



Informatica® PowerExchange CDC Publisher
1.3

User Guide

Informatica PowerExchange CDC Publisher User Guide

1.3

December 2022

© Copyright Informatica LLC 2017, 2022

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties. Required third party notices are included with the product.

The information in this documentation is subject to change without notice. If you find any problems in this documentation, report them to us at infa_documentation@informatica.com.

Informatica products are warranted according to the terms and conditions of the agreements under which they are provided. INFORMATICA PROVIDES THE INFORMATION IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

Publication Date: 2022-12-15

Table of Contents

Preface	6
Informatica Resources.	6
Informatica Network.	6
Informatica Knowledge Base.	7
Informatica Documentation.	7
Informatica Product Availability Matrices.	7
Informatica Velocity.	7
Informatica Marketplace.	7
Informatica Global Customer Support.	7
Chapter 1: PowerExchange CDC Publisher Overview	9
Product Overview.	9
PowerExchange CDC Publisher Data Sources and Targets.	10
PowerExchange CDC Publisher Architecture.	11
CDC Publisher.	12
Alternative Configurations of PowerExchange and PowerExchange CDC Publisher.	13
PowerExchange CDC Publisher Utilities.	17
General Data-Streaming Limitations.	17
Chapter 2: Installing and Upgrading PowerExchange CDC Publisher	19
Installation Overview.	19
Software Requirements.	20
Installing PowerExchange CDC Publisher.	21
Installed Directories.	21
Post-Installation Tasks.	22
Configuring Environment Variables.	23
Controlling Access to PowerExchange CDC Publisher Files.	23
Upgrading PowerExchange CDC Publisher.	24
Uninstalling PowerExchange CDC Publisher.	25
Chapter 3: PowerExchange CDC Publisher Key Concepts	26
Publisher-Generated Messages for Data Delivery.	26
Checkpointing and Guaranteed Delivery.	29
Storing Checkpoints in Kafka.	30
Restart and Recovery.	31
Chapter 4: PowerExchange Change Capture Environment	33
About the PowerExchange Change Capture Environment.	33
Preparing the PowerExchange Change Capture Environment.	34
Configuring Connectivity to PowerExchange.	35

Chapter 5: Target Messaging Systems.....	37
Apache Kafka Targets.	37
Preparing the Apache Kafka Environment for Use with CDC Publisher.	37
Apache Kafka Data-Streaming Considerations.	38
If You Use Confluent Schema Registry on a Kafka Target.	39
If You Use Kerberos Authentication for Client Communication with the Kafka Server.	40
MapR Streams Targets.	40
Preparing the MapR Environment for Use with CDC Publisher.	40
MapR Data-Streaming Considerations.	42
 Chapter 6: Configuring PowerExchange CDC Publisher.....	 43
Configuration Overview.	43
Customizing the PowerExchange CDC Publisher Configuration Files.	44
Common Configuration Properties.	45
Formatter Configuration Properties.	45
Connector Configuration Properties.	51
PowerExchange Extract Configuration Properties.	56
Filtering Extraction Maps and Source Objects.	59
Configuring CDC Publisher Message Logging.	61
Configuring CDC Publisher to Support Multiple Partitions in Individual Topics.	62
Generating Composite Message Keys for Source Tables.	62
Customizing Target Message Output.	64
Configuring the Custom Pattern Formatter.	65
If Informatica Data Engineering Streaming Will Consume Data from a Target.	66
 Chapter 7: Streaming Change Data.....	 68
Before You Start Data Streaming.	68
Starting the PwxCDCPublisher Utility.	68
Operational Considerations.	69
 Chapter 8: Monitoring PowerExchange CDC Publisher.....	 71
Monitoring Overview.	71
Determining the Status of CDC Publisher Processes.	72
Generating Statistics on CDC Publisher Processing.	73
 Chapter 9: Administering PowerExchange CDC Publisher.....	 75
Administration Overview.	75
Reporting the Avro Format Definitions for Source Tables.	76
Handling Changes to Source Tables and Extraction Maps.	76
Adding a Source Table to a Change Data Stream.	78
Encrypting a Password.	79
Shutting Down the CDC Publisher.	80

Restarting a PowerExchange CDC Publisher Change Data Stream.	80
Appendix A: Command Reference for the Command-Line Utilities.	82
Command Reference Overview.	82
PwxCDCPublisher Utility - Command and Parameters.	83
PwxCDCInfo Utility - Command and Parameters.	84
PwxCDCAdmin Utility - Command and Parameters.	86
Appendix B: Avro Schema Formats.	90
Types of Avro Schema Formats.	90
Avro Fields That Define a Source Column.	91
Avro Flat Schema Format.	91
Avro Nested Schema Format.	92
Avro Generic Schema Format.	92
Avro Wrapper Schema Format.	92
Appendix C: Custom Pattern Formats.	94
Custom Pattern Usage Considerations.	94
Syntax of Table-Specific Custom Pattern Files.	94
Section Tags.	95
Column Attributes.	95
Event Tags.	100
Special Attribute Tags.	101
Special Pattern Tags.	102
Formatter Properties and Custom Patterns.	104
Custom JSON Examples.	104
Custom JSON in Avro Format Example.	108
Custom Pattern Syntax Validation.	109
Custom Pattern Data File.	113
Appendix D: Message Reference.	116
Message Reference Overview.	116
CDCPUB_10001 to CDCPUB_16001.	117
CDCPUB_10001 to CDCPUB_16001.	117
CPPSVC_00100 to CPPSVC_00103.	149
LOGCFG_10001 to LOGCFG_10006.	150
PWXLIC_00110 to PWXLIC-00999.	151
PWXRDR-01000 to PWXRDR-01301.	152
Index.	156

Preface

The *PowerExchange CDC Publisher User Guide* describes how to configure, use, and administer the PowerExchange CDC Publisher software for streaming change data to target messaging systems such as Apache Kafka. After the change data is published to the target messaging system, it can be read by consumer applications for a variety of purposes.

PowerExchange CDC Publisher is an add-on option of the Informatica PowerExchange product.

The User Guide covers the following topics:

- General product architecture
- Product limitations
- Installation and upgrades
- PowerExchange CDC considerations
- Target considerations
- Configuration parameters
- Common administrative tasks
- PowerExchange CDC Publisher command-line utilities
- Supported Avro schema formats
- Messages

This guide assumes that users are familiar with the target messaging system, the PowerExchange CDC product, and the PowerExchange CDC data sources from which change data is captured.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.

- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at <https://search.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <http://network.informatica.com> and select the eSupport option.

CHAPTER 1

PowerExchange CDC Publisher Overview

This chapter includes the following topics:

- [Product Overview, 9](#)
- [PowerExchange CDC Publisher Data Sources and Targets, 10](#)
- [PowerExchange CDC Publisher Architecture, 11](#)
- [Alternative Configurations of PowerExchange and PowerExchange CDC Publisher, 13](#)
- [PowerExchange CDC Publisher Utilities, 17](#)
- [General Data-Streaming Limitations, 17](#)

Product Overview

The PowerExchange CDC Publisher is a Java-based tool that streams change data that has been captured from a PowerExchange data source to a target messaging system such as Apache Kafka. This tool is licensed as an option of the PowerExchange CDC product.

Typically, the PowerExchange CDC Publisher runs continuously as a Linux daemon or as a Windows foreground process until you stop it. It acts as a client of both the PowerExchange system and target messaging system. You can run the PowerExchange CDC Publisher on any Linux or Windows system in your environment, including a system that is remote from the data source, target messaging system, and PowerExchange Logger for Linux, UNIX, and Windows.

The PowerExchange CDC Publisher retrieves change data from the PowerExchange Logger log files. The CDC Publisher process creates a child extraction process that connects to the system that contains the PowerExchange Logger log files to read change data.

The PowerExchange Logger logs units of work (UOWs) in commit order. By default, the PowerExchange CDC Publisher maintains the order of DML change operations from the source when streaming change data to the target messaging system. However, if you configure CDC Publisher to stream data to multiple partitions in a single topic, CDC Publisher cannot ensure that change operations are written to the target messaging system in the same order that they were received from the source.

You can run the PowerExchange Logger in continuous mode or batch mode. However, if you configure the PowerExchange Logger to run in batch mode and stop at the "end of log," the CDC Publisher streams the change data in bursts as the batch Logger process makes the changes available in the log files.

When establishing a change data stream, the PowerExchange CDC Publisher performs the following processing:

1. Retrieves a list of extraction map names that match the schema name that you specified.
2. If you defined filtering criteria for source tables or objects, selects the extraction maps that match your filters for use in extraction processing.
3. Begins extracting change data.
4. When the first change for an extraction map is received, generates an Avro schema for the source object.
5. Formats the extracted source change records into messages based on the Avro schemas or custom pattern formats that you define.
6. Connects to the target messaging system as a producer to send the formatted messages to target topics.

After the change data is available in the target messaging system, consumer applications, such as Informatica Data Engineering Streaming, can consume the data for a variety of purposes. The consumer applications must have copies of the Avro schemas that PowerExchange CDC Publisher generated to decode the messages.

The PowerExchange CDC Publisher includes optional utilities for monitoring and administering a CDC Publisher process and for generating legible copies of the Avro schemas for consumer application use. You can run these utilities locally or from a remote system. To control access to the utility script files, you must use file system security. PowerExchange CDC Publisher does not provide security on the utility script files or administrative functions.

PowerExchange CDC Publisher Data Sources and Targets

PowerExchange CDC Publisher can process change data captured from relational and nonrelational PowerExchange change data sources on heterogeneous platforms and stream that data to target messaging systems.

Supported source types:

- On z/OS:
 - Batch VSAM
 - CICS/VSAM
 - Db2 for z/OS
 - IMS
- Note:** PowerExchange CDC Publisher does not process data from PowerExchange Adabas, Datacom, or IDMS sources on z/OS.
- On IBM i (previously called i5/OS):
 - Db2 for i
 - On Linux, UNIX, or Windows (LUW):
 - Db2 for LUW
 - Microsoft SQL Server (Windows only)

- MySQL
- PostgreSQL (Linux and Windows only)
- Oracle

Supported target types:

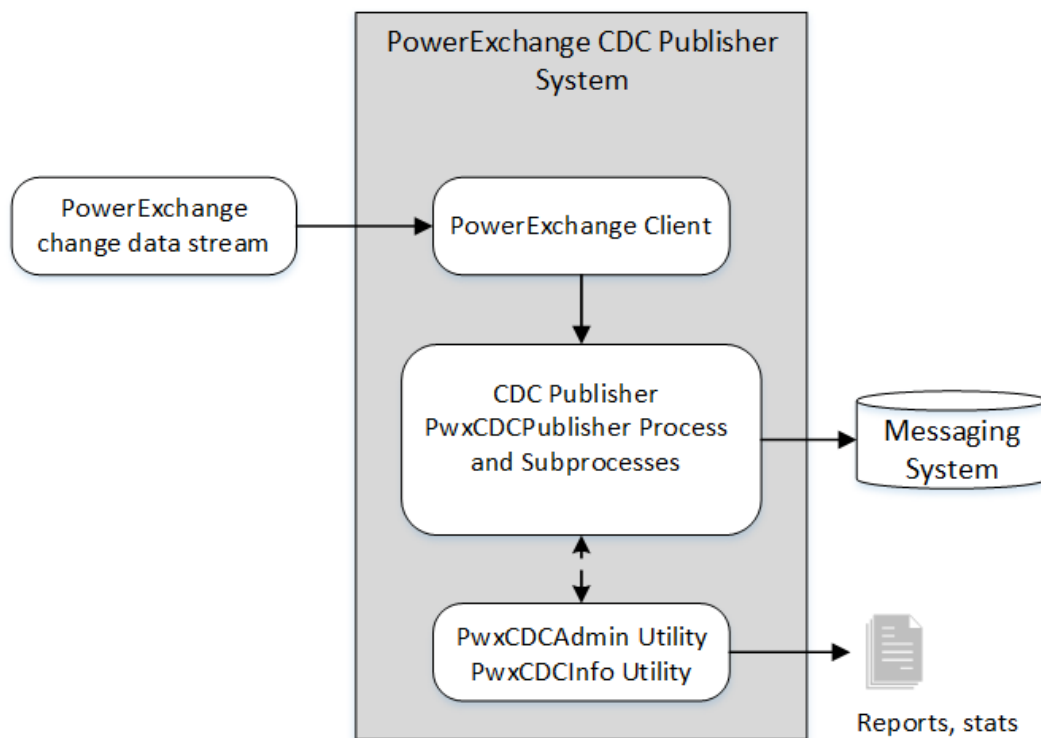
- Apache Kafka on Linux or Windows
- MapR Streams with the Apache Kafka API on Linux or Windows

PowerExchange CDC Publisher Architecture

The PowerExchange CDC Publisher contains a Java-based CDC Publisher component that interacts with local PowerExchange client code to retrieve change data from PowerExchange. The CDC Publisher formats the change data into messages and streams the messages to the target messaging system.

The PowerExchange CDC Publisher also includes utilities for administering a CDC Publisher process and data stream, displaying the status of each process, reporting processing statistics, and generating copies of the Avro schemas for review or for consumers of the data sent to the messaging system.

The following image shows the general architecture of the PowerExchange CDC Publisher:



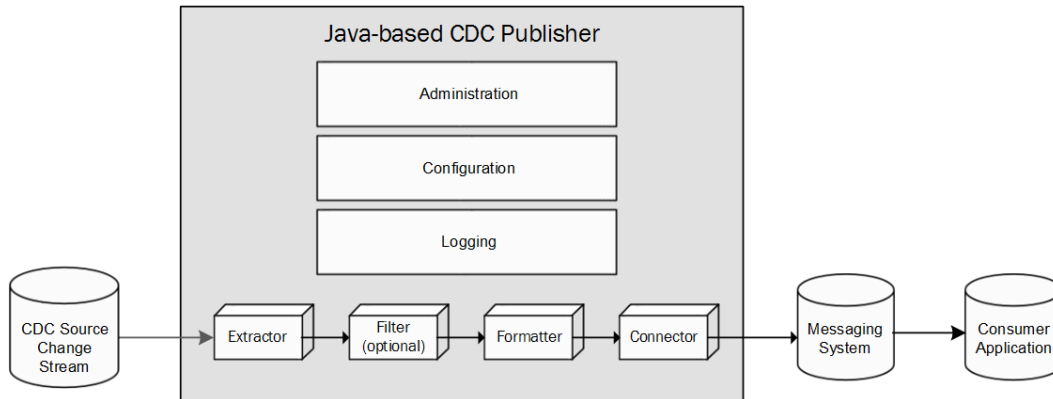
CDC Publisher

The Informatica CDC Publisher is a Java-based tool that is used to stream change data to a target messaging system such as Apache Kafka.

The CDC Publisher contains the following components that move data:

- The CDC Publisher *Extractor* consumes a stream of change data from the source. The incoming data records include schema information, row-based data changes, and transactional boundary metadata. The Extractor performs the following functions:
 - Assigns a sequence ID that is both repeatable and increasing to each change data record.
 - Interacts with the component that supplies the streamed data.
 - Ignores records that are older than the current restart point.
 - Verifies that data is in an expected format.
 - Places the results on an outbound queue for Formatter processing.
- The *Filter* component optionally filters the extracted change data based on lists of source objects to include or exclude that you specify.
- The *Formatter* receives change data from the CDC Publisher Extractor, formats the data based on the generated Avro schema of the selected format (flat, nested, or generic) for inclusion in messages, and sends the formatted messages to the Connector.
- The optional *Custom Pattern Formatter* is an extension of the Formatter. When configured, the Custom Pattern Formatter takes the output from the Formatter and formats the data based on user-defined patterns to customize the resulting Kafka messages.
- The *Connector* reads the formatted messages from the formatter process and connects to the target messaging system to apply the messages. The Connector applies the message data in a consistent, ordered, and recoverable manner.

The following image shows the basic architecture of the Java-based CDC Publisher:



Alternative Configurations of PowerExchange and PowerExchange CDC Publisher

When determining how to implement PowerExchange CDC Publisher in your PowerExchange environment, consider the following guidelines:

- The PowerExchange CDC Publisher is a client of both the PowerExchange source system and the target messaging system. It can run on any Linux or Windows machine in your environment, including a system that is remote from the data source, target, and PowerExchange Logger for Linux, UNIX, and Windows systems.
- The PowerExchange CDC Publisher does not run on platforms other than Linux or Windows.
- PowerExchange must be installed on the source system, PowerExchange Logger system, and PowerExchange CDC Publisher system. The CDC Publisher uses the PowerExchange client code from the local PowerExchange installation to communicate with PowerExchange Listener instances on other machines.
- The PowerExchange CDC Publisher requires the PowerExchange Logger. The PowerExchange CDC Publisher reads change records from PowerExchange Logger log files. Run a PowerExchange Listener instance on the same machine as the PowerExchange Logger. Then configure the PowerExchange client on the CDC Publisher system to connect to the PowerExchange Listener on that machine to extract change records.
- If the PowerExchange Logger is remote from the data source, you must configure remote logging of change data, as described in the *PowerExchange CDC Guide for Linux, UNIX, and Windows*.

The following configurations of PowerExchange and PowerExchange CDC Publisher components are typical:

Configuration 1: All components run on the source system

The data source, PowerExchange Logger for Linux, UNIX, and Windows, PowerExchange Listener, and PowerExchange CDC Publisher run on the same Linux or Windows machine. All of the configuration files reside on this machine, including the PowerExchange dbmover configuration file, PowerExchange Logger pwxcl.cfg configuration file, all CDC Publisher configurations files, and the extraction maps. PowerExchange and the CDC Publisher can share the same dbmover configuration file.

This configuration can be used if the data source is on a Linux or Windows system. This configuration cannot be used if the data source is on an IBM i (i5/OS), UNIX, or z/OS system.

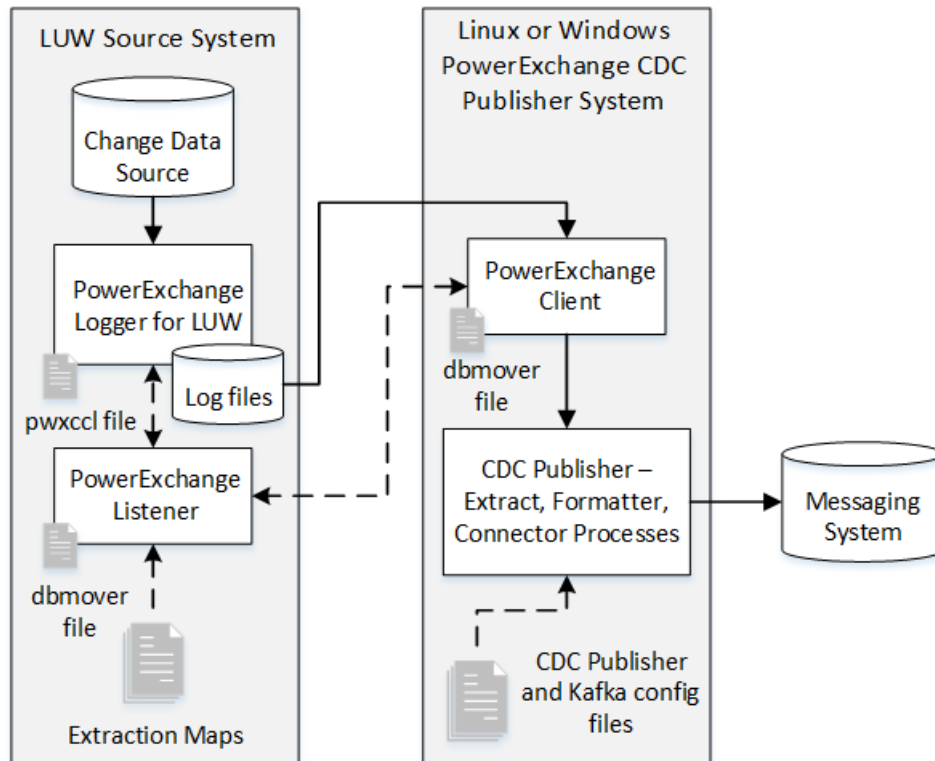
Configuration 2: The PowerExchange Logger runs on the source system and the CDC Publisher runs on a separate system

You can use this configuration if the data source and PowerExchange Logger run on a Linux, UNIX, or Windows system. This configuration is also suitable when you want to add CDC Publisher functionality to an existing PowerExchange environment in which the PowerExchange Logger for Linux, UNIX, and Windows logs change data locally on the source system.

Because the PowerExchange Logger extracts change data from the data source locally, you do not need to configure remote logging. The PowerExchange client on the CDC Publisher system connects to the PowerExchange Listener on the PowerExchange Logger system to retrieve the change data. The source system and CDC Publisher systems must have separate dbmover configuration files. The dbmover configuration file on the CDC Publisher system includes a NODE statement that points to the source host.

This configuration can be used if the data source is on a Linux, UNIX, or Windows system. This configuration cannot be used if the data source is on an IBM i or z/OS system.

The following image shows this configuration:



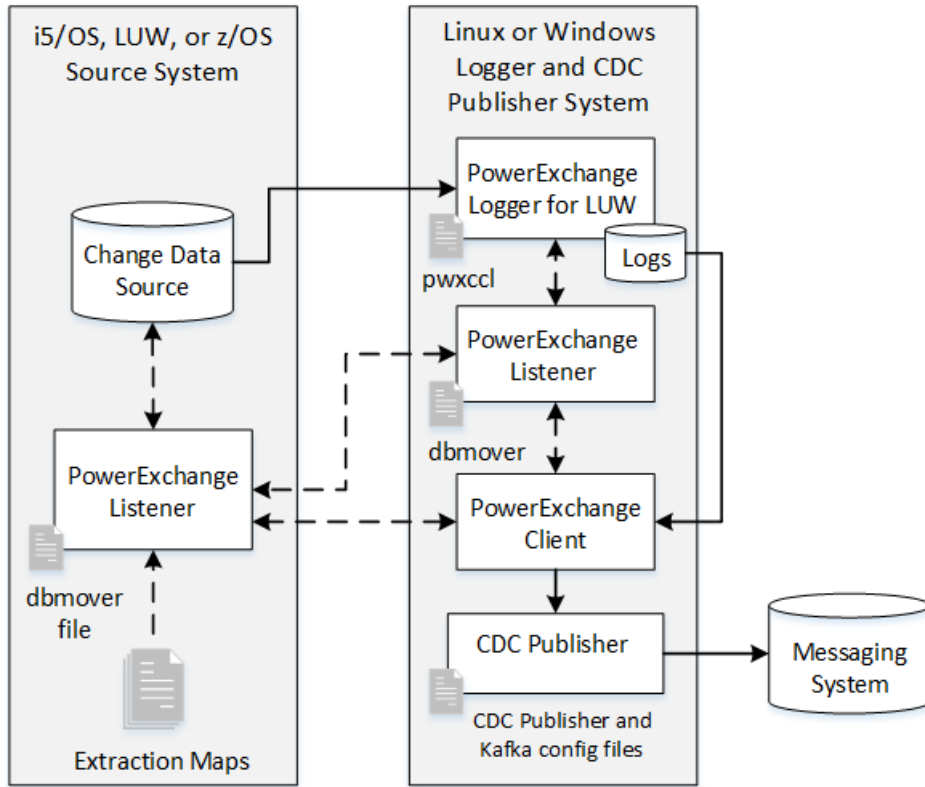
Note: In this image, *change data source* can be Db2 LUW transaction logs, Oracle redo logs, or a PostgreSQL database, MySQL database, or a Microsoft SQL Server distribution database.

Configuration 3: The PowerExchange Logger and CDC Publisher run on a system that is remote from the source

The PowerExchange Logger for Linux, UNIX, and Windows and PowerExchange CDC Publisher components run on the same Linux or Windows system. Because the PowerExchange Logger is remote from the data source, you must configure PowerExchange for remote logging, as described in the *PowerExchange CDC Guide for Linux, UNIX, and Windows*. The source system and the Logger and CDC Publisher system must have separate dbmover configuration files. The dbmover configuration file on the Logger and CDC Publisher system includes a NODE statement that points to the source host.

