears next to Step 4 to indicate it is complete.

---

**Step 5: Defining Transactions**

You can create transactions that move data between a control system and an enterprise application or database.

To define a transaction:

1. Click Step 5. The Transaction Definition dialog box appears.

2. Provide a name for the transaction, and then select the data object to which you want to bind (the process of mapping a column in a database table to a data point) the data points. The data object’s columns or parameters appear in the list of available bindings near the bottom of the dialog box.

   **NOTE** Each transaction name must be unique.

   You can bind individual data points or an expression to a column/field/parameter in a data object. Double-click a non-bound entry in the Data Object Column to open the Filter and Select Data Points dialog box or right-click and select Filter and Select Data Points from the menu. You can limit the list of data points by a
connector and/or device. Bind the data point to the data object by dragging it from the Filter and Select Data Points dialog box and dropping it onto the appropriate Data Object Column in the Transaction Definition dialog box.

From the Transaction Definition dialog box, open the Trigger and Storage Parameters dialog box to specify the events that will initiate your transactions and timeout values. For more detailed information about transactions, see "Creating Transactions (page 75)".

In a configuration that uses online edits, you can edit existing or create new transactions while the configuration is running. When you have pending edits, you can view the differences between the current and pending definitions on the Transaction Differences dialog box. If the pending edits that you created have caused pending edit alerts, you can view them on the Pending Edit Alerts dialog box. For a detailed description of the information displayed on these dialog boxes, see "Understanding Online Edits (page 89)".

Step 6: Verifying Transactions

To verify multiple completed transactions, click Verify on the Configuration Checklist. You can also verify transactions individually from the Transaction Definition dialog box, which provides informational messages or warnings about the configuration.

Miscellaneous

In this section you will learn about the following:

- Viewing Configuration Properties (page 38)
- Starting Configurations (page 38)
- Stopping Configurations (page 40)
- Starting and Stopping Connectors (page 40)
- Monitoring Configurations (page 40)
Chapter 3  Exploring the FactoryTalk Transaction Manager User Interface

Viewing Configuration Properties

To view the properties of a configuration:

1. In the configuration item tree, select the configuration for which you want to view the properties.

2. Do either of the following:
   - On the menu, click Configuration > Properties.
   - Right-click the configuration name, and on the shortcut menu click Configuration Properties.

The Configuration Properties dialog box appears. You can access all configuration level settings, such as enterprise connector options and error logging levels, from individual tabs in this dialog box.

Starting Configurations

If you are starting a FactoryTalk Transaction Manager configuration, you must be logged into a Microsoft Windows account that has administrative privileges for all computers that are part of the FactoryTalk Transaction Manager system. This is required by the Microsoft Windows Service Control Manager to be able to start and stop FactoryTalk Transaction Manager services.

Only the Transaction Control Manager service or the FactoryTalk Transaction Manager service can run at one time. Further, the Transaction Control Manager service or the FactoryTalk Transaction Manager service can run only one FactoryTalk Transaction Manager configuration at a time. The configuration runs until it is stopped.

To start a FactoryTalk Transaction Manager configuration do either of the following:

- In the Configuration tree, right-click the configuration you want to start, and on the shortcut menu click Start Configuration.
- In the Configuration tree, select the configuration you want to start, and then on the toolbar click .
If you are using a distributed configuration, make sure all remote computers that are used in the configuration are running and available before you start the configuration. If the software encounters a remote computer that is not running or available, it will proceed to the next running and available computer. If your configuration includes RSView32 or FactoryTalk View SE, ensure that these products are running on the appropriate host computer and the project that is used by the configuration is loaded and running before you start the configuration.

Once you have started a configuration manually, you can set the configuration to start automatically.

To set a configuration to start automatically:

1. Go to Start > All Programs > Control Panel > Administrative Tools > Services. The Services window appears.

2. From the list of services, select either the FactoryTalk Transaction Manager Transaction and Control Manager service or the FactoryTalk Transaction Manager service (depending on your configuration), and double-click the service. The Properties dialog box appears.

3. From the Start-up box, select Automatic.

4. Click OK.

5. Repeat steps 2 through 4 for each connector service in your configuration.

NOTE The Transaction Control Manager service and the FactoryTalk Transaction Manager service cannot be set to Auto-start at the same time.
Stopping Configurations

To stop a configuration do either of the following:

- In the Configuration tree, right-click the configuration you want to stop, and on the shortcut menu, click **Stop Configuration**.

- In the Configuration tree, select the configuration you want to stop, and then on the toolbar click **Stop**.

You cannot stop the configuration by exiting FactoryTalk Transaction Manager or logging off from the computer. For more information, see "Exploring Advanced Topics (page 103)".

Starting and Stopping Connectors

To start an individual connector when a configuration is running, in the Configuration tree, right-click the connector, and on the shortcut menu click **Start Connector Service**.

To stop an individual connector when a configuration is running, in the Configuration tree, right-click the connector, and on the shortcut menu click **Stop Connector Service**.

Monitoring Configurations

You can monitor transactions as they are executed while a configuration is running. Click **Transaction Monitor** on the toolbar to view a summary of cumulative activity organized by transaction.

The following columns are available in the summary:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>The total number of transactions that have been triggered.</td>
</tr>
<tr>
<td>Passed</td>
<td>The number of transactions that have completed without errors.</td>
</tr>
<tr>
<td>Failed</td>
<td>The number of transactions that have failed.</td>
</tr>
<tr>
<td>% Passed</td>
<td>Passed/Total * 100.</td>
</tr>
<tr>
<td>Cached</td>
<td>The number of transactions currently in Transaction Cache files.</td>
</tr>
<tr>
<td>Database Passed</td>
<td>The number of transactions that have been successfully executed by the database.</td>
</tr>
</tbody>
</table>
Understanding FactoryTalk Transaction Manager

External Files

FactoryTalk Transaction Manager generates several types of external files while creating and running a configuration. You can set the location of these files from the FactoryTalk Transaction Manager user interface. You can determine where the configuration files are stored by viewing the Configuration Properties dialog box. For more information on viewing configuration properties, see Viewing Configuration Properties (page 38).

FactoryTalk Transaction Manager generates the following types of external files:

- **Configuration files (*.dat)** - FactoryTalk Transaction Manager stores all information associated with a single configuration in a set of *.dat files. The configuration files have fixed file names, so each configuration generates a set of identically named files. For this reason, configuration files must be stored in a unique directory.

- **Cached Transaction files (*.rsl)** - FactoryTalk Transaction Manager transactions can update their target database directly from cached transaction files. These files contain completed transactions that are applied to the database as a group. The use of cached transaction files is set individually for each transaction, but the files are associated with an enterprise connector. To edit cached transaction file properties, open the Configuration Properties dialog box, select the Cache tab and...
double-click a connector to open the Enterprise Connector Options dialog box.

- **Log files (*.log)** - Each of the FactoryTalk Transaction Manager services generates log files when the configuration is running. The level of error messages contained in these files is set at the configuration level. You can specify error log file parameters from the FactoryTalk Transaction Manager user interface by selecting **Define > Error Logging**. All log files can be viewed by selecting **View > Error Log Files**.

- **SQL files (*.sql)** - FactoryTalk Transaction Manager generates these SQL files as a result of a failed connection or database error. To include this data in the database, use a database maintenance utility.

- **RSQ files (*.rsq)** - FactoryTalk Transaction Manager uses these compressed files to backup configurations. An .rsq file is a result of a backup. You can restore an .rsq file by using the restore command.

- **RPB files (*.rpb)** - These files are generated when you use the Time-series Data Compression enterprise connector. FactoryTalk Historian Classic uses these files to save uncompressed, partial block information.

- **RFB files (*.rfb)** - These files are generated when you use the Time-series Data Compression enterprise connector. FactoryTalk Historian Classic uses these files to save compressed, full block information prior to storage in the database.

**Using the Service Console**

The Service Console is a Microsoft Management Console (MMC) snap-in that provides administrative functions for the Information Services Manager and other RSBizWare servers. The Service Console is capable of accessing and controlling RSBizWare servers that are running on machines across the network or on the local machine. It supports system user administration, server
administration, and any custom functionality specific to the RSBizWare component that is being managed.

FactoryTalk Transaction Manager can be managed via the Service Console when used as a part of a FactoryTalk Historian Classic or FactoryTalk Metrics applications. For more information, refer to the Administration Guide.

**NOTE** The Service Console is not installed with FactoryTalk Transaction Manager, but it is installed with the RSBizWare products.
In this chapter you will learn about the following:

- Introducing FactoryTalk Transaction Manager Services (page 45)
- Control Connectors (page 46)
- Enterprise Database Connectors (page 49)
- Enterprise Application Connectors (page 50)
- Enterprise Connector Options (page 51)
- FactoryTalk Transaction Manager Service (page 52)
- Transaction Control Manager Service (page 53)
- Configuration Server (page 53)

FactoryTalk Transaction Manager is designed to run as several services. During design time, the FactoryTalk Transaction Manager user interface sends information to the Configuration Server which writes to the configuration files. At run time, the other FactoryTalk Security services run in the background of the computer(s) involved in the configuration, similar to other Microsoft Windows services. For more information about configuring services, refer to "Exploring the FactoryTalk Transaction Manager User Interface (page 23)".

This chapter describes the types of control and enterprise connectors you must define during design time.
Control Connectors

The control connector services manage the interaction between the industrial control system and the FactoryTalk Transaction Manager service in the FactoryTalk Transaction Manager. The control connector services communicate with the data server using the appropriate protocol. You can use the following control connectors:

- FactoryTalk Live Data (page 46).
- DDE (page 46).
- RSLinx Classic OPC (page 47).
- RSView32 (page 48).
- Generic OPC (page 48).

Rockwell Software recommends using the FactoryTalk Live Data control connector for most user applications. If your application requires the use of any other control connector, please review and consider the following sections before selecting an alternate control connector.

FactoryTalk Live Data

The FactoryTalk Live Data control connector service is used to interface with data items provided by the FactoryTalk Live Data servers. The FactoryTalk Directory provides a common name space for factory automation products from Rockwell Software, allowing all applications to use the same naming convention and giving you the capability to browse available data points. FactoryTalk Live Data provides services that allow the efficient transfer of high-speed manufacturing data between processes in the system.

In a configuration that uses online edits, the Transaction Control Manager service performs the duties of the FactoryTalk Transaction Manager service and inherits the functionality of the FactoryTalk Live Data control connector. For more information about performing online edits, see "Understanding Online Edits (page 89)".
The DDE control connector service can be used for legacy connections to DDE servers or to provide functionality that is not supported by the OPC or FactoryTalk Live Data specification.

**NOTE**
The DDE server must be running on the same computer as the associated control connector. FactoryTalk Transaction Manager does not support NetDDE.

The FactoryTalk Transaction Manager user interface cannot query remote DDE servers to retrieve configured DDE topic information. When the DDE control connector is located on a remote FactoryTalk Transaction Manager computer, manually enter DDE topic names on the FactoryTalk Transaction Manager DDE Data Point Configuration dialog box. Perform the following steps.

1. Define a configuration on the local computer that contains the FactoryTalk Transaction Manager service.
2. Run the FactoryTalk Transaction Manager user interface on the remote computer and select a configuration that is using existing configuration files on the local computer.
3. Use the FactoryTalk Transaction Manager user interface on the remote computer to configure the DDE topics and items.

When you have finished, exit the FactoryTalk Transaction Manager user interface on the remote computer and continue creating configurations on the local computer.

**RSLinx Classic OPC**

The RSLinx Classic OPC control connector service can be used for data items that reside in Allen-Bradley Programmable Logic Controllers (PLC), with the exception of the Logix family of controllers. The RSLinx Classic OPC control connector must be on the same computer on which RSLinx Classic is running.
OPC recognizes when a controller sends a message to a client with the same value, but does not forward that message to the client. If you are using MSG instructions to trigger transactions, you must ensure that the value is different in each MSG instruction. Additionally, you can use the DDE control connector to forward all values to your application.

RSView32

The RSView32 control connector service can be used to interface with tags provided by RSView32 applications. The FactoryTalk Transaction Manager RSView32 control connector must be on the same computer on which the RSView32 project is running. The FactoryTalk Transaction Manager user interface can browse for RSView32 project tags on remote computers, but when the transactions are running, the RSView32 control connector must be on the same computer as the RSView project.

The RSView32 control connector can be used to collect data from either memory tags or device tags. Device tags are updated every 300 milliseconds.

NOTE If you need updated tags at a faster rate, use the FactoryTalk Live Data, RSLinx Classic OPC, or Generic OPC control connectors to retrieve data from the devices directly.
To interface with tags provided by FactoryTalk View SE, use the FactoryTalk Live Data control connector.

Generic OPC

The Generic OPC control connector service can be used to interface with items provided by any OPC server that conforms to the OPC custom interface specifications. The Generic OPC connector is an OPC client that supports OnDataChange subscription callback using either:

- **IAdviseSink** – for OPC 1.0A-compliant servers.
- **IConnectionPoints** – for OPC 2.0-compliant servers.

The Generic OPC connector service tries to establish the IAdviseSink method, and then tries the IConnectionPoints method.
Writing data to OPC items is performed using Asynchronous Writes. FactoryTalk Transaction Manager also supports Asynchronous Reads from Device as an option for OPC servers which support this method.

**Enterprise Database Connectors**

The enterprise database connector services manage FactoryTalk Transaction Manager interaction between a database and the Transaction Control Manager service or FactoryTalk Transaction Manager service. Each enterprise database connector service can manage connections with multiple databases on multiple computers. You can use the following enterprise connectors:

- Microsoft OLE DB (page 49).
- ODBC (page 49).
- Oracle OCI (page 50).

For transactions that update existing records, the data values that are used to look up a record in the database may update zero or more records in the database. If one or more records is selected, the transaction updates all of the selected records. If no records are selected, the transaction converts from Update to Insert mode, and the transaction is inserted into the database. This is not an error condition.

**Microsoft OLE DB**

The Microsoft OLE DB connector allows you to browse a Microsoft SQL Server database without a DSN.
The ODBC enterprise database connector allows you to connect with virtually any ODBC-compliant database including Microsoft Access, Microsoft SQL Server 2000 or later, IBM DB2, and Sybase.

The ODBC enterprise connector requires a system data source name (DSN) to connect to an ODBC data source. When you create an ODBC database connection, you must enter a valid system DSN for the ODBC server. A system DSN is available to all users and Microsoft Windows services.

The Oracle Call Interface (OCI) enterprise database connector allows you to connect to database objects from an Oracle server. If you are using Oracle OCI in FactoryTalk Transaction Manager, you need to locally install either of the following Oracle services:

- SQL*NET 2.3x or later
- Net8
- Oracle Net

You can then connect to an Oracle database versions 9i through 11g.

The enterprise application connector services manage FactoryTalk Transaction Manager interaction with the enterprise application connectors (Microsoft COM+, Time-series Data Compression, or FactoryTalk Metrics), and the Transaction Control Manager service or FactoryTalk Transaction Manager service.

You can use the following enterprise application connectors:

- Microsoft COM+ (page 51).
- Time-series Data Compression (page 51).
- FactoryTalk Metrics (page 51).
Microsoft COM+

The Microsoft COM+ enterprise application connector allows you to interface with Microsoft COM+ application components. These components must be integrated into a Microsoft Windows COM+ environment.

Time-series Data Compression

FactoryTalk Metrics

The FactoryTalk Metrics enterprise application connector is only used with FactoryTalk Metrics and stores data into the Service Console. The FactoryTalk Metrics enterprise application connector can only be configured through the Service Console. For more information, see the Administration Guide.

Enterprise Connector Options

You can set additional options for enterprise connectors in the Enterprise Connector Options dialog box.

To access the Enterprise Connector Options dialog box do either of the following:

- In the Configuration Checklist dialog box:
  1. Click Step2. The Connector Definition dialog box appears.
  2. Click Options. The Enterprise Connector Options dialog box appears.

- In the Configuration tree:
  1. Right-click the selected configuration, and on the shortcut menu click Configuration Properties. The Configuration Properties dialog box appears.
  2. On the Connectors tab, double-click the appropriate enterprise connector. The Connector Definition dialog box appears.
  3. Click Options. The Enterprise Connector Options dialog box appears.
You can set the following options:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of real-time threads</td>
<td>Specifies the number of real-time threads used by this connector. Increasing this value permits multiple real-time transactions to execute simultaneously in the enterprise connection with each thread having its own database connection. Do not increase this value unless the real-time transactions are not executing to the database fast enough. To make a transaction use real-time threads, select the Use Real Time Thread option in the Transaction Definition dialog box. For more information see Step 5: Defining Transactions (page 36).</td>
</tr>
<tr>
<td>SQL buffer size</td>
<td>Specifies the size of the SQL buffer in kilobytes. The SQL buffer specifies the number of bytes necessary to build the command that will be executed. The default value only needs to be modified if database errors occur and the SQL file shows only a partial command.</td>
</tr>
<tr>
<td>Maximum transactions per file</td>
<td>Indicates that a file is sent to the enterprise connector for processing when Maximum Transactions Per File or Maximum Time Between Files value is reached, whichever comes first. Setting this value to a field greater than 1 allows the enterprise connector to use array inserts on databases that support them. Array inserts increase database performance by allowing multiple inserts/updates in a single database command.</td>
</tr>
<tr>
<td>Maximum time between files</td>
<td>Indicates that a file is sent to the enterprise connector for processing when the Maximum Time Between Files or Maximum Transactions Per File is reached, whichever comes first.</td>
</tr>
</tbody>
</table>

**FactoryTalk Transaction Manager Service**

The FactoryTalk Transaction Manager service performs the following tasks:

- Controls the execution of all FactoryTalk Transaction Manager transactions.
- Collects and sends data to and from all connector services.
- Controls the scheduling and execution of the transactions.
- Controls data manipulation, if required.
Transaction Control Manager Service

The Transaction Control Manager is a service that controls and executes FactoryTalk Transaction Manager transactions contained in a configuration, but with the additional functionality of the FactoryTalk Live Data control connector embedded in it. In a configuration enabled for editing, the Transaction Control Manager replaces the separate FactoryTalk Transaction Manager and control connector services.

Configuration Server

The Configuration Server is a service that runs continuously to provide a single interface to the configuration files (with the .dat file extension) which make up the FactoryTalk Transaction Manager configuration. The Configuration Server simplifies access to the configuration files by filtering all changes to the files and by communicating with other FactoryTalk Transaction Manager services. A collection of all changes that affect a configuration is recorded in an audit trail (via either FactoryTalk Diagnostics or the Configuration Server *.log file).

Since the service is always running, functions such as configuration diagnostics and remote file browsing are easier. One of the benefits of the Configuration Server is consolidated file access. The Configuration Server is the focal point for all interactions with, and manipulation of the configuration files.
Defining Data Points

In this chapter you will learn about the following:

- Introducing Data Points (page 55)
- FactoryTalk Live Data Data Points (page 57)
- OPC Data Points (page 64)

Introducing Data Points

Data points are specific data locations or registers in the control system that are made available to FactoryTalk Transaction Manager transactions. The software can read information from, and write to data points through the embedded FactoryTalk Live Data control connector (Transaction Control Manager service) or a control connector (FactoryTalk Transaction Manager service). The control connector then communicates with a data server (based on OPC, DDE, or FactoryTalk) that communicates with the control system devices. Once defined, a data point can be used by multiple transactions.

Using FactoryTalk Transaction Manager, you can assign the following attributes to a data point:

- Item string or address.
- Data point name.
- Data type.
- Number of elements.
- Mode and Scan rate.
• Retrieval timeout.
• Substitution options.

These attributes are assigned in Step 3 of the Configuration Checklist by defining data points in the Data Point Definition dialog boxes. In these dialog boxes you can:

• Browse for data locations serviced by a data server.
• Add data points to the configuration.
• Edit existing data points.
• Delete unwanted data points.

The following figure shows a sample FactoryTalk Data Point dialog box.
Notice the following fields on the status bar at the bottom of the FactoryTalk Data Point dialog box:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration status</td>
<td>The status of the current configuration.</td>
</tr>
<tr>
<td>Rows</td>
<td>The number of rows selected before you saved, assembled, or canceled pending edits.</td>
</tr>
<tr>
<td>Points</td>
<td>The number of data points that specific row represents. For example, two rows may be two different data points or they may be the current definition and the pending definition of the same data point.</td>
</tr>
<tr>
<td>Passed</td>
<td>The number of successful operations.</td>
</tr>
<tr>
<td>Disregarded</td>
<td>The number of rows on which FactoryTalk Transaction Manager does not allow the action to be performed. For example, selecting a row that does not have a pending edit and clicking Cancel Edits.</td>
</tr>
<tr>
<td>Failed</td>
<td>Operations that did not take place because of an internal error.</td>
</tr>
</tbody>
</table>

For more information on online edits, see "Understanding Online Edits (page 89)"

FactoryTalk Transaction Manager uses FactoryTalk Live Data to communicate with the FactoryTalk Live Data servers. With access to the FactoryTalk Directory, the FactoryTalk Transaction Manager service can browse available data items directly. This is the only method that can be used to transfer data to and from FactoryTalk View SE or RSLinx Enterprise, and is the recommended method to transfer data to and from RSLinx Classic or RSView32.

In a configuration that uses online edits, you can use only FactoryTalk data points. For further details about using FactoryTalk data points in a configuration that uses online edits, see "Understanding Online Edits (page 89)"

Rockwell Software recommends using the FactoryTalk Live Data control connector for most user applications. The majority of this chapter will focus on information related to FactoryTalk Live Data.
data points. For information on all of the other data point types, see "OPC Data Points (page 64)".

**Selecting a Collection Mode**

You can select a collection mode for FactoryTalk data points. This section provides more detail about the following collection modes:

- Scheduled (page 58)
- Device Scheduled (page 58)
- Unscheduled (page 59)

**Scheduled: Maintain the Current Subscribed Value**

In the scheduled collection mode, each data point is continuously scanned at a configurable rate. The data server sends any change in value or quality to the control connector. The control connector retains the current value in a buffer, and provides it to the FactoryTalk Transaction Manager service when requested. The Transaction Control Manager service also buffers the data internally and uses it as needed.

The data server reads the value of the data point at the Subscription Scan Rate that is set in the FactoryTalk Data Point dialog box; if a new value is found, it is sent to the data client. If the value has not changed in the data server, no data is transferred. Thus network bandwidth is not used to transmit the same value between the data server and the control connector.

Typically, this collection mode is used to support a transaction that logs data constantly and rapidly (for example, logging a data point every second while an assembly line is running).

**Device Scheduled: Request the Current Value From the Device**

In the device scheduled collection mode, the data points are not scanned, and the data is not buffered in the control connector or data server. The data server reads directly from the device. Each value is read from the controller only when the Transaction Control Manager service or FactoryTalk Transaction Manager service
requests it from the control connector. The value is then retrieved from the controller and passed to the control connector.

Typically, this collection mode is used to support a transaction that logs data infrequently.

This collection mode is driven by the data server, and each data point is continuously scanned at the specified rate. When the server detects a change to a data point value or quality, it sends the value to the control connector, which passes it to the Transaction Control Manager service or the FactoryTalk Transaction Manager service. This collection mode is event-based, not time-based.

NOTE Keep in mind that any unsolicited message sent by the controller that does not include a change in value or quality will never be sent to FactoryTalk Transaction Manager.

The Subscription Scan Rate specifies how fast new data can arrive. For example, if the Subscription Scan Rate is 150 milliseconds, the data server cannot send changes faster. Even if the underlying data value is changing faster, the control connector can only see the current value every 150 milliseconds.

Typically, this collection mode is used for a transaction trigger that executes at a high or low transition, or exceeds a valid range.

In your application, you can quickly add multiple consecutive data points (for example, N7:0 through N7:6).

To add multiple consecutive data points:

- In the FactoryTalk Data Point dialog box, select the data points from the Contents of window.
- Click Add Selected Tags.

The data points are added to the FactoryTalk data point grid.

Unscheduled: Send Subscribed Value Whenever It Changes

Consecutive Data Point and Data Block Support
You can create a single data point with blocked data, such as a data point that has multiple consecutive elements. For example, if you want to create a data point with 10 elements, type `N7:0,L10`. This feature applies to pure data table files and not to structures (for example, `T3:0.ACC,L8` is not supported). The maximum size of a data block passed to the connector from a data point is 512 bytes for transactions without online edits, and 4096 bytes for transactions with online edits. If you defined a contiguous set of ASCII registers from a PLC processor, this data point contains a complete string (when used in a transaction). On the other hand, if you defined multiple consecutive elements with numeric data types in the contiguous registers of the PLC processor, you have an array. You can use the Parse() expression to extract each element from the block data.

**NOTE**

This feature also applies to the RSLinx Classic OPC, Generic OPC, and DDE control connectors.

**Selecting Timeout Properties**

Keep the following sections in mind when selecting timeout properties.

**Data Valid**

Regardless of the trigger mechanism, once a transaction is started, the Transaction Control Manager service or FactoryTalk Transaction Manager service checks each required data point to determine if the value in its local cache is still valid.

A data point value is not valid if the data valid time has passed or is set to zero. The following calculation demonstrates the validity requirements.

\[ \text{Data is valid} = \text{Time Received} + \text{Data Valid value} < \text{Current Time} \]

Once the data point value is no longer valid, the Transaction Control Manager or FactoryTalk Transaction Manager service requests the data point from a data server, and starts a timer for the retrieval of the data point. If the data point is not returned by the
time specified in the data retrieval parameter, the rules of substitution are applied.

When a data point is unscheduled, it is never requested, and the transaction fails if the data valid time has expired. Unscheduled data points do not have substitution values. If a data point is to be used as a transaction trigger as well as in other transactions, set the data valid parameter to a value that is large enough to keep the value fresh. Alternatively, you can create another data point using the same address, and make it a scheduled data point so that the Transaction Control Manager FactoryTalk Transaction Manager service can request it as needed.

**NOTE** An unscheduled trigger is considered to be fresh when it starts a transaction.

If the Transaction Control Manager or FactoryTalk Transaction Manager service starts a transaction, and determines that it has already requested a data point but not received a reply yet, it will not request the data point again; it will use the same value for both transactions when the value is received.

**Data Retrieval Timeout**

The data retrieval timeout is measured from the transaction start to the data point arrival. The data point timeout may be affected if there is communication latency between any of the following:

- The Transaction Control Manager service and the data server
- The FactoryTalk Transaction Manager service and the control connector, as well as the control connector and the data server

An example of this latency type is using a device-scheduled data point in FactoryTalk, where the following actions are performed:

1. The Transaction Control Manager service or FactoryTalk Transaction Manager service requests the data point from the control connector.
2. The control connector sends a notification to the FactoryTalk Live Data server.

3. The FactoryTalk Live Data server sends a read request to the controller, which in turn returns the data value.

4. The FactoryTalk Live Data server sends the value to the control connector, which sends it to the Transaction Control Manager service or FactoryTalk Transaction Manager service.

If the data point is not returned by the time specified in the data retrieval parameter, the rules of substitution are applied.

All data points need to have valid values to allow a transaction to complete successfully. If a data point is not retrieved and times out, the Transaction Control Manager service or FactoryTalk Transaction Manager service uses a substitution value. Unscheduled data points do not have substitution policies. If unscheduled data points are invalid, the transaction will fail.

**To select a substitution option:**

1. In the Data Point Definition dialog box, right-click the data point, and select Edit Selected Collection Parameters.

2. Choose one of the following substitution options for scheduled data points:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Substitution</td>
<td>Specifies that the transaction fails if valid data is not available for this data point. This occurs when the Transaction Control Manager service or FactoryTalk Transaction Manager service times out waiting for data, or when you have bad quality data.</td>
</tr>
<tr>
<td>Substitute Previous Value</td>
<td>Instructs the FactoryTalk Transaction Manager service to use the last good value for this data point.</td>
</tr>
<tr>
<td>Substitute Value</td>
<td>Allows you to specify the substitution value to use.</td>
</tr>
</tbody>
</table>

A substitution value of Null (default) causes the enterprise database connector to leave the value for the column empty. A data point that
has a Null substitution policy causes a transaction to fail if the data point is used in an expression. A Null value is not a null string or a zero. It is a value that does not exist; therefore, the expression evaluator cannot use it to calculate a result.

**Preventing Stale and Mismatched Data**

Stale data is data that no longer matches the value in the control system. Mismatched data refers to a set of data in which individual data values from different times were collected, and they are not synchronized. Several strategies exist to eliminate stale and mismatched data, depending on the transaction type. For scheduled transactions, data may be read while values are changing. For most applications, this should not be a concern (except for the case of high-speed data changes). If the data is changing at a high rate, you need to switch the transactions to unscheduled.

Unscheduled transactions offer better protection against stale and mismatched data via the use of ladder logic. The controller dictates when the data is read, and it can lock values into its registers prior to triggering the transaction. This helps prevent stale data from being read as long as the data valid time is set to zero for a non-trigger data point.

The best way to prevent stale or mismatched data is to use unscheduled data point blocks. All data is handled as a single unit, managed by the controller and parsed into separate units using the Parse function. This method may also reduce traffic on the controller network because the data is not scanned at a constant rate.

**Specifying Quality**

For the FactoryTalk Live Data connectors, choose one of the bad quality options if you want the Transaction Control Manager service or FactoryTalk Transaction Manager service to use bad quality values. Otherwise, select Use Substitution Option for Bad Quality in the Edit Collection Parameters dialog box. If you do not select a bad quality value, the Transaction Control Manager or FactoryTalk Transaction Manager service will use one of the
substitution policies listed earlier in this chapter when it receives a bad quality value for this data point.

If you choose to allow bad quality values, use the QualityOf() expression in the Expression Editor dialog box to bind the quality value to a column in your database. In addition, the bad quality status is saved to the Transaction Control Manager service log file or the control connector’s log file (when using the FactoryTalk Transaction Manager service) at an error level. The QualityOf() expression provides the following quality values:

<table>
<thead>
<tr>
<th>Server returns:</th>
<th>QualityOf() returns:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bad (0)</td>
<td>1</td>
</tr>
<tr>
<td>Uncertain (1)</td>
<td>2</td>
</tr>
<tr>
<td>N/A (2)</td>
<td>3</td>
</tr>
<tr>
<td>Good (3)</td>
<td>0</td>
</tr>
</tbody>
</table>

**NOTES**

This function only applies to FactoryTalk Live Data and OPC data servers. If you are using a DDE data server, a Good value is always returned. If you are using the bad quality substitution option and the controller is switched off, set the Data Retrieval property to a value large enough so that the server has enough time to respond.

**OPC Data Points**

OPC data points are typically used by the RSLinx Classic OPC, Generic OPC, and RSView32 connectors, but can also be used by FactoryTalk Live Data connectors.

“The OPC Specification is a non-proprietary technical specification that defines a set of standard interfaces based upon Microsoft’s OLE/COM technology. The application of the OPC standard interface facilitates the interoperability between automation/control applications, field systems/devices and business/office applications.” For further information, visit the OPC Foundation web site (https://opcfoundation.org/).
RSLinx Classic OPC Data Points

RSLinx Classic must run as a service (not as an application) on the computer that is running the RSLinx Classic OPC control connector. When FactoryTalk Transaction Manager uses OPC to communicate with RSLinx Classic, RSLinx Classic 2.52.00 is recommended.

In the RSLinx Classic OPC Data Point dialog box, data points are sorted by the associated connector, server, and host server. If you select a different OPC server, only the data points with that server are displayed; this also applies when you select a new server or connector. You should not create a new RSLinx Classic OPC control connector to collect data from data points in different OPC servers. Instead, select the new server and add the data points you want to include.

Generic OPC Data Points

The Generic OPC control connector service is an OPC client. It communicates with an OPC server according to OPC standards. In the OPC Data Points dialog box, you can browse a list of servers supported by the Generic OPC control connector. If the OPC server supports tag browsing, you can browse for the desired OPC items to create data points. The Generic OPC control connector has full DCOM support, which allows the OPC server to reside on a remote PC provided that the DCOM security is set correctly.