This configuration accommodates data sources on IBM i, UNIX, or z/OS as well as data sources on Linux and Windows.

The following image shows this configuration:



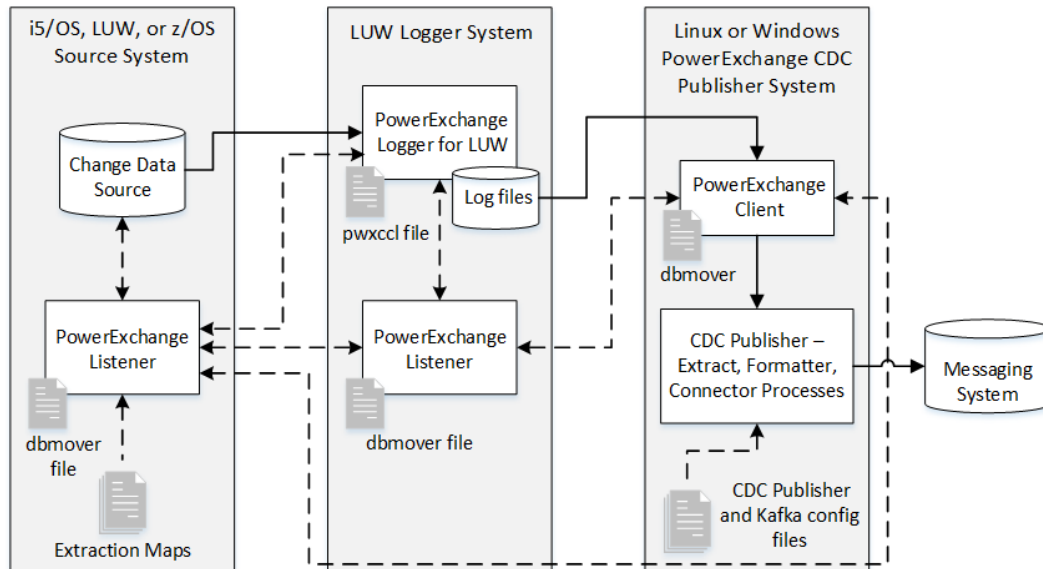
Note: In this image, *change data source* is broadly defined as including PowerExchange Logger for z/OS log files or PowerExchange Condense files on z/OS, journal receivers or PowerExchange Condense files on IBM i (i5/OS), Db2 LUW transaction logs, Oracle redo logs, or a PostgreSQL database, MySQL database, or a Microsoft SQL Server distribution database.

Configuration 4: The data source, PowerExchange Logger, and PowerExchange CDC Publisher run on three separate machines

You can use this configuration to process change data from a source on a platform such as i5/OS or z/OS on which the PowerExchange Logger for Linux, UNIX, and Windows and PowerExchange CDC Publisher cannot run. This configuration is also suitable when you want to add CDC Publisher functionality to an existing PowerExchange environment that uses remote logging of change data to PowerExchange Logger for Linux, UNIX, and Windows log files.

On each system, define a PowerExchange dbmover configuration file. The dbmover configuration file on the CDC Publisher system includes a **NODE** statement that is used to connect to the source system to read extraction maps and another **NODE** statement that is used to connect to the Logger system to retrieve change data from the Logger log files. The CDC Publisher system also contains the PowerExchange CDC Publisher configuration files and the `Kafka producer.properties` file that the CDC Publisher uses to connect to Kafka.

The following image shows this configuration, including the locations of the extraction maps and the PowerExchange, CDC Publisher, and Kafka configuration files:



Note: In this image, *change data source* is broadly defined as including PowerExchange Logger for z/OS log files or PowerExchange Condense files on z/OS, journal receivers or PowerExchange Condense files on IBM i (i5/OS), Db2 LUW transaction logs, Oracle redo logs, or a PostgreSQL database, MySQL database, or a Microsoft SQL Server distribution database.

The general process flow is:

1. PowerExchange captures change data from the local data source based on capture registrations.
2. PowerExchange remotely logs the change data to the PowerExchange Logger log files on another system. The PowerExchange Logger logs changes from successful units of work (UOWs) in commit order.
3. The PowerExchange client on the PowerExchange CDC Publisher system retrieves change data from the PowerExchange Logger log files, maintaining the commit order.
4. The CDC Publisher extracts the change data for source objects of interest, formats the data into messages, and connects to the target messaging system to transmit the messages.

PowerExchange CDC Publisher Utilities

To interact with PowerExchange CDC Publisher, you must use its command-line utilities. The primary utility that initiates and processes the change data stream is the PwxCDCPublisher utility. The other utilities, PwxCDCInfo and PwxCDCAdmin, are for monitoring and administering the CDC Publisher system.

The following table briefly describes each utility:

Utility	Description
PwxCDCPublisher	The main PowerExchange CDC Publisher utility for streaming change data from PowerExchange to the target messaging system. The utility extracts change data from PowerExchange, filters the data if filtering criteria is defined, formats the data into messages, connects to the target, and publishes the messages to the target topics. These processes comprise a streaming <i>path</i> . The utility usually runs as a long-running process or daemon.
PwxCDCAdmin	The PowerExchange CDC Publisher utility for issuing administrative commands to a PwxCDCPublisher process that is running. Commands are available for: <ul style="list-style-type: none">- Shutting down the CDC Publisher.- Clearing Avro schemas from cache so that you can refresh the extraction maps for source tables with added or removed columns, thereby enabling PowerExchange CDC Publisher to regenerate the Avro schemas the next time data is received for the tables.- Re-reading all extraction maps from PowerExchange to get new or changed extraction map definitions for source tables.- Reporting the current Avro schemas in a format that is suitable for viewing or consumer application use.- Validating custom pattern syntax that is used by the Custom Pattern Formatter.
PwxCDCInfo	The PowerExchange CDC Publisher utility for reporting the color-coded status of a PwxCDCPublisher process and each of its main subprocesses: "GREEN" for running, "RED" for stopped, or "YELLOW" for a potential problem. The utility can also report statistics and state information for each topic and for internal attributes that are intended for diagnostic use.

The command syntax for each utility is described in [Appendix A, "Command Reference for the Command-Line Utilities" on page 82](#).

General Data-Streaming Limitations

Review the following items to determine if the PowerExchange CDC Publisher is appropriate for your environment:

- PowerExchange CDC Publisher can perform source table-level filtering. However, CDC Publisher does not perform source column-level filtering or row-level filtering. All source columns or fields are included in every message. For each change record or row received, the CDC Publisher generates a message.
- The PowerExchange CDC Publisher is certified with a single source system and a single target messaging system (a one-to-one topology). The source tables or objects must be all of the same source type and registered under a single source instance. Topologies that have a single source and multiple target messaging systems (a fan-out topology) or that have multiple sources and a single target messaging system (a fan-in topology) are not currently supported. For more information, contact Informatica Global Customer Support or your product specialist.

- The version of PowerExchange that is installed on the PowerExchange CDC Publisher system must match the PowerExchange version that is installed on other systems in the PowerExchange change capture environment.
- By default, the CDC Publisher expects that the PowerExchange Logger for Linux, UNIX, and Windows is configured to extract both before-images and after-images of data. If the PowerExchange Logger is configured to extract only after images, the before images do not appear in the Avro messages. In this case, set the following properties when you configure the CDC Publisher configuration files:
 - Extract.pwxUpdateImageOption=AI
 - Formatter.avroIncludeBeforeImage=false. The setting of false disables the generation of the Avro fields for before images.
- For data sources that were registered in PowerExchange based on a COBOL copybook, the CDC Publisher uses the low-level fields but not the group that contains the fields. To handle this situation, you can use a PowerExchange expression or let the consumer application determine the method to use.
- For Oracle sources, if you specify ORACLE_UNHANDLED_NUMASCHAR=Y in the DBMOVER configuration file that the PowerExchange Navigator uses, PowerExchange converts NUMBER columns that have a precision greater than 28 or an undefined length and FLOAT columns that have a precision greater than 15 significant digits to VARCHAR(172) strings when creating capture registrations and extraction maps. The CDC Publisher does not convert the VARCHAR(172) columns back to numeric values.
- Avro represents all character strings in UTF-8 character encoding.
- If DDL changes are made to source tables, the CDC Publisher ignores the DDL changes and retains the existing message structure until you take action to update the Avro schema. For more information, see [“Handling Changes to Source Tables and Extraction Maps” on page 76](#).

CHAPTER 2

Installing and Upgrading PowerExchange CDC Publisher

This chapter includes the following topics:

- [Installation Overview, 19](#)
- [Software Requirements, 20](#)
- [Installing PowerExchange CDC Publisher, 21](#)
- [Installed Directories, 21](#)
- [Post-Installation Tasks, 22](#)
- [Upgrading PowerExchange CDC Publisher, 24](#)
- [Uninstalling PowerExchange CDC Publisher, 25](#)

Installation Overview

Informatica provides a platform-specific .zip file or .tar.gz file for installing PowerExchange CDC Publisher on a supported Linux or Windows system. To install PowerExchange CDC Publisher, you simply unzip or extract the compressed installation file.

Before you install PowerExchange CDC Publisher, perform the following tasks:

- Get the PowerExchange CDC Publisher installation file for your operating system and a PowerExchange license key that enables the PowerExchange CDC Publisher option.
- Confirm that your system meets the software requirements of PowerExchange CDC Publisher.
- Check that you have sufficient disk space on your Linux or Windows system. Informatica recommends the following minimum amounts:
 - 50 MB free disk space for the initial installation
 - 200 MB for the contents of each PowerExchange CDC Publisher instanceX subdirectory that you configure

After you install PowerExchange CDC Publisher, you must perform some initial system-setup tasks to prepare the PowerExchange CDC Publisher environment, such as configuring environment variables.

Software Requirements

Verify that the system where you plan to install PowerExchange CDC Publisher meets the product operating system and software requirements.

Tip: The Product Availability Matrix (PAM) for this product version provides the latest information on supported versions of operating systems and software. If you are an Informatica Network member, you can access the current PAM at

<https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Supported Operating Systems

PowerExchange CDC Publisher is certified to run on the following operating systems:

- Red Hat Enterprise Linux versions 6.5, 7.4 (minimum certified)
- SUSE Linux 12
- Windows 10
- Windows Server 2012 R2
- Windows Server 2016

Supported PowerExchange Versions

PowerExchange CDC Publisher works with the following PowerExchange CDC versions, including the latest hotfix on a version:

- 10.4.1
- 10.4.0
- 10.5.0
- 10.5.1
- 10.5.2

For supported versions of the PowerExchange data sources, see the *PowerExchange Installation and Upgrade Guide* for your PowerExchange version.

Supported Apache Kafka Target Versions

PowerExchange CDC Publisher works with Apache Kafka versions up to V2.6.0, from any available binary download.

Notes:

- The Scala version is immaterial.
- If you use Confluent Schema Registry in the Kafka target environment, Confluent versions up to 5.5 are supported.

Supported MapR Streams Target Versions

PowerExchange CDC Publisher supports the following MapR Streams versions with the Apache Kafka API:

- 6.1
- 6.0

Other Software Requirements

On the PowerExchange CDC Publisher system, ensure the following software is also installed:

- 64-bit Java JRE 8 (root directory `jre1.8.0_version`) or later

- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (Windows only)

Note: The PwxCDCPublisher script looks for the msvcr120.dll file in the Windows path in the Path environment variable.

Installing PowerExchange CDC Publisher

To install PowerExchange CDC Publisher on a 64-bit Linux or Windows system, you simply extract the compressed installation file that Informatica delivers for the operating system in the pwxcdcpublisher_100_cd.zip file.

1. Transfer the platform-specific installation file that you received from Informatica to a directory on your Windows or Linux system.

Informatica provides the pwxcdcpubv.r.m_win_x64.zip file for the Windows installation and the pwxcdcpubv.r.m_linux_x64.tar.gz file for the Linux installation. Retain the installation file so that you can use it again if you need to reinstall the PowerExchange CDC Publisher.

2. Unzip or extract the installation file to a location of your choice.

The PowerExchange CDC Publisher root installation directory and subdirectories are created. The root installation directory has a unique version-specific name that will not be overwritten by subsequent upgrade installations.

3. Verify that the expected PowerExchange CDC Publisher directories and files are present.

Installed Directories

After you install PowerExchange CDC Publisher, the following directories and subdirectories exist:

```
pwxcdcpub_root_installation_directory
  bin
  instanceA
    checkpoint
    config
    logs
    overflow
    reports
  libs
  samples
```

The following table describes these directories:

Directory	Description
<i>pwxcdcpub_root_installation_directory</i>	The product root installation directory that is created when you unzip or extract the compressed installation file for your operating system. The installation provides a version-specific root installation directory name. You can optionally change it to another unique name.
bin	Contains the script files for running the PowerExchange CDC Publisher utilities: PwxCDCAdmin, PwxCDCInfo, and PwxCDCPublisher.

Directory	Description
instanceA	Contains the checkpoint, config, logs, overflow, and reports subdirectories for the default instance of PowerExchange CDC Publisher. You can create additional instance directories under other names such as instanceB or MyInstance, each with their own checkpoint, config, logs, overflow, and reports subdirectories.
checkpoint	Contains the checkpoint files that record the sequence number of the last change operation that was successfully sent to the target messaging system. These files are used to restart apply processing from the last change operation processed, thereby avoiding missing or duplicate changes on the target.
config	Contains the customized PowerExchange CDC Publisher configuration files that the PowerExchange CDC Publisher utilities use. Also contains the Log4j logging configuration file, PwxCDCPubLog4j.xml file, that configures the logging of PowerExchange CDC Publisher messages. You must copy the configuration files from the samples directory to this config directory to customize and use the files.
logs	Contains the active and archived log files with the PowerExchange CDC Publisher message output. The log files also contain PowerExchange error messages and optionally some PowerExchange informational messages.
overflow	Contains PowerExchange CDC Publisher cache files in temporary storage.
reports	Contains output from PwxCDCAdmin Report command. This command reports the Avro schemas that have been generated for source tables. These schemas can be used by consumer applications.
libs	Contains the PowerExchange CDC Publisher shared libraries that include the .jar files and the .dll or .so files.
samples	Contains the sample PowerExchange CDC Publisher configuration files. Tip: Informatica recommends that you retain the sample files "as is" so that they are available for re-use. Copy the files to the config subdirectory of the instance you intend to configure and then customize the file copies there.

Post-Installation Tasks

You must prepare some initial setup tasks to prepare the system for PowerExchange CDC Publisher.

Perform the following tasks before running Powerexchange CDC Publisher:

- Set the environment variables that PowerExchange CDC Publisher uses.
- Ensure that the PWX_LICENSE environment variable points to the correct license.key file.
- Configure file system security to control access to PowerExchange CDC Publisher files, as needed.

Configuring Environment Variables

After you install PowerExchange CDC Publisher, verify that the required environment variables are correctly set.

The following table summarizes the environment variable settings for a CDC Publisher environment:

Environment Variable	Setting
JAVA_HOME	The path to the base directory for your 64-bit JRE installation on the PowerExchange CDC Publisher system, for example: <code>C:\Program Files\java\jre1.8.0_144</code> .
KAFKA_CLIENT_LIBS	<p>For an Apache Kafka target, enter the path to the local Kafka client libs directory on the PowerExchange CDC Publisher system. This directory contains the Kafka .jar files. For example: <code>C:\kafka_2.11-0.11.0.0\libs</code>.</p> <p>Note: If you use Confluent Schema Registry in the Kafka target environment and want the CDC Publisher messages to be validated against the registered Avro schema, ensure that the Kafka client libs directory also contains the contents from the following Confluent directories: <code>/confluent-v.r.m/share/java/confluent-common</code> and <code>/confluent-v.r.m/share/java/kafka-serde-tools</code>.</p> <p>For a MapR Streams target, enter the path to the MapR lib directory that contains both the MapR client libraries and MapR Kafka client libraries. For example: <code>C:\opt\mapr\lib</code>.</p> <p>If the CDC Publisher runs on the MapR server, this path points to the local MapR lib directory that contains both client libraries. If the CDC Publisher runs on a host that is remote from the MapR server, the local MapR lib directory contains the MapR client libraries but not the MapR Kafka client libraries. In this case, you must copy the <code>kafka-clients-version-mapr-nnnn.jar</code> file from the MapR server to the local MapR lib directory.</p>
MAPR_HOME	The path to the MapR server or to the client installation on the local machine. For example: <code>C:\opt\mapr</code> .
PWX_CONFIG	The path and file name of the PowerExchange DBMOVER configuration file on the system where the PowerExchange CDC Publisher runs.
PWX_LICENSE	The path and file name of the PowerExchange license key file that enables use of the PowerExchange CDC Publisher option for the Kafka or MapR target messaging system. Note: You can change the path and file name of the license key file that is delivered with the PowerExchange CDC Publisher, provided that you specify the new path and file name in this environment variable.
PWXPUB_HOME	The path to the PowerExchange CDC Publisher root installation directory, for example: <code>C:\pwxcdcpub100_win_x64</code> . Define this environment variable if you plan to run the PowerExchange CDC Publisher utility scripts by using batch files instead of running the scripts directly from the <code>pwxcdcpub_root_installation\bin</code> subdirectory. In each batch file, you can reference <code>%PWXPUB_HOME%</code> instead of the full path to the root installation directory.

Controlling Access to PowerExchange CDC Publisher Files

PowerExchange CDC Publisher relies on file system security to control access to files and scripts. You are responsible for configuring this type of security on your operating system.

To protect against unauthorized use of any PowerExchange CDC Publisher script in the `pwxcdcpub_root_installation\bin` subdirectory, configure file-system privileges to control who has access

to these scripts. For example, to provide userA with permission to run the PwxCDCInfo script but *not* run the PwxCDCAdmin or PwxCDCPublisher script, you could use the following strategy:

1. Grant read-only access to all files in the PowerExchange CDC Publisher installation to userA, except the script files in the bin subdirectory.
2. Copy the PwxCDCInfo script to another directory and grant execute permissions on that instance of the script to userA .
3. Set the PWXPUB_HOME environment variable for userA to point to the PowerExchange CDC Publisher root installation directory.

This environment variable setting is required to run PowerExchange CDC Publisher scripts from a location other than the `pwxcdcpub_root_installation\bin` subdirectory.

For more information about setting up file-system security, contact your Security Administrator.

Upgrading PowerExchange CDC Publisher

If PowerExchange CDC Publisher is currently installed, you can use this procedure to upgrade to a new PowerExchange CDC Publisher version.

The upgrade procedure adds the new installation without deleting or overwriting the previous installation. This strategy allows you to copy the information defined for your existing instances to the new installation incrementally, as needed. Also, if you need to fall back to the older version, it is still available until you uninstall it.

Before you upgrade the product, perform the following tasks:

- Read the *PowerExchange CDC Publisher Release Notes* to determine if any upgrade issues exist.
 - Verify that your system is compatible with the software requirements of the new version.
1. Download the PowerExchange CDC Publisher installation file for your operating system to a directory other than the current PowerExchange CDC Publisher installation directory.
Tip: Retain the installation file so that you can use it to upgrade other PowerExchange CDC Publisher installations if necessary.
 2. Shut down the PwxCDCPublisher process if it is running. To do so, run the PwxCDCAdmin command with the SHUTDOWN parameter.
Note: You do not need to stop PowerExchange CDC processing or Apache Kafka.
 3. Unzip or extract the installation file for the new PowerExchange CDC Publisher version. Typically, the file is extracted to a directory other than the existing PowerExchange CDC Publisher installation directory.
The software is extracted into a version-specific root installation directory so that it does not overwrite your previous installation. You can rename the new installation directory, for example, from "pwxcdcpubvrm.win_x64" to "PowerExchangeCDCPublisher_vrm." Alternatively, you can rename the previous installation directory and then use its original name as the name of the new installation so that you do not have to change the PWXPUB_HOME environment variable value, if set.
 4. If you defined the PWXPUB_PUB environment variable, ensure that the environment variable points to the new installation.
 5. If you have a new license key file, ensure that the PWX_LICENSE environment variable points to the new file.

6. Copy the instanceX directories that you use from the previous installation to the new installation. Confirm that all of the subdirectories under each instance are also copied, including the config subdirectory that contains the PowerExchange CDC Publisher configuration files.

Note: You can copy the instanceX directories to the new installation incrementally, as they are needed.

7. If the new CDC Publisher version contains new or changed configuration properties, update the CDC Publisher configuration files in the config subdirectory as needed.

To determine if CDC Publisher properties changed, see the PowerExchange CDC Publisher documentation.

8. To verify that the upgrade succeeded, restart the PowerExchange CDC Publisher process and then run the PwxCDCInfo command with the STATUS parameter.

By default, the Publisher process starts from the last checkpoint position. If you want to start the Publisher process from a different position, issue the PwxCDCPublisher command with the RESTART parameter. For more information, see [“PwxCDCPublisher Utility - Command and Parameters” on page 83](#).

For information about running the PwxCDCInfo command with the STATUS parameter, see [“PwxCDCInfo Utility - Command and Parameters” on page 84](#).

Uninstalling PowerExchange CDC Publisher

To remove PowerExchange CDC Publisher from your system, you can simply delete the top-level installation directory.

1. Shut down the PowerExchange CDC Publisher process by running the PwxCDCAdmin utility with the SHUTDOWN parameter.
2. Delete the PowerExchange CDC Publisher root installation directory.
3. Verify that the PowerExchange CDC Publisher root installation directory and all of its subdirectories and files have been removed.
4. Optionally, remove the environment variables that you added for PowerExchange CDC Publisher.

CHAPTER 3

PowerExchange CDC Publisher Key Concepts

This chapter includes the following topics:

- [Publisher-Generated Messages for Data Delivery, 26](#)
- [Checkpointing and Guaranteed Delivery, 29](#)
- [Storing Checkpoints in Kafka, 30](#)
- [Restart and Recovery, 31](#)

Publisher-Generated Messages for Data Delivery

PowerExchange CDC Publisher creates a message for each change record that it receives for a source table or object and sends the message to the target messaging system.

Note: PowerExchange CDC Publisher does not generate messages for DDL operations or for UOW begin records.

The format of all messages is based on the Avro schemas that the PowerExchange CDC Publisher generates. To generate an Avro schema, the PowerExchange CDC Publisher uses the PowerExchange extraction map for the source object.

Message Content

A message contains extracted change data and metadata. The following table describes the fields that are included in a generated message:

Field	Description
DTL__xxx	The PowerExchange-generated DTL__ columns that have been added to extraction maps by default or by the PowerExchange user. For more information about these columns, see the <i>PowerExchange Navigator User Guide</i> .
INFA_OP_TYPE	The DML operation type (INSERT, UPDATE, or DELETE) that was extracted from the source.
INFA_TABLE_NAME	The source <i>mapname.tablename</i> from the extraction map name. This value identifies the source object for which change data was extracted.

Field	Description
INFA_SEQUENCE	A sequence number that the CDC Publisher assigns to the change record.
<i>source_column_name</i>	The after image of a source column to which a change operation was applied.
<i>source_column_name_Present</i>	An indicator of whether the column contains a value from a change operation.
<i>source_column_name_BeforeImage</i>	The before image value of an updated source column.
<i>source_column_name_BeforeImage_Present</i>	An indicator of whether the before image of an updated source column is present.

For UPDATE operations, the Avro messages include both the before image and after image. You can set the `Formatter.avroIncludeBeforeImage` property to `false` in the `cdcPublisherAvro.cfg` configuration file to not generate Avro fields for before images.

Note: If you specify `CAPTURE_IMAGE=AI` in the PowerExchange Logger `pxxcl.cfg` file to capture after images only, no before image data will be available.

Message Key

PowerExchange CDC Publisher can produce messages that include a key value that consists of a specific character string or the source table name. The target messaging system can use the key value to send messages to a single partition in any target topic. To use this type of key value, specify the `Connector.kafkaMessageKey` property in the `cdcPublisherKafka.cfg` configuration file. For more information, see [“Connector Configuration Properties” on page 51](#).