RSView32 Data Points

RSView32 6.10.16 or higher supports transmitting data via OPC. In RSView32 6.10.16 through 6.3.xx, the OPC browsing was not enabled. Therefore, if you use one of those versions of RSView32, you need to use the FactoryTalk Transaction Manager RSView32 connector to provide the browsing capability. If you use RSView32 6.40.00 or later, it is recommended to use the FactoryTalk Live Data connector. Alternatively, you can use the RSView32 connector or the Generic OPC connector.

To connect to data points using the RSView32 connector when the project does not reside on the FactoryTalk Transaction Manager user interface computer, you need to install RSView32 locally, in
order to provide the correct registry keys for remote browsing. The same applies when using RSView32 6.40.00 or later and the OPC connector. In the RSView32 project, use the RTDataServerOn command to turn on the data server and allow FactoryTalk Transaction Manager to read RSView32 tags. To provide the write access to RSView32 tags, you need to use the RTDataWriteEnable command.

**NOTE**

FactoryTalk View SE users need to use the FactoryTalk Live Data connector in FactoryTalk Transaction Manager. When using RSView32 data points and the FactoryTalk Live Data connector, use the FTDataServerOn and FTDataWriteEnable commands in the RSView32 project.

### DDE Data Points

To create a DDE data point, enter an Item String or import a tag or symbolic name from a tag file. RSLogix 5/500, AI5 project files, as well as CSV and TSV files are supported.

In the **DDE Data Point** dialog box, data points are sorted by the associated connector, server, and topic. If you select a different topic, only the data points with that topic are displayed. The same applies when selecting a new server or connector. If you select a new tag file after data points have been created for a connector/server/topic combination, the tag file is cascaded to all data points for that connector/server/topic. You do not need to create a new DDE control connector to collect data from data points in different topics. Instead, select the new topic and add the data points that you want to include.
Chapter 6

Defining Data Objects

In this chapter you will learn about the following:

- Introducing Data Objects (page 67)
- Enterprise Database Objects (page 68)
- Enterprise Application Objects (page 70)
- Enterprise Connector Error Handling (page 70)
- Inserting and Updating Data Table Records (page 72)
- Stored Procedures (page 72)

Introducing Data Objects

A FactoryTalk Transaction Manager data object references a specific object in an enterprise system. For enterprise database connectors, this object can be a database table, view, or stored procedure. For enterprise application connectors, this object can be a FactoryTalk Metrics function or a Microsoft COM+ method. The objects can be used in transactions. A data object may contain many columns or parameters that are used as a unit. Enterprise database objects can also include an action (for example, inserting or updating a record).

In FactoryTalk Transaction Manager, you can define data objects in the Data Object Definition dialog box. To access the dialog box, click Checklist on the Configuration menu, and then click Step 4.

You can assign the following attributes to a data object:

- Data object name
- Enterprise system connection
• Mode (Insert or Update)
• Database tables, views, or stored procedures
• Database columns or parameters

The target database tables, views, and stored procedures must already exist in the database to be used in data objects. You can also create new stored procedures using the Stored Procedure Wizard (for Microsoft SQL Server 7.0 or later and Oracle only).

**NOTE** If you modify target database tables, views, or stored procedures in the database after you create a data object in FactoryTalk Transaction Manager, you need to click Apply in the Data Object Definition dialog box so that the data object recognizes the database modifications.

Enterprise database connectors (ODBC, Oracle OCI, or Microsoft OLE DB) can communicate with databases located on other computers. Multiple data objects can reference the same database using a connection to the database system.

**Enterprise Database Objects**
The following sections describe how to connect to database objects for the enterprise database connectors.

**Oracle Call Interface (OCI) Data Objects**

The Oracle Call Interface (OCI) allows you to connect to database objects from an Oracle server. In the OCI Data Object Definition dialog box, you can create, modify, or delete FactoryTalk Transaction Manager data objects. When you create a data object, you need to enter a valid connection for the Oracle OCI server.

FactoryTalk Transaction Manager can communicate with a local or remote Oracle database via the Oracle network of client tools, such as SQL*NET, Net8, or Oracle Net Services.

**NOTE** 
You can install SQL*NET/Net8 on the computer that contains the enterprise connector, and communicate with remote Oracle databases running on any operating system supported by Oracle.

**Microsoft SQL Server Data Objects**

The Microsoft OLE DB connector service allows you to connect to database objects from a Microsoft SQL Server 2000, Microsoft SQL Server 2005, or Microsoft SQL Server 2008 R2. You can browse a Microsoft SQL Server 2000, SQL Server 2005, or SQL Server 2008 R2 database without entering a DSN.

**ODBC Data Objects**

The ODBC connector service allows you to connect to database objects using an ODBC 2.0 or a newer version of a compliant server. You need to use a valid system DSN, which is available to all Microsoft Windows services.

If the FactoryTalk Transaction Manager user interface resides on a computer that is remote from the computer(s) containing the enterprise connector, the system DSN list comes from the computer where the enterprise connector resides.

**NOTE** 
Rockwell Software does not recommend using ODBC data objects if you have an Oracle database. Instead, use Oracle OCI data objects.
Chapter 6  Defining Data Objects

**Enterprise Application Objects**

The following sections describe how to connect to database objects for the enterprise application connectors.

**Microsoft COM+ Data Objects**

FactoryTalk Transaction Manager communicates with COM+ components through the COM+ enterprise connector service. You can code the COM+ components using standard program development tools such as Microsoft Visual Basic or Microsoft Visual C++ (6.00). The COM+ components provide reusable functionality for large enterprise systems. The components reside on local or remote servers running Microsoft Windows.

Data is passed between the client applications and the remote server components in arguments that can accept many different types of data. FactoryTalk Transaction Manager restricts the arguments that it can utilize to scalar variables (variables that contain a single value). The data types that can be used are String, Integer, Long Integer, Single Float, Double Float, Byte, Date, and Boolean (True/False).

To create a data object, select a COM+ method in the COM+ Connection Definition dialog box.

**FactoryTalk Metrics Data Objects**

You can only connect to FactoryTalk Metrics data objects from the FactoryTalk Metrics Server. This connector is different from the other connectors because it preprocesses the data before sending it to the database. For more information, see the FactoryTalk Metrics User Guide.

**Enterprise Connector Error Handling**

The following are the most likely FactoryTalk Transaction Manager error conditions that can affect the enterprise connector:

- **Lost connection with the enterprise connector** - The Transaction Control Manager service or FactoryTalk Transaction Manager service cannot communicate with the enterprise connector. This may occur because the enterprise connector service has stopped running, or the enterprise connector is located on another computer and the
communication between the computers has been interrupted. The Transaction Control Manager or FactoryTalk Transaction Manager service will create cache transaction files for transactions that use historical logging (if the **Use Cached Transaction Files** check box is selected in the **Transaction Definition** dialog box). Therefore, you should store configuration cache files on the same computer as the Transaction Control Manager or FactoryTalk Transaction Manager service.

- **Lost connection with the enterprise database** - The enterprise connector cannot communicate with the database. This may occur because the database service has stopped running, or the database is located on another computer and the communication between the computers has been interrupted. The enterprise database connector stores the data to be reapplied when the database connection returns.

  **NOTE** If you select the **Use Cached Transaction Files** check box in the **Transaction Definition** dialog box, a *.sql file is generated, which you need to manually apply to the database. The software keeps generating *.rsl files until the connection returns. FactoryTalk Transaction Manager does not process any further *.rsl files if there is no connection.

- **Lost connection with Microsoft COM+** - The enterprise application connector cannot communicate with Microsoft COM+. This may occur because the FactoryTalk Transaction Manager COM+ service has stopped running, or the Microsoft COM+ component has stopped working. Microsoft COM+ may respond by creating a *.txt file, which displays the insert method.

To access diagnostic information about the currently running configuration, click **Transaction Monitor** on the **View** menu while the configuration is running. The **Transaction Monitor** dialog box appears.
Chapter 6  Defining Data Objects

The following columns display failed transactions:

- The Failed column - Lists transactions that were not successfully processed.
- The Database Failed column - Lists transactions that were processed correctly, but could not be applied to the database due to a database error.

For more information on error conditions that can affect enterprise connectors, as well as how FactoryTalk Transaction Manager handles transaction types and storage methods, see "Creating Transactions (page 75)."

Inserting and Updating Data Table Records

When logging data to a table, you can use FactoryTalk Transaction Manager to directly insert data (creating new records in the table), or update existing data in the table.

In the Data Object Definition dialog box, the Insert mode is selected by default. When you select the Update mode, some of the data points may be used as criteria for selecting rows to be updated, while other data points will be used to update the values in the selected columns. If no rows match the given criteria, the data will be inserted into a new row.

Stored Procedures

A stored procedure is a user-defined function or program that is executed in the database. It can consist of any of the components of a structured language that enable you to define data behavior. A stored procedure can be simple like a single select command, or complicated like validating all data before it is inserted into a table.

A stored procedure works like a function that is stored in a database. Most databases provide a comprehensive stored procedure language...
that combines the data query capabilities of SQL and some kind of procedural control (e.g., the "If...Then" statement). As most kinds of function calls, a stored procedure can have both inputs and outputs. In a transaction that is connected to a stored procedure, the values bound to inputs are collected from the control system, and the outputs are returned to the control system.

When you select the **Stored Procedure** option in the **Data Object Definition** dialog box under **Database**, FactoryTalk Transaction Manager queries the database for all stored procedures accessible by the chosen database user account, and they appear in the **Stored Procedure** list. When you select a stored procedure, the input and output parameters are displayed under **Parameters**. Since all parameters for a stored procedure are required, the software automatically adds them under **Data Object Columns** when they are selected.

To simplify the creation of simple stored procedures, FactoryTalk Transaction Manager provides the Stored Procedure Wizard for SQL Server and Oracle databases. To access the wizard, select the **Stored Procedure** option, and then click **Wizard**. For complicated stored procedures and/or debugging of stored procedures, use database-specific tools.
Creating Transactions

In this chapter you will learn about the following:

- Introducing Transactions (page 75)
- Transaction Types (page 77)
- Transaction Timeout (page 82)
- Transaction Completion (page 82)
- Using the Expression Editor (page 84)

**Introducing Transactions**

FactoryTalk Transaction Manager transactions move data between the control system and the enterprise system. Binding is the process of mapping:

- A column in a database table to a data point in a control system, expression, or a literal string.
- A parameter in a stored procedure to a data point, expression, a literal string, or a null value.

A configuration can contain any number of transactions, each running independently based on a trigger event. Although the number of transactions in a configuration is unlimited, you cannot exceed the tag count for which you are licensed (see Activation Options). In a configuration that uses online edits, you can edit existing or create new transactions while the configuration is running. For more information, see "Understanding Online Edits (page 89)".
You can assign attributes to a transaction in the **Transaction Definition** dialog box.

**To access the dialog box:**

1. On the **Configuration** menu, click **Checklist**.
   The **Configuration Checklist** dialog box appears.
2. Enter the configuration name and then click **Step 5**.
   The **Transaction Definition** dialog box appears.

![Transaction Definition dialog box]

In the dialog box, you can assign the following transaction attributes:

- Transaction name
- Transaction timeout
- Data object name
- Bindings of data object elements to data points/expressions
- Trigger event
- Transaction (cached or real-time) and storage options
Creating Transactions

Transaction Types

FactoryTalk Transaction Manager supports two types of transactions, regardless of the trigger mechanism:

- Unidirectional transactions
- Bidirectional transactions

For either transaction type, you can optionally specify the transaction result to be bound back to a control system.

Unidirectional Transactions

Unidirectional transactions in FactoryTalk Transaction Manager use information from the control system to add records to a database table, update the contents of existing records, or call stored procedures that do not contain output parameters.

Unidirectional transactions are commonly used to log production data to a database, including:

- The monitoring of performance
- The sampling of quality analysis
- The collecting of real-time production information
- The tracking of material consumption
- The product tracking
- The reporting of the end of job/batch/shift

Unidirectional transactions are the simplest transaction type. They can be used to perform the database Insert or Update commands. They can also be used to perform simple stored procedures, provided no output parameters or return codes exist. In most applications, unidirectional transactions account for the majority of transaction volume. Data may be collected frequently from a large number of data points.

Bidirectional Transactions

Bidirectional transactions in FactoryTalk Transaction Manager take data from the control system, and then call a stored procedure,
which exercises some logic and provides the software with output values that can be written back to the control system. It is possible to set up bidirectional transactions so that the data is not sent to the database, but downloaded from the database to a control system.

Bidirectional transactions are the most powerful transaction type, because they allow transactions to interact with a database stored procedure or Microsoft COM+ method. Bidirectional transactions let you perform:

- The downloading of product parameters.
- The dynamic routing.
- The dynamic production scheduling.
- The controller centralizing.
- The production floor interacting.
- The warehousing of automated storage and retrieval.

A bidirectional transaction with input/output bindings implements a data transfer from the enterprise system and sends it to the control system, binding the input and output parameters of the database stored procedure. The control system data serves as an input to a stored procedure. The results of a stored procedure can be written back to data points in the control system (enabling the creation of sophisticated transactions that allow a high degree of interaction between a database system and the factory floor).

For transactions with output bindings, the **Real Time Thread Storage** option needs to be selected in the **Transaction Definition** dialog box, because the procedure or method must complete before the data can be returned to the Transaction Control Manager service or FactoryTalk Transaction Manager service. FactoryTalk Transaction Manager allows you to specify the number of database threads used by a connector to improve performance.
When creating a transaction using input/output bindings, parameters are bound in the same manner as columns in a table. The input data points are collected, expressions are evaluated, and the procedure is called. If the procedure succeeds, output parameters are written to the control system.

Transactions with input/output bindings use the following parameter types:

- **Input parameters.** In a stored procedure, these parameters must have a binding (a data point, expression, or a null). Right-click a data object parameter and select **Bind Data Point** or **Bind Null Value**. A transaction must have all input parameters bound to it before it can be enabled. Binding a null value to an input eliminates the need to gather a value from the database.

- **Output parameters.** These parameters do not require a binding. The results of the bound value are displayed in the **Bound Value** column. If an output parameter is not bound, or is bound to a null, the value is ignored and the column is empty. A procedure must succeed before output parameters can be returned to the control system. Normally, you bind outputs to a data point in the control system; upon a successful completion of a transaction, a value is written from the stored procedure to the data point.

- **Input/output parameters.** These parameters are handled as a single binding unless they are separated using the **Separate Input/output** option. This option allows separate data points to be bound to the same procedure parameter. This implies that the address from which the input parameter is derived is different from the address to which the output parameter is written. This also implies that if an input/output parameter is used only for the output, then the input portion can be bound to a null value. Leaving an input/output parameter bound to a single data point causes the value to be read prior to executing
the procedure, and the output value to be written upon the transaction completion.

The Microsoft SQL Server RETURN_CODE contains data that can be bound as an output to a stored procedure. This value is only available if the procedure executes successfully. A successful return code does not guarantee a successful transaction as the transaction is not yet complete.

A bidirectional or unidirectional transaction with a transaction result binding implements a data value that is written back to the control system; the data value allows you to determine whether the transaction has been completed successfully. The control system can then take appropriate actions based on the success or failure of the transaction. For example, consider a high-liability manufacturing environment where a verifiable quality record is a requirement for each item produced. At various steps in the manufacturing process, a transaction may send a test result to be stored in the central database. If the test result is not recorded successfully, the part on the production line may become worthless.

The transaction result, which notifies the control system that the transaction was successful, is sent variably based on the transaction type. In a real-time transaction (where data flows straight to the database), the transaction result is sent upon the data being successfully logged into the database. However, in the case of a cached transaction, where a cache is used to ensure data integrity, the transaction result is sent once the data has been logged into the cache, and not necessarily into the database. The transaction result is used to confirm that the test result was recorded. If it indicates that the test result was not recorded, the control system can respond appropriately by alerting the operator, changing the part’s routing, or retrying the transaction. The following are two examples:

- Validated data logging
- Closed-loop quality tracking
The Transaction Result Binding option allows a transaction to return a transaction result code to the control system. To enable the option, select the Bind Transaction Result check box in the Transaction Definition dialog box, and then in the list select a data point to accept the transaction result code. The control system should take the appropriate action, depending on the transaction result.