PowerExchange CDC Publisher can also generate a composite key value based on rules that you define for each source table in a rule-configuration file. You must specify the file in the `Formatter.captureColumnValuesFile=pathFilename` property in the `cdcPublisherAvro.cfg` configuration file. For more information, see [“Formatter Configuration Properties” on page 45](#) and [“Generating Composite Message Keys for Source Tables” on page 62](#).

Note: If you use the Custom Pattern Formatter, the `Formatter.captureColumnValuesFile` property also affects custom patterns. For more information, see [“Formatter Properties and Custom Patterns” on page 104](#).

Avro Message Format and Encoding

PowerExchange CDC Publisher produces messages in Avro format with the encoding type that you select.

You control the Avro encoding by setting the `Formatter.avroEncodingType` configuration property in the `cdcPublisherAvro.cfg` configuration file. You can specify an encoding type of **JSON**, **binary**, or **none**. The CDC Publisher uses the specified encoding type when serializing records in an Avro format. Set the encoding type to **none**, which indicates no explicit encoding, if you use a third-party encoding schema, such as the Confluent Schema Registry encoding schema.

The PowerExchange CDC Publisher generates an Avro schema when it receives change data for a source table or object. The PowerExchange CDC Publisher can generate Avro schemas in one of the following formats, depending on how you set the `Formatter.avroSchemaFormat` property in the `cdcPublisherAvro.cfg` configuration file:

- *Flat*. Lists all Avro fields in one Avro record. A unique Avro schema is generated for each source object.

- *Nested*. Organizes each type of information in a separate Avro record. A unique Avro schema is generated for each source object.
- *Generic*. Generates an Avro schema that accommodates more than one source object. The source column names are included in each record, which allows the generic schema to be independent of any source object. All of the PowerExchange-generated DTL_XXX columns are included as metadata.

Additionally, you can "wrap" a flat, nested, or generic schema with an Avro schema that acts as a header that contains metadata followed by the underlying Avro schema for the source object. To do so, set the `Formatter.avroWrapperSchemaFormat` property to `avroWrapperSchemaFormatV1` in the `cdcPublisherAvro.cfg` file. The wrapper schema format contains the following Avro fields:

- Sequence number of the change record
- Change operation type
- Source `mapname_tablename` from the extraction map name
- The "wrapped" Avro schema of the type specified in the `Formatter.avroSchemaFormat` property

All messages based on the wrapper schema have the same four-field format.

You can use a generic or wrapper schema to represent multiple source tables. The consumer of data in the target topics examines the metadata to determine which source table is represented and discover the source table structure. The consumer can then parse the underlying data and put it into the proper Avro format. Consider using a generic or wrapper schema when you want to send messages with change data from multiple source objects to a single Kafka topic. The topic is identified in the `Connector.kafkaTopic` property.

A flat or nested schema pertains to a specific source table. It defines the columns in the source table. Consider using a flat or nested schema when you want to send change data from a specific source object to the Kafka topic that is generated for that source object. In this case, set the `Connector.kafkaTopic` property to `USE_TABLE_NAME`.

Supported Avro Types

The Avro schemas that the CDC Publisher generates support all Avro primitive types except `FLOAT` or `DOUBLE`, which are represented as strings. Also, if you use Avro logical types for dates, decimal values, times, or timestamps, the CDC Publisher makes a best-effort attempt to process the logical types under the following conditions:

- A `Formatter.avroUseLogicaltype` property is set to `true` in the `cdcPublisherAvro.cfg` configuration file, where `type` is `DateType`, `DecimalType`, `TimeMillisType`, `TimeMicrosType`, `TimestampMillisType`, or `TimestampMicrosType`.
- The source field is defined in the extraction map with a compatible data type, scale, and precision.

Note: The CDC Publisher-generated Avro schemas do not convert a timestamp to a date type.

The Publisher-generated Avro schemas do not support complex types.

Avro Schema Usage Considerations

Consumer applications that read data from the messaging system must have a copy of the Avro schema to decode the message. You can use the `PwxCDCAdmin` utility to generate Avro schemas in a legible format that consumer applications can use. For more information, see ["Reporting the Avro Format Definitions for Source Tables" on page 76](#).

If you change the structure of a source table or object and update the extraction map, or if you change any of the Avro-related properties in the `cdcPublisherAvro.cfg` configuration file, the PowerExchange CDC Publisher does not automatically update any existing Avro schema. You can use the `PwxCDCAdmin` utility to clear the existing Avro schema from cache and then regenerate the Avro schema. For more information, see ["Handling Changes to Source Tables and Extraction Maps" on page 76](#).

Custom Pattern Formats

If the Formatter does not meet your message format requirements, you can use the Custom Pattern Formatter, which is an extension of the Formatter to create user-defined patterns that customize the resulting Kafka messages. The Custom Pattern Formatter uses the definitions you create to format the message output. You can create patterns in any text-based form without restrictions, for example, in comma separated (CSV), JSON, or other text-based format. The custom patterns use tags that are similar to HTML tags. The custom patterns describe how the change data is presented in the message output. You can provide a custom pattern to describe the output for source tables, and you can also specify a default pattern for tables for which a specific pattern is not defined. Custom patterns support date, time, and timestamp conversions using masks, and data type overrides. For more information, see [“Configuring the Custom Pattern Formatter” on page 65](#).

Checkpointing and Guaranteed Delivery

PowerExchange CDC Publisher generates a checkpoint file after it sends the first change operation. As data streaming progresses, the CDC Publisher saves information about the last change operation processed to the checkpoint file. This checkpoint information is used to resume CDC Publisher apply processing after the CDC Publisher is restarted.

The CDC Publisher uses only one checkpoint file for each instance. By default, the file is named "checkpoint" and is created in the "checkpoint" subdirectory of an instance. You can change the file name and directory by specifying the `Connector.kafkaFileCheckpointFileName` and `Connector.kafkaCheckpointFileDirectory` properties in the `cdcPublisherKafka.cfg` configuration file.

The checkpoint file contains checkpoint information only for the last change operation processed. The checkpoint format is specific to the CDC Publisher so do not edit the checkpoint information.

When CDC Publisher writes a checkpoint depends on whether you set the `Connector.kafkaProducerGuaranteeDelivery` property to true or false in the `cdcPublisherKafka.cfg` file. With the default value of true, the CDC Publisher uses *guaranteed delivery* to help avoid message loss and duplication. CDC Publisher writes a checkpoint after each message containing a change operation is successfully delivered to the target messaging system. CDC Publisher overrides some CDC Publisher connector properties to force a single stream of change operations to be delivered one at a time to a single topic. Also, CDC Publisher sets the `retries` parameter in the Kafka Streams configuration to 0, unless the `Kafka.enable.idempotence` parameter is set to true in the `producer.properties` file. When idempotence is enabled, CDC Publisher honors the existing `retries` parameter value, which must be greater than 0 for exactly-once delivery using the idempotence feature. If you set this property to false and the target messaging system terminates while changes are in flight, CDC Publisher does not skip any change operations after the change stream is restarted but might deliver duplicate messages to the target messaging system.

Note: To send change data to multiple partitions in any single topic on the target messaging system, see [“Configuring CDC Publisher to Support Multiple Partitions in Individual Topics” on page 62](#).

The following considerations pertain to using checkpoints and checkpoint files:

- If connectivity to the target messaging system is lost or the CDC Publisher terminates, the CDC Publisher process will restart from the checkpoint position that is recorded in the checkpoint file by default. In this situation, some messages might be duplicated on the target messaging system. To guarantee that messages are not duplicated, ensure that the `Connector.kafkaProducerGuaranteeDelivery` property is set to true.

- If the PowerExchange CDC Publisher process ends abnormally, the checkpoint value in the checkpoint file might not be accurate in the following situations:
 - The existing checkpoint value does not reflect the last change operation because the change stream terminated after the CDC Publisher sent a message to the target messaging system and before the target acknowledged the message as received. In this case, the CDC Publisher still restarts from the existing checkpoint value. Some messages that were previously sent to the target messaging system might be resent.
 - The checkpoint file is corrupted. This situation can occur if an attempt to write a checkpoint value to the checkpoint file failed or did not complete. In this case, delete the checkpoint file. Then configure a restart point by setting the `Extract.restart1` and `Extract.restart2` properties in the `cdcPublisherPowerExchange.cfg` file. When you restart the CDC Publisher, use the `RESTART=FROM_CONFIG` parameter. If you do not configure a specific restart point, the CDC Publisher restarts from the oldest point in the log files, as if `RESTART=FROM_BEGINNING` is specified.

Storing Checkpoints in Kafka

By default, PowerExchange CDC Publisher generates a local checkpoint file after it sends the first change operation. As data streaming progresses, CDC Publisher saves information about the last change operation processed to the checkpoint file.

If you set the connector property `Connector.checkpointsInTarget` to true in the `cdcPublisherKafka.cfg` file, CDC Publisher stores checkpoints in the Kafka headers and periodically in the checkpoint file, but they might be out of synch. In this case, the checkpoint file becomes a backup in case topics that contain checkpoint information are purged or become missing. For more information, see [“Considerations for Storing Checkpoints in Kafka” on page 30](#).

Checkpoint information is written to each Kafka message that CDC Publisher writes. The backup checkpoint file is written at the intervals set in one or both of the following properties in the `cdcPublisherKafka.cfg` configuration file:

- `Connector.checkpointMessageFrequency`
- `Connector.checkpointTimeFrequency`

Note: If neither property is specified with a value greater than 0, no periodic checkpoints are written to the backup checkpoint file.

If you set the `Connector.checkpointsInTarget` property to true, when the CDC Publisher restarts, it reads the topics to find the last checkpoint that was stored and then uses it to restart.

For more information about connector properties, see [“Connector Configuration Properties” on page 51](#).

Considerations for Storing Checkpoints in Kafka

Before using Kafka to store checkpoints, review the following considerations.

- The Kafka version must be 0.11.0.2 or later.
- CDC Publisher assumes that all of the messages in a topic are from the same CDC Publisher instance. If multiple CDC Publisher instances write to the same topics in the same Kafka target, corruption of the target data can occur.
- To shut down the CDC Publisher, you must run the `PWXCDCAAdmin` utility with the `SHUTDOWN` parameter. The `SHUTDOWN` parameter forces CDC Publisher to synchronize the last written Kafka checkpoint with the backup checkpoint file.

If you do not use the PwxCDCAdmin SHUTDOWN command to shut down CDC Publisher, duplicate data might be written to CDC Publisher on startup of CDC Publisher if the Kafka topics are empty, deleted, or corrupted. If no Kafka headers are found on startup of CDC Publisher, CDC Publisher uses a restart point based on the checkpoint information in the backup checkpoint file. Depending on the latency of the checkpoint, more duplicate data might be written on CDC Publisher restart.

- When CDC Publisher starts, it collects information about the existing Kafka topics and topic partitions. The latest Kafka checkpoint information is used as a starting point for PowerExchange CDC capture or as the starting point for CDC Publisher extraction of changes from the Logger log files. If previously written messages are not found because of topic removal or content truncation, checkpoint information might be compromised and CDC Publisher might not restart from the correct location. This situation can result in the incorrect startup of CDC Publisher as follows:
 - If CDC Publisher does not find the topic that contains the latest checkpoint for restart, it uses the last checkpoint from another topic. In this case, the restart point is compromised, which might cause CDC Publisher to send duplicate data to the target.
 - If CDC Publisher does not find topics that contain checkpoint data, which indicates that all topics were deleted or truncated, a restart point cannot be determined. Because no restart point exists that indicates which data has not been processed, data might be lost.

If the last checkpoint value written to Kafka is missing because the topic to which data was last written has been deleted or truncated, you can optionally use the backup checkpoint file contents to restart CDC Publisher, as described in ["Restarting a PowerExchange CDC Publisher Change Data Stream" on page 80](#).

- If you need to change the CDC Publisher ID that is specified by the connector.checkpointPublisherId property, first shut down CDC Publisher by using the PwxCDCAdmin SHUTDOWN command and ensure that a backup checkpoint file exists. When you change the CDC Publisher ID, the CDC Publisher restart process uses the backup checkpoint file to determine the restart point because it will not find the new CDC Publisher ID in the Kafka topics.

After you change the ID, restart the CDC Publisher. The restart process then uses the backup checkpoint file to determine the restart point. Thereafter, in new messages that CDC Publisher writes to the target, it uses the new CDC Publisher ID.

Restart and Recovery

You might need to restart a PowerExchange CDC Publisher process under normal conditions or after a CDC Publisher process, PowerExchange component, or the target messaging system terminates unexpectedly.

The following list describes the default restart behavior under these circumstances:

- If you restart the CDC Publisher process under normal conditions or after a PowerExchange Logger or Listener failure, the CDC Publisher process restarts from the last change operation sent, as recorded in the checkpoint file, by default.
- If you restart the CDC Publisher process after it ends abnormally, it restarts from the last change operation sent, as recorded in the checkpoint file, by default. If a change operation was sent but has not been acknowledged as received by the target messaging system, the change record is resent.
- If you restart the CDC Publisher process after the target messaging system ends abnormally, CDC Publisher apply processing restarts based on the last change operation recorded in the checkpoint file by default. If you disabled guaranteed delivery by specifying Connector.kafkaProducerGuaranteeDelivery=false in the cdcPublisherKafka.cfg configuration file, some changes might be re-sent to the target. If guaranteed delivery is enabled, no changes are duplicated or missed.

You can override the default CDC Publisher restart behavior by entering the PwxCDCPublisher command with the RESTART parameter at the command line. You can configure the RESTART parameter to restart the CDC Publisher process from the earliest available change record in the PowerExchange Logger log files (FROM_BEGINNING) or from a specific position in the change stream (FROM_CONFIG), which you specify in the Extract.restart1 and Extract.restart2 properties in the cdcPublisherPowerExchange.cfg file. For more information, see ["Restarting a PowerExchange CDC Publisher Change Data Stream" on page 80](#).

Note: To cold start the CDC Publisher process FROM_BEGINNING, you must delete the checkpoint file. Otherwise, all change operations prior to or equivalent to the checkpointed change operation are *not* extracted and sent to the target.

CHAPTER 4

PowerExchange Change Capture Environment

This chapter includes the following topics:

- [About the PowerExchange Change Capture Environment, 33](#)
- [Preparing the PowerExchange Change Capture Environment, 34](#)
- [Configuring Connectivity to PowerExchange, 35](#)

About the PowerExchange Change Capture Environment

Before you can run PowerExchange CDC Publisher processes, you must perform some prerequisite tasks in PowerExchange and configure connectivity to PowerExchange. This section summarizes these tasks.

For information about configuring PowerExchange to capture change data from data sources, see the Powerexchange documentation set for your PowerExchange version.

Preparing the PowerExchange Change Capture Environment

You must perform some PowerExchange configuration tasks for the PowerExchange CDC Publisher, even if you are already using PowerExchange to capture change data from data sources.

Perform some or all of the following configuration tasks:

Task	Reference
<p>On the CDC Publisher machine, install a PowerExchange version that the CDC Publisher supports, unless you will run the CDC Publisher on an existing PowerExchange Navigator or PowerExchange Logger system.</p>	<p>Chapter "Installing and Upgrading PowerExchange on Linux and UNIX" or "Installing and Upgrading PowerExchange on Windows" in the <i>PowerExchange Installation and Upgrade Guide</i></p>
<p>Ensure that a local copy of the PowerExchange license key file that includes the PowerExchange CDC Publisher option for the target messaging system exists. This PWX_LICENSE environment variable should point to this file.</p>	<p>-</p>
<p>In the PowerExchange Navigator, if capture registrations and extraction maps for the source tables or objects do not yet exist, create a capture registration for each source table or object. In the new registrations, set the Status option to Active and the Condense option to Part. The PowerExchange Navigator generates an extraction map for each capture registration that you create.</p>	<p>Chapters "Registration Groups and Capture Registrations" and "Extraction Groups and Extraction Maps" in the <i>PowerExchange Navigator User Guide</i></p>
<p>On the PowerExchange CDC Publisher system, configure a dbmover configuration file that includes the following statements:</p> <ul style="list-style-type: none"> - A NODE statement that points to the PowerExchange Listener where the PowerExchange Logger for Linux, UNIX, and Windows runs. This node name must match the Extract.pwxNodeLocation property in the cdcPublisherPowerExchange.cfg file. If you use remote logging of change data to PowerExchange Logger log files, the node name must also match the CAPTURE_NODE value in the PowerExchange Logger configuration file, pwxcl.cfg. - If the extraction maps and PowerExchange Logger are not on the same system, define another NODE statement that points to the system that contains the extraction maps. This node name must match the Extract.pwxXmapLocation property value in the cdcPublisherPowerExchange.cfg file. - If the PowerExchange Listener or PowerExchange Logger is not on the same system as the CDC Publisher, optionally include a set of SVCNODE and CMDNODE statements for the Listener and for the Logger to be able to issue commands to these PowerExchange components from the local CDC Publisher system. 	<p>Chapter "DBMOVER Configuration" in the <i>PowerExchange Reference Manual</i> and the source-specific chapter in the PowerExchange CDC guide for your source platform.</p>

Task	Reference
<p>On the PowerExchange Logger machine, configure a PowerExchange Logger configuration file, pwxcl.cfg, if one does not already exist.</p> <p>Also, in the dbmover configuration file on the PowerExchange Logger system, define a CAPX CAPI_CONNECTION statement for continuous extraction of change data from PowerExchange Logger log files.</p>	<p>Chapter "PowerExchange Logger for Linux, UNIX, and Windows" in the <i>PowerExchange CDC Guide for Linux, UNIX, and Windows</i></p>
<p>If the PowerExchange data source is on IBM i or z/OS, configure remote logging of source data to PowerExchange Logger log files, if it is not already configured. If the data source is on a Linux, UNIX, or Windows machine, remote logging is optional.</p>	<p>Chapter "Remote Logging of Data" in the <i>PowerExchange CDC Guide for Linux, UNIX, and Windows</i>. Ignore steps related to the PowerCenter Integration Service and connection attributes.</p>

Configuring Connectivity to PowerExchange

The PowerExchange CDC Publisher must be able to connect to the system where the PowerExchange Logger for Linux, UNIX, and Windows log files reside and to the system with the source extraction maps. To configure connectivity to PowerExchange, you must specify certain PowerExchange dbmover statements and CDC Publisher properties.

DBMOVER Configuration Statements

If the CDC Publisher and PowerExchange Logger are on the same system, they can share the same local PowerExchange dbmover configuration file.

If the CDC Publisher is remote from the PowerExchange Logger or source system, you must create a dbmover configuration file on the CDC Publisher system. In this dbmover configuration file, configure one or both of the following connectivity statements:

- If the CDC Publisher system is remote from the PowerExchange Logger system, define a NODE statement that points to the PowerExchange Logger host system.
- If the CDC Publisher is remote from the source system where the extraction maps reside, define a NODE statement that points to the source host with the extraction maps.

Also ensure that the PWX_CONFIG environment variable points to the local dbmover configuration file on the CDC Publisher system.

For more information about the NODE statement, see the *PowerExchange Reference Manual*.

CDC Publisher Configuration Properties

In the cdcPublisherPowerExchange.cfg file on the CDC Publisher system, specify the following properties:

- If the location of the extraction maps is different from the Extract.pwxNodeLocation location from which change data will be extracted, specify the Extract.pwxXmapLocation property. In this property, specify the node name from the NODE statement that points to the location of the extraction maps. If a user ID and password are required to access this node, also specify the Extract.pwxXmapUserId and Extract.pwxXmapPwd properties.
- In the Extract.pwxNodeLocation property, specify the node name from the NODE statement that points to the PowerExchange Listener on the system from which change data will be extracted. If a user ID and

password are required to access this node, also specify the Extract.pwxNodeId property and the Extract.pwxNodePwd or Extract.pwxNodeEPwd property.

- In the Extract.pwxCapiConnectionName property, specify the NAME value from the CAPX CAPI_CONNECTION statement that is in the dbmover configuration file on the PowerExchange Logger system. This statement is used to continuously extract change data from PowerExchange Logger log files.

For more information about these properties, see [“PowerExchange Extract Configuration Properties” on page 56](#).

CHAPTER 5

Target Messaging Systems

This chapter includes the following topics:

- [Apache Kafka Targets, 37](#)
- [MapR Streams Targets, 40](#)

Apache Kafka Targets

Apache Kafka is an open-source distributed data streaming platform that is scalable and fault tolerant and able to process large amounts of data quickly. It can serve as an interim staging area for data that will be consumed by various types of downstream consumer applications.

Producer applications, such as the PowerExchange CDC Publisher, can publish streams of change records as messages to Kafka topics. Kafka makes messages available with low latency to consumer applications.

A topic is a collection of messages to which consumer applications subscribe based on the topic name. A topic is divided into multiple partitions, each containing a set of messages in an unchangeable sequence. The use of partitions enables parallel processing of topic data by splitting the data across multiple Kafka broker servers. Partitions also allow consumer applications to read data from the same topic in parallel.

For information about Kafka use cases, configuration, and implementation, see the Apache Kafka documentation at <https://kafka.apache.org/>.

Preparing the Apache Kafka Environment for Use with CDC Publisher

You must perform a few tasks on the PowerExchange CDC Publisher system and on the Kafka target system to prepare for using Apache Kafka with PowerExchange CDC Publisher.

1. Download the Kafka client libraries to the PowerExchange CDC Publisher system.
The version of the Kafka client libraries must match the version of the Kafka target. The Kafka client libraries do not need to be copied under the PowerExchange CDC Publisher root installation directory.
2. On the PowerExchange CDC Publisher system, define the `KAFKA_CLIENT_LIBS` environment variable.
This environment variable points to the **libs** subdirectory of the local Kafka client installation. The **libs** directory contains Kafka .jar files.
3. Ensure that the CDC Publisher can access the Kafka `producer.properties` file. The file can be on the local CDC Publisher host or a shared device.

This file contains connectivity information that CDC Publisher Connector will use to communicate with the Kafka target.

You can optionally rename the file name. When you configure the PowerExchange CDC Publisher configuration files later, you will set the `Connector.kafkaProducerPropertiesFile` property to point to this `producer.properties` file.

4. On the Kafka target system, if Kafka is not configured to automatically generate Kafka topics, create the Kafka topic or topics to which the PowerExchange CDC Publisher will publish the messages that contain change data. Ensure that the topics are created before you run the PowerExchange CDC Publisher process.

You can create a single topic for all source tables or create a separate topic for each source table. When you configure the PowerExchange CDC Publisher configuration files later, set the `Connector.kafkaTopic` property to specify either the name of single topic to which all messages will be published or the option `USE_TABLE_NAME` to publish messages to a separate topic for each source table.

Note: By default, the `auto.create.topics.enable` parameter in the `Kafka.server.properties` file is set to `true`. With this setting, the Kafka broker automatically generates the topic or topics if they do not exist the first time messages are sent to the topics. You need to manually create the topics only if the `auto.create.topics.enable` parameter has been set to `false`.

For more information about configuring a Apache Kafka system, see the Apache Kafka documentation.

Apache Kafka Data-Streaming Considerations

The following data-streaming considerations and limitations apply to Kafka targets:

- The `Kafka.producer.properties` file controls PowerExchange CDC Publisher connectivity to Kafka.
- The PowerExchange CDC Publisher supports Kafka primitive datatypes except `FLOAT` and `DOUBLE`. Also, the CDC Publisher tries to convert logical types for dates, times, timestamps, and decimal values on a best effort basis if you set the `Formatter.avroUseLogicalType` property to `true` in the `cdcPublisherAvro.cfg` configuration file, where `type` is `Date`, `Decimal`, `TimeMillis`, `TimeMicros`, `TimestampMillis`, `TimestampMicros`. The CDC Publisher does not support complex types.
Note: The `FLOAT` and `DOUBLE` types are not supported because the CDC Publisher converts all source data to characters in Kafka messages. Any subsequent attempt to convert the character data to `FLOAT` or `DOUBLE` would decrease data accuracy and precision.
- To minimize the risk of duplicate or missing messages on the Kafka target after a network outage, ensure that the `Connector.kafkaProducerGuaranteeDelivery` property is set to `true`, the default setting. Guaranteed delivery provides the highest level of data integrity but might degrade performance. You can disable guaranteed delivery by setting the `Connector.kafkaProducerGuaranteeDelivery` property to `false` in the `cdcPublisherKafka.cfg` configuration file.
- If you use Confluent Schema Registry in the Kafka target environment, you can enable CDC Publisher to communicate with Confluent Schema Registry when sending formatted messages. Confluent Schema Registry can intercept and validate the messages against the registered Avro schema. For more information, see [“If You Use Confluent Schema Registry on a Kafka Target” on page 39](#).
- Except for the underscore character (`_`), PowerExchange CDC Publisher removes non-alphanumeric characters from a topic name by default. You can change this behavior, if needed. For more information, see the `Connector.kafkaTopicAllowSpecialCharacters` property description in [“Connector Configuration Properties” on page 51](#).

If You Use Confluent Schema Registry on a Kafka Target

If you use Confluent Schema Registry in a Kafka target environment, when PowerExchange CDC Publisher publishes its formatted messages, it allows Confluent Schema Registry to intercept the messages and validate them against the registered Avro schema.

To enable PowerExchange CDC Publisher communication with Confluent Schema Registry, perform the following tasks:

- On the CDC Publisher system, configure the `KAFKA_CLIENT_LIBS` environment variable to point to a local directory that contains the Kafka client `.jar` file and the contents of the following Confluent serialization libraries:

```
/confluent-v.r.m/share/java/confluent-common  
/confluent-v.r.m/share/java/kafka-serde-tools
```

- Ensure that the Kafka producer.properties file contains the following entries for the Confluent implementation of the key and value serializers:

```
key.serializer=io.confluent.kafka.serializers.KafkaAvroSerializer  
value.serializer=io.confluent.kafka.serializers.KafkaAvroSerializer  
schema.registry.url=http://host:port
```

The `host` and `port` variables represent the host name and port number on which Confluent Schema Registry listens.

- In the CDC Publisher `cdcPublisherKafka.cfg` configuration file, ensure that the `Connector.kafkaProducerPropertiesFile` property points to the customized Kafka producer.properties file that contains the Confluent implementation of the key and value serializers.
- In the CDC Publisher `cdcPublisherAvro.cfg` configuration file, ensure that the `Formatter.avroEncodingType` property is set to **none**. This value allows the Avro records to be passed to Confluent Schema Registry without encoding, as required.

By default, when CDC Publisher sends the first message to a Kafka topic, Confluent Schema Registry checks whether the Avro schema for the message is already in its registry. If the schema is not registered, Confluent Schema Registry automatically adds the schema to its registry.

Alternatively, you can manually add Avro schema to the Confluent Schema Registry before you start publishing messages with CDC Publisher. To get the Avro schema definitions for your source tables, run the `PwxCDCAdmin` utility with the `REPORT=FORMAT` parameter. Then use the resulting schema definitions to register the schema in Confluent Schema Registry. In some situations, Confluent Schema Registry requires the entire schema to be entered as a single string delimited by double-quotation marks. In this case, you must use an escape character before the quotation marks. To do so, perform the following steps:

1. In the `cdcPublisherAvro.cfg` configuration file, set the `Formatter.avroDisplaySchemaWithEscapedQuotes` property to `true`.
2. Run the `PwxCDCAdmin` utility with the `REPORT=FORMAT` parameter to generate the schema definitions. The reported schema definitions then include the `\` escape character before the double-quotation marks, for example, `/"schema_string/"`.

Note: PowerExchange CDC Publisher does not support any schema registries other than Confluent Schema Registry.

If You Use Kerberos Authentication for Client Communication with the Kafka Server

You can enable PowerExchange CDC Publisher, as a producer client, to communicate securely with an Apache Kafka server that uses Kerberos for client authentication.

If you use Kerberos authentication in your Kafka target environment, perform the following tasks to allow CDC Publisher to securely communicate with the Kafka server:

- Edit the PwxCDCPublisher startup script to include the local Java Authentication and Authorization Service (JAAS) login configuration file for the Kerberos environment. In the RUN environment variable, before the reference to SimpleServerForQAdapter, add a line for the `java.security.auth.login.config` property, which points to the JAAS configuration file on the local machine where the CDC Publisher runs. For example:

```
RUN="$RUN -Djava.security.auth.login.config=/opt/kafka_client_jaas.conf"  
RUN="$RUN com.informatica.rat.pwxqadapter.SimpleServerForQAdapter"
```

- Ensure that the Kafka producer.properties file on the CDC Publisher system includes the following entries:

```
security.protocol=SASL_PLAINTEXT  
sasl.mechanism=GSSAPI  
sasl.kerberos.service.name=xxxx
```

The `xxxx` variable represents the value of the following option in the JAAS configuration file:

```
principal="xxxx/..."
```

Also ensure that the `Connector.kafkaProducerPropertiesFile` property in the `cdcPublisherKafka.cfg` configuration file points to this `producer.properties` file.

For more information, see the Apache Kafka security documentation at <http://kafka.apache.org/documentation/#security>.

MapR Streams Targets

MapR Streams is a distributed messaging system for streaming event data at a large scale. MapR Streams targets use the Apache Kafka API to receive data from applications that act as Kafka producers.

PowerExchange CDC Publisher, as a producer application, publishes streams of messages that contain change data to MapR topics in a manner that is similar to how CDC Publisher publishes changes to Apache Kafka. However, the tasks that you perform to prepare the MapR target environment are slightly different.

For example, MapR organizes topics into streams. You must define the stream that contains the topics to which messages will be sent before you run the CDC Publisher process and create or generate topics. Also, you must set some environment variables.

For information about Kafka use cases, configuration, and implementation, see the Apache Kafka documentation at <https://mapr.com/docs/>.

Preparing the MapR Environment for Use with CDC Publisher

On the PowerExchange CDC Publisher system, you must perform a few tasks to configure the MapR target for use with CDC Publisher.

1. Download the MapR client libraries.

The version of the MapR client libraries must match the version of the MapR target.

2. Define the MAPR_HOME environment variable, if it does not already exist.

This environment variable points to the MapR client installation on the local machine. For example:

```
C:/opt/mapr
```

3. Download the MapR kafka-client .jar file to the PowerExchange CDC Publisher system, if the file is not present in the local MapR lib subdirectory.

The MapR client installation does not include the kafka-clients-*version*-mapr-*nnnn*.jar file. If the CDC Publisher runs on a MapR client machine, you must copy the kafka-clients-*version*-mapr-*nnnn*.jar file from the MapR server installation to the local \$MAPR_HOME/lib directory. If the CDC Publisher runs on the MapR server, the MapR lib directory includes the MapR Kafka client libraries, which makes this step unnecessary.

Note: For MapR release 6.0.1, the Kafka client library jar file name is kafka-clients-1.0.1-mapr-1803.jar. For MapR release 6.1.0, the jar file names is kafka-clients-1.1.1-mapr-1808.jar.

4. Define the KAFKA_CLIENT_LIBS environment variable to point to the **lib** subdirectory of the local MapR client installation that contains the MapR client and MapR Kafka client .jar files.

For example:

```
KAFKA_CLIENT_LIBS =$MAPR_HOME/lib
```

5. Ensure that the CDC Publisher can access the Kafka-formatted producer.properties file.

This file can be on the local CDC Publisher host or on a shared device. The file contains information that the CDC Publisher Connector uses with the Kafka API to communicate with the MapR target. The file must contain the following minimum entries:

```
value.serializer=org.apache.kafka.common.serialization.ByteArraySerializer
key.serializer=org.apache.kafka.common.serialization.ByteArraySerializer
```

For more information about options supported in the MapR implementation of the Kafka producer properties file, see

https://maprdocs.mapr.com/52/MapR_Streams/configuration_parameters_for_producers.html.

6. Create or identify the MapR stream that includes the topics to which the CDC Publisher will publish messages.

The CDC Publisher supports publishing messages to topics within a single MapR stream. You must specify the stream name in the Connector.kafkaMaprStreamName property in the CDC Publisher cdcPublisherKafka.cfg file.

7. On the MapR target system, if MapR is not configured to automatically generate topics, create the MapR topic or topics to which the PowerExchange CDC Publisher will publish messages. Complete this step before you run the PowerExchange CDC Publisher process.

You can create a single topic for all source tables or create a separate topic for each source table. When you configure the PowerExchange CDC Publisher configuration files later, set the Connector.kafkaTopic property to specify either the name of the single topic or the option USE_TABLE_NAME to use a separate topic for each source table.

Note: By default, a topic is created automatically when a producer creates the first message for it and the topic does not already exist. If you disabled automatic creation of topics by setting the autcreate parameter to false when creating the stream, you must manually create the topics.

For more information about configuring a MapR system, see the MapR documentation.

MapR Data-Streaming Considerations

If you stream change data to MapR Streams targets that use the Kafka API for producers, the following considerations apply:

- The MapR implementation of the Kafka API for producers is delivered as part of the MapR server installation in a separate .jar file named `kafka-clients-version-mapr-nnnn.jar`, where *nnnn* varies by MapR version. This file is required to access MapR by using the Kafka API definitions. If you run CDC Publisher on a MapR client machine, copy this .jar file from the MapR server to the local lib subdirectory that contains the MapR client libraries.
- The PowerExchange CDC Publisher supports MapR topics that are defined under a single MapR stream name. A CDC Publisher instance can use only one MapR stream name.
- PowerExchange CDC Publisher does not interact with or implement any feature of the MapR-DB Change Data Capture (CDC) system.

CHAPTER 6

Configuring PowerExchange CDC Publisher

This chapter includes the following topics:

- [Configuration Overview, 43](#)
- [Customizing the PowerExchange CDC Publisher Configuration Files, 44](#)
- [Filtering Extraction Maps and Source Objects, 59](#)
- [Configuring CDC Publisher Message Logging, 61](#)
- [Configuring CDC Publisher to Support Multiple Partitions in Individual Topics, 62](#)
- [Generating Composite Message Keys for Source Tables, 62](#)
- [Customizing Target Message Output, 64](#)
- [If Informatica Data Engineering Streaming Will Consume Data from a Target, 66](#)

Configuration Overview

After you install PowerExchange CDC Publisher and prepare the PowerExchange source environment and the target messaging system, configure the parameters in the PowerExchange CDC Publisher configuration files according to your data streaming requirements.

By default, under the PowerExchange CDC Publisher root installation directory, a single instanceA directory exists and contains checkpoint, config, logs, overflow, and reports subdirectories. You can create additional instance subdirectories, each with a unique directory name and separate set of subdirectories and configuration files. Consider using multiple instances if you want to process changes from a large number of source objects in multiple data streams. In this case, configure each instance to use a different set of extraction maps to process changes for different source objects. When you run the PowerExchange CDC Publisher process, specify which instance to use in the INSTANCE parameter.

For each instance that you create, perform the following configuration tasks:

1. Copy all of the PowerExchange CDC Publisher configuration files from the `\pwxcdcpub_root_installation\samples` subdirectory to the `\pwxcdcpub_root_installation\instanceX\config` subdirectory. PowerExchange CDC Publisher uses the configuration files that are in the config subdirectory.
2. Customize the properties in the PowerExchange CDC Publisher configuration files that are in the config subdirectory. See [“Customizing the PowerExchange CDC Publisher Configuration Files” on page 44](#).

3. If you do not want to stream change data based on all of the extraction maps that match the extraction-map schema name that you specify, enter filtering criteria in the `Extract.captureConsumerExcludeNameList` property, `Extract.captureConsumerIncludeNameList` property, or both in the `cdcPublisherPowerExchange.cfg` configuration file. See [“Filtering Extraction Maps and Source Objects” on page 59](#).
4. Customize the Log4j 2 logging of PowerExchange CDC Publisher messages, if necessary. Usually, the sample `PwxCDCPubLog4j.xml` configuration file that PowerExchange CDC Publisher supplies can be used without modification. See [“Configuring CDC Publisher Message Logging” on page 61](#).
5. If the Informatica Data Engineering Streaming product will consume the messages with change data that the PowerExchange CDC Publisher sends to the target messaging system, configure the Formatter parameters for treating fields with a binary datatype as string data and for using binary-encoded messages. See [“If Informatica Data Engineering Streaming Will Consume Data from a Target” on page 66](#).

Customizing the PowerExchange CDC Publisher Configuration Files

PowerExchange CDC Publisher provides sample configuration files in the `pwxcdcpub_root_installation/samples` subdirectory. These files contain the basic CDC Publisher properties that you will need to set. For special circumstances, Informatica technical staff might request that you set additional properties.

The following sample configuration files are provided:

- **cdcPublisherAvro.cfg**. Contains properties that specify the Avro schema format and the data serialization method and encoding for messages that contain change data.
- **cdcPublisherCommon.cfg**. Contains properties that define the PowerExchange CDC Publisher host and port.
- **cdcPublisherKafka.cfg**. Contains properties related to the target messaging systems.
- **cdcPublisherPowerExchange.cfg**. Contains properties related to accessing PowerExchange extraction maps and change data.

You can copy the sample configuration files to the `pwxcdcpub_root_installation/instanceX/config` subdirectory and then customize the copies of the files. The files contain comments that describe each property. The PowerExchange CDC Publisher will read only the configuration files that are in the `config` subdirectory.

For PowerExchange CDC Publisher to recognize a file as a configuration file, the file must have the file name extension `.cfg`. Also, the file content must begin with `# Configuration` on the first or second line in the file.

Note: No sample configuration file is available for the Custom Pattern Formatter. You create the Custom Pattern Formatter configuration files in a location that the CDC Publisher can access. They do not require the `# Configuration` comment.

You can include all configuration properties in one file or create alternative subsets of properties in multiple configuration files. Informatica recommends that you use multiple configuration files that logically group the properties, similar to the sample configuration files. You can then copy and re-use the properties that do not change across multiple instances and migration paths, and customize only the properties that need to change because of changes to your CDC Publisher environment. For example, you might want to customize the PowerExchange-related extraction properties and Kafka target properties when migrating PowerExchange CDC Publisher from a test environment to a production environment but retain the Avro-related properties.

Important: Do not include the same property in multiple configuration files. If the same property appears with different values in multiple configuration files, the PowerExchange CDC Publisher will issue an error.

Common Configuration Properties

The sample `cdcPublisherCommon.cfg` file contains properties that identify the PowerExchange CDC Publisher host and port.

Property Descriptions

Common.pwxCDCPublisherHost

The host name or IP address of the machine where the PowerExchange CDC Publisher server runs. Enter **localhost** if you plan to run the `PwxCDCAdmin` and `PwxCDCInfo` utilities on the same machine as the PowerExchange CDC Publisher server.

Tip: If you plan to run the `PwxCDCAdmin` or `PwxCDCInfo` utilities on a system that is remote from the PowerExchange CDC Publisher server, enter a public IP address or host name.

Default value is **localhost**.

Common.pwxCDCPublisherPort

Required. The port number that the PowerExchange CDC Publisher uses on the specified host to listen for command and control requests.

No default value is provided.

Common.logPwxConfigContents

Indicates whether to write `CDCPUB_10067` messages that report the contents of the PowerExchange `dbmover` configuration file on the CDC Publisher system to the message log. The default value of `true` causes these messages to be written to the log. If any `dbmover` statements contain sensitive information such as passwords, you can set this property to `false` to prevent the messages from being logged.

Default value is **true**.

Common.ReportSuffix

Sets the file name suffix for all report files generated by the `PwxCDCAdmin` utility.

To use another override value for a specific utility run, add the `SUFFIX` option for the `REPORT` parameter in the utility command. For example, you might need to specify an override if a downstream application that consumes the report expects a specific report file extension. For more information, see [“PwxCDCAdmin Utility - Command and Parameters” on page 86](#).

The default suffix is `.rpt`.

Formatter Configuration Properties

The sample `cdcPublisherAvro.cfg` file contains configuration properties that define the format of the generated Avro schema, the encoding type to use for serializing the Avro records to be included in messages, and several optional Formatter settings.

Property Descriptions

The following properties are in the sample `cdcPublisherAvro.cfg` file:

Note: `Formatter.avroSchemaFormat=avroFlatSchemaFormatV1` is the only setting that is supported for custom patterns.

Formatter.formatterType

The type of data serialization formatter to use for messages. The only valid value is **Avro**.

Formatter.avroSchemaFormat

Required. The Avro schema format that the PowerExchange CDC Publisher uses to generate the Avro schema that will determine the structure of the message values. Valid values are:

- **avroFlatSchemaFormatV1**. Structures messages by using a flat Avro schema format, which lists all Avro fields in one Avro record. A unique Avro schema is generated for each source object, which contains the Avro field definitions.
- **avroNestedSchemaFormatV1**. Structures messages by using a nested Avro schema format, which provides a main Avro record that contains a separate nested record for each type of Avro field.
- **avroGenericSchemaFormatV1**. Structures messages in a generic manner that accommodates any source object definition. All source columns are represented by an array. Each array entry contains column data and metadata. The source column names are included in each data record, allowing the generic schema to be independent of the source table.

Note: Do not use this format type with the Custom Pattern Formatter.

No default value is provided.

Note: You can "wrap" a flat, nested, or generic schema by setting the `Formatter.avroWrapperSchemaFormat` property to `avroWrapperSchemaFormatV1`. The schema then consists of four fields for each source object.

Use a generic or wrapper schema to allow a single Avro schema to represent multiple source tables. For more information about the schema formats, see [Appendix B, "Avro Schema Formats" on page 90](#).

Formatter.avroEncodingType

Required. The Avro encoding type that the CDC Publisher Formatter uses to serialize the Avro records to be included in messages. Valid values are:

- **binary**. Use binary encoding to serialize Avro records.
- **json**. Use JSON to serialize Avro records.
- **none**. Do not use any explicit encoding type. Specify this option only if you use Confluent Schema Registry in a Kafka target environment.

No default value is provided.

The following additional properties can also be included in the `cdcPublisherAvro.cfg` file at your discretion:

Formatter.avroBinaryAsString

Controls whether change data with a binary datatype is represented as string data in Avro messages. Set this property to true if the data will be consumed by applications that do not support binary data, such as Informatica Data Engineering Streaming. The default value is false.

Formatter.avroExcludeDTLColumns

Excludes all or selected PowerExchange-generated metadata columns that have the `DTL__` prefix from the messages.

- To exclude all `DTL__` columns from the formatted results, enter only the asterisk (*) wildcard character in the following format: `Formatter.avroExcludeDTLColumns=*`

- To exclude one or more individual DTL__ columns from the formatted results, enter the column names using a comma (,) separator. For example:

```
Formatter.avroExcludeDTLColumns=(DTL__CAPXACTION, DTL__CAPXRESTART1,
DTL__CAPXRESTART2, DTL__CAPXUSER, DTL__CAPXUOW, DTL__CAPXTIMESTAMP,
DTL__CAPXROWID)
```

Formatter.avroIncludeBeforeImage

Controls whether the generated Avro schema and messages include a field for before-image data. Set this property to true to include this field. Set this property to false to not include this field.

If you include the before-image field, the field is populated with data for UPDATE operations, if you set the Extract.pwxUpdateImageOption property to enable the extraction of before-image data from the PowerExchange change stream. For DELETE and INSERT operations, the field is not populated with data.

The default value is true.

Formatter.avroIncludeInfaBigIntSequence

Controls whether the Avro Formatter generates an internal sequence value for each captured change record, which you can use to filter or sort messages in a change data stream. The generated value is in string format but can fit in a big integer column, if needed. For each captured change record, the Avro Formatter adds a column named INFA_BIGINT_SEQUENCE in the output messages to hold the generated sequence string. Valid values are:

- **false**. Do not generate a sequence string and do not add the INFA_BIGINT_SEQUENCE column in the formatted messages.
- **true**. Generate the sequence string and include it in the INFA_BIGINT_SEQUENCE column.

The default value is false.

When Formatter.avroIncludeInfaBigIntSequence is set to true, CDC Publisher generates a sequence string in the following form:

timestamp overflow + timestamp + sequence overflow + sequence

For example: 015859610896870000000000000000000003

The following elements make up the string:

- **Timestamp overflow**. A 1-byte value that indicates whether the timestamp has overflowed its maximum value. The value 1 indicates an overflow occurred, and the value 0 indicates no overflow occurred.
- **Timestamp**. A 13-byte timestamp value in string format. The timestamp is set once and remains consistent as long as the CDC Publisher process exists.
- **Sequence Overflow**. A 1-byte value that indicates that the sequence number has overflowed its maximum value. The value 1 indicates an overflow occurred, and the value 0 indicates no overflow occurred.
- **Sequence**. A 19-byte ascending sequential value for each record that is created. This number starts from 0 when the CDC Publisher process starts.

The combination of the timestamp overflow, timestamp, sequence overflow, and sequence number provides a unique ascending value for each record.

Usage considerations:

- The generated sequence strings are not repeated if you restart the CDC Publisher process. The strings are unique for each CDC Publisher run. You can use the sequence strings at the target to compare and determine the original order in which they were processed at the source database.

- Start time values on restart are sequential and ascending but not consecutive. Within one CDC Publisher run, the sequence number is ascending but might have gaps.
- You cannot use the sequence strings to determine if operations are missing, delivered, or received at the target.

Formatter.avroBinaryStringRepresentationType

If you set the `Formatter.avroBinaryAsString` property to true or use a generic Avro format, indicates whether binary data is represented as a hexadecimal string or base64 string. Valid values are:

- **hexadecimal**
- **base64**

The default value is base64.

Formatter.avroCheckZeroScale

Before writing data to Kafka messages, PowerExchange CDC Publisher converts numeric data types to decimal data types for fields that have a scale value of zero or null, by default. If you want to retain the original data type, as defined in PowerExchange, set this property to true. Valid values are:

- **true**. Do not convert a numeric data type to a decimal data type when a field has a zero or null scale value. Retain the original data type.
- **false**. Convert a numeric data type to a decimal data type when a field has a zero or null scale value.

The default value is false.

Formatter.avroDisplaySchemaWithEscapedQuotes

If you use Confluent Schema Registry in a Kafka target environment and need to manually add an Avro schema to the registry as a single string that is delimited by double-quotation marks, set this parameter to true to use a backslash (/) as the escape character that precedes the double-quotation marks. Then run the `PwxCDCAdmin` utility with the `REPORT=FORMAT` parameter to generate a schema definition that includes the escape character before each delimiter, for example, `/"schema_string/"`. You can then use the generated schema definition to add the schema to Confluent Schema Registry. The default value is false, which disables the use of escaped double-quotation marks in generated schema.

Formatter.avroSchemaPrintDefaultFields

Controls whether Avro schemas include the "default" fields. If you need to reduce the schema size, you can set this property to false to exclude the default fields. The default value is true, which includes the default fields.

Formatter.avroSchemaPrintDocFields

Controls whether Avro schemas include the "doc" fields. The doc fields include metadata such as the CDC and PowerExchange datatypes, precision, and scale. If you need to reduce the schema size, you can set this property to false to exclude the doc fields. The default value is true, which includes the doc fields.

Formatter.avroSchemaPrintPretty

Controls whether Avro schemas include spaces and line feeds to improve legibility. If you need to reduce the schema size, you can set this property to false to exclude the spaces and lines feeds. The default is true, which includes the spaces and line feeds.

Formatter.avroUseOriginalLogicalTimeConvertToUtc

PowerExchange CDC Publisher represents Avro logical date and time values in UTC by default. Previously, CDC Publisher converted Avro logical date and time values to long epoch values by using a

UTC time zone. If you want to revert to the previous behavior, set the `Formatter.avroUseOriginalLogicalTimeConvertToUtc` property to true. Valid values are:

- **true**. Use the original behavior which converts Avro logical date and time values to long epoch values by using a UTC time zone.
- **false**. Recommended. CDC Publisher does not convert values.

The default value is false.