The transaction result code is a 16-bit integer. Bit 0 is the least significant, Bit 15 is the most significant. The transaction result code consists of the following:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit 0: the Done Bit</td>
<td>A value of 1 indicates the transaction has completed, 0 indicates that it has not. No information is given on the transaction's success or failure.</td>
</tr>
<tr>
<td>Bit 1- The Error Bit</td>
<td>A value of 1 indicates that there was an error, 0 indicates that no errors occurred.</td>
</tr>
<tr>
<td>Bits 2 through 15- The Error Code</td>
<td>If Bit 1 contains 1, the bits contain the error code. Otherwise, the bits are zeros. If you have chosen to use the transaction result code, you can run the FactoryTalk Transaction Manager error utility from the Startup menu. This eliminates the need to decode the integer to determine the FactoryTalk Transaction Manager error code.</td>
</tr>
</tbody>
</table>

To convert the error code to a proper error number:

1. Click Start > Programs > Rockwell Software > FactoryTalk Transaction Manager > Error Messages.
   
   The FactoryTalk Transaction Manager Error Messages dialog box appears.

2. Enter the transaction result code number (binary or decimal).

3. Select Bind Transaction Result Error.

4. Click Apply.
For example, a transaction result code of 0000000000000001 (binary) or 1 (decimal) indicates that the transaction has been completed without errors.

If the transaction result code is 0000010111101111 (1519 decimal), the FactoryTalk Transaction Manager error code is 33147.

**Transaction Timeout**

The transaction timeout parameter specifies how long the Transaction Control Manager service or FactoryTalk Transaction Manager service should wait for a transaction to complete. The timeout for an unscheduled transaction does not affect other copies of the same transaction, because multiple copies of an unscheduled transaction can be executed at the same time.

**NOTE** Setting the transaction scan rate for a scheduled transaction lower than the transaction timeout may cause the second transaction to fail, because only one scheduled transaction can run at a time.

**Transaction Completion**

It is important to determine when the Transaction Control Manager service or FactoryTalk Transaction Manager service considers a transaction to be complete, because the services run only one occurrence of a scheduled transaction at a time. The next occurrence cannot start until the current transaction is complete. Additionally, if a transaction has a transaction result binding, the transaction result is written when the transaction completes.

**Cached Transactions**

With the Use Cache Transaction Files option selected in the Transaction Definition dialog box under Transaction Options, unidirectional transactions complete as soon as the data is written to the cache file. The data is not stored to the database yet, but it is on the disk.

**Real-time Transactions**

With the Use Real Time Thread option selected in the Transaction Definition dialog box under Transaction Options, unidirectional transactions complete when the Transaction Control Manager service or FactoryTalk Transaction Manager service
receives a reply from the enterprise connector that the data has been stored. This means that the time necessary for the enterprise system to store the values is included in the transaction completion time. This may cause the transaction to time out if the Transaction Control Manager service or FactoryTalk Transaction Manager service has not received the reply from the enterprise connector in the allotted time. The transaction timeout does not determine if the data was logged to the database. The transaction can time out prior to sending the data to the database or after the data was successfully stored.

The table below shows how FactoryTalk Transaction Manager handles each transaction type and storage method.

<table>
<thead>
<tr>
<th>Transaction type</th>
<th>Transaction storage method</th>
<th>Lost connection with enterprise connector</th>
<th>Lost connection with enterprise system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unidirectional</td>
<td>Cached transaction files</td>
<td>Cache files are applied when the connection is restored.</td>
<td>One cache file can be converted to an .sql file and returns Database Failed. Remaining cache files are processed when connection is restored.(1)</td>
</tr>
<tr>
<td>Unidirectional</td>
<td>Real time thread</td>
<td>The transaction fails and transaction data is lost.</td>
<td>Transaction returns Database Failed, transaction data is written to an .sql file.(2)</td>
</tr>
<tr>
<td>Bidirectional</td>
<td>Real time thread</td>
<td>The transaction fails and transaction data is lost.</td>
<td>Transaction returns Database Failed, transaction data is written to an .sql file.(2)</td>
</tr>
<tr>
<td>Transaction Result Binding</td>
<td>Cached transaction files</td>
<td>The transaction data is written to a cache file. The transaction returns a successful result to the controller. The cache files are processed when the connection is restored.</td>
<td>Transaction returns a successful result. One cache file can be converted to an .sql file and returns Database Failed. Remaining cache files are processed when the connection is restored.(1)</td>
</tr>
<tr>
<td>Transaction Result Binding</td>
<td>Real time thread</td>
<td>The transaction fails and transaction data is lost. The failure is sent to the controller.</td>
<td>Transaction returns Database Failed, transaction data is written to an .sql file.(2)</td>
</tr>
</tbody>
</table>

(1) If multiple database connections are defined in the configuration, the cache files continue processing. All data for the failed
Data is stored in separate .sql files, depending on whether the Use Real Time Thread or the Use Cache Transaction Files option was selected in the Transaction Definition dialog box. This allows you to recover the data.

For more information on the enterprise connector error handling, see "Defining Data Objects (page 67)".

**Bidirectional Transactions**

Bidirectional transactions with input/output parameters are not complete until the control connector acknowledges that the control system has received the transaction output data. The transaction timeout includes the time it takes the data server to connect to the control system, write the data values, and return a response to the control connector. Transactions with outputs may have latency in both the enterprise connector and the control connector.

**Transactions With Bound Transaction Results**

Bidirectional transactions that use a transaction result binding are not complete until the Transaction Control Manager service or FactoryTalk Transaction Manager service receives a reply from the control connector that the bound result was written. If a transaction times out after the bind transaction result is sent to the control connector, all data is moved successfully to the controller, and a message indicates that the transaction failed.

**Database Triggers**

Database triggers are functions that are executed by the database whenever the triggering operation occurs. For example, a trigger can be set in such a way that whenever a value is inserted into a table, the data is verified, and then another value is updated with the verified data. The amount of time necessary to process the trigger and the associated function is charged against the transaction timeout. In this case, the database insert does not return control to the enterprise connector until the insert and its triggered function are complete. In other words, a real-time transaction is considered complete when the FactoryTalk Transaction Manager transaction and any database triggers caused by the transaction are complete.
Using the Expression Editor

Use the Expression Editor to define calculations or formulas with mathematical operators and functions that can be bound to a data object. To display the Expression Editor, right-click a data object in the Transaction Definition dialog box, and then select Bind Expression.

Logical and Mathematical Operators

Mathematical operators define simple expressions, which perform calculations that are evaluated after all data has been collected (or substituted). There is also a set of operators for both bitwise and logical operations. Once an expression has been defined, the syntax and semantics are checked to determine if it can be evaluated at runtime. After the data points have been collected, their current values are used to evaluate the expression. The results are then passed to the database for processing.
The Expression Editor provides several functions for storing the current time. Select time functions can be expressed in either Coordinated Universal Time (UTC) or system local time.

The TimestampOf() function logs the time when a data point is read. This may differ from the time the transaction executed because the data point may have a data valid time of greater than zero.

The TransTimestamp() function returns the transaction execution time. This may differ from the time the data was inserted into the database, because completed transactions may be buffered in cached transaction (*.rsl) files.

The MTimestampOf() and MTransTimestamp() functions log the millisecond portions of the two previous times. These can be stored in separate columns, which permits the accurate trend analysis for databases that do not store time values to the millisecond.

The OPCTimeStampOf() and MOPCTimeStampOf() functions return an OPC timestamp value that indicates when the OPC data server (or FactoryTalk data server) received (or read) the data from the controller. If the data server is RSLinx Classic, this is the time when RSLinx Classic provided data to FactoryTalk Transaction Manager. If the data server is RSLinx Enterprise, this is the time when RSLinx Enterprise acquired the data from the controller. This value is accurate to the nearest second.

The Expression Editor supports a data point range syntax that compares a data point value over a series of transactions. Each time a transaction runs, a new value is added to the data range for a given transaction. The expression is then evaluated using the range of values. This allows the expression to calculate an average of the previous 10 transaction values (avg(datapoint[0,9]), as well as calculate the minimum (min) or maximum (max) value of a data point over several transactions.
NOTE You must run the transaction to reflect the changes in the historical values. However, you do not need to store the results.

Using the Store on Every N Transactions option, you can collect the data needed for an average, but not store the data to the database. If a transaction executes every second but only stores its data every 60 times, and there is an \( \text{avg}(\text{datapoint}[0,59]) \), the value that is stored once a minute is the average of the values taken every second.

In a running configuration, if you make changes to transactions (or their bound data points) using the Data Point Range function in conjunction with the Avg function in an expression, the transaction resets, or behaves as though it is starting for the first time when you assemble the pending edits. For more information, see "Understanding Online Edits (page 89)".

**Parse Function**

The Parse function in the Expression Editor ensures that all the data for a transaction is synchronized. The input for the Parse function is a block of data, and the output is a parsed subset. This allows the control system to manage all the data into a single data point, which can be sent to the control connector using an unsolicited message. The control connector then sends the data as a single unit to the Transaction Control Manager service or FactoryTalk Transaction Manager service. The service uses the data point arrival as a trigger, and then parses out the individual values as needed. Since the control system gathered all of the data into a single block and the block was sent to the service as a single unit, it is synchronized. The Parse function can also be used on scheduled data points and in scheduled transactions to separate data values from a single data point.
In this chapter you will learn about the following:

- Introducing Online Edits (page 89)
- Understanding Online Edit Concepts (page 90)
- Creating a Configuration That Uses Online Edits (page 96)

### Introducing Online Edits

The ability to change data points and transactions in a running FactoryTalk Transaction Manager configuration is known as online edits. The use of online edits allows continuing the data collection in your automation system, while you add new or modify existing data points and transactions.

Online edits include current and pending edits. A current edit reflects how a data point or a transaction is currently defined in the running configuration. A pending edit reflects a new definition of a data point or transaction after you change and save it (before it has been assembled). Pending edits do not affect the currently running configuration until they are assembled. When assembled, the pending edits replace the current definition (i.e., become the new current definition).

When discussing online edits, you need to understand the difference between how data points and transactions are defined in the currently running configuration, and what changes take place when you assemble.
## Understanding Online Edit Concepts

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuration that uses online edits</td>
<td>A configuration you can change while it is running.</td>
</tr>
<tr>
<td>Current definition</td>
<td>The definition of a currently running transaction or data point.</td>
</tr>
<tr>
<td>Pending definition</td>
<td>The changes made to data points or transactions in a running configuration that uses online edits. Pending edits must be saved before they can be assembled. Pending edits do not affect the running configuration until they have been assembled.</td>
</tr>
<tr>
<td>Assemble pending edits</td>
<td>The process of changing the running, current definition of a transaction or data point in a configuration that uses online edits to the pending definition. You cannot assemble pending edits until you save them first.</td>
</tr>
<tr>
<td>Cancel pending edits</td>
<td>The process of removing pending edits made to data points or transactions. Since pending edits do not affect the running configuration until they have been assembled, the definition of a data point or a transaction reverts to the current definition, and there is no effect on the running configuration.</td>
</tr>
<tr>
<td>Pending edit alerts</td>
<td>Informational messages that describe the side effects of the pending data points or transaction edits made on running transactions.</td>
</tr>
<tr>
<td>Transaction Control Manager</td>
<td>Similar to the FactoryTalk Transaction Manager service, but with the additional functionality of the FactoryTalk Live Data control connector embedded in it. In a configuration that uses online edits, the Transaction Control Manager service replaces the separate FactoryTalk Transaction Manager service and FactoryTalk Live Data control connector services.</td>
</tr>
</tbody>
</table>
Online Edits Workflow

The following diagram illustrates the multi-step process of changing a configuration that uses online edits.

The running configuration is not affected until the end of the process when the pending edits are assembled. Saving, reviewing, or canceling pending edits prior to assembling only affects the configuration files, not the running configuration itself.

Configurations That Use Online Edits

A configuration that uses online edits allows you to add new data points and transactions, as well as change existing data points and transactions while it is running. These changes are referred to as pending edits. In defining configurations that use online edits, you automatically use the Transaction Control Manager, which communicates exclusively with FactoryTalk Live Data data servers.
If you wish to convert existing configurations to configurations that use online edits, you need to migrate the existing configurations that use RSLinx Classic, RSView32, and OPC connectors to configurations that use the FactoryTalk Live Data connector. For more information about available migration methods and tools, refer to the product CD or Rockwell Automation Knowledgebase Web site. Go to https://rockwellautomation.custhelp.com/, and search by entering keywords such as FactoryTalk Transaction Manager, RSSql, or migration.

In a running configuration that uses online edits, you can:

- Create new and modify existing data points.
- Create new and modify existing transactions.
- Enable and disable transactions.
- Save and assemble pending edits.
- View differences between current transaction definitions and transactions with pending edits.

In a running configuration that uses online edits, you cannot:

- Modify configuration properties.
- Modify error logging properties.
- Add new, modify, or delete existing enterprise connectors.
- Modify enterprise or control connector properties.
- Use any control connector except FactoryTalk Live Data.
- Modify database connection properties.
- Add new, modify, or delete existing data objects.
- Modify a starting or stopping event definition.
- Change data point names or modes (scheduled, unscheduled, or device-scheduled).
• Delete data points.
• Delete transactions (although you can disable them).

NOTE If you have a running configuration with pending edits and you stop it, you cannot make any further changes to the items with pending edits until you cancel or assemble the pending edits.

Assembling Pending Edits

Assembling pending edits is the process of making the running configuration aware of the changes made to data points or transactions. The process may affect different types of pending edits, depending on the location in which they are assembled. To start assembling pending edits, you need to save them first; otherwise, a warning message appears.

You can assemble pending edits in the following locations:

• The main FactoryTalk Transaction Manager window. When assembling pending edits from the toolbar in the main FactoryTalk Transaction Manager window, all pending edits for transactions and data points are assembled.

• The FactoryTalk Data Point dialog box. Assembling in this dialog box affects only the selected data point rows and data point pending definitions. Transaction pending definitions are not assembled.

• The Pending Transaction Definition dialog box. Assembling in this dialog box affects only the pending definitions for the transaction you are currently viewing. Data point pending definitions are not assembled.

• The Pending Edit Alerts dialog box. When assembling transaction or data point pending edits that have caused side effects you may not be aware of, FactoryTalk Transaction Manager prompts you to review the pending edits. If you click Assemble All in the Pending Edit Alerts dialog box, all data point and transaction pending edits are
assembled (including the ones that have not caused alerts), and they are no longer displayed in the dialog box.

For more information about assembling pending edits, refer to the FactoryTalk Transaction Manager online help.

**Canceling Pending Edits**

Canceling pending edits is the process of removing pending edits made to data points and transactions in a running configuration that uses online edits. Canceling pending edits does not affect the running configuration. Data points or transactions return to their original definitions.

You can cancel pending edits in the FactoryTalk Data Point, Pending Transaction Definition, and Pending Edit Alerts dialog boxes. To cancel pending edits, you need to save them first. You cannot cancel pending edits if they are already assembled.

**Pending Edit Alerts**

Pending edit alerts are messages informing you that a transaction with the state information (i.e., internal buffers containing information about the current and previous transaction states) will lose the information.

**Occurrence Conditions of Pending Edit Alerts**

Some transactions require the state information for their correct execution. If you make some changes to the transactions or the bindings they use, the transactions reset or behave as though they are starting for the first time when you assemble pending edits. FactoryTalk Transaction Manager warns you that the transaction will be reset by displaying a pending edit alert message.

A transaction requires the state information if it has any of the following characteristics:

- Ignore First Unscheduled Event
- Transactions Stores Data On Number of Completed Transactions
- Transactions Stores Data On Data Change and/or Rate
• An expression using the DIFF function
• An expression using the Data Point Range function used in conjunction with the Min, Max, or Avg functions

A pending edit alert occurs if your enabled transaction includes any of the characteristics listed above, and you:

• Perform any of the following tasks:
  • Trigger type
  • Unscheduled event data point trigger
  • Add a binding
  • Delete a binding
  • Data point to expression
  • Expression to data point
  • Data type of binding
  • Data point in a binding
  • Expression in a binding
  • Order of bindings.
  • Merge input/output parameters in a binding.
  • Separate input/output parameters in a binding.