Formatter.avroWrapperSchemaFormat

Enables the use of an Avro "wrapper" schema format. The wrapper schema can be used to describe any source object. The wrapper, or parent, schema consists of four fields for each source object: the sequence number of the change record, source table name, change operation type, and the "wrapped" Avro child schema expressed as a large string. The consumer application can then parse the underlying data and put it in the proper Avro format for the source object. To use a wrapper schema format, set this property to `avroWrapperSchemaFormatV1`. No default value is provided. For more information, see ["Avro Wrapper Schema Format" on page 92](#).

Formatter.avroUseLogicalDateType

Formatter.avroUseLogicalDecimalType

Formatter.avroUseLogicalTimeMillisType

Formatter.avroUseLogicalTimeMicrosType

Formatter.avroUseLogicalTimestampMillisType

Formatter.avroUseLogicalTimestampMicrosType

If you use Avro logical types for dates, decimal values, times, or timestamps and want the CDC Publisher to make a best-effort attempt to process these logical types, set this property to true. The following sets of properties are mutually exclusive so specify one property or the other but not both:

- `Formatter.avroUseLogicalTimeMillisType` and `Formatter.avroUseLogicalTimeMicrosType`
- `Formatter.avroUseLogicalTimestampMillisType` and `Formatter.avroUseLogicalTimestampMicrosType`

The default value for each of these properties is false.

Note: If you set a property to true, make sure that the source fields are defined in the extraction map with a compatible data type, scale, and precision.

Formatter.captureColumnValuesFile

Identifies the full path and file name of the file in which you optionally define rules that the Formatter uses to generate a composite message key for each source table. Each rule specifies the column or columns to be included in the generated message key. PowerExchange CDC Publisher includes the generated message key in the messages that it sends to the target messaging system. The target messaging system writes the messages that contain the same key value to the same partition in the target topics. For example, if the rule identifies the DEPT column as the message key for a source table, all records that contain a specific DEPT value, such as "Finance," will be sent to the same topic partition. When using a message key to write messages to the same topic partition, the target messaging system can maintain the order in which the messages were received from the CDC Publisher. For information about defining rules, see ["Generating Composite Message Keys for Source Tables" on page 62](#).

Formatter.formatterAddTimestampColumn

Indicates whether the PowerExchange CDC Publisher adds a timestamp column to the generated Avro schema and formatted output messages to represent the date and time at which the Formatter processed the incoming change records. Valid values are:

- **false**. Do not add the timestamp column.

- **true**. Add the timestamp column.

Default is false.

If you set this property to true, you can optionally specify the column name, timestamp format, time zone, and source in the following properties: `Formatter.formatterAddedTimestampColumnName`, `Formatter.formatterAddedTimestampColumnFormat`, `Formatter.formatterAddedTimestampColumnTimezone`, and `Formatter.formatterAddedTimestampUseSource` properties.

Formatter.formatterAddedTimestampColumnFormat

If `Formatter.formatterAddTimestampColumn` is set to true, you can use this property to specify a date and time pattern string that indicates the format of the timestamp values in the added timestamp metadata column. Enter any date and time pattern that the Java class `SimpleDateFormat` supports for formatting dates and times. For more information, see <https://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>. Default is `yyyy/MM/dd HH:mm:ss.SSS`.

Formatter.formatterAddedTimestampColumnTimezone

If `Formatter.formatterAddTimestampColumn` is set to true, you can use this property to control the time zone in which the timestamp value in the added timestamp metadata column is reported. Valid values are:

- **local**. The local time zone where the CDC Publisher runs.
- **UTC**. Coordinated Universal Time.

Default is **local**.

Formatter.formatterAddedTimestampUseSource

If `Formatter.formatterAddTimestampColumn` is set to true, you can use this property to indicate whether the added timestamp column contains PowerExchange-generated `DTL__CAPXTIMESTAMP` timestamp values included in captured source records. Valid values are:

- **false**. Do not write the `DTL__CAPXTIMESTAMP` values to the added timestamp column.
- **true**. Write the `DTL__CAPXTIMESTAMP` values to the added timestamp column.

Default is false.

Formatter.formatterAddedTimestampColumnName

If the `Formatter.formatterAddTimestampColumn` property is set to true, you can use this property to specify the name of the added timestamp metadata column. This column will appear in the generated Avro schema and formatted output messages. Enter an alphanumeric string. Default is `INFA_TIME_CREATED`.

Formatter.generateCommitDML

Indicates whether the Formatter generates messages for transaction commit operations. Also indicates whether the Formatter generates a commit message for each source table that was updated by the committed transaction or generates one commit message for all of the updated tables by using the schema of the last updated table. Valid values are:

- **none**. Do not generate messages for commit operations.
- **LAST_TABLE**. Generate a single commit message for all source tables that the transaction updated. The Formatter generates the commit message by using the Avro schema of the last source table that was updated by the transaction.

- **ALL_TABLES.** Generate a commit message for each source table that was updated by the transaction. Consider using this option if you configured CDC Publisher to generate one topic per source table.

Default is **none**.

If you enable the generation of commit messages, you can optionally set the `Connector.kafkaCommitDmlTopic` and `Connector.kafkaCommitDmlTopicFiltering` properties in the `cdcPublisherKafka.cfg` file.

Custom Pattern Formatter Configuration Properties

If you use the Custom Pattern Formatter, you can add the following configuration properties to the `cdcPublisherAvro.cfg` file.

Formatter.customPatternFileList

Required. Identifies the full path and file name of the file that contains the paths to the table-specific custom pattern files for the Custom Pattern Formatter:

```
Formatter.customPatternFileList=path/filename
```

For more information, see [“Configuring the Custom Pattern Formatter” on page 65](#).

Formatter.customPatternDataFile

Optional. Identifies the full path and file name of the custom pattern data file:

```
Formatter.customPatternDataFile=path/filename
```

If you want to simulate the output from CDC Publisher, create a custom pattern data file. This file is a local file that contains data for each table definition defined in the Custom Pattern Formatter configuration file. For more information, see [“Custom Pattern Data File” on page 113](#) and the `CHECKCUSTOMPATTERNS` parameter description in [“PwxCDCAdmin Utility - Command and Parameters” on page 86](#).

Connector Configuration Properties

The sample `cdcPublisherKafka.cfg` file contains configuration properties related to the Apache Kafka or MapR Streams target.

Property Descriptions

The following properties are in the sample `cdcPublisherKafka.cfg` configuration file:

Connector.kafkaProducerPropertiesFile

Required. The path and file name of the Kafka producer.properties file that PowerExchange CDC Publisher uses to communicate with Kafka. This file is typically in the `/Kafka_installation/config` directory.

No default value is provided.

Connector.kafkaTopic

Required. The Kafka topic or topics to which PowerExchange CDC Publisher sends messages that contain extracted change data. Enter a specific topic name if you want to send all messages to a single topic, or enter `USE_TABLE_NAME` to direct the target messaging system to use a separate topic for each source table. A generated source-specific Kafka topic uses the `mapname_tablename` portion of the full extraction map name as the topic name. The full extraction map name has the format `schema.mapname_tablename`. For MapR targets, the topic name includes the stream name in the format `stream_path_name:mapname_tablename`.

Note: By default, the target messaging system automatically generate topics if the topics do not exist the first time messages are sent to it. You can disable the automatic generation of topics by setting the `auto.create.topics.enable` parameter to `false` in Kafka or by setting the `autocreate` parameter to `false` in MapR. If you do so, you must manually create the topics before CDC Publisher starts publishing messages.

No default value is provided.

Connector.queueType

Required. The type of target messaging queue to which PowerExchange CDC Publisher streams messages. Valid values are:

- **kafka** for an Apache Kafka target
- **maprkafka** for a MapR Streams target that uses the Kafka API for producers

The following additional properties can also be included in the `cdcPublisherKafka.cfg` file at your discretion:

Connector.checkpointMessageFrequency

Controls the frequency of writing checkpoints to the checkpoint file. Specify the number of target messages that must be written before a checkpoint is taken. The `Connector.checkpointMessageFrequency` setting works in tandem with the `Connector.checkpointTimeFrequency` setting.

Any number 0 and greater is valid. For more information, see [“Considerations for Storing Checkpoints in Kafka” on page 30](#).

Important: The checkpoint value is used on CDC Publisher restart to determine where to start reading from the change stream. Frequent checkpoints reduce the number of duplicate messages that might be sent when you restart CDC Publisher. Less frequent checkpoints reduce overhead while increasing the number of duplicates that might be sent on restart.

The default is 0. No checkpoint will be taken.

Connector.checkpointPublisherId

If you set the `Connector.checkpointsInTarget` property to `true`, this property is required to specify the logical name of the CDC Publisher instance to use for writing checkpoint information to Kafka message topic headers. When PowerExchange CDC Publisher starts, the specified value is compared to the `checkpointPublisherId` value in the Kafka headers.

Important: Enter a name that uniquely identifies the CDC Publisher instance. If the name is not unique, the checkpoint information might be shared across topics, which can cause data corruption on the target.

Valid values are any string that uniquely identifies the CDC Publisher instance.

No default value is provided.

Connector.checkpointTimeFrequency

Controls the frequency at which checkpoints are written to a checkpoint file. Specifies the number of seconds that must elapse before a checkpoint is written to the checkpoint file. The `Connector.checkpointTimeFrequency` setting works in tandem with the `Connector.checkpointMessageFrequency` setting.

Any number 0 and greater is valid. For more information, see [“Considerations for Storing Checkpoints in Kafka” on page 30](#).

Important: The checkpoint value is used on CDC Publisher restart to determine where to start reading from the change stream. Frequent checkpoints reduce the number of duplicate messages that might be

sent when you restart CDC Publisher. Less frequent checkpoints reduce overhead while increasing the number of duplicates that might be sent on restart.

The default is 0. No checkpoint will be taken.

Connector.checkpointsInTarget

Controls whether PowerExchange CDC Publisher stores checkpoints for CDC restart processing in Kafka headers on the target or in a local checkpoint file. Kafka version 0.11.0.2 or later is required. Valid values are:

- **true**. Store checkpoint information in Kafka.

Note: Do not use this option if you have a Kafka version that does not support headers.

- **false**. Store checkpoint information in a checkpoint file instead of Kafka.

Default is false.

With this property, you can also specify the following related connector configuration properties:

- **Connector.checkpointMessageFrequency**. Optional.
- **Connector.checkpointPublisherId**. Required.
- **Connector.checkpointTimeFrequency**. Optional.

Connector.kafkaCheckpointFileDirectory

A name for the CDC Publisher instance subdirectory to which the CDC Publisher writes checkpoint files. The default value is "checkpoint." Use this property to override the default subdirectory name.

Connector.kafkaCommitDmlTopic

Indicates whether to send all commit messages to a single topic that you specify or to send commit messages to the topic or topics that the `Connector.kafkaTopic` property identifies. Valid values are:

- *topic_name*. The name of the Kafka topic to which the CDC Publisher sends all commit messages. Use this option to send commit messages to a topic that is different from the topic that the `Connector.kafkaTopic` property identifies.
- **default**. Send commit messages to the topic or topics as identified in the `Connector.kafkaTopic` property. If the `Connector.kafkaTopic` property specifies `USE_TABLE_NAME`, the commit messages, along with the data messages, are sent to the source-specific topics that are generated for each source table that was by the transaction.

Default value is the **default**.

To use this property, you must enable the generation of commit messages in the **Formatter.generate.CommitDML** property in the `cdcPublisherAvro.cfg` file.

Connector.kafkaCommitDmlTopicFiltering

Indicates whether to filter commit messages before they are sent to the topic or topics that receive messages for DML operations in a transaction. Valid values are:

- **none**. Do not filter the commit messages to be sent to topics. Send all of the commit messages to each topic that receives messages for the DML operations in the transaction.
- **MAX_ONE_PER_TOPIC**. Send only one commit message to each topic that receives messages for the DML operations in the transaction.

Default is **MAX_ONE_PER_TOPIC**.

To use this property, you must enable the generation of commit messages in the **Formatter.generate.CommitDML** property in the `cdcPublisherAvro.cfg` file.

Connector.kafkaConsumerPropertiesFile

Optional. The path and file name of the Kafka consumer.properties file that PowerExchange CDC Publisher uses to communicate with Kafka.

If you do not specify a value and PowerExchange CDC Publisher needs to establish a connection as a Kafka consumer, the producer properties will be used with the appended or overridden values needed for the PowerExchange CDC Publisher consumer requirements.

This file is typically in the `/Kafka_installation/config` directory.

No default value is provided.

Connector.kafkaFileCheckpointFileName

Controls the name of the file to which the CDC Publisher writes checkpoints. This file is located in the subdirectory that the `Connector.kafkaCheckpointFileDirectory` property specifies. Enter a specific file name or `DEFAULT`. If you enter `DEFAULT`, the CDC Publisher uses the default file name of "checkpoint." The default property value is `DEFAULT`.

Connector.kafkaMaprStreamName

Required for MapR Streams targets. The path and name of the existing MapR stream that contains the topic or topics to which the CDC Publisher will publish messages.

Note: In MapR, the stream name is combined with the topic name in the format `path_and_name_of_stream:topic_name` to identify a topic. The CDC Publisher combines the stream name that you specify in this property with the topic name that you specify in the `Connector.kafkaTopic` property.

Connector.kafkaMessageKey

Identifies the key value to include in the messages that the CDC Publisher producer delivers to the target messaging system. You can specify a key value or use the source table name as the key value. Valid values are:

- *string*. Specify a character string to use as the message key in all messages. The messages will be sent to the same partitions in the target topics.
- **USE_TABLE_NAME**. Use the source table name as the message key in all messages. Messages for a specific source table will be sent to the same partitions in the target topics.

If you omit this property, no message key is sent to the target messaging system.

To generate a composite message key for a source table before the messages are sent to a target topic, see ["Generating Composite Message Keys for Source Tables" on page 62](#).

Connector.kafkaProducerGuaranteeDelivery

Controls whether the CDC Publisher uses *guaranteed delivery* to reduce the possibility that a message is sent to the target messaging system more than once or in the wrong order. Valid values are:

- **true**. Enables guaranteed delivery. CDC Publisher overrides some CDC Publisher connector properties to force a single stream of messages to be synchronously delivered one at a time to a single partition in any target topic. This delivery mode avoids message loss and duplication. CDC Publisher writes a checkpoint after each message is acknowledged as successfully delivered to the target.
- **false**. Disables guaranteed delivery. If the target messaging system terminates while changes are in flight, CDC Publisher might deliver duplicate messages to the target topics after the change stream is restarted.

Default is true.

Connector.kafkaProducerPartitionID

The numeric partition ID that the CDC Publisher instance assigns to each message that is sent to the target messaging system. CDC Publisher uses this partition ID across all messages and target topics.

Valid values are -1 through 32767.

The value of -1 causes the target messaging system to select the partition ID. If you configure a message key in the Connector.kafkaMessageKey property, the target messaging system can use the message key to assign messages to partitions.

If you enter a valid value greater than -1, CDC Publisher writes data to the partition that has specified partition ID. Ensure that a partition with the specified ID exists in all of the topics to which the CDC Publisher will write messages.

Default is -1.

Connector.kafkaTopicAllowSpecialCharacters

Controls whether the period (.) and dash (-) special characters can be used in Kafka target topic names. Also determines whether topic names can begin with a number.

Valid values are:

- **true**. Allows the period (.) and dash (-) characters to be used in topic names. Also, allows a number to be the first character in topic names.
- **false**. Causes topic names that include a period or dash character to be either automatically adjusted or rejected. If you set Connector.kafkaTopic=USE_TABLE_NAME to derive topic names from source table names, any period or dash in a table name is translated to an underscore (_) character. If you set Connector.kafkaTopic to a specific topic name that includes period or dash, the topic name is rejected. Also, with this setting, topic names cannot begin with a number.

Default is **false**.

Connector.logConnectorStatsOnExit

Enables PowerExchange CDC Publisher to write runtime statistics to the message log when the Publisher process shuts down. The statistics are printed in message CDCPUB_15034. The statistics are the same as those produced by the PWXDCINFO STATS=TOPIC command.

- **true**. Prints runtime statistics at shutdown.
- **false**. Does not print runtime statistics at shutdown.

Default is **false**.

Connector.restartCheckpointSource

Allows an override of where the checkpoint information is to be derived from. Use this property if checkpoint sources become corrupted and an override is needed to specify where checkpoints are to be obtained. Valid values are:

- **all**. Checkpoints can be collected from the target and the backup checkpoint file.
- **target**. Acquire the checkpoint from the target.
- **file**. Acquire the checkpoint from the backup file.

Default is all.

Connector.sendMaintainOrder

Controls whether the CDC Publisher maintains the order in which change operations were retrieved from the source when sending messages to the target messaging system. Valid values are:

- **true**. Sends messages synchronously to the same partition in each target topic in the order that the change operations were retrieved from the source.
- **false**. Sends messages asynchronously to one or more partitions in target topics as soon as possible, without regard for the order in which the change operations were retrieved from the source.

Default is **true**.

PowerExchange Extract Configuration Properties

The sample `cdcPublisherPowerExchange.cfg` file contains properties that the PowerExchange CDC Publisher uses to extract change from PowerExchange and to retrieve PowerExchange extraction map definitions of source objects. The following list describes the properties in the sample file. It also includes some additional properties that you might want to use.

Property Descriptions

The following properties are in the sample `cdcPublisherPowerExchange.cfg` file:

Extract.pwxCapiConnectionName

Required. The NAME parameter value in the CAPX CAPI_CONNECTION statement in the PowerExchange DBMOVER configuration file. This CAPI_CONNECTION statement is used to connect to the PowerExchange change data stream to read data from PowerExchange Logger log files.

No default value is provided.

Extract.pwxExtractionMapSchemaName

Required. The PowerExchange-generated schema name that is included in the names of the PowerExchange extraction maps for the source objects. This schema name has the format *xnninstance* or *dnninstance*, where "u" indicates a user-defined extraction map, "d" indicates the default extraction map generated from the capture registration, *nn* is a two-digit value that represents the source type, and *instance* is the instance value from the PowerExchange registration group for the data source. The schema name is the first part of the full extraction map name: *xnninstance.mapname_tablename*. This property is used to select the extraction maps that the CDC Publisher will use to extract change data.

Note: All extraction maps with names that match the specified schema name or mask are selected. If you want to use a subset of these extraction maps, define the `Extract.captureConsumerExcludeNameList` or `Extract.captureConsumerIncludeNameList` property to filter the extraction maps.

No default value is provided.

Extract.captureConsumerIncludeNameList

A list of the extraction maps that match the specified schema name and that you want to explicitly *include* in extraction processing. The PowerExchange CDC Publisher process will use only these included extraction maps. Any other extraction maps that were selected based on the extraction-map schema name are ignored. For each extraction map to include, enter the full extraction map name in the format *xnninstance.mapname_tablename*. Use a comma (,) separator between the values, for example:

```
Extract.captureConsumerIncludeNameList=xnninstance.mapname_tablename1,  
xnninstance.mapname_tablename2,...
```

You can include the asterisk (*) wildcard anywhere in the value to represent one or more characters, thereby creating a mask that matches multiple extraction maps.

Extract.captureConsumerExcludeNameList

A list of the extraction maps that match the specified schema name but that you want to *exclude* from extraction processing. For each extraction map to exclude, enter the full extraction map name in the format *xnninstance.mapname_tablename*. Use a comma (,) separator between the values, for example:

```
Extract.captureConsumerExcludeNameList=xnninstance.mapname_tablename1,  
xnninstance.mapname_tablename2,...
```

You can include the asterisk (*) wildcard anywhere in the value to represent one or more characters, thereby creating a mask that matches multiple extraction maps.

Extract.pwxNodeLocation

Required. The node name (first positional parameter) in the NODE statement that is used to connect to the PowerExchange Listener for change data extraction. This statement is defined in the PowerExchange DBMOVER configuration file. This Listener runs on the system with the PowerExchange Logger for Linux, UNIX, and Windows log files.

No default value is provided.

Extract.pwxNodeId

A user ID that is authorized to access the node that contains the PowerExchange Logger log files from which change data will be extracted. Define this property only if access to the node specified in the Extract.pwxNodeLocation property is secured by a user ID and password. If you use remote logging of source data to PowerExchange Logger for Linux, UNIX, and Windows log files that reside off the source system, this property value matches the CAPTURE_NODE_UID parameter value in the PowerExchange Logger configuration file, pwxcl.cfg.

No default value is provided.

Extract.pwxNodePwd

A clear-text or encrypted password that is associated with the user ID specified in the Extract.pwxNodeId property. To encrypt the password, run the PwxCDCAdmin utility with the ENCRYPT parameter. When you enter the encrypted password in this property, include the "ENC:" prefix.

Define this property only if access to the node specified in the Extract.pwxNodeLocation property is secured by a user ID and password. If you use remote logging of source data to PowerExchange Logger for Linux, UNIX, and Windows log files that reside off the source system, this property value matches the CAPTURE_NODE_PWD parameter value in the PowerExchange Logger configuration file, pwxcl.cfg.

If you use PowerExchange-encrypted passwords, specify the Extract.pwxNodeEPwd property instead.

No default value is provided.

Extract.pwxNodeEPwd

An encrypted password that is associated with the user ID specified in the Extract.pwxNodeId property. This password must have been encrypted by using the PowerExchange Navigator Encrypt Password facility.

Define this property only if access to the node specified in the Extract.pwxNodeLocation property is secured by a user ID and password. If you use remote logging of source data to PowerExchange Logger for Linux, UNIX, and Windows log files that reside off the source system, this property value matches the CAPTURE_NODE_EPWD parameter value in the PowerExchange Logger configuration file, pwxcl.cfg.

If you use clear-text passwords or passwords that have been encrypted by the CDC Publisher PwxCDCAdmin ENCRYPT command, specify the Extract.pwxNodePwd property instead.

No default value is provided.

Extract.pwxXmapLocation

The node name (first positional parameter) in the NODE statement that is used to connect to the PowerExchange Listener on the source system to retrieve extraction maps for source objects. This statement is defined in the local PowerExchange DBMOVER configuration file. If you use remote logging of source data to PowerExchange Logger for Linux, UNIX, and Windows log files that reside off the source machine, this property value matches the CAPTURE_NODE parameter value in the PowerExchange Logger configuration file, pwxcl.cfg.

Note: If the data source and extraction maps are on the same node as the PowerExchange Listener and Logger, you do not need to specify this property. You can specify only the Extract.pwxNodeLocation property to connect the PowerExchange Listener that handles both requests for extraction maps and requests for change data extraction.

No default value is provided.

Extract.pwxXmapUserId

A user ID that is authorized to access the node that contains the PowerExchange extraction maps for source objects. Define this property only if access to the node specified in the Extract.pwxXmapLocation property is secured by a user ID and password. If you use remote logging of source data to PowerExchange Logger for Linux, UNIX, and Windows log files that reside off the source system, this property value matches the CAPTURE_NODE_UID parameter value in the PowerExchange Logger configuration file, pwxcl.cfg.

No default value is provided.

Extract.pwxXmapPwd

A clear-text or encrypted password that is associated with the user ID specified in the Extract.pwxXmapUserId property. To encrypt the password, run the PwxCDCAdmin utility with the ENCRYPT parameter. When you enter the encrypted password in this property, include the "ENC:" prefix.

Define this property only if access to the node specified in the Extract.pwxXmapLocation property is secured by a user ID and password. If you use remote logging of source data to PowerExchange Logger for Linux, UNIX, and Windows log files that reside off the source system, this property value matches the CAPTURE_NODE_PWD parameter value in the PowerExchange Logger configuration file, pwxcl.cfg.

No default value is provided.

The following additional properties can also be included the cdcPublisherPowerExchange.cfg file at your discretion:

Extract.pwxOffloadOption

Controls whether PowerExchange offloads some change data processing from the PowerExchange Listener system where the data resides to the CDC Publisher system. Valid values are:

- **N.** Disable offloading processing. Typically, this value is acceptable.
- **Y.** Enable offload processing.

Default is N.

Extract.pwxReadLatency

The maximum number of seconds that PowerExchange waits for more change data before flushing data to the PowerExchange client on the CDC Publisher machine.

Valid values are 2 through 86400. Default is 2.

Extract.pwxReturnLogInfoMessages

Controls whether the informational messages that PowerExchange returns to the CDC Publisher are written to the PwxCdcPublisher.out file in the logs subdirectory. Set this property to true to log the

PowerExchange informational messages, or set this property to false to not log these informational messages. Default is false.

Extract.pwxUpdateImageOption

Indicates how UPDATES that are read from the source are published to the CDC Publisher target. Valid values are:

- **AI.** Process each update as a single UPDATE record that contains the after image only.
- **TU.** Process each update as a single UPDATE record that contains both the before image and after image.

Note: To have the before-image values included in the generated messages to be sent to the target, you must also specify `Formatter.avroIncludeBeforeImage=true` in the `cdcPublisherAvro.cfg` configuration file.

- **BA.** Process each update as a DELETE record that contains the before image followed by an INERT record that includes the after image.

Default is TU.

Important: If the PowerExchange Logger for Linux, UNIX, and Windows is configured to log only after images, you must set this property to AI. Otherwise, PowerExchange issues a fatal error.

Extract.restart1

The PowerExchange sequence token value that you want to use to restart the CDC Publisher process from a specific point in the change stream. For this property value to be used, you must restart the PwxCdcPublisher process with the `COLDSTART=Y` or `RESTART=FROM_CONFIG` parameter. You must also specify the `Extract.restart2` property. No default value is provided.

Extract.restart2

The PowerExchange restart token value that you want to use to restart the CDC Publisher process from a specific point in the change stream. For this property value to be used, you must restart the PwxCdcPublisher process with the `COLDSTART=Y` or `RESTART=FROM_CONFIG` parameter. You must also specify the `Extract.restart1` property. No default value is provided.

Filtering Extraction Maps and Source Objects

PowerExchange CDC Publisher uses the `pwxExtractionMapSchemaName` property value to select the PowerExchange extraction maps to use for extracting change data from source objects. You can filter this set of extraction maps and the source objects that are associated with them by specifying filtering criteria. You can enter filtering criteria that exclusively includes or excludes extraction maps.

To filter extraction maps, enter one or both of the following properties in the `cdcPublisherPowerExchange.cfg` configuration file:

- **Extract.captureConsumerIncludeNameList.** Specifies a subset of the selected extraction maps to explicitly *include* in extraction processing. For each extraction map to include, enter the full extraction map name in the format `xnninstance.mapname_tablename` with a comma (,) separator between values, as follows:

```
Extract.captureConsumerIncludeNameList=xnninstance.mapname_tablename1,  
xnninstance.mapname_tablename2,...
```

You can use the asterisk (*) wildcard anywhere in the value to represent one or more characters, thereby creating a mask that matches multiple extraction maps. The PowerExchange CDC Publisher process will

use the included extraction maps and ignore any other extraction maps that matched the extraction-map schema name.

Use this parameter when you want to explicitly include some of the selected extraction maps and ignore all of the other extraction maps that matched the `pwxExtractionMapSchemaName` value.

- **Extract.captureConsumerExcludeNameList.** Specifies a subset of the selected extraction maps to *exclude* from extraction processing. For each extraction map to exclude, enter the full extraction map name in the format `xnninstance.mapname_tablename`. Use a comma (,) separator between values. You can include the asterisk (*) wildcard anywhere in the value to represent one or more characters, thereby creating a mask that matches multiple extraction maps.

Use this property when you want to use most of the extractions maps that matched the `pwxExtractionMapSchemaName` value and exclude only those that match the exclusion criteria in this property.

You can use the `Extract.captureConsumerIncludeNameList` and `Extract.captureConsumerExcludeNameList` properties together. For example, use an `Extract.captureConsumerIncludeNameList` mask to select a subset of extraction maps and then use the `Extract.captureConsumerExcludeNameList` to narrow the extraction maps in that subset.

For example, assume the following properties are specified:

- `Extract.pwxExtractionMapSchemaName=d8district1`
- `Extract.captureConsumerIncludeNameList=d8district1.store*`
- `Extract.captureConsumerExcludeNameList=d8district1.store1_sales, d8district1.store1_customers, d8district1.store1_items`

In this case, all extraction maps that have names beginning with the `xnninstance` value, or schema name, of "d8district1" are selected. Of the selected extraction maps, you explicitly include only those that have `mapname` values that begin with "store." From the subset of included extraction maps, you explicitly exclude the three extraction maps for map "store 1."

Notes:

- The PowerExchange CDC Publisher does not automatically exclude extraction maps that are associated with capture registrations that have the status of Inactive, History, or Suspended. If no change data is available, the CDC Publisher does not publish change data for the registered source objects. If you will never use these deactivated capture registrations or their associated extraction maps to capture and publish change data, you can exclude the extraction maps in the CDC Publisher to reduce resource consumption. Before you define the exclusion filter, ensure that any change data that was previously captured for the source object has been published to the target messaging system. After you define the exclusion filter, you must restart the CDC Publisher process for the filter to take effect.
- If you use PowerExchange Logger for Linux, UNIX, and Windows log groups, run a PowerExchange CDC Publisher instance for each log group. In CDC Publisher, filter the extraction maps that each CDC Publisher instance uses for extraction processing based on the capture registrations defined for the log group that the instance is processing. Note that the CDC Publisher uses extraction maps for data extraction, whereas the log groups are associated with capture registrations.

Configuring CDC Publisher Message Logging

The PowerExchange CDC Publisher uses the Apache Log4j 2 API to implement message logging and tracing.

The PwxCDC Publisher provides a sample Log4j logging configuration file in XML format, PwxCDCPubLog4j.xml, in the `\pwxcdcpub_root_installation\samples` subdirectory. The sample file contains the Log4j 2 configuration properties that the PowerExchange CDC Publisher uses.

Note: For information about Log4j 2 logging configuration properties, see the Apache Log4j 2 documentation.

You must copy the sample PwxCDCPubLog4j.xml file from the samples subdirectory to the `\pwxcdcpub_root_installation\instanceA\config` subdirectory. If you use multiple instances, copy the file to the config subdirectory for each instance. Do not rename the file. Usually, the PwxCDCPubLog4j.xml file in the config subdirectory can be used without modification, even for instances other than instanceA. However, if you want trace messages to be logged, you must change the following Logger statement to specify `level="all"`:

```
<Logger name="trace" level="all" additivity="false" includeLocation="false">
  <AppenderRef ref="TraceFile"/>
</Logger>
```

If requested by Informatica Global Customer Support, you can enable additional Logger instances in the PwxCDCPubLog4j.xml file.

Based on the PwxCDCPubLog4j.xml file configuration, as delivered, Log4j 2 logs CDC Publisher messages to files in the `\pwxcdcpub_root_installation\instanceA\logs` subdirectory. If you use instances other than instanceA, the PwxCDCPubLog4j.xml file is automatically updated with the current instance name when you start the PwxCDCPublisher process so that messages will be logged to the logs subdirectory of that instance.

Also, based on the delivered PwxCDCPubLog.xml file, Log4j 2 generates log files for PowerExchange CDC Publisher messages in the `\pwxcdcpub_root_installation\instance\logs` subdirectory using the following naming conventions:

- The active log file is named PwxCdcPublisher.out.
- Archive log files are generated using the following file-name format:
PwxCdcPublisher.out.yyyy-mm-dd-hh-n.gz, where *yyyy-mm-dd* is year-month-day, *hh* is hour of the day, and *n* is a generated number. In the a sample Log4j logging configuration file, the maximum number of archive log files is set to 999999.

The active log is archived when the maximum file size is reached or when the CDC Publisher is restarted. The Log4j default maximum size is 10 MB. You can change this maximum size in the PwxCDCPubLog4j.xml file. Each time you restart the CDC Publisher, the previous log file is archived and a new active log file is created.

For your convenience, any error and warning messages that PowerExchange returns to the CDC Publisher are also written to the PwxCdcPublisher.out file in the logs subdirectory by default. Optionally, you can configure the CDC Publisher to also log PowerExchange informational messages to the PwxCdcPublisher.out file by specifying `Extract.pwxReturnLogInfoMessages=true` in the `cdcPublisherPowerExchange.cfg` configuration file.

Configuring CDC Publisher to Support Multiple Partitions in Individual Topics

You can configure PowerExchange CDC Publisher to allow the messages that contain change operations to be written to multiple partitions in any target topic on the target messaging system.

By default, CDC Publisher streams all messages that contain change operations to the same partition in any target topic. This behavior ensures that the order in which the change operations were received from the source is maintained when CDC Publisher writes the messages to the target topics on the target messaging system. If you allow CDC Publisher to write messages to multiple partitions in any target topic, the order in which changes were retrieved from the source is *not* maintained when writing messages to the target topics.

Note: Depending on how you configure the `Connector.kafkaTopic` property, a target topic receives changes for all source tables or for a specific source table. In either case, the configuration requirements for using multiple partitions are the same.

To allow CDC Publisher to support multiple partitions in the targets topics, configure the following properties in the Connector configuration file, `cdcPublisherKafka.cfg`:

Connector.kafkaProducerGuaranteeDelivery=false

Controls whether the CDC Publisher uses guaranteed delivery to reduce the possibility that a message is sent to the target messaging system more than once or in the wrong order. With the default value of `true`, CDC Publisher overrides some CDC Publisher settings to ensure guaranteed delivery. Set this property to `false` to disable guaranteed delivery and be allowed to set the `Connector.sendMaintainOrder` property to `false`.

Connector.kafkaProducerPartitionID=-1

The numeric partition ID that the CDC Publisher instance assigns to each message that is sent to the target messaging system. CDC Publisher uses this partition ID across all messages and target topics. When sending messages to target topics with multiple partitions, use the default value of `-1` to allow the target messaging system to determine the partition ID.

Connector.sendMaintainOrder=false

Controls whether CDC Publisher sends messages in the order in which the change operations were retrieved from the source or sends the messages as soon as possible without regard to order. When sending messages to target topics with multiple partitions, CDC Publisher cannot ensure that the order of change operations from the source is maintained. In this scenario, set this property to `false`.

Generating Composite Message Keys for Source Tables

You can configure rules to generate a composite message key that includes one or more columns for each source table. PowerExchange CDC Publisher can then include the generated message key in the messages that it sends to the target messaging system. The target messaging system can use the message key to write messages with a specific key value to the same partition in a multi-partitioned topic.

When writing messages with a specific key value to the same partition, the target messaging system can maintain the order in which the change records were received from the CDC Publisher.

Note: If you increase the number of partitions in a target topic, the source data might be written to a different topic partition than before, even if you configured generated message keys.

For the Formatter to generate a message key for each source table, you must create a rule-configuration file in a text editor and then specify the file in the `Formatter.captureColumnValuesFile=pathFilename` property in the `cdcPublisherAvro.cfg` file.

Rule Syntax:

Use the following syntax to define rules in the rule-configuration file:

```
rule=(xnninstance.mapname_tablename,column1,column2,column3,... )
additional rules...
[delimiter=character]
[tableNotFound=ABORT]
[trace={true|false}]
[updateUseImage={before|after}]
```

To include comments in the file, begin each comment line with the number (#) sign. For example:

```
#This text is for informational purposes only.
```

Parameter Descriptions:

- **rule.** Defines a rule for generating a composite message key for a source table. In each rule, first identify the extraction map for the source table in the format `xnninstance.mapname_tablename`, where `xnninstance` is the schema name that PowerExchange generates for the extraction map, `mapname` is the extraction map name, and `tablename` is the source table name. Then specify the names of one or more table columns that comprise the message key. Ensure that the columns are defined in the extraction-map schema. CDC Publisher compares the specified column names to the extraction-map schema. If any column is not found in the schema, processing ends abnormally and multiple CDCPUB-14019 messages are written to the trace file.

You can define multiple rules in the same rule-configuration file.

When generating the message key, the Formatter uses the character representation of each column value followed by the delimiter character. Each column value and delimiter are appended to the composite key value in the order in which the columns appear in the rule definition. The composite key is then used as the Kafka message key for a record. The position of any column that has empty or null values in the message key is represented by the delimiter character only.

- **delimiter.** *Optional.* Specifies a single character that will be used as a delimiter after each key column value in generated message keys. You can specify this parameter only once in the rule-configuration file. Default is the semicolon (;).
- **tableNotFound.** *Optional.* Set this parameter to ABORT to cause the Formatter to stop processing any source table data when the table does not have a rule definition in the rule-configuration file. Each source table must have a rule definition for the generation of the composite message key to succeed. You can specify this parameter only once in the rule-configuration file.

If you do not specify this parameter, the `Connector.kafkaMessageKey` property in the `cdcPublisherKafka.cfg` file can be used to create the message key for any source table that does not have a rule definition. Otherwise, the default rule in the target messaging system parameters determines the key to use for a record.

You can specify this parameter only once in the rule-configuration file.

- **trace.** *Optional.* Enables or disables tracing for the generation of message keys based on rule definitions. Valid values are:
 - **true.** Enables tracing for message key generation based on rule definitions. If message key generation is successful, CDC Publisher writes message CDCPUB_14018 to the trace data set. This message identifies the generated key value, the source table, and the change stream location of the key generation event.
 - **false.** Disables tracing for message key generation based on rule definitions.

You can specify this parameter only once in the rule-configuration file.

Default is **false**.

- **updateUseImage**. *Optional*. For source update operations only, indicates whether to use the before-image or after-image of the update row when generating the message key. Valid values are:

- **before**. Use the before-image of the data, as it appeared before the update operation.

- **after**. Use the after-image of the data, as it appeared after the update.

You can specify this parameter only once in the rule-configuration file.

Default is **before**.

Example

You define the following two rules:

```
rule=(shipping.address,ADDRESSNO,STREET,CITY)
rule=(hr.employees,EMP,ADDRESSNO,STREET,CITY,COUNTRY)
tableNotFound=ABORT
trace=false
delimiter=;
```

For the shipping.address table, assume that the ADDRESSNO column value is "123" and the STREET column value is "Park Lane" in the data records. The CITY column is null. Based on these values, the following message key is generated for the table:

```
123;Park Lane;;
```

Note that the key columns appear in the same order that they are specified in the rule definition.

For the hr.employees table, assume that the EMP column value is "Fred", the ADDRESSNO column value is "12345", the STREET column value is "Main Street", the CITY column value is "New York", and the COUNTRY column value is "US". Based on these values, the following message key is generated for the table:

```
Fred;12345;Main Street;New York;US;
```

If the Formatter tries to process data for any table that does not have a rule definition, processing ends abnormally because the tableNotFound property is set to ABORT.

Customizing Target Message Output

The Custom Pattern Formatter is an extension of the Formatter. The Custom Pattern Formatter customizes the Formatter Avro message output based on custom patterns that you define to meet your requirements. You can define patterns in any text-based form, such as comma separated (CSV), JSON, or other text-based format. The custom patterns use tags that are similar to HTML tags. The custom patterns describe the text output.

You can create table-specific custom patterns. For example, you can perform the following actions:

- Generate customized JSON output. For more information, see [“Custom JSON in Avro Format Example” on page 108](#).
- Define a default pattern for tables for which a table-specific custom is not defined. For more information, see [“Configuring the Custom Pattern Formatter” on page 65](#).
- Define patterns that customize the message output for date, time, and timestamp columns. You can override the data type or change the date or time format. For more information, see [“Column Attributes” on page 95](#).

Configuring the Custom Pattern Formatter

To use custom patterns, you must create a Custom Pattern Formatter configuration file. This configuration file contains a list of source tables and the locations of the custom pattern files for each table. The custom pattern files contain the custom pattern definitions for each source table.

Custom Pattern Formatter Configuration File

In the Custom Pattern Formatter configuration file, define the source table names and the full path and name of the file that contains the custom pattern definitions for each source table.

The configuration file uses the following tags:

Tag	Description
<table>	Defines the table name for table-specific custom patterns or specifies the keyword DEFAULT. Specify the DEFAULT keyword to indicate the custom pattern file or the formatter CDC Publisher should use for tables that are not specified.
<formatter>	If you specify DEFAULT in a <table> tag, you can specify this <formatter> tag. It specifies the default formatter to use when CDC Publisher encounters a table that is not listed in the Custom Pattern Formatter configuration file. The only valid value is AVRO. When AVRO is specified and the table is not found in the list, the results will be written to Kafka in the Avro message format. This option allows some tables to use the Custom Pattern Formatter while other tables use the existing Avro format.
<path>	Specifies the path and name of the file that contains the custom pattern definition for the table specified in the <table> tag or the path of the file that contains the default custom pattern definition.

Place the configuration file in a location that is accessible to PowerExchange CDC Publisher. To specify the location of the file, specify the Formatter configuration property `Formatter.customPatternFileList` in the `cdcPublisherAvro.cfg` file. For example:

```
Formatter.customPatternFileList=pwxcDCpub_root_installation\instance\config\  
customFormatterPatternsFile.list
```

For more information, see [“Custom Pattern Formatter Configuration Properties” on page 51](#).

Configuration File Example: Table-Specific Patterns with a DEFAULT Pattern

In the following example, the Custom Pattern Formatter configuration file contains three <table> definitions:

- `my_table1.pattern` contains the custom patterns for `my.table1`.
- `my_table2.pattern` contains the custom patterns for `my.table2`.
- `DEFAULT` indicates that any table that does not have a table-specific custom pattern file should use the default custom pattern file called `generic.pattern`.

```
<table>my.table1</table><path>\pwxcDCpub_root_installation\instance\config\my_table1.pattern</path>  
<table>my.table2</table><path>\pwxcDCpub_root_installation\instance\config\my_table2.pattern</path>  
<table>DEFAULT</table><path>\pwxcDCpub_root_installation\instance\config\generic.pattern</path>
```

Configuration File Example: Table-Specific Patterns with a DEFAULT Pattern That Specifies Avro Format

In the following example, the Custom Pattern Formatter configuration file contains a DEFAULT table specification that specifies the Avro format type:

```
<table>my.table1</table><path>\pwxcDCpub_root_installation\instance\config\my_table1.pattern</path>  
<table>my.table2</table><path>\pwxcDCpub_root_installation\instance\config\my_table2.pattern</path>  
<table>DEFAULT</table><formatter>AVRO</formatter>
```

Based on the last statement, the formatted results for any table that is not specified in a <table> tag will be generated in the Avro format.

Table-Specific Custom Pattern Files

Provide a custom pattern file to describe the output by source tables. A custom pattern file contains user-defined patterns for specific tables. A custom pattern file can contain patterns that are used by more than one table or it can specify patterns for a specific table. Patterns are characterized by change data event types. Within a pattern, event type sections contain specifications that describe what PowerExchange CDC Publisher should do for the event type. You can define a custom pattern file with one or more sections. Each section in the file is independent. You do not need to define every event type. For example, if you want to write inserts for a table to Kafka, you do not need to specify the schema event, transaction event, update event, delete event, and commit event sections in the pattern within the file.

For example, assume that the following configuration property is set in the cdcPublisherAvro.cfg file:

```
Formatter.customPatternFileList=pwxcddpub_root_installation\instance\config\mytables.list
```

The mytables.list file contains the locations of the custom pattern files for each table and has the following contents:

```
<table>tableA</table><path>\pwxcddpub_root_installation\instance\config\tableA.pattern</path>  
<table>DEFAULT</table><path>\pwxcddpub_root_installation\instance\config\generic.pattern</path>
```

The tableA.pattern file contains the custom patterns for table A, and the generic.pattern file contains the default custom patterns for any table that does not have a specific custom pattern file.

For information about the format and syntax of custom pattern files, see [Appendix C, “Custom Pattern Formats” on page 94](#).

If Informatica Data Engineering Streaming Will Consume Data from a Target

PowerExchange CDC Publisher streams change data that PowerExchange captured in near real time to target messaging systems such as Apache Kafka. Informatica Data Engineering Streaming can then consume the change data from the target message queue and use it for a variety of purposes. For example, Data Engineering Streaming can use the change data to generate near-real-time fraud detection alerts or customize sales offers at point-of-sale.

If the Data Engineering Streaming product will consume the change data that the PowerExchange CDC Publisher sends to a target messaging system, use the following PowerExchange CDC Publisher configuration guidelines:

- Data Engineering Streaming cannot consume data from fields that have a binary data type. Configure the PowerExchange CDC Publisher to send data from binary fields as string data by setting the following properties in the cdcPublisherAvro.cfg configuration file:
 - **Formatter.avroBinaryAsString=true**. With this setting, binary data is represented as string data in the generated Avro messages.
 - **Formatter.avroBinaryStringRepresentationType={base64|hexadecimal}**. When **Formatter.avroBinaryAsString=true**, this property determines whether to use base64 or hexadecimal strings to represent binary data. Default is base64.
- Data Engineering Streaming cannot consume JSON-encoded Avro messages. To use binary-encoded messages, specify **Formatter.avroEncodingType=binary** in the cdcPublisherAvro.cfg configuration file.

- As a consumer application, Data Engineering Streaming must have copies of the Avro schemas for the source tables to properly interpret the change data in the messages. You can use the REPORT=FORMAT parameter of the PwxCDCAdmin utility to report the existing Avro schemas in a legible format for use by consumer applications. If no Avro schemas have been generated for the source tables, the utility attempts to create the Avro schemas based on the properties in the cdcPublisherAvro.cfg configuration file. For more information, see [“PwxCDCAdmin Utility - Command and Parameters” on page 86](#).
- If you try to import an Avro schema that the PowerExchange CDC Publisher generated for a very large table and that is larger than 65535 bytes into Data Engineering Streaming, the Scala compiler issues a Java exception related to the scala.tools.asm package. This problem occurs because the Scala code does not handle literals greater than 65535 bytes in size. To circumvent this problem, you can configure the PowerExchange CDC Publisher to generate Avro schema in a minimized format by specifying some or all of the following properties in the cdcPublisherAvro.cfg configuration file:
 - **Formatter.avroSchemaPrintPretty={true|false}**. Set this property to false to *not* include the spaces and line feeds that are intended to improve legibility in the generated Avro schemas. Default value is true, which causes the spaces and line feeds to be included.
 - **Formatter.avroSchemaPrintDocFields={true|false}**. Set this property to false to *not* report the "doc" fields in the generated Avro schemas. The doc fields include metadata such as the CDC and PowerExchange datatypes, precision, and scale. Default value is true, which causes this information to be included.
 - **Formatter.avroSchemaPrintDefaultFields={true|false}**. Set this property to false to *not* include the "default" fields in the generated Avro schemas. Default value is true, which causes the default fields to be included.

CHAPTER 7

Streaming Change Data

This chapter includes the following topics:

- [Before You Start Data Streaming, 68](#)
- [Starting the PwxCDCPublisher Utility, 68](#)
- [Operational Considerations, 69](#)

Before You Start Data Streaming

To begin streaming change data from a PowerExchange source to a target messaging system, you start the main PwxCDCPublisher utility.

The PwxCDCPublisher utility can run as a Linux daemon in the foreground or background or as a Windows process in the foreground.

Before you start the PwxCDCPublisher utility, ensure that the following prerequisites are met:

- Active capture registrations and extraction maps exist for the data source objects.
- The PowerExchange Listener and PowerExchange Logger for Linux, UNIX, and Windows are running.
- PowerExchange has captured change records from the data sources and logged them in PowerExchange Logger log files.
- The target messaging system is running.

Starting the PwxCDCPublisher Utility

You can start the PwxCDCPublisher utility from a command prompt at the *pwxcdcpub_root_installation\bin* subdirectory or by using batch file, provided that file system security allows you to run this script. This procedure describes how to start the utility from a command prompt.

Before you start the PwxCDCPublisher, make sure the following prerequisites are met:

- Ensure that the PowerExchange Listener instances in the PowerExchange environment are running.
- If the target type is Apache Kafka, ensure that Apache Kafka and Zookeeper are running.

- If the target type is MapR Streams, ensure that the MapR server is running.
- ▶ At the command prompt, enter the PwxCDCPublisher command. Include the optional INSTANCE parameter if you want to run the PwxCDCPublisher process using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`.

Use the following syntax:

```
PwxCDCPublisher [INSTANCE=override_instance_directory] [NOHUP]
```

On Linux, you can include NOHUP to run the PwxCDCPublisher as a daemon in the background.