• Change any of the following data point parameters:
  • Number of elements in an array
  • Size of string
  • Address of a data point
  • Data type of a data point

In the Pending Edit Alerts dialog box, you can review the transactions that have pending edit alerts. The transactions which do not have alerts are not displayed in the dialog box. You can
assemble or cancel the selected transactions or all the transactions in the configuration (even the ones not displayed in the dialog box).

Creating a Configuration That Uses Online Edits

To create a configuration that uses online edits:

- Use the FactoryTalk Live Data control connector exclusively
- Select the Enable Online Edits check box

Editing Data Points In a Running Configuration That Uses Online Edits

You can edit data points (or create pending edits) in a running configuration by adding new data points or editing existing data points in the FactoryTalk Data Point dialog box. The FactoryTalk Connector and Application fields cannot be edited.

Adding New Data Points

To add new data points, navigate to the correct area in your FactoryTalk Directory in the Select Tags group and double-click to open the folder/area.

Select a tag in the Contents of window, and then click Add Selected Tag(s).
The new data points appear in red in the data point grid. This new data point is considered a pending edit.

In creating new data points and saving them, saving pending edits does not affect the running configuration because they have not been used in a transaction. Assembling new data points that are not used in a configuration that uses online edits does not affect the running configuration because FactoryTalk Transaction Manager does not collect data for data points that are not used. Data points must be assembled before they can be used in a new transaction.

**Editing Existing Data Points**

While editing existing data points, you may change any of the data point parameters except the data point name and mode (scheduled, unscheduled or device-scheduled). To open the Edit Collection Parameters of Selected Row(s) dialog box and change the properties of data points, use one of the methods described below:

- Select the data point row in the data point grid, right-click and select Edit Selected Collection Parameters from the menu.
- Double-click in the data point row to which you want to make the change.
- Select the data point row in the data point grid and select Create Edits.
- Select the data point row in the data point grid, and copy or paste data points from Excel.

After you have finished modifying the data point parameters, you must save the changes before you close the FactoryTalk Data Point dialog box.

**Saving Data Point Pending Edits**

When you are finished adding new data points or changing existing data points, you must save the pending edits. Select the data point rows in the data point grid and click **Save Edits** or select the data point rows in the grid, right-click and select Save Selected Edit(s).
from the menu. You must always save pending edits before you assemble.

**Assembling Data Point Pending Edits**

To make the pending edits effective in the running configuration, you must assemble them. On the FactoryTalk Data Point dialog box, select the data point rows in the grid and click **Assemble Edits** or select the data point rows in the grid, right-click and select **Assemble Edits** from the menu.

It is not critical to assemble pending edits at a specific time in the online edits process. If you have already created data point pending edits, you can close the FactoryTalk Data Point dialog box without assembling and proceed to make changes to transactions on the Pending Transaction Definition dialog box. However, data point pending edits do not display on the Pending Transaction Definition dialog box until they have been assembled. It is a good idea to save and assemble data point pending edits before creating transaction pending edits. Assembling new data points does not affect a running configuration because the new data points are not used in a running transaction yet.

You can also click **Assemble Edits** on the main FactoryTalk Transaction Manager user interface to assemble data point pending edits.

**Canceling Data Point Pending Edits**

To cancel pending edits, select the data point rows in the grid and click **Cancel Edits** or select the data point rows in the grid, right-click and select **Cancel Edits** from the menu. You can cancel pending edits after you have saved. You cannot cancel pending edits after you have assembled.

**Editing Transactions In a Running Configuration That Uses Online Edits**

When you open the Transaction Definition dialog box to edit an existing transaction, notice that the fields are all disabled. You must first click **Create Edits** to begin making changes to the transaction. Notice that the title changes to Pending Transaction Definition dialog box. Then you can change transaction bindings and
parameters such as scan rate or timeout, and even enable or disable the transaction. If you want to change the transaction name or the data object that the transaction uses, you must create a new transaction. When you are finished modifying the transaction, click **Save Edits** to save your changes. You must always save pending edits before you assemble. The following figure is an example of the Transaction Definition dialog box.

In the main <FFTM> window you can edit multiple transactions in a running configuration that uses online edits. Select one or more transactions in the Transaction Definition View in the right pane, right-click, and select one of the following menu options.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Transaction or Disable Transaction</td>
<td>A new pending edit row is displayed with the new state. The pending edit is automatically saved, but it must be assembled manually.</td>
</tr>
<tr>
<td>Edit Transaction Parameters</td>
<td>The parameters displayed in the <strong>Trigger and Storage Parameters</strong> dialog box are FactoryTalk Transaction Manager default values, not the values of the selected transactions. Once again, the pending edit is automatically saved, but it must be assembled manually.</td>
</tr>
</tbody>
</table>
Saving Transaction Pending Edits

When you finish changing the transaction, you need to save the pending edits. You must save pending edits before assembling. Saving pending edits does not affect the running configuration, it only saves the pending definition.

Assembling Transaction Pending Edits

To make the pending edits effective in the configuration, you must assemble them. In the **Pending Transaction Definition** dialog box, click **Assemble Edits**. You can also click **Assemble Edits** in the main FactoryTalk Transaction Manager window to assemble transaction pending edits.

**NOTE**

There may be a delay between the time you assemble the pending definition and the time the pending definition actually becomes effective, regardless of when the FactoryTalk Transaction Manager user interface indicates the changes have been made.

The rules governing when the actual transaction is changed are complex, because they take into consideration the management of the currently running transactions. Keep the following in mind:

- If a transaction is not currently being executed, the software will apply pending edits immediately.
- If a transaction is currently being executing, the software will wait until the transaction has been completely executed or it’s timeout has been reached before applying pending edits. No additional transaction instances will be executed until the pending edits have been applied.

The Transaction Control Manager log file contains the entry displaying the time the transaction in question has been assembled. To view the log file in the main FactoryTalk Transaction Manager window, select the Transaction Control Manager in the Configuration tree, and then click **Log Files** on the toolbar.
Viewing Transaction Differences

In the **Transaction Differences** dialog box, you can see the differences between the current definition and the pending definition for a specific transaction. You need to save pending edits before you view the transaction differences. To view the differences, click **Show Differences** in the Pending Transaction Definition dialog box.

Transaction properties are displayed at the top of the dialog box. Bindings (including the address location of the data points, not just the data point names) are displayed at the bottom of the dialog box. The default option is All, but you may select Differences to display only the properties or bindings that are different between the current and pending definitions. You can view the differences between the current and pending transaction definitions any time after you save, but before you cancel or assemble pending edits.

You can also view transaction differences in the Transaction Definition View in the main FactoryTalk Transaction Manager window. To view the differences, right-click a transaction and select **Show Transaction Differences**.
Canceling Transaction Pending Edits

You cannot cancel pending edits after they have been assembled. Canceling pending edits does not affect the running configuration; it only removes the pending definition. If you stop a running configuration with pending edits, you cannot make any further changes to the configuration until you cancel or assemble the pending edits.

To cancel pending edits after saving them in the Pending Transaction Definition dialog box, click Cancel Edits.
Chapter 9

Advanced Topics

In this chapter you will learn about the following:

- Remote User Interface (page 103)
- Distributed Configurations (page 105)
- Increasing Performance (page 110)

Remote User Interface

A remote FactoryTalk Transaction Manager user interface is used to configure FactoryTalk Transaction Manager services and configuration (.dat) files to run on another computer. For example, the FactoryTalk Transaction Manager user interface runs on Computer A, but the FactoryTalk Transaction Manager services and FactoryTalk Transaction Manager configuration files are located on Computer B. No additional licensing is required to perform this function remotely.

**NOTE** Although you can use the Demo or Trial versions of FactoryTalk Transaction Manager to configure a remote user interface, you cannot run a configuration using any of those versions.

The remote FactoryTalk Transaction Manager user interface communicates directly with the Configuration Server, which then reads from and writes to the configuration files.
Remote browsing allows you to browse DSNs and Oracle connection strings that cannot be found on the local computer. All the browsing for FactoryTalk data points and databases is performed in the context of Computer B.

### Configuring the Remote User Interface

To configure the remote user interface:

1. While running the FactoryTalk Transaction Manager user interface on the remote computer (the computer not running the Configuration Server), right-click the computer name in the Configuration tree and select Register Configuration Server.

   The Register Configuration Server dialog box appears.

2. Type the host computer name and then click Register.

   **NOTE**

   If you are not using the FactoryTalk Live Data connector, we recommend that you do not attempt to use a remote FactoryTalk Transaction Manager user interface for configuring your data points.

   When using the remote FactoryTalk Transaction Manager user interface, you need to ensure that all of the computers involved in your configuration belong to the same FactoryTalk Directory.

   To ensure the proper functioning of the FactoryTalk Security, the local FactoryTalk Transaction Manager user interface and remote FactoryTalk Transaction Manager user interface need to share the same Configuration Server in the FactoryTalk Directory.
A distributed configuration exists when FactoryTalk Transaction Manager services are used on different computers. The advantage of using a distributed configuration is that the processing of large amounts of data can be distributed across multiple computers. This is very useful when a single computer processor is not able to handle the increased amount of data, or you want to use multiple control connectors or enterprise connectors of the same type.

You can create a distributed configuration in step 2 of the Configuration Checklist (page 31), by choosing different computers to run different FactoryTalk Transaction Manager services.

A FactoryTalk Transaction Manager Professional license is required to distribute control and enterprise connector services among multiple computers. You need to run the FactoryTalk Server Activation software on the network computer(s) that will act as the activation server(s), and run the FactoryTalk Client Activation software on each of the client computers. Then you will direct the client computers to the activation server computer.

**NOTE** The license must not be installed on a mapped drive, otherwise the Transaction Control Manager service or FactoryTalk Transaction Manager service will not be able to use it.

The FactoryTalk Transaction Manager user interface must be logged into a Microsoft Windows account that has administrative privileges for all the computers being part of the FactoryTalk Transaction Manager system. This requirement gives the FactoryTalk Transaction Manager user interface access to the Microsoft Windows Service Control Manager, in order to start and stop the FactoryTalk Transaction Manager services.

To create a distributed configuration:

1. In Step 1 of the Configuration Checklist (page 31), define the FactoryTalk Transaction Manager configuration files. The
path that you use must be on the same machine as the Configuration Server.

2. Select the control and enterprise connector services.

3. In Step 2 of the Configuration Checklist, define the host computer on which each service will run. You should have FactoryTalk Transaction Manager installed on each of the host computers before you begin.

### Using UNC Paths

When defining distributed configurations, you should use the Universal Naming Convention (UNC) for the error log (.log) and transaction cache (.rsl) file paths. Follow the format below:

```
\servername\sharename\path
```

For example:

```
\Computer 1\c$\rssql_config
```

**NOTE:** It is important that the account you are currently logged into (and the one that FactoryTalk Transaction Manager services will run as) has read and write privileges to the shared folder.

### Changing the Transaction Cache File Path

To change the transaction cache file:

1. Select the configuration name in the tree, and then on the Configuration menu, click Properties.

   The Configuration Properties dialog box appears.

2. On the Cache tab, double-click the connector.

   The Enterprise Connector Options dialog box appears.

3. Under Cached Transaction Files, change the file path in the Path box.

   **NOTE:** The transaction cache file must be located on the same computer as the FactoryTalk Transaction Manager service.
To change the error log file:

1. Select the configuration name in the tree, and then on the Configuration menu, click Properties.
   
The Configuration Properties dialog box appears.

2. Click the Error Log tab.

3. Under File Messages, change the file path in the Path box.

   **NOTE**
   
   To help reduce the network traffic, you can store the error log files on the remote FactoryTalk Transaction Manager computer. Unfortunately, you may not be able to view those log files from the local FactoryTalk Transaction Manager computer; in such a case, view the files directly on the remote FactoryTalk Transaction Manager computer.

---

**Data Point Buffering**

Simple logging applications include many locations for buffering data. This means that there is a risk of discrepancies between the values in your controller and the values in your database. The sections that follow describe different areas in which data point buffering can be used.

**Buffering In the Controller**

Values may change between scans, or, more likely, the data server may read a series of related values while the controller is updating them. In such a case, the values will not be synchronized. This problem can be solved by either blocking the data in a single message sent by the controller or by making sure that the controller does not trigger the transaction until all the values are set. You should also use a transaction result to alert the controller that the values have been read and can now be changed.

**Buffering In the FactoryTalk Transaction Manager Control Connector**

The FactoryTalk Transaction Manager control connector maintains a copy of the data. Data servers, which may have separate copies of the data, send changed values to the control connector. Once in the control connector, the data is either sent to the FactoryTalk Transaction Manager service (unscheduled data points) or is buffered until it is requested (scheduled data points).
Unscheduled data points sent to the FactoryTalk Transaction Manager service are used in currently running transactions, as transaction triggers, or used to update the current value table of the FactoryTalk Transaction Manager service. If the data points are not immediately needed, they can be overwritten before the data is used.

Scheduled data points are only used when a transaction is executed. Scheduled data points are stored in the control connector until the FactoryTalk Transaction Manager service requests them. If the transaction is executed slower than the data is changing in the controller, the data may be overwritten in the control connector’s buffer. In some circumstances, lost data is acceptable (for example, when the temperature is recorded every five minutes, but fluctuates every few seconds). The control connector is aware of every change, while the FactoryTalk Transaction Manager service is only aware of the current value every five minutes.

The Transaction Control Manager service has a local data point cache that contains the current values for every data point and the time the value was collected.

If the Data Valid option in the FactoryTalk Data Point Definition dialog box is set to zero, data is not requested from the FactoryTalk Live Data server, because the server automatically provides updated data values when they change. Hence, the values for the Transaction Control Manager services are always correct.

Transactions buffer their own values, and only request values from the master value table of the Transaction Control Manager service when the Data Valid timeout occurs. The exceptions to this rule is device scheduled data points, which are read from the controller by the FactoryTalk Live Data server once it receives the read request from the Transaction Control Manager service. Another exception is that the FactoryTalk Transaction Manager service never requests the current value of an unscheduled data point from the FactoryTalk Live Data server.

Buffering In the Transaction Control Manager Service
The FactoryTalk Transaction Manager service has a local data point cache that contains the current values for every data point and the time the value was collected.

If the **Data Valid** option in the **FactoryTalk Data Point Definition** dialog box is set to zero, data is requested from the control connector every time it is needed. Using this data, the FactoryTalk Transaction Manager service knows which data points are still fresh (i.e., still in the valid timeout range), and which data points need to be requested from the control connector. Note that the data is not read from the controller at this time, but from the control connector's buffers. The exception to this rule is device scheduled data points, which are read from the controller by the data server once it receives the read request from the control connector. Another exception is that the FactoryTalk Transaction Manager service never requests the current value of an unscheduled data point from the control connector.

Individual transaction buffers are maintained for each transaction that is running. If two copies of the same transaction are running at the same time, the data values from the second transaction do not overwrite the data values from the first one. Only unscheduled transactions can execute more than one copy of the same transaction at the same time.

Complete transactions that are not configured for real-time storage are buffered in cached transaction (*.rsl) files prior to storage in the database. Therefore, data will not be available from a query until it has been removed from the cache file and written to the database.

It is possible to control the rate that the cache transaction files are applied at. In the **Transaction Definition** dialog box, you can set the number of complete transactions to a smaller value, or decrease the time between cache transaction files. This will improve the timeliness of the data in your database, however, the load on your database will increase.
Increasing Performance

The sections that follow discuss recommendations for increasing performance when running FactoryTalk Transaction Manager configurations.

Control System

For control system, adhere to the following recommendations:

- Store data in the control system in consecutive locations. This allows the control data server to read and write the entire block of data one time, instead of reading and writing data several times for each transaction.

- If you must use DDE, select AdvanceDDE instead of DDE, because AdvanceDDE is faster than CF_Text.

- Enable the data server optimization of the control system for reading and writing data.

- Use event-driven communication (unsolicited messages) instead of fast polling. For example, if the control data does not change often, set the control system to send data only when it changes, instead of continuously polling.

- If you must poll control data, use an appropriate poll rate (e.g., do not poll every 10 milliseconds for data that is saved every 10 seconds).