Tip: For the full command syntax, including other optional PwxCDCPublisher parameters, see ["PwxCDCPublisher Utility - Command and Parameters" on page 83](#).

The startup output is displayed, for example:

```
-----
PowerExchange CDC Publisher from Informatica LLC 1.0.0
Copyright (c) 2014-2017 Informatica LLC. All rights reserved. See patents at
https://www.informatica.com/legal/patents.html.
INSTANCE = instanceA. CDC Publisher Started. Listening on host:port USHOST1:26987
-----
Running...
PWX-07122 DTLCLI connected using DTLConnect PWX Version: 10.1.1, Patch Level: V1011_B05_5095439,
Local code pages: Control=UTF-8 (41) Data=ISO-8859 (1) SQL=UTF-8 (41).
PWX-33304 Connection: 10.17.3.191:64393, Type: CAPXRT, Task ID: 0, Operation timeout: (-) secs,
Heartbeat interval: 30 secs, Session: (-)
PWX-33304 Connection: 10.17.3.191:64399, Type: CAPXRT, Task ID: 0, Operation timeout: (-) secs,
Heartbeat interval: 30 secs, Session: (-)
PWX-31517 To retrieve extraction maps, PowerExchange uses the value for Map Location,
'mylaptop_v1000',
from the PWX CDC application connection.
PWX-33304 Connection: 10.17.3.191:64393, Type: CAPXRT, Task ID: 0, Operation timeout: (-) secs,
Heartbeat interval: 30 secs, Session: (-)
PWX-07122 DTLCLI connected using DTLConnect PWX Version: 10.1.1, Patch Level: V1011_B05_5095439,
Local code pages: Control=UTF-8 (41) Data=ISO-8859 (1) SQL=UTF-8 (41).
```

To verify that the PowerExchange CDC Publisher is active and processing change records, run the PwxCDCInfo utility. For more information, see ["Monitoring Overview" on page 71](#).

Operational Considerations

The following operational considerations apply to PowerExchange CDC Publisher change data streaming:

- You can configure multiple PowerExchange CDC Publisher "instances," each with its own set of checkpoint files, configuration files, log files, and Avro schemas. When you start a PwxCDCPublisher process, you specify the name of the instance to use. The instance name corresponds to a directory within the PowerExchange CDC Publisher installation. By default, the PowerExchange CDC Publisher installation creates the "instanceA" directory and its subdirectories. Consider adding instances to process change data in multiple streams for different sources or different targets or for load balancing.
- The PowerExchange CDC Publisher reads extraction map information only at startup. If you edit the capture registrations and the associated extraction maps for the data sources, PowerExchange CDC Publisher does not automatically update the Avro schema. You must run the PwxCDCAdmin utility with the CLEAR parameter to remove the old Avro schema from cache. You can then run the utility again with RESET parameter to re-read the extraction maps and cache the updated Avro schema. After you restart the PwxCDCPublisher process, it can use the updated Avro schema in cache to generate messages in the latest schema format for transmittal to the target messaging system. For more information about PwxCDCAdmin utility parameters, see ["PwxCDCAdmin Utility - Command and Parameters" on page 86](#).
- If you start the PwxCDCPublisher process before the PowerExchange Logger for Linux, UNIX, and Windows has written at least one source change record to its log files, the Publisher Extractor process might end

with an error. This issue is caused by a PowerExchange Logger reader limitation that pertains to the first run of the PowerExchange Logger.

- PowerExchange CDC Publisher reads changes from successful units of work (UOWs) in commit order. By default, the CDC Publisher maintains the order of DML change operations received from the source when streaming messages that contain the change data to the target messaging system. However, if you configure CDC Publisher to stream messages to multiple partitions in any topic, CDC Publisher cannot ensure that change operations are written to the target messaging system in the same order that they were received from the source.
- The CDC Publisher processes a stream of change records in a manner that is independent from transaction start and end points. If the CDC Publisher terminates or connectivity to the target messaging system is lost before all of the changes in a transaction are processed, the target will have only some of the changes until you restart the CDC Publisher so that it can send the remaining changes in the transaction. The CDC Publisher process will restart from the last checkpoint position recorded in the checkpoint file by default.
- The PowerExchange CDC Publisher can filter source tables or objects by using the filtering criteria you specify in the `Extract.captureConsumerIncludeNameList` and `Extract.captureConsumerExcludeNameList` properties in the `cdcPublisherPowerExchange.cfg` configuration file. However, the CDC Publisher cannot filter source columns.
- The PowerExchange CDC Publisher does not automatically exclude extraction maps that are associated with capture registrations that have the status of Inactive, History or Suspended. If no change data is available, the CDC Publisher does not publish change data for the registered source objects. If you will never use these deactivated capture registrations or their associated extraction maps to capture and publish change data, you can exclude the extraction maps in the CDC Publisher to reduce resource consumption. Before you define the exclusion filter, ensure that any change data that was previously captured for the source object has been published to the target messaging system. After you define the exclusion filter, you must restart the CDC Publisher process for the filter to take effect. If you plan to reactivate a suspended registration, first remove the exclusion filter for the associated extraction maps. Then restart the CDC Publisher process and reactivate the registration.

CHAPTER 8

Monitoring PowerExchange CDC Publisher

This chapter includes the following topics:

- [Monitoring Overview, 71](#)
- [Determining the Status of CDC Publisher Processes, 72](#)
- [Generating Statistics on CDC Publisher Processing, 73](#)

Monitoring Overview

You can use the PwxCDCInfo utility to report PowerExchange CDC Publisher status information and internal statistics for CDC Publisher processes, updated target topics, and CDC Publisher internal attributes.

The PwxCDCInfo utility can report the following types of monitoring information:

- The overall status of the PowerExchange CDC Publisher process and the status of each major subprocess (Extractor, Formatter, and Connector). The status is designated by a color indicator: "GREEN" for running, "YELLOW" for a warning situation, and "RED" for stopped.
- Information about the target topics that the CDC Publisher updated, including the topic name, the associated source table name, the number of messages sent to the topic, and the last change operation type and sequence ID.
- The PowerExchange Logger log file, a .cnd file, that the CDC Publisher is reading.
- Statistics for diagnostic use, by internal attribute and CDC Publisher child process. You can use the STATS=INTERNAL command while the CDC Publisher instance is running to determine the most recent log file CDC Publisher has read or the log file it is currently reading. For more information, see ["Generating Statistics on CDC Publisher Processing" on page 73](#).

Also, the PowerExchange CDC Publisher messages that are written to the logs subdirectory provide information about errors, potential errors, and processing status. For more information, see [Appendix D, "Message Reference" on page 116](#).

Determining the Status of CDC Publisher Processes

To determine the overall status of PowerExchange CDC Publisher processes, run the PwxCDCInfo utility with the STATUS parameter. To provide status information for each CDC Publisher process (Extractor, Formatter, and Connector), include the optional DETAIL parameter.

You can use this information to determine if a CDC Publisher process stopped or has a potential problem.

Before you issue the PwxCDCInfo command, ensure that the PowerExchange CDC Publisher process is running in a separate command line window.

- ▶ At the command prompt, enter the PwxCDCInfo command with the STATUS parameter. Include the optional INSTANCE parameter if the Publisher process is using an instance other than the default "instanceA" at `/pwxpub_root installation/instanceA.`

Use the following syntax:

```
PwxCDCInfo STATUS [DETAIL [VERBOSE]] [INSTANCE=override_instance_directory]
```

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the PWXPUB_HOME environment variable.

For the full command syntax, see ["PwxCDCInfo Utility - Command and Parameters" on page 84.](#)

The command output is displayed on screen. Optionally, you can direct the output to a file.

- If you entered `PwxCDCInfo STATUS`, the output shows the overall status of all PowerExchange CDC Publisher processes by using a color indicator and brief description. For example:

```
PROCESS STATE GREEN All processes are running
```

The other process states are:

- YELLOW. A warning condition exists. The data stream might be restricted or not flowing.
- RED. The CDC Publisher processes are not running or cannot be found.

- If you entered `PwxCDCInfo STATUS DETAIL`, the output shows the status color indicator for each CDC Publisher child process. For example:

PARENT	CHILD	ATTRIBUTE	VALUE	DESCRIPTION
Extract_Connector	Extract	TRAFFIC LIGHT	GREEN	
Extract_Connector	Formatter	TRAFFIC LIGHT	GREEN	
Extract_Connector	Connector	TRAFFIC LIGHT	GREEN	

- If you entered `PwxCDCInfo STATUS DETAIL VERBOSE`, the output shows the status color indicator for each CDC Publisher child process with a description of the valid status values. For example:

PARENT	CHILD	ATTRIBUTE	VALUE	DESCRIPTION
Extract_Connector	Extract	TRAFFIC LIGHT	GREEN	Similar to a traffic light, a red value indicates an error, a green value indicates the process is running as expected, and a yellow value indicates a mixed state (not an error and not running as expected).
Extract_Connector	Formatter	TRAFFIC LIGHT	GREEN	Similar to a traffic light, a red value indicates an error, a green value indicates the process is running as expected, and a yellow value indicates a mixed state (not an error and not running as expected).
Extract_Connector	Connector	TRAFFIC LIGHT	GREEN	Similar to a traffic light, a red value indicates an error, a green value indicates the process is running as expected, and a yellow value indicates a mixed state (not an error and not running as expected).

Generating Statistics on CDC Publisher Processing

You can use the PwxCDCInfo utility with the STATS parameter to generate statistics and state information for the topics to which the CDC Publisher sent messages and for CDC Publisher internal attributes for diagnostic use.

The statistics are cumulative from the start time of the PowerExchange CDC Publisher process. You can reset the internal statistics to their original values to determine the processing activity from the reset point going forward.

Before you issue the PwxCDCInfo command, ensure that the PowerExchange CDC Publisher process is running in a separate command line window.

- ▶ At the command prompt, enter the PwxCDCInfo command with the STATS=TOPIC or STATS=INTERNAL parameter. Include the optional INSTANCE parameter if the Publisher process is using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`.

Use the following syntax:

```
PwxCDCInfo STATS={TOPIC|INTERNAL} [DETAIL [VERBOSE]] [RESET]
[INSTANCE=override_instance_directory]
```

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the PWXPUB_HOME environment variable.

Tip: For the full command syntax, see ["PwxCDCInfo Utility - Command and Parameters" on page 84](#).

The command output is displayed on screen. Optionally, you can direct the output to a file.

If you entered **STATS=TOPIC**, the output shows processing information for each topic that the CDC Publisher updated, including a count of messages sent to the topic and the last change operation type and sequence ID. The following example shows information for one updated topic:

```
Topic updated:          abcd0006_SRC_ALLCHAR_16K
PartitionID of the topic: 0
Count of messages sent: 8
Partition offset returned: 7
Message key:
Source table name:      d8collnam1.abcd0006_SRC_ALLCHAR_16K
Type of last change event: INSERT_EVENT
Sequence ID of last change: 2,PWX_GENERIC,1,,2,3,D4002CA219131A00000000000000002CA21913190000001F000037
C3000017000124000100000000000100000000,00,0000018DF3288319
```

Note: The CDC Publisher generates the Sequence ID value and uses it internally to track the last change applied to the topic.

You can use the STATS=TOPIC RESET command to clear the statistics.

If you entered **STATS=INTERNAL**, the output shows internal statistics and state information, by attribute, for each type of PowerExchange CDC Publisher child process. This information is primarily for diagnostic use by Informatica Global Customer Support and other technical staff. The following example shows a small portion of this report:

PARENT	CHILD	ATTRIBUTE	VALUE	DESCRIPTION
Extract	Extract Capture Consumer	ErrorState	false	
Extract	Extract Capture Consumer	IsRunningState	true	
Extract	Extract Capture Consumer	IsSleepingState	true	
Extract	Extract Capture Consumer	SleepDuration	2729102	
Extract	Extract Capture Consumer	InputRecordsRead	4761	
Extract	Extract Capture Consumer	InputBytesRead	10976086	
Extract	Extract Capture Consumer	LastSequenceIn	2,PWX_GENERIC,1,,2,3,...	
Extract	Extract Capture Consumer	EventsSkippedOldSequence	1	
Extract	Extract Capture Consumer	InputSchemaRecordsRead	10	
Extract	Extract Capture Consumer	InputDMLRecordsRead	4728	

Tip: The STATS=INTERNAL output is displayed in a wide format. For easier reading, direct the output to a file.

You can use STATS=INTERNAL RESET command to set the statistics back to their original values.

If you entered **STATS=INTERNAL VERBOSE**, the output shows the same type of diagnostic information as the **STATS=INTERNAL** report with an additional description of each attribute. The following example shows a small portion of this report:

PARENT	CHILD	ATTRIBUTE	VALUE	DESCRIPTION
Extract	Extract Capture Consumer	ErrorState	false	is True if the process is currently in error, false otherwise.
Extract	Extract Capture Consumer	IsRunningState	true	True if the processing is running, false if it has stopped running.
Extract	Extract Capture Consumer	IsSleepingState	true	True if the processing is sleeping, false if not.
Extract	Extract Capture Consumer	SleepDuration	2596322	If sleeping, the current duration of that sleep (milliseconds).
Extract	Extract Capture Consumer	InputRecordsRead	4761	The current number of input records consumed by this process.
Extract	Extract Capture Consumer	InputBytesRead	10976086	The current number of bytes of input consumed by this process.
Extract	Extract Capture Consumer	LastSequenceIn	2,PWX_GENERIC,1,,2,...	The sequence value of the last operation read into this process.
Extract	Extract Capture Consumer	EventsSkippedOldSequence	1	The number of incoming events skipped because their sequence indicates they were previously processed.
Extract	Extract Capture Consumer	InputSchemaRecordsRead	10	The current number of input records of type Schema consumed by this process.
Extract	Extract Capture Consumer	InputDMLRecordsRead	4728	The current number of input records of type DML (inserts, updates, deletes and truncates) consumed by this process.

You can use the **STATS=INTERNAL** command while the CDC Publisher instance is running to determine the most recent log file CDC Publisher has read or the log file it is currently reading. The following example shows a portion of the report:

```
2020-10-07 15:18:49,078 [pool-3-thread-3 ] INFO [PWXRDR_01201] [PWX-04512 Opening captured file /ausqa/pwx/v1040orad/capture/linux/mlx121q1_mp2/*zcaps_and_alldataTypes.CND.CP200910.T0052001*.]
2020-10-07 15:18:49,078 [pool-3-thread-3 ] INFO [PWXRDR_01201] [PWX-04512 Opening captured file /ausqa/pwx/v1040orad/capture/linux/mlx121q1_mp2/*zcaps_and_alldataTypes.CND.CP200910.T0104001*.]
2020-10-07 15:18:49,078 [pool-3-thread-3 ] INFO [PWXRDR_01201] [PWX-04512 Opening captured file /ausqa/pwx/v1040orad/capture/linux/mlx121q1_mp2/*zcaps_and_alldataTypes.CND.CP200910.T0212001*.]
2020-10-07 15:18:49,078 [pool-3-thread-3 ] INFO [PWXRDR_01201] [PWX-04512 Opening captured file /ausqa/pwx/v1040orad/capture/linux/mlx121q1_mp2/*zcaps_and_alldataTypes.CND.CP200910.T0226001*.]
```

CHAPTER 9

Administering PowerExchange CDC Publisher

This chapter includes the following topics:

- [Administration Overview, 75](#)
- [Reporting the Avro Format Definitions for Source Tables, 76](#)
- [Handling Changes to Source Tables and Extraction Maps, 76](#)
- [Adding a Source Table to a Change Data Stream, 78](#)
- [Encrypting a Password, 79](#)
- [Shutting Down the CDC Publisher, 80](#)
- [Restarting a PowerExchange CDC Publisher Change Data Stream, 80](#)

Administration Overview

To perform administrative tasks in PowerExchange CDC Publisher, you primarily use the PwxCDCAdmin utility. Check that your file system security allows you to have access to the PwxCDCAdmin utility script.

This section covers the following PwxCDCAdmin administrative tasks:

- Reporting the current Avro format definitions in a legible format, which can be used for validation and by consumer applications
- Handling changes to the source table structure, such as added or dropped columns
- Adding a source table to the change stream of a PowerExchange CDC Publisher process
- Encrypting a password for use in the pwxNodePwd or pwxXmapPwd property in the cdcPublisherPowerExchange.cfg configuration file
- Shutting down a PowerExchange CDC Publisher process

This section also describes how to restart the PwxCDCPublisher utility after it shuts down or ends abnormally using an override restart position in the PowerExchange Logger log records.

Reporting the Avro Format Definitions for Source Tables

Use the PwxCDCAdmin utility with the REPORT=FORMAT parameter to generate the Avro format definitions, or schemas, for all source tables or a specific source table in the `\instance\reports` subdirectory.

The generated format definitions are available in a legible format for viewing and can be used by consumer applications to read the messages that the PowerExchange CDC Publisher sent to the target messaging system. The generated schemas reflect the flat, nested, or generic format that you set in the `Formatter.avroSchemaFormat` parameter setting in the `cdcPublisherAvro.cfg` configuration file.

Before you issue the PwxCDCAdmin command, ensure that the PowerExchange CDC Publisher process is running in a separate command line window.

1. At the command prompt, enter the PwxCDCAdmin command with the REPORT=FORMAT and TABLE parameters.

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the `PWPUB_HOME` environment variable.

In the TABLE parameter, indicate whether to generate the schemas for all tables or a specific table. Also include the optional INSTANCE parameter if you are running the PwxCDCPublisher process using an instance other than the default "instanceA" at `/pwxcdcpub_root_installation/instanceA`.

Use the following syntax:

```
PwxCDCAdmin REPORT=FORMAT TABLE={ALL|mapname_tablename}
[INSTANCE=override_instance_directory]
```

In this syntax, `mapname_tablename` is the map name and table name portion of the extraction map name, where the extraction map name has the format `xnninstance.mapname_tablename`. For more information about the command syntax, see ["PwxCDCAdmin Utility - Command and Parameters" on page 86](#).

2. To verify that the Avro format definitions were generated for the source tables or table, check the reports subdirectory under the instance that you are using.

Handling Changes to Source Tables and Extraction Maps

PowerExchange CDC Publisher does not update the Avro schemas if the source table structure and extraction map definition change. The extraction map definition must change if the source columns or fields for which change data is extracted are added or removed.

Use the following procedure to add or remove source columns or fields and properly resume PowerExchange CDC Publisher processing.

1. On the source system, stop change activity (inserts, updates, and deletes) on the source table.
2. Verify that the change data based on the current table definition has been published to the target messaging system or downstream consumer application. Use the following methods:
 - If you know the change operation that was last applied to the source table, verify that the target messaging system or consumer application has received that change.

- In the change records that the target messaging system or consumer application has received, verify that the DTL__CAPXTIMESTAMP value is later than the time at which change activity against the source table was stopped.

If neither the target messaging system nor consumer application has received the expected change records, verify that the CDC Publisher is running. To do so, run the `PwxCDCInfo STATUS` command. The expected result is "PROCESS STATE GREEN All processes are running."

3. Shut down the PowerExchange Logger for Linux, UNIX, and Windows.

For more information, see the *PowerExchange CDC Guide for Linux, UNIX, and Windows*.

4. Optionally, clear the Avro schemas that match the existing source extraction maps from cache if you want to force the generation of updated Avro schemas that reflect the source table changes.

Note: If you do not perform this step, CDC Publisher will continue to use the cached Avro schemas that do not reflect the source table changes.

To clear the Avro schemas from cache, enter the `PwxCDCAdmin` command with the `CLEAR=FORMAT` and `TABLE` parameters at the command prompt. Use the following syntax:

```
PwxCDCAdmin CLEAR=FORMAT TABLE={ALL|mapname_tablename}
[INSTANCE=override_instance_directory]
```

Include the optional `INSTANCE` parameter if you are using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`. The `mapname_tablename` variable is the map name and table name portion of the new extraction map name, where the extraction map name has the format `xnninstance.mapname_tablename`. For more information about the command syntax, see ["PwxCDCAdmin Utility - Command and Parameters" on page 86](#).

The command produces output such as shown in the following example:

```
INSTANCE = instanceA. Connecting to host:port USHOST1:26987
-----
COMMAND      RESULT  RESULT CODE  DESCRIPTION
CLEAR_FORMAT      1          The number of Avro Schema definitions cleared
```

5. Shut down the `PwxCDCPublisher` process by using the `PwxCDCAdmin SHUTDOWN` command. For more information, see ["Shutting Down the CDC Publisher" on page 80](#).

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the `PWPUB_HOME` environment variable. For more information, see ["Shutting Down the CDC Publisher" on page 80](#).

6. In the PowerExchange Navigator, open the original capture registration and set its status to **History**. For more information, see the *PowerExchange Navigator User Guide*.
7. Alter the source table or object to add or remove columns.
8. In the PowerExchange Navigator, re-create the capture registration and associated extraction map for the source table. Activate the capture registration.
9. If you set any PowerExchange CDC Publisher properties that refer to the `mapname_tablename` portion of the extraction map name, update the properties to refer to the `mapname_tablename` portion of the regenerated extraction map name.

Note: The extraction map name has the format `xnninstance.mapname_tablename`.

For example, perform this step if you defined the `Extract.captureConsumerExcludeNameList` and `Extract.captureConsumerIncludeNameList` filter properties.

10. On the PowerExchange CDC Publisher system, restart the `PwxCDCPublisher` process as you would under normal conditions. For more information, see ["Restarting a PowerExchange CDC Publisher Change Data Stream" on page 80](#).

11. To verify that the CDC Publisher recognizes the schema change and that the Avro schema is properly updated, perform the following substeps:

- a. To get the updated extraction map information, enter the PwxCDCAdmin command with the RESET=FORMAT parameter at the command prompt.

Use the following syntax:

```
PwxCDCAdmin RESET=FORMAT [INSTANCE=override_instance_directory]
```

Include the optional INSTANCE parameter only if you are using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`.

Alternatively, you can wait until the CDC Publisher first receives change data for the updated table. At that point, the CDC Publisher automatically caches the extraction map and generates the Avro schema.

- b. To force the generation of the updated Avro schema so that you can verify that it properly reflects the changed table structure, enter the PwxCDCAdmin command with the REPORT=FORMAT and TABLE parameters.

Use the following syntax:

```
PwxCDCAdmin [REPORT=FORMAT TABLE={ALL|mapname_tablename}]  
[INSTANCE=override_instance_directory]
```

In this syntax, *mapname_tablename* is the map name and table name portion of the new extraction map name. Include the optional INSTANCE parameter only if you are using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`.

This command generates new Avro schema in the `\instance\reports` subdirectory. The schema is generated in a legible format. The generated schema can be used for verification purposes and by consumer applications.

12. Warm start the PowerExchange Logger.

Adding a Source Table to a Change Data Stream

To add a new source table or object to a PowerExchange CDC Publisher change data stream, you must create a PowerExchange extraction map definition for the new table and restart the PwxCDCPublisher process. Optionally, you can generate an Avro format definition (schema) for the table so that you can verify it.

Note: If you do not generate the Avro schema, the CDC Publisher will automatically generate it when change data is received for the new source table.

1. Shut down the PwxCDCPublisher process by issuing the PwxCDCAdmin SHUTDOWN command.

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the `PWX_PUB_HOME` environment variable. For more information, see ["Shutting Down the CDC Publisher" on page 80](#).

2. Shut down the PowerExchange Logger for Linux, UNIX, and Windows.

For more information, see the *PowerExchange CDC Guide for Linux, UNIX, and Windows*.

3. In the PowerExchange Navigator, create a capture registration and associated extraction map for the new source table. Activate the capture registration.

For more information, see the *PowerExchange Navigator User Guide*. In the capture registration, set the **Status** to **Active** and the **Condense** option to **Part**.

4. Restart the PwxCDCPublisher process. For more information, see ["Restarting a PowerExchange CDC Publisher Change Data Stream" on page 80](#).

5. To retrieve extraction map information for all source tables from PowerExchange, including the new table, enter the PwxCDCAdmin command with the RESET=FORMAT parameter at the command prompt. Include the optional INSTANCE parameter if you are using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`.