Database

For database, adhere to the following recommendations:

- Use a commercial database (e.g., Microsoft SQL Server) rather than a personal database (such as Microsoft Access). If you decide to use a Microsoft SQL Server, upgrade it to the version 2000 or later.

- Distribute the database to a different computer than the one running FactoryTalk Transaction Manager.

- Use an appropriate data model for your application.
• Configure the database. Take the time to understand when to use indices and how to archive data. If need be, consult your database administrator for assistance.

• Optimize queries, triggers, and stored procedures executed by FactoryTalk Transaction Manager.

**FactoryTalk Transaction Manager**

For FactoryTalk Transaction Manager, adhere to the following recommendations:

• Use unscheduled data change as transaction triggers instead of scheduled transaction execution.

• If you must use Dynamic Data Exchange (DDE), use constant data receipt (hot links) rather than one-off data receipt (cold links) as the main type of communication with the control data server.

• Use the FactoryTalk Device Scheduled collection mode, which improves data accuracy and reduces network traffic. However, note that this collection mode will cause transactions to run slower.

• Distribute both control connectors and enterprise connectors on multiple computers.

• When using real-time transactions, use multiple real-time threads. The disadvantage is that this consumes extra connections to the database. Some databases are licensed based on the number of simultaneous connections.

• Use the Oracle OCI connection instead of the ODBC connection to the Oracle database.

• Use cached transaction files instead of real-time threads. This enhances the performance of commercial databases that allow inserting array values.

• Modify the parameters of cached transaction files (the number of transactions per log file and the time between caching the
files) to optimize the scheduling and volume of transactions issued to your database.

- Disable debug error logging in FactoryTalk Transaction Manager.

- When collecting data from RSLinx Enterprise and FactoryTalk View, some controller values may exist in RSLinx Enterprise and in the FactoryTalk View tag database. Collect these points from RSLinx Enterprise, not from FactoryTalk View. This allows RSLinx Enterprise to optimize the data collection from the controller by reading the data once and passing it to FactoryTalk View and FactoryTalk Transaction Manager.

- When collecting data from RSLinx Classic and RSView32, some controller values may exist in RSLinx Classic and in the RSView32 tag database. Collect these points from RSLinx Classic, not from RSView32. This allows RSLinx Classic to optimize the data collection from the controller by reading the data once and passing it to RSView32 and FactoryTalk Transaction Manager.

- Delete unused data objects and database connections. The enterprise connector automatically tries to connect to these databases even if they are not used in the configuration.

- If a controller register is assigned to more than one data point in a transaction, use the same data point name so that the Transaction Control Manager service or FactoryTalk Transaction Manager service does not have to request the data more than once.

**Hardware and Operating Environment**

For hardware and operating environment, adhere to the following recommendations:

- Use the fastest CPU, the most RAM, and the fastest disk controller as possible, as well as multiple fast disks.
• Run the FactoryTalk Transaction Manager services, especially the Transaction Control Manager service or FactoryTalk Transaction Manager service, on a multi-processor computer. The FactoryTalk Transaction Manager services are multi-threaded and can take advantage of multiple-processors.

• Optimize the Ethernet traffic. Use a local subnetwork based on switched Ethernet.

• Increase the Ethernet data rate to either 10Mbit or 100Mbit.

• Turn off the ODBC trace and SQL trace facilities.

• Place the Microsoft Windows operating system and paging files, the cached transaction files, and the database along with its associated files on separate physical disks.
Chapter 10

FactoryTalk Transaction Manager
Sample Applications

In this chapter you will learn about the following:

- External Trigger Sample Application (page 115)
- Application Contents (page 115)
- Running the Sample Application (page 116)

External Trigger Sample Application

The External Trigger sample application illustrates how an external application can trigger a FactoryTalk Transaction Manager transaction. You can use external triggering to create a custom user interface to the FactoryTalk Transaction Manager application, or to integrate FactoryTalk Transaction Manager functionality into an existing software system.

The Extras directory on the FactoryTalk Transaction Manager product CD contains the External Trigger sample application discussed in this chapter.

Application Contents

The sample application contains the following elements:

- A Microsoft Excel spreadsheet named ExternalTriggerSample.xls. The integer number in cell A1 is the data point that is sent to the database.
- A Microsoft SQL database with a table named ExternalTriggerSample containing three columns:
  - ExternalTriggerDatabaseID (an AutoNumber field)
• ExcelValue
• Timestamp

• A FactoryTalk Transaction Manager configuration that contains a single unidirectional transaction named ExternalTriggerTransaction. The transaction obtains a data value from cell A1 in the Excel spreadsheet, and appends a record named ExternalTriggerSample in the Microsoft SQL database. The transaction is defined so that it can be triggered by an external application.

• Three sample programs (in Microsoft Visual Basic, C Language, and Microsoft Visual C++) that allow you to trigger a transaction by entering a transaction name.

Running the Sample Application

To run the application:

1. Open the ExternalTriggerSample.xls Excel spreadsheet and then enter an integer value in the A1 cell. This is the data value used by the transaction.

2. Create the ExternalTriggerSample table using the provided ExternalApplication.sql script.

3. Create a System DSN (use ODBC data sources on the Microsoft Windows Control Panel) named ExtTrigger that points to the Microsoft SQL table (the ExternalTriggerSample table).

4. Restore the FactoryTalk Transaction Manager configuration contained in the ExternalTrigger.rsq file using the correct host name paths. Start the configuration and wait until all traffic light indicators turn green.

5. Select the programming environment that you will use, and open the corresponding sample program from its appropriate subdirectory (VB_Example.vbp, C_Example.dsw, Visual_CPP_Example.dsw).
6. Run the program and enter the ExternalTriggerTransaction transaction name.

**NOTE** If you are using the Visual_CPP_Example.dsw, click **Trigger** to trigger the transaction.

7. View the records that have been created in the database.

A new row is created each time a transaction is triggered.

You can change the data value in the Excel spreadsheet, or select another option on the sample application screen.

**NOTE** If you are using the Visual_CPP_Example.dsw, you can continue to click **Trigger**.

The sample illustrates the following two methods in the FactoryTalk Transaction Manager Application Program Interface (API) for triggering transactions:

- **RSSqlUnconnectedTrigger()** function: The only parameter is the name of the transaction to be triggered (case-sensitive). The function establishes a connection to the FactoryTalk Transaction Manager service, sends the trigger request, and stops the connection. The function works well when the number of trigger requests is small. The Visual Basic version of this call is **RSSqlUnconnectedTriggerVB()**.

- **RSSqlConnectedTrigger()** function: The only parameter is the name of the transaction to be triggered (case-sensitive). In addition, the function requires that the calling function use the **RSSqlConnect()** and **RSSqlDisconnect()** functions to manage the connection. The function is suitable for applications in which a large number of transactions must be triggered. The Visual Basic version of this call is **RSSqlConnectedTriggerVB()**, and the related calls are **RSSqlConnectVB()** and **RSSqlDisconnectVB()**.

For more information, refer to section “API Calls” in the FactoryTalk Transaction Manager online help.
Chapter 11

FactoryTalk Transaction Manager and Microsoft COM+ Objects

In this chapter you will learn about the following:

- Introducing FactoryTalk Transaction Manager and Microsoft COM+ Objects (page 119)
- Creating the Remote Component (page 120)
- Creating the Client Application (page 121)
- Installing the Remote Component (page 121)
- Setting Up the Remote Client (page 123)
- Creating the Microsoft COM+ Setup Program (page 124)
- Moving the Client Sample Application (page 125)
- Including the COM+ Enterprise Application Connector In a FactoryTalk Transaction Manager Configuration (page 125)
- Defining the COM+ Enterprise Application Connector (page 125)
- Defining the COM+ Data Object (page 126)
- Code Sample A (ComSampleVB) (page 127)
- Code Sample B (ClientSampleVB) (page 128)

Introducing FactoryTalk
Transaction Manager and
Microsoft COM+ Objects

FactoryTalk Transaction Manager logs data to many different databases. Additionally, FactoryTalk Transaction Manager can call stored procedures in a database, as well as call Microsoft COM+ objects used in building multi-tiered, distributed applications. The
examples that follow present the steps required to create and use a simple COM+ object in a FactoryTalk Transaction Manager configuration using Microsoft Visual Basic. Some sections are optional, but they are included to demonstrate the ability to reuse and distribute COM+ objects.

You need to fulfill the following prerequisites before you create the remote component:

- Install Microsoft Windows.
- Verify administrator logon privileges.
- Install Microsoft Visual Basic.
- Verify COM+ activation in FactoryTalk Transaction Manager.

To create the remote component:

1. Open Visual Basic and create a new ActiveX DLL project.
2. Select Project > References to add a reference to the Microsoft ActiveX Data Objects 2.5 and COM+ Services Type libraries.
3. Select Project > Project Properties. Click the General tab and change the project name to “ComSampleVB.” Make sure that the Threading Model is Apartment Threaded.
4. Recreate Code Sample A at the end of this section in the General Declarations section of the default class module. Notice the use of the required ObjectContext object.

   1. Declare an object (in this case, ctxObject).
   2. Set the object using GetObjectContext().
   3. Follow GetObjectContext with your code.
   4. End the code with either a SetComplete or SetAbort method on ctxObject.
5. Save the project and compile it into a .DLL file.

### Creating the Client Application

**NOTE** A client application is used to test a remote component outside of FactoryTalk Transaction Manager. A remote component can be used by more than one client. The process is optional.

**To create the client application:**

1. Create a new Visual Basic Standard EXE project.
2. Select `Project > Project Properties`. Click the General tab and change the project name to `ClientSampleVB`.
3. Add a command button to the standard form.
4. Copy Code Sample B at the end of this section and paste it into the General Declarations section of the form.
5. Save the project.

### Installing the Remote Component

**To install the remote component:**

1. Go to `Start > All Programs > Administrative Tools > Component Services`.
   
The Component Services Console appears.

   **TIP** If you do not see the Administrative Tools on the Start menu, right-click the taskbar and select `Properties`. Click the `Advanced` tab, and then click `Display Administrative Tools`.

2. In the left pane of the `Component Services` dialog box, expand `Component Services`, and then expand `Computers`, `My Computer`, and `COM+ Applications`.
3. Right-click `COM+ Applications` and select `New > Application`.
   
The COM+ Application Wizard appears.
4. Click `Next`.
   
The `Install or Create a New Application` page appears.
5. Click `Create an Empty Application`. 
The Create Empty Application page appears.

6. In the Enter a name for the new application box, type ComSampleVB.

7. Under Activation Type, select Server application, and then click Next.

The Set Application Identity page appears.

8. Under Account, select Interactive User - the current logged on user, and then click Next.

The Add Application Roles page appears.

9. (Optional) Click Add role, and then type the name for it in the Role dialog box that appears.

The new role appears on the Add Application Roles page under Roles.

If you wish to remove a role, select the desired item under Roles, and then click Remove.

10. Click Next.

The Add Users to Roles page appears.

**NOTE** If at least one role has been added, you can add users to the role(s). This step is optional.

11. Under the desired role name, click Users, and then click Add.

The Select Users or Groups dialog box appears.

12. Click Finish to complete the process and close the wizard.

The new ComSampleVB application appears in the tree in the left pane of the Component Services dialog box.

13. Under ComSampleVB, right-click Components, and then select New > Component.

The COM+ Component Install Wizard appears.

14. Click Next.

The Import or install a component page appears.
15. Click **Install new component(s)**.

   The **Select files to install** dialog box appears.

16. Browse to the directory where the .dll file was saved when creating the remote component (page 120).

17. In the **Select files to install** dialog box, select **ComSampleVB.dll**, and then click **Open**.

   The **ComSampleVB.dll** file appears under **Files to install**.

18. Click **Next**, and then click **Finish**.

   **NOTE** You should be able to run the client code “ClientSampleVB” from the Visual Basic environment on the server computer successfully.

### Setting Up the Remote Client

**NOTE** This process is optional.

A Microsoft COM+ application is usually part of a larger application (N-tier) or system that affects many areas of a business. You need to decide how to distribute components across the servers, and what clients will have access to them. We assume that you have a thorough understanding of the target environment (user accounts/groups, server names, etc.). For descriptions of general concepts and detailed information about administering COM and COM+ applications, refer to the online help for the Component Services application.

You can install a Microsoft COM+ application using the setup program provided by the application vendor or the in-house developer, or manually installing and configuring the COM+ application using the Component Services application.

For more information about installing a COM+ application, refer to Installation Tasks in the online help for the Component Services application.
The procedures that follow explain how to export the remote COM+ component, move the client project “ClientSampleVB.vbp” into Visual Basic, and then run the client application in the Visual Basic environment.

Creating the Microsoft COM+ Setup Program

NOTE  This process is optional.

To create the COM+ setup program:

1. Go to Start > All Programs > Administrative Tools > Component Services.
   The Component Services Console appears.

2. In the left pane of the Component Services dialog box, expand Component Services, and then expand Computers, My Computer, COM+ Applications, and finally the ComSampleVB application folder.

3. Right-click ComSampleVB and select Export.
   The COM+ Application Export Wizard appears.

4. Click Next.
   The Application Export Information page appears.

5. Type or browse for the full path and filename where you will save the Microsoft Install (MSI) file.

6. Under Export as, select Application proxy – Install on other machines to enable access to this machine.

7. Click Finish to close the wizard.

8. Go to the directory to which you exported the application.

9. Navigate to the MSI and MSI.CAB files created as the result of the export.

10. Move the files to the remote (client) computer and then run the MSI file.
Do not run the program on the server. The program correctly registers the remote components on the client computer.

**NOTE** This process is optional.

**Moving the Client Sample Application**

**To move the Client Sample Program:**

1. With the Microsoft Windows Explorer, navigate to the directory in which the client application is stored.
   
   The **Form1.frm** and **ClientSampleVB.vbp** files appear in the directory.

2. Move the files to the remote (client) computer.

3. Open the .VBP file up in the Visual Basic environment.

   **NOTE** You should now be able to run the ClientSampleVB application on the remote client computer, and execute the COM+ Server component “ComSampleVB” on the server.

**Including the COM+ Enterprise Application Connector In a FactoryTalk Transaction Manager Configuration**

**To include the Microsoft COM+ enterprise application connector in a FactoryTalk Transaction Manager configuration:**

1. Open FactoryTalk Transaction Manager.

2. Right-click the desired configuration in the tree, and select **Define Configuration**.
   
   The FactoryTalk Transaction Manager Configuration dialog box appears.

3. Under **Enterprise Connector Services**, select the **Microsoft COM+** check box.

4. Click **Apply** to save the parameters, and then click **Close**.

**Defining the COM+ Enterprise Application Connector**

**To define the COM+ enterprise application connector:**

1. Open FactoryTalk Transaction Manager.
2. Right-click the desired configuration in the tree, and select Define Connector.

The Connector Definition dialog box appears.

3. In the Connector Service list, select COM+ Connectors.

4. Type the connector name, host computer name, user name, and password.

5. Click Apply to save the parameters, and then click Close.

### Defining the COM+ Data Object

The remote component must already be installed on the COM+ server before you can define a FactoryTalk Transaction Manager COM+ data object. For more information, see “Installing the remote component (page 121)”. If you are on a client computer, the application proxy must already be installed. For more information, see “Setting up the remote client (page 123)”. 

To define the COM+ Data Object:

1. Open FactoryTalk Transaction Manager.

2. On the Configuration menu, click Checklist.

   The Configuration Checklist dialog box appears.

3. Make sure that all previous steps have been completed (a complete step is displayed with a yellow or green mark next to it).


5. The Compression Data Object Definition dialog box appears.

6. In the Connector list, select the name of the connector.

7. In the Name box, type the name of the data object.

8. If the desired COM+ connection is not displayed under Database, click 

   The Database Connection Definition dialog box appears.
9. Select a connection name from the drop-down list or type it in the **Connection Name** box.

**NOTE** The **User Name** and **Password** fields are disabled because they are not supported in the initial release of Microsoft COM+.

10. Select a COM+ server from the drop-down list. The COM+ applications (or proxies) installed on the computer display in the COM+ Applications area.

11. Select a COM+ interface from the list, click Apply, then click Close to return to the COM+ Data Object Definition dialog box.

12. Select a COM+ Method from the drop-down list. The method parameters that can be included in the specified FactoryTalk Transaction Manager data object display in the Parameters column.

13. Click **Apply** to save the parameters, and then click **Close**.

You are now ready to use Microsoft COM+ components in a FactoryTalk Transaction Manager transaction.

---

**Code Sample A**

*ComSampleVB*

The following code is referenced in the “Creating the remote component” procedure.

```vba
Option Explicit
Public Function Get_VB_Sample_Data( _
  ByVal strKeyID As String, _
  Optional ByRef strData As String, _
  Optional ByRef intData As Integer, _
  Optional ByRef lngData As Long, _
  Optional ByRef sngData As Single, _
  Optional ByRef dblData As Double, _
  Optional ByRef bytData As Byte, _
  Optional ByRef dtData As Date, _
  Optional ByRef bolData As Boolean) _
  As Long
  ' Declare an object variable as ObjectContext
  Dim ctxObject As ObjectContext 'Required
  On Error GoTo errorhandler
  ' Set the object variable using GetObjectContext()
  Set ctxObject = GetObjectContext() 'Required
  ' Put all business code below
  strData = "VB Sample Data"
  intData = 32767
  lngData = 32768
```

---
Code Sample B
(ClientSampleVB)

The following code is referenced in the procedure described in “Moving the client sample application (page 125)”.

Option Explicit
Private Sub Command1_Click()
Dim obj As Object
Dim szKey As String
Dim szString As String
' Create an instance of the remote component
Set obj = CreateObject("ComSampleVB.Class1")
' Put a value into the required argument and
' pass it to the remote component
szKey = "ignored"
' Call the remote component
' and display the return value
MsgBox obj.Get_VB_Sample_Data(szKey, szString)
' Display the returned data
MsgBox "String Data: " & szString
' Destroy the instance of the remote component
Set obj = Nothing
End Sub
Private Sub Form_Load()
Command1.Caption = "Call COM+ VB Sample"
End Sub
Securing FactoryTalk Transaction Manager Using FactoryTalk Security

FactoryTalk Security is intended to improve the security of your automation system by limiting access to the users with a legitimate need. FactoryTalk Security authenticates user identities and authorizes user requests to access a FactoryTalk-enabled system. These security services are fully integrated with the FactoryTalk Directory and are included as part of the FactoryTalk Services Platform that is installed with many Rockwell Software products.

For more information on configuring or overriding security services using FactoryTalk Security, see the FactoryTalk Security online help. By default, the help file is located in the following directory:

C:\Program Files (x86)\Common Files\Rockwell\HelpFTSecurityEN.chm

Please keep the following in mind when configuring FactoryTalk Transaction Manager for use with FactoryTalk Security:

- FactoryTalk Transaction Manager inherits its security settings from Network (also called Distributed) applications and/or the FactoryTalk Network Directory. Any changes that you make via FactoryTalk Security affect FactoryTalk Transaction Manager and all other products that are connected to the same FactoryTalk Directory computer.

If you must change the FactoryTalk Directory computer location, reboot your computer to synchronize the Configuration Server with the FactoryTalk Directory computer.

- The Configuration Server is the only FactoryTalk Transaction Manager-specific component to which you can apply security considerations when using FactoryTalk Transaction Manager with FactoryTalk Security.
permissions via the FactoryTalk Administration Console. One Configuration Server (in FactoryTalk Transaction Manager) is equivalent to one computer (in FactoryTalk). All configurations that are displayed under a single Configuration Server have the same security settings.

**NOTE**
For information on product-specific, FactoryTalk Security-related permissions that are necessary for external components used by FactoryTalk Transaction Manager (e.g., FactoryTalk Live Data), refer to the online help for respective components.

- FactoryTalk Transaction Manager inherits its FactoryTalk Security settings from the computer that hosts the Configuration Server. The settings must be configured using the FactoryTalk Administration Console.

**NOTE**
The computer that hosts the Configuration Server may be configured to inherit from a higher level in the FactoryTalk Directory (e.g., Application or Area). For more information, refer to the FactoryTalk online help.

- In addition to the standard Read and Write permissions, FactoryTalk Transaction Manager also supports one custom action: Start, Stop, and Assemble Configurations. For more information, see "FactoryTalk Security Permissions To Perform FactoryTalk Transaction Manager Tasks (page 136)", "Writing Product-Specific Security Privileges From a Previous Release To a File (page 137)".

- All FactoryTalk Transaction Manager-specific privilege information from previous releases will be ignored in existing configurations, and it will not be converted directly to FactoryTalk Security attributes. You can view all FactoryTalk Transaction Manager-specific privilege information using the tool described in "Writing Product-Specific Security Privileges From a Previous Release To a File (page 137)", and then you can use that data to establish similar permissions in the FactoryTalk Administration Console.

- FactoryTalk Security is set in the FactoryTalk Directory. Therefore, if you move a configuration from one FactoryTalk
Directory to another FactoryTalk Directory, your FactoryTalk Security permissions will not be maintained.

- If the administrator changes your individual security permissions in the FactoryTalk Administration Console, or if your user permissions are altered, the changes will be reflected in FactoryTalk Transaction Manager without requiring you to log off and log on to FactoryTalk Security. However, if the administrator makes changes in the FactoryTalk Administration Console to a group membership (of which you are a member), you need to log off and then log on again to FactoryTalk Security to enforce the permission changes.

- FactoryTalk Transaction Manager includes the Security menu that allows you to log on and log off from FactoryTalk Security from within FactoryTalk Transaction Manager, as well as check the permissions of the current user (by clicking Permissions on the Security menu).

- If you want to share configuration tasks across multiple computers, the FactoryTalk Directory servers on all of the computers must match. For example, if the FactoryTalk Directory server and the FactoryTalk Transaction Manager Configuration Server are on your computer (computer A), and you want to communicate with another computer (computer B) that is configured to use another FactoryTalk Directory server, data can be shared between computer A and computer B only if the FactoryTalk Directory servers are the same.

- All security permissions must be assigned in the FactoryTalk Administration Console.

- The user name and password associated with the FactoryTalk Security permissions to the FactoryTalk Transaction Manager user interface may be completely independent from the user name and password associated with each connector.
For more information on configuring FactoryTalk Security, see the FactoryTalk Security online help, section "About security". By default, the help file is located in the following directory:

C:\Program Files (x86)\Common Files\Rockwell\HelpFTSecurityEN.chm

**Using FactoryTalk Transaction Manager With Single Sign-on (SSO)**

The single sign-on (SSO) is a FactoryTalk Security policy setting that allows you to log on to the first product that you run in the FactoryTalk system, and then to be automatically logged on (without being prompted) to each subsequent Rockwell Software product that you run, using the same user account and password.

**NOTE**

If you want to use FactoryTalk Transaction Manager and FactoryTalk Security effectively, you must have a FactoryTalk user account in the FactoryTalk Directory.

When you log on to FactoryTalk Transaction Manager, all ties between FactoryTalk Transaction Manager and the machine-wide SSO are cut. It means that any changes made to the machine-wide SSO user do not affect the user that is already logged on to FactoryTalk Transaction Manager.

When you log off from FactoryTalk Transaction Manager, the machine-wide SSO user is not affected either.

When you log on to FactoryTalk Transaction Manager, the SSO behavior varies depending on the user account that you are using:

- **If you have a Windows-linked user account in the FactoryTalk Directory.**
  If you already have a Windows-linked user account in the FactoryTalk Directory, you will be logged on as the SSO user, even if FactoryTalk Transaction Manager does not recognize you as the SSO user at startup.
  If you log off from FactoryTalk Transaction Manager by clicking **Logoff** on the **Security** menu, and then you are
logging on again by clicking **Logon**, you will be logged on to FactoryTalk Transaction Manager automatically.

- **If you have a FactoryTalk user account and are logged on to the FactoryTalk Directory.**

If you are already logged on to the FactoryTalk Directory with your FactoryTalk user account, and you are logging on to FactoryTalk Transaction Manager, you will be logged on as the SSO user, because FactoryTalk Transaction Manager will recognize you as the SSO user at startup.

If you log off from FactoryTalk Transaction Manager by clicking **Logoff** on the **Security** menu, and you are logging on again by clicking **Logon**, you will be logged on to FactoryTalk Transaction Manager automatically.

- **If you do not have a Windows-linked user account.**

If you are not recognized by FactoryTalk Security as the SSO user, and you do not have a Windows-linked user account in the FactoryTalk Directory, you will be prompted to type your FactoryTalk user name and password at the FactoryTalk Transaction Manager startup in the **Log On to FactoryTalk** dialog box.

If you log off from FactoryTalk Transaction Manager by clicking **Logoff** on the **Security** menu, and you are logging on again using the **Logon** option, you will be prompted to type your FactoryTalk user name and password in the **Log On to FactoryTalk** dialog box.

**TIP**

For more information on user account types and the way they interact with FactoryTalk Security, see the FactoryTalk Security online help, section "About user, computer, and group accounts".

By default, the online help is located in the following directory:

*C:\Program Files (x86)\Common Files\Rockwell\HelpFTSecurityEN.chm*
To perform specific tasks in FactoryTalk Transaction Manager, you need to have specific FactoryTalk Security permissions. Use the following table to determine the initial action (indicated by “x”) that you or your administrator must take in the FactoryTalk Administration Console to ensure that you will be able to perform the specified FactoryTalk Transaction Manager tasks.

<table>
<thead>
<tr>
<th>To perform this FactoryTalk Transaction Manager task:</th>
<th>Select the following item on the Explorer pane in the FactoryTalk Administration Console:</th>
<th>Right-click the selected item on the Explorer pane in the FactoryTalk Administration Console, select Security, and then set the following actions in the Security Settings dialog box that is displayed:</th>
</tr>
</thead>
<tbody>
<tr>
<td>View a configuration.</td>
<td>Computer in the Computers group that hosts the FactoryTalk Transaction Manager Configuration Server.</td>
<td>FactoryTalk Transaction Manager &gt; Start, Stop, and Assemble Transactions</td>
</tr>
<tr>
<td></td>
<td>FactoryTalk Area (located under Network &gt; App) that you want to browse for tags. (1)</td>
<td>X</td>
</tr>
<tr>
<td>Change a non-running configuration.</td>
<td>Computer in the Computers group that hosts the FactoryTalk Transaction Manager Configuration Server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>FactoryTalk Area (located under Network &gt; App) that you want to browse for tags. (1)</td>
<td>X</td>
</tr>
<tr>
<td>Change a running configuration (i.e., perform online edits).</td>
<td>Computer in the Computers group that hosts the FactoryTalk Transaction</td>
<td>X</td>
</tr>
</tbody>
</table>
To perform this FactoryTalk Transaction Manager task: | Select the following item on the Explorer pane in the FactoryTalk Administration Console: | Right-click the selected item on the Explorer pane in the FactoryTalk Administration Console, select Security, and then set the following actions in the Security Settings dialog box that is displayed:

<table>
<thead>
<tr>
<th>FactoryTalk Transaction Manager &gt; Start, Stop, and Assemble Transactions</th>
<th>Common &gt; List Children</th>
<th>Common &gt; Read</th>
<th>Common &gt; Write</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager Configuration Server.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer in the Computers group that hosts the FactoryTalk Transaction Manager Configuration Server.</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Start or stop a configuration.</td>
<td>Run a service that is not FactoryTalk Live Data enabled.</td>
<td>Run a service that is FactoryTalk Live Data enabled (Transaction Control Manager or FactoryTalk Live Data connector) for read only access to the controller. (2)</td>
<td>FactoryTalk Area (located under Network &gt; App) that you want to browse for tags.</td>
</tr>
</tbody>
</table>

(1) If you specify permissions at the application level in the FactoryTalk Administration Console, they will be inherited by all of the areas included in the application. You can also set the permissions for each area separately. For more information, see the FactoryTalk Security online help.

(2) The user specified in the control connector must be a Microsoft Windows-linked user. This Microsoft Windows-linked user does not have to be defined via FactoryTalk Security.

Writing Product-Specific Security Privileges From a Previous Release To a File

Prior to FactoryTalk Transaction Manager 9.00.00, the software used its own security mechanism to secure configurations. If you used this product-specific security implementation to secure your configurations in a previous release, you can now use a utility to write your previous FactoryTalk Transaction Manager security settings to a file.
When using the tool, you can specify the file name and location. The file will be formatted as follows:

- Line 1 - the file title.
- Line 2 - the Configuration Server name.
- Line 3 - the configuration name.
- Line 4 and subsequent lines - the security level, the user or group name, and any file notification options.

For example:

```
-------------------------------------------------------------------------------
FactoryTalk Transaction Manager Security Privilege Information
Configuration Server = ussewbob
Configuration = TrackerSecretPrivs
-------------------------------------------------------------------------------
ADMIN Users/Groups
Count = 1
BOB
-------------------------------------------------------------------------------
MODIFY Users/Groups
Count = 0
-------------------------------------------------------------------------------
VIEW Users/Groups
Count = 1
Mail - FactoryTalk Transaction Manager Beta
```

After generating the file, you can review the old product-specific security privileges and use the content to establish new FactoryTalk Security permissions in the FactoryTalk Administration Console.

For more information on writing product-specific security privileges to a file, see "FactoryTalk Security Permissions To Perform FactoryTalk Transaction Manager Tasks (page 136)".
Mapping Old Product-Specific Security Privileges To New FactoryTalk Security Permissions

The following table provides information on the way the old product-specific security privileges map to the new FactoryTalk Security permissions.

<table>
<thead>
<tr>
<th>If you used this old product-specific security privilege:</th>
<th>It has been replaced by this FactoryTalk Security permission or FactoryTalk Transaction Manager custom action:</th>
<th>This security permission allows you to:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSSQL_ADMIN</td>
<td>FactoryTalk Transaction Manager &gt; Start, Stop, and Assemble Configurations (page 136)</td>
<td>Start, stop, or assemble (on-line editing) a running configuration.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This attribute also requires you to have the <strong>Common &gt; Read</strong> and <strong>Common &gt; Write</strong> permissions.</td>
<td></td>
</tr>
<tr>
<td>RSSQL_MODIFY</td>
<td>Common &gt; Write (page 136)</td>
<td>Make offline changes to a configuration.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: This attribute also requires you to have the <strong>Common &gt; Read</strong> permission.</td>
<td></td>
</tr>
<tr>
<td>RSSQL_VIEW</td>
<td><strong>Common &gt; Read</strong> (page 136)</td>
<td>View a configuration.</td>
</tr>
<tr>
<td>No privilege</td>
<td>No permissions needed.</td>
<td>View the following FactoryTalk Transaction Manager options (since security permissions have not been configured):</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Security &gt; Logon</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configuration &gt; Exit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Help Menu (all enabled)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Configuration Tree (empty)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• System View Graphic</td>
</tr>
</tbody>
</table>
Chapter 13

Glossary

Assemble Pending Edits
The process of changing the running, current definition of a transaction or data point in a configuration that uses online edits to the pending definition. In order to assemble pending edits, you need to save them first.

Audit Trail
A record of changes made to a FactoryTalk Transaction Manager service in the FactoryTalk Transaction Manager configuration, including the information by whom and when the changes were made, as compiled by the Configuration Server log file. The changes are displayed in FactoryTalk Diagnostics.

Binding
The relationship between a single data object element (a table column or a stored procedure parameter) and its corresponding data point or expression in a transaction.

Bound Value
The data to be written to or read from a table column or a stored procedure parameter.

Configuration
A FactoryTalk Transaction Manager configuration consists of a set of transactions, as well as the connectors, data points, and data objects required to implement the transactions. All configuration information is stored in configuration files. You can create an unlimited number of configurations, but the FactoryTalk Transaction Manager can run only one configuration at a time.

Configuration Server
The Configuration Server is a service that runs continuously to provide a single interface to the configuration files (with the .dat file extension) which make up the FactoryTalk Transaction Manager configuration. The Configuration Server simplifies access to the configuration files by filtering all changes to the files and by communicating with other FactoryTalk Transaction Manager
services. A collection of all changes that affect a configuration is recorded in an audit trail (via either FactoryTalk Diagnostics or the Configuration Server *.log file).

**Configuration That Uses Online Edits**
A configuration using the Transaction Control Manager service to communicate exclusively with FactoryTalk Live Data servers. The configuration can be changed while it is running.

**Control Connector**
A service that moves data between a data server in the control system and the FactoryTalk Transaction Manager service.

**Control System**
Typically includes a network of controllers and/or HMI servers that collect data from machines in a manufacturing plant and control their operation.

**Data Object**
A subset of columns in a database table, a database view, or a set of stored procedure parameters, along with the database connection information to access the database. A FactoryTalk Transaction Manager transaction acts on a single data object, so all of the necessary database information for a transaction must be contained in a single data object. Data objects are defined in enterprise connectors.

**Data Point**
Data locations in the control system. Data points are associated with control connectors and also contain collection parameters and other attributes. Data points can serve as transaction triggers, supply input data for transactions, and receive data as an output from a transaction.

**Database View**
A filter on selected fields in database table(s) outside the FactoryTalk Transaction Manager software.

**DDE Control Connector**
A service that moves data between the FactoryTalk Transaction Manager service and a DDE or AdvanceDDE server.
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSN</td>
<td>An acronym for Data Source Name (i.e., the name of the database being used). A system DSN is available to all users and Microsoft Windows services, while a user DSN is available only to the user who configured it. The ODBC enterprise database connector requires a system DSN to connect to an ODBC data source.</td>
</tr>
<tr>
<td>Enterprise Connector</td>
<td>A service that moves data between the FactoryTalk Transaction Manager service and database(s) or an enterprise system.</td>
</tr>
<tr>
<td>Expression Editor</td>
<td>A FactoryTalk Transaction Manager utility used to create expressions prior to binding them in a transaction.</td>
</tr>
<tr>
<td>FactoryTalk Metrics</td>
<td>FactoryTalk Security is intended to improve the security of your automation system by limiting access to the users with a legitimate need. FactoryTalk Security authenticates user identities and authorizes user requests to access a FactoryTalk-enabled system. These security services are fully integrated with the FactoryTalk Directory and are included as part of the FactoryTalk Services Platform that is installed with many Rockwell Software products.</td>
</tr>
<tr>
<td>Enterprise Application Connector</td>
<td></td>
</tr>
<tr>
<td>FactoryTalk Security</td>
<td></td>
</tr>
<tr>
<td>FactoryTalk Transaction Manager User Interface</td>
<td>The user interface that you use to create, run, control, and monitor FactoryTalk Transaction Manager configurations.</td>
</tr>
<tr>
<td>Generic OPC Control Connector</td>
<td>A service that moves data between the FactoryTalk Transaction Manager service and an OPC-compliant server.</td>
</tr>
<tr>
<td>Local FactoryTalk Transaction Manager Computer</td>
<td>A computer that is running the FactoryTalk Transaction Manager user interface.</td>
</tr>
<tr>
<td>Microsoft COM+ Enterprise Application Connector</td>
<td>A standard for designing distributed n-tier application systems. Microsoft COM+ builds on standard COM and incorporates new versions of tools, such as Microsoft Transaction Server (MTS) and Microsoft Message Queues (MSMQ).</td>
</tr>
</tbody>
</table>
**Microsoft OLE DB Enterprise Database Connector**  
A service that moves data between the FactoryTalk Transaction Manager service and Microsoft SQL Server.

**OCI**  
An acronym for Oracle Call Interface. OCI is an Application Programming Interface (API) used for developing software that can interface natively to Oracle databases.

**ODBC**  
An acronym for Open Database Connectivity. ODBC is a widely accepted API for database access that is based on the Call-Level Interface (CLI) specifications from X/Open and ISO/IEC APIs, and uses Structured Query Language (SQL) as its database access language.

**ODBC Enterprise Database Connector**  
A service that moves data between the FactoryTalk Transaction Manager service and an ODBC-compliant database. The ODBC enterprise database connector is currently written to the ODBC version 2.0 specification, and should support any ODBC driver that is version 2.0 compliant or greater.

**OLE DB**  
A Component Object Model (COM)–based database architecture that provides universal data integration over an enterprise network (from mainframe to desktop), regardless of the data type.

**Oracle OCI Enterprise Database Connector**  
A service that uses OCI to move data between the FactoryTalk Transaction Manager service and an Oracle SQL*NET–compliant database.

**Pending Edits**  
Changes made to data points or transactions in a configuration that uses online edits. Pending edits must be saved before they can be assembled. Pending edits do not affect the running configuration until they have been assembled.

**Pending Edit Alerts**  
Informational messages describing that transactions with state information will lose the state information when the associated data point or transaction is assembled.
| **Remote FactoryTalk Transaction Manager Computer** | A computer that is used to configure FactoryTalk Transaction Manager services and configuration (.dat) files to run on another computer (local FactoryTalk Transaction Manager computer). The computer communicates directly with the Configuration Server, which then writes to the configuration files. The FactoryTalk Transaction Manager user interface does not run on the computer. |
| **RSLinx Classic OPC Control Connector** | A service that moves data between the FactoryTalk Transaction Manager service and an RSLinx Classic server. |
| **RSView32 Control Connector** | A service that moves data from an RSView32 project to the FactoryTalk Transaction Manager server in RSView32. |
| **SQL** | An acronym for Structured Query Language. SQL is an ANSI/ISO standard language for querying, updating, inserting, deleting, controlling access to, and defining storage containers for data. |
| **Table** | In relational database terms, a unit of storage containing columns and rows with specific names and data types. |
| **Tag** | A collection of information for a single data point. |
| **Time-series Data Compression Enterprise Database Connector** | A service that compresses data from the control system using a lossless algorithm to conserve space. This connector can be configured and used only by FactoryTalk Historian Classic. |
| **Transaction** | An exchange of data between data points and a data object. Transactions also include triggering information and other attributes that govern its behavior. Transactions can be unidirectional or bidirectional. |
| **Transaction Control Manager Service** | The Transaction Control Manager is a service that controls and executes FactoryTalk Transaction Manager transactions contained in a configuration, but with the additional functionality of the FactoryTalk Live Data control connector embedded in it. In a configuration enabled for editing, the Transaction Control Manager... |
replaces the separate FactoryTalk Transaction Manager and control connector services.

| **FactoryTalk Transaction Manager Service** | A service that controls and executes FactoryTalk Transaction Manager transactions contained in a configuration. |
| **Transaction Result Binding** | Implements data transfer from the control system to the enterprise system (and possibly the other way round, logging a return code to the control system that is bound to a data point, and reports the success or failure of the transaction). The control system can then take the appropriate action based on the success or failure of the transaction. |
| **Transaction Result Code** | A data value assigned to a transaction by FactoryTalk Transaction Manager in order to communicate successful transaction completion. |
| **UNC** | An acronym for Universal Naming Convention. |
| **Unidirectional Transactions** | Unidirectional transactions are transactions that use information from the control system to add records to a database table, or to update the contents of existing records. They do not return data to the control system. |
## Index

### A
- Adding New Data Points 96
- Advanced Topics 105
- Application Contents 117
- Assemble Pending Edits 143
- Assembling Data Point Pending Edits 98
- Assembling Pending Edits 93
- Assembling Transaction Pending Edits 100
- Audit Trail 143

### B
- Bidirectional or Unidirectional Transactions With Transaction Bindings 80
- Bidirectional Transactions 78, 84
- Binding 143
- Bound Value 143
- Buffering In Cached Transaction Files 111
- Buffering In the Controller 109
- Buffering In the FactoryTalk Transaction Manager Control Connector 109
- Buffering In the FactoryTalk Transaction Manager Service 111
- Buffering In the Transaction Control Manager Service 110

### C
- Cached Transactions 82
- Canceling Data Point Pending Edits 98
- Canceling Pending Edits 94
- Canceling Transaction Pending Edits 102
- Changing the Error Log File Path 109
- Changing the Transaction Cache File Path 108
- Code Sample A (ComSampleVB) 129
- Code Sample B (ClientSampleVB) 130
- Configuration 143
- Configuration and Connector Status 28

### D
- Data Object 144
- Data Point 144
- Data Point Buffering 109
- Data Point Range and Advanced Functions 86
- Data Retrieval Timeout 61
- Data Valid 60
- Database 112
- Database Triggers 84
- Database View 144
- DDE 46
- DDE Control Connector 144
- DDE Data Points 66
- Defining Data Objects 67
- Defining Data Points 55

Configuration Checklist 31
Configuration Server 16, 53, 143
Configuration Server Status 29
Configuration That Uses Online Edits 144
Configuration Tree 27
Configurations That Use Online Edits 91
Configuring the Remote User Interface 106
Consecutive Data Point and Data Block Support 59
Considerations When Using FactoryTalk Transaction Manager With FactoryTalk Security 133
Control Connector 144
Control Connectors 15, 46
Control System 112, 144
Creating a Configuration That Uses Online Edits 96
Creating a Distributed Configuration 107
Creating the Client Application 123
Creating the Microsoft COM+ Setup Program 126
Creating the Remote Component 122
Creating Transactions 75
<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining the COM+ Data Object 128</td>
</tr>
<tr>
<td>Defining the COM+ Enterprise Application Connector 127</td>
</tr>
<tr>
<td>Device Scheduled Request the Current Value From the Device 58</td>
</tr>
<tr>
<td>Distributed Configurations 107</td>
</tr>
<tr>
<td>Distributed Installations 21</td>
</tr>
<tr>
<td>Documentation Feedback 156</td>
</tr>
<tr>
<td>DSN 144</td>
</tr>
</tbody>
</table>

**E**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editing Data Points In a Running Configuration That Uses Online Edits 96</td>
</tr>
<tr>
<td>Editing Existing Data Points 97</td>
</tr>
<tr>
<td>Editing Transactions In a Running Configuration That Uses Online Edits 98</td>
</tr>
<tr>
<td>Enterprise Application Connectors 50</td>
</tr>
<tr>
<td>Enterprise Application Objects 70</td>
</tr>
<tr>
<td>Enterprise Connector 145</td>
</tr>
<tr>
<td>Enterprise Connector Error Handling 70</td>
</tr>
<tr>
<td>Enterprise Connector Options 51</td>
</tr>
<tr>
<td>Enterprise Connectors 16</td>
</tr>
<tr>
<td>Enterprise Database Connectors 49</td>
</tr>
<tr>
<td>Enterprise Database Objects 69</td>
</tr>
<tr>
<td>Establishing Microsoft Windows Privileges 107</td>
</tr>
<tr>
<td>Exploring the FactoryTalk Transaction Manager User Interface 23</td>
</tr>
<tr>
<td>Expression Editor 145</td>
</tr>
<tr>
<td>External Trigger Sample Application 117</td>
</tr>
</tbody>
</table>

**F**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryTalk 17</td>
</tr>
<tr>
<td>FactoryTalk Live Data 46</td>
</tr>
<tr>
<td>FactoryTalk Live Data Data Points 57</td>
</tr>
<tr>
<td>FactoryTalk Metrics 51</td>
</tr>
<tr>
<td>FactoryTalk Metrics Data Objects 70</td>
</tr>
<tr>
<td>FactoryTalk Metrics Enterprise Application Connector 145</td>
</tr>
</tbody>
</table>

**G**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generic OPC 48</td>
</tr>
<tr>
<td>Generic OPC Control Connector 145</td>
</tr>
<tr>
<td>Generic OPC Data Points 65</td>
</tr>
<tr>
<td>Getting Started 21</td>
</tr>
<tr>
<td>Glossary 143</td>
</tr>
</tbody>
</table>

**H**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware and Operating Environment 114</td>
</tr>
</tbody>
</table>

**I**

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Including the COM+ Enterprise Application Connector In a FactoryTalk Transaction Manager Configuration 127</td>
</tr>
<tr>
<td>Increasing Performance 112</td>
</tr>
<tr>
<td>Inserting and Updating Data Table Records 72</td>
</tr>
<tr>
<td>Installation Assistance 156</td>
</tr>
<tr>
<td>Installing and Activating FactoryTalk Transaction Manager 21</td>
</tr>
<tr>
<td>Installing the Remote Component 123</td>
</tr>
<tr>
<td>Intended Audience 19</td>
</tr>
<tr>
<td>Introducing Data Objects 67</td>
</tr>
</tbody>
</table>

**Index**

148 Rockwell Automation Publication RSSQL-UM001I-EN-P-June 2014
Introducing Data Points 55
Introducing FactoryTalk Transaction Manager and Microsoft COM+ Objects 121
Introducing FactoryTalk Transaction Manager Services 45
Introducing Online Edits 89
Introducing Transactions 75

L
Licensing Required For a Distributed Configuration 107
Local FactoryTalk Transaction Manager Computer 145
Logical and Mathematical Operators 85

M
Mapping Old Product-Specific Security Privileges To New FactoryTalk Security Permissions 140
Menu Bar 25
Microsoft COM+ 50
Microsoft COM+ Data Objects 70
Microsoft COM+ Enterprise Application Connector 145
Microsoft OLE DB 49
Microsoft OLE DB Enterprise Database Connector 145
Microsoft SQL Server Data Objects 69
Miscellaneous 39
Monitoring Configurations 42
Moving the Client Sample Application 127

N
New Product Satisfaction Return 156

O
Occurrence Conditions of Pending Edit Alerts 94
OCI 146
ODBC 49, 146
ODBC Data Objects 69
ODBC Enterprise Database Connector 146
OLE DB 146
Online Edits Workflow 90
OPC Data Points 64
Oracle Call Interface (OCI) Data Objects 69
Oracle OCI 50
Oracle OCI Enterprise Database Connector 146

P
Parse Function 87
Pending Edit Alerts 94, 146
Pending Edits 146
Preventing Stale and Mismatched Data 63

R
Real-time Transactions 82
Remote FactoryTalk Transaction Manager Computer 146
Remote User Interface 105
Rockwell Automation Support 156
RSLinx Classic OPC 47
RSLinx Classic OPC Control Connector 147
RSLinx Classic OPC Data Points 64
RSView32 48
RSView32 Control Connector 147
RSView32 Data Points 65
Running the Sample Application 118

S
Saving Data Point Pending Edits 97
Index

Saving Transaction Pending Edits 100
Scheduled
  Maintain the Current Subscribed Value 58
Securing FactoryTalk Transaction Manager
Using FactoryTalk Security 133
Selecting a Collection Mode 58
Selecting a Substitution Option 62
Selecting Timeout Properties 60
Setting Up the Remote Client 125
Specifying Quality 63
SQL 147
Starting and Stopping Connectors 41
Starting Configurations 39
Starting FactoryTalk Transaction Manager 23
Status Bar 31
Step 1
  Defining and Naming a New Configuration 34
Step 2
  Defining Connectors 35
Step 3
  Defining Data Points 36
Step 4
  Defining Data Objects 37
Step 5
  Defining Transactions 37
Step 6
  Verifying Transactions 39
Stopping Configurations 41
Stored Procedures 72

Transaction 147
Transaction Completion 82
Transaction Control Manager Service 13, 53, 147
Transaction Result Binding 148
Transaction Result Code 148
Transaction States 30
Transaction Timeout 82
Transaction Types 77
Transactions 17
Transactions With Bound Transaction Results 84

U
UNC 148
Understanding FactoryTalk Transaction Manager Concepts 13
Understanding FactoryTalk Transaction Manager External Files 42
Understanding FactoryTalk Transaction Manager Services 45
Understanding Online Edit Concepts 90
Understanding Online Edits 89
Unidirectional Transactions 77, 148
Unscheduled
  Send Subscribed Value Whenever It Changes 59
Using FactoryTalk Transaction Manager With Single Sign-on (SSO) 136
Using the Expression Editor 85
Using the Service Console 44
Using UNC Paths 108

V
Viewing Configuration Properties 39
Viewing Transaction Differences 101
W
Welcome To FactoryTalk Transaction Manager 11
What Can FactoryTalk Transaction Manager Do For Me? 12

What Is FactoryTalk Transaction Manager? 11
Where Can I Go for Help? 19
Workspace 29
Writing Product-Specific Security Privileges From a Previous Release To a File 139
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/.


www.rockwellautomation.com