Use the following syntax:

```
PwxCDCAdmin RESET=FORMAT [INSTANCE=override_instance_directory]
```

Alternatively, you can wait until the CDC Publisher first receives change data for the new table. At that point, the CDC Publisher automatically caches the extraction map and generates the Avro schema.

6. If you specified the PwxCDCAdmin RESET=FORMAT command and want to force the generation of the Avro schema for the new table so that you can verify that it properly reflects the table structure, enter the PwxCDCAdmin command with the REPORT=FORMAT and TABLE parameters at the command prompt. The TABLE parameter is required. Include the optional INSTANCE parameter if you are using an instance other than the default "instanceA" at `/pwxcdcpub_root installation/instanceA`.

Use the following syntax:

```
PwxCDCAdmin REPORT=FORMAT  
TABLE=mapname_tablename[INSTANCE=override_instance_directory]
```

In this syntax, `mapname_tablename` represents the table name portion of the new extraction map name, where the extraction map name has the format `xnninstance.mapname_tablename`.

This command generates the updated Avro schema in the `\instance\reports` subdirectory. The schema is generated in a legible format. The generated schema can be used for verification purposes and by consumer applications.

7. Warm start the PowerExchange Logger.

Encrypting a Password

You can encrypt a password to be entered in the `Extract.pwxXmapPwd` or `Extract.pwxNodePwd` property in the `cdcPublisherPowerExchange.cfg` configuration text file to protect the password from unauthorized viewing. Use the PwxCDCAdmin utility with the ENCRYPT parameter.

Note: Do not use this method to encrypt a password that will be entered in the `Extract.pwxNodeEPwd` property. This property value must be encrypted with the PowerExchange Navigator Encrypt Password facility.

- At the command prompt, enter the PwxCDCAdmin command with the ENCRYPT parameter.

Use the following syntax:

```
PwxCDCAdmin ENCRYPT=nonencrypted_password
```

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the `PWXPUB_HOME` environment variable.

The PwxCDCAdmin utility generates and displays an encrypted password in the format `ENC:encrypted_password`, where "ENC:" is the prefix that the CDC Publisher uses to identify an encrypted password value. When you enter the encrypted password in the `Extract.pwxXmapPwd` or `Extract.pwxNodePwd` property, include the prefix, for example:

```
Extract.pwxXmapPwd=ENC:AC/YB0tjDJDR2DBdAUXgqDQ==
```

Shutting Down the CDC Publisher

Use the PwxCDCAdmin utility with the SHUTDOWN parameter to stop the PowerExchange CDC Publisher process.

If messages are being written to topics when the shutdown request is made, the connection to the target messaging system remains open so that the write operations can complete before the CDC Publisher shuts down. If CDC Publisher internal processes are processing change data, the processes ignore the in-progress changes and shut down cleanly. The processing of these in-progress changes will occur after the CDC Publisher process is restarted, provided that you use the default restart position.

Before you issue the PwxCDCAdmin command, ensure that the PowerExchange CDC Publisher process is running in a separate command line window.

- ▶ At the command prompt, enter the PwxCDCAdmin command with the SHUTDOWN parameter. Include the optional INSTANCE parameter if you are shutting down a PwxCDCPublisher process that is using an instance other than the default "instanceA" at */pwxcdcpub_root installation/instanceA*.

Use the following syntax:

```
PwxCDCAdmin SHUTDOWN [INSTANCE=override_instance_directory]
```

You can issue the command from the *pwxcdcpub_root_installation\bin* subdirectory or from another location if you specified the PWXPUB_HOME environment variable. For more information about the command syntax, see ["PwxCDCAdmin Utility - Command and Parameters" on page 86](#).

The following output from shutdown processing is displayed:

```
INSTANCE = instanceA. Connecting to host:port USHOST1:26987
-----
Shutdown request has been executed...
Waiting for server to shutdown. Wait duration 0:00:11 as of 2017/09/22 11:29:19
Shutdown completed. Server connection is no longer available.
```

Restarting a PowerExchange CDC Publisher Change Data Stream

If a PowerExchange CDC Publisher process is shut down or ends for any reason, you can use the PwxCDCPublisher utility to restart it from the default restart position in the PowerExchange Logger for Linux, UNIX, and Windows log files or from an override restart position that you set by using the PwxCDCPublisher RESTART or COLDSTART parameter.

By default, the first time you start the PwxCDCPublisher for a particular instance, it starts from the earliest position in the PowerExchange Logger log files. Subsequent PwxCDCPublisher runs start from the last position of the prior run unless you specify an override position. If you plan to override this default position by using the RESTART FROM_BEGINNING or FROM_CONFIG option or the COLDSTART=Y setting, first delete the existing checkpoint file.

- ▶ At the command prompt, enter the PwxCDCPublisher command without the RESTART or COLDSTART parameter to restart the Publisher process from the default restart position. To override the default restart position, include the optional RESTART or COLDSTART parameter. Also include the optional INSTANCE parameter if you are restarting a PwxCDCPublisher process that is using an instance other than the default "instanceA" at */pwxcdcpub_root installation/instanceA*.

You can issue the command from the `pwxcdcpub_root_installation\bin` subdirectory or from another location if you specified the `PWXPUB_HOME` environment variable.

Use the following syntax:

```
PWXCDCPUBLISHER [RESTART={FROM_BEGINNING|FROM_END|FROM_CONFIG}] [COLDSTART={Y|N}]  
[INSTANCE=override_instance_directory]
```

In this syntax:

- **RESTART.** Overrides the default restart position. Select one of the following options to specify the location in the change stream from which to restart Publisher processing:
 - **FROM_BEGINNING.** Extract change data from the earliest record that is available in the PowerExchange Logger log files, which is the same as the default start position for the first run of the PowerExchange CDC Publisher.
 - **FROM_END.** Extract change data beginning from the last processed record in the PowerExchange Logger log files.
 - **FROM_CONFIG.** Extract change data from the specific change stream position that you identify in the `Extract.restart1` and `Extract.restart2` properties in the `cdcPublisherPowerExchange.cfg` configuration file. This option is equivalent to setting `COLDSTART=Y`.

Note: The `Extract.restart1` and `Extract.restart2` properties specify the PowerExchange sequence token and restart token values, respectively. For more information about these PowerExchange token values, see the PowerExchange CDC documentation or contact Informatica Global Customer Support.
- **COLDSTART.** Set this parameter to `Y` to cold start the `PwxCDCPublisher` process from the specific position in the PowerExchange Logger log files that is identified by the `Extract.restart1` and `Extract.restart2` properties. This setting is equivalent to `RESTART=FROM_CONFIG`. Do not specify the `COLDSTART=Y` parameter with the `RESTART` parameter.

For more information about the command syntax, see [“PwxCDCAdmin Utility - Command and Parameters” on page 86](#).

The following example output is from a `PWXCDCPublisher RESTART=FROM_END` command:

```
INSTANCE = instanceA. CDC Publisher Started. Listening on host:port USHOST1:26987  
-----  
PWX-07122 DTLCLI connected using DTLConnect PWX Version: 10.1.1, Patch Level:  
V1011_B05_5095439, Local code pages: Control=UTF-8 (4I) Data=ISO-8859 (1) SQL=UTF  
-8 (4I).  
Running...  
PWX-33304 Connection: 10.17.3.191:49363, Type: CAPXRT, Task ID: 0, Operation  
timeout: (-) secs, Heartbeat interval: 30 secs, Session: (-)  
PWX-33304 Connection: 10.17.3.191:49369, Type: CAPXRT, Task ID: 0, Operation  
timeout: (-) secs, Heartbeat interval: 30 secs, Session: (-)  
PWX-33304 Connection: 10.17.3.191:49363, Type: CAPXRT, Task ID: 0, Operation  
timeout: (-) secs, Heartbeat interval: 30 secs, Session: (-)  
PWX-31517 To retrieve extraction maps, PowerExchange uses the value for Map  
Location, 'mylaptop_vl000', from the PWX CDC application connection.  
PWX-33304 Connection: 10.17.3.191:49363, Type: CAPXRT, Task ID: 0, Operation  
timeout: (-) secs, Heartbeat interval: 30 secs, Session: (-)  
PWX-07122 DTLCLI connected using DTLConnect PWX Version: 10.1.1, Patch Level:  
V1011_B05_5095439, Local code pages: Control=UTF-8 (4I) Data=ISO-8859 (1) SQL=UTF  
-8 (4I).
```

APPENDIX A

Command Reference for the Command-Line Utilities

This appendix includes the following topics:

- [Command Reference Overview, 82](#)
- [PwxCDCPublisher Utility - Command and Parameters, 83](#)
- [PwxCDCInfo Utility - Command and Parameters, 84](#)
- [PwxCDCAdmin Utility - Command and Parameters, 86](#)

Command Reference Overview

This command reference describes the proper syntax, usage, and limitations of commands for controlling the PowerExchange CDC Publisher command-line utilities.

Commands for the following PowerExchange CDC Publisher utilities are covered:

- **PwxCDCPublisher utility** - The main utility for streaming change data from PowerExchange to the target messaging system. The utility extracts change data from PowerExchange, filters the data by source object if filtering criteria is defined, formats the data into messages, connects to the target, and publishes the messages to the target. The utility initiates a data stream *path* that is composed of all of these processes.
- **PwxCDCAdmin utility** - The utility for issuing administrative commands to an active PwxCDCPublisher process.
- **PwxCDCInfo utility** - The utility for reporting the status of an active PwxCDCPublisher process and each of its main subprocesses, statistics and state information for each target topic, and internal attributes for diagnostic use.

The following table describes the conventions that the command reference uses to indicate the proper command syntax:

Convention	Description	Example
Monospaced font	Indicates the command syntax or a command example.	PWXCDCPUBLISHER HELP
<i>Italics</i>	Indicates a variable in the syntax for which you enter a specific value.	[ENCRYPT= <i>password_value</i>]

Convention	Description	Example
Square brackets []	Indicates an optional parameter or subparameter.	[SHUTDOWN]
A vertical bar between items	Indicates that you enter only one of the items.	TABLE={ALL <i>mapname_tablename</i> }
Curly brackets { } around multiple items	Indicates that you must enter at least one of the items within the brackets.	RESTART={FROM_BEGINNING FROM_END FROM_CONFIG}
<u>Underlining</u>	Indicates the default value	COLDSTART={Y <u>N</u> }

To run a command, navigate to the `bin` subdirectory that contains the utility script. Then enter the script name with the appropriate parameters at the command line.

You can run the PwxCDCAdmin and PwxCDCInfo utilities on a system that is remote from host where the PowerExchange CDC Publisher process is running. In this scenario, you must install the PowerExchange CDC Publisher on the remote system as well as on the CDC Publisher system. Also, in the configuration file on the remote system, you must set the `Common.pwxCDCPublisherHost` property to the public IP address of the CDC Publisher system and define the `PWXPUB_HOME` environment variable.

PwxCDCPublisher Utility - Command and Parameters

Use the PwxCDCPublisher utility to create and run a change data stream from PowerExchange to a supported target.

Also, the PwxCDCPublisher utility has optional parameters that allow you to specify an instance subdirectory other than `instanceA` and to indicate the change stream position from which the PwxCDCPublisher will restart.

Syntax

The PwxCDCPublisher command has the following syntax:

```
PWXCDCPUBLISHER
[HELP|?]
[INSTANCE=override_instance_directory]
[RESTART={FROM_BEGINNING|FROM_END|FROM_CONFIG}]
[COLDSTART={Y|N}
```

All parameters are optional. Use a single space as a separator between parameters.

Parameter Descriptions

HELP or ?

Displays help information about the PwxCDCPublisher parameters.

INSTANCE=*override_instance_subdirectory*

Overrides the default name or location of the instance subdirectory, which is `pwxcdcpub_root installation\instanceA`. You can enter the fully qualified path of the subdirectory or a location that is relative to the root installation directory that is specified in the `PWXPUB_HOME` environment variable.

For example, the following values specify the same directory location if PWXPUB_HOME=C:\Informatica\PwxCdcPublisherInstall.1.0.0.windows.64bit:

```
instanceB
C:\Informatica\PwxCdcPublisherInstall.1.0.0.windows.64bit\instanceB
```

RESTART={[FROM_BEGINNING|FROM_END|FROM_CONFIG]}

Overrides the default change stream position from which the PwxCDCPublisher process restarts after it is stopped or ends abnormally. To indicate the change stream position, enter one of the following options:

- **FROM_BEGINNING.** Extract change data from the earliest record that is available in the PowerExchange Logger log files.
- **FROM_END.** Extract change beginning from the last processed record in the PowerExchange Log log files.
- **FROM_CONFIG.** Extract change data from the specific change stream position that you identify in the Extract.restart1 and Extract.restart2 properties in the Extractor configuration file. This option is equivalent to setting COLDSTART=Y.

Note: The Extract.restart1 and Extract.restart2 properties specify the PowerExchange sequence token and restart token values, respectively. For more information about these PowerExchange token values, see the PowerExchange CDC documentation or contact Informatica Global Customer Support.

By default, the first time you start the PwxCDCPublisher, it starts from the earliest position in the PowerExchange Logger log files, as if FROM_BEGINNING was specified. Subsequent PwxCDCPublisher runs start from the last position of the prior run. If you want it to start from a different position, use the RESTART parameter. If you use the RESTART FROM_BEGINNING or FROM_CONFIG option, first delete the existing checkpoint file. Do not also specify the COLDSTART parameter.

COLDSTART={Y|N}

Indicates whether to cold start the PwxCDCPublisher process. If you specify COLDSTART=Y, the process begins extraction processing from the change stream position that you specify in the Extract.restart1 and Extract.restart2 properties in the cdcPublisherPowerExchange.cfg configuration file. The COLDSTART=Y setting is equivalent to the RESTART=FROM_CONFIG setting. COLDSTART=N indicates a cold start will not occur.

Note: Issue either the COLDSTART or RESTART parameter. Do not specify both parameters.

Default value is N.

PwxCDCInfo Utility - Command and Parameters

Use the PwxCDCInfo utility to display summary or detailed status information and internal processing statistics for a PwxCDCPublisher process.

Tip: The STATS INTERNAL and STATUS DETAIL output is displayed in a wide format. For easier reading, direct the output to a file.

Syntax

The PwxCDCInfo command has the following syntax:

```
PWXCDCINFO
[HELP|?]
[INSTANCE=override_instance_directory]
[STATUS [DETAIL [VERBOSE]]]
```

```
[STATS={TOPIC|INTERNAL [VERBOSE]}
[RESET]
```

All parameters are optional. Use a single space as a separator between parameters.

Parameter Descriptions

HELP or ?

Displays help information about the PwxCDCInfo parameters.

INSTANCE=override_instance_subdirectory

Overrides the default name or location of the instance subdirectory, which is *pwxcdcpub_root installation\instanceA*. You can enter the fully qualified path of the subdirectory or a location that is relative to the root installation directory that is specified in the PWXPUB_HOME environment variable. For example, the following values specify the same directory location if PWXPUB_HOME=C:\Informatica\PwxCdcPublisherInstall.1.0.0.windows.64bit:

```
instanceB
C:\Informatica\PwxCdcPublisherInstall.1.0.0.windows.64bit\instanceB
```

STATUS [DETAIL [VERBOSE]]

Indicates the status of the PwxCDCPublisher processes.

You can enter the STATUS parameter in any of following ways:

- **STATUS**. Reports the overall status of all PowerExchange CDC Publisher processes by using a color indicator and brief description. For example:

```
PROCESS STATE      GREEN      All processes are running
```

Other process states are: YELLOW for a warning situation, or RED for a CDC Publisher subprocess that stopped or could not be found. If the main CDC Publisher process terminates or is unavailable for connection, error messages are reported instead of a process state.

- **STATUS DETAIL**. Reports the status color indicator for each of the major PowerExchange CDC Publisher internal processes, including the Extract, Filter, Formatter, and Connector processes. The following example shows the status of the Extract internal process:

```
PARENT      CHILD      ATTRIBUTE      VALUE      DESCRIPTION
Extract_Connector  Extract  TRAFFIC LIGHT  GREEN
```

- **STATUS DETAIL VERBOSE**. Reports the status color indicator and a brief description of the status for each of the major PowerExchange CDC Publisher internal processes. The following example shows the verbose status information for the Extract internal process:

```
PARENT      CHILD      ATTRIBUTE      VALUE      DESCRIPTION
Extract_Connector  Extract  TRAFFIC LIGHT  GREEN      Similar to a traffic
light, a red value indicates an error, a green value indicates the process is
running as expected, and a yellow value indicates a mixed state (not an error and
not running as expected).
```

Note: VERBOSE can be specified only with the DETAIL parameter.

STATS={TOPIC|INTERNAL [VERBOSE]} [RESET]

Reports information about the topics that the CDC Publisher has updated or internal statistics and state information for PowerExchange CDC Publisher child processes (Extract, Formatter, and Connector). The statistics are cumulative since the Publisher process started. You can use the optional RESET parameter to refresh the topic information or reset the statistics counters to their original values.

Do not specify both the TOPIC and INTERNAL parameters.

You can enter the STATS parameter in any of following ways:

- **STATS=TOPIC.** Reports the following information for each topic that the CDC Publisher updated:
 - Updated topic name and partition ID
 - Number of messages sent to the topic
 - Partition offset returned
 - Source table name
 - Last change operation type and sequence ID
- **STATS=TOPIC RESET.** Clears the statistics for the updated topics.
- **STATS=INTERNAL.** Reports detailed statistics and state information, by internal attribute, for each type of PowerExchange CDC Publisher child process. This information is primarily for diagnostic use by Informatica Global Customer Support and other technical staff. The following example shows the report format:

PARENT	CHILD	ATTRIBUTE	VALUE	DESCRIPTION
Extract 2018	Extract Capture Consumer	InputRecordsRead		
Extract	Extract Capture Consumer	InputBytesRead	12560180	
Extract	Extract Capture Consumer	CurrentLoggerFile	/pwx/CONDENSE/Files.CND.CP200723.T0832001.	

- **STATS=INTERNAL VERBOSE.** Reports detailed statistics and state information, by internal attribute, for each of the major PowerExchange CDC Publisher internal processes with a description of the attribute. The following example shows the verbose report format:

PARENT	CHILD	ATTRIBUTE	VALUE	DESCRIPTION
Extract	Extract Capture Consumer	InputRecordsRead	2018	The current number of input records consumed by this process.
Extract	Extract Capture Consumer	InputBytesRead	12560180	The current number of bytes of input consumed by this process.

Note: Do not specify INTERNAL VERBOSE with the RESET parameter.

- **STATS=INTERNAL RESET.** Resets the internal attribute values to their initial values at CDC Publisher startup.

PwxCDCAdmin Utility - Command and Parameters

Use the PwxCDCAdmin utility to perform administrative functions for the PowerExchange CDC Publisher process.

With the PwxCDCAdmin command and its parameters, you can perform the following tasks:

- Report the Avro schemas, or format definitions, for all source tables or a selected source table for viewing or consumer application use.
- Clear the generated Avro schemas for source tables from cache. The CDC Publisher uses these schemas to format data in the messages to be sent to the target.
- Re-read extraction map definitions for all data sources from PowerExchange and cache any new and updated information.
- Shut down the PwxCDCPublisher process after it completes current work in progress.
- Validate custom pattern files.

Tip: You can use file-system security to limit rights on the PwxCDCAdmin utility script.

Syntax

```
PWXCDCADMIN  
[HELP|?]  
[INSTANCE=override_instance_directory]  
[SHUTDOWN]  
[CLEAR=FORMAT TABLE={ALL|mapname_tablename}]  
[RESET=FORMAT]  
[REPORT=FORMAT TABLE={ALL|mapname_tablename} [SUFFIX=.file_extension]  
[ENCRYPT=cleartext_password]  
[CHECKCUSTOMPATTERNS]
```

All parameters are optional. Use one or more spaces as the separator between parameters.

Parameter Descriptions

HELP or ?

Displays help information about the PwxCDCAdmin parameters.

INSTANCE=override_instance_subdirectory

Overrides the default name or location of the instance subdirectory, which is *pwxcdcpub_root installation\instanceA*. You can enter the fully qualified path of the subdirectory or a location that is relative to the root installation directory that is specified in the *PWPUB_HOME* environment variable. For example, the following values specify the same directory location if *PWPUB_HOME=C:\Informatica\PwxCdcPublisherInstall.1.0.0.windows.64bit*:

```
instanceB  
C:\Informatica\PwxCdcPublisherInstall.1.0.0.windows.64bit\instanceB
```

SHUTDOWN

Shuts down the PowerExchange CDC Publisher process cleanly. If messages are being written to topics when the shutdown request is made, the connection to the target remains open until all of the sent messages are acknowledged by the target as received. Then the CDC Publisher shuts down.

CLEAR=FORMAT TABLE={ALL|mapname_tablename}

Clears the Avro schemas, or format definitions, for all source tables or a selected source table from cache. Use this parameter if the Avro format definition for one or more source tables has changed and is no longer valid for CDC Publisher processing. When the CDC Publisher next receives change data for a changed source table, it regenerates the Avro schema. Alternatively, you can force the Avro schema to be regenerated by using the *REPORT=FORMAT* parameter.

The *TABLE* parameter is required and has no default value. Set the *TABLE* parameter to either *ALL* or a specific *mapname_tablename* value from an extraction map name.

RESET=FORMAT

Re-reads extraction map definitions for all data sources from PowerExchange and caches them. PowerExchange CDC Publisher uses the extraction map information to generate new Avro schemas when either change data is next received for the source tables associated with the extraction maps or when you issue the PwxCDCAdmin command with the *REPORT=FORMAT* parameter.

If you add a source table, you can use the *RESET=FORMAT* parameter to get the extraction map definition for the new table right away so that you can generate the Avro schema for verification purposes, before change data is received from the table. Alternatively, you can wait until change data from the new table is first received. At that point, the CDC Publisher caches the extraction map and generates the Avro schema.

When you run the PwxCDCAdmin command with this parameter, the PwxCDCAdmin task for the command times out after 5 minutes by default. If the task times out before command processing is finished, ask Informatica Global Customer Support to extend the timeout interval.

REPORT=FORMAT TABLE={ALL|*mapname_tablename*} [SUFFIX=*.file_extension*]

Reports the existing Avro schemas, or format definitions, for either all source tables or a selected source table in the `/pwxcdcpub_root installation/instanceX/reports` directory. These Avro schemas are produced in a legible format. The schemas can be used for validation purposes or by consumer applications. If no Avro schemas exist for one or more source tables, the utility generates the schemas based on the Formatter properties in the `cdcPublisherAvro.cfg` configuration file and cached extraction map definitions. If the extraction maps are not in cache, you can use the RESET parameter to re-read them from PowerExchange.

The TABLE parameter is required and has no default value. Set the TABLE parameter to either ALL or a specific *mapname_tablename* value from an extraction map name. If this value includes special characters, the CDC Publisher replaces the special characters with the replacement character that is specified in the `Formatter.avroNameReplacementValue` parameter. Avro does not support special characters.

Add the SUFFIX parameter to override the file name extension value specified in configuration parameter `Common.ReportSuffix`. For example, adding the following setting specifies that the report file extension of `.txt` should be used instead of the `Common.ReportSuffix` value:

```
SUFFIX=.txt
```

For more information, see [“Common Configuration Properties” on page 45](#).

Tip: To determine the *mapname_tablename* value, look up the extraction map name in the PowerExchange Navigator. The extraction map name has the format `xnninstance.mapname_tablename`. Use the *mapname_tablename* portion in the TABLE parameter. If the extraction map name does not include the *tablename*, change records have not yet been extracted based on the extraction map. For more information, see the *PowerExchange Navigator User Guide*.

ENCRYPT=*cleartext_password*

Encrypts a clear-text password so that you can enter an encrypted password in the `Extract.pwxXmapPwd` or `Extract.pwxNodePwd` property in the `cdcPublisherPowerExchange.cfg` configuration file to protect the password from unauthorized viewing. The `PwxCDCAdmin` utility generates and displays encrypted passwords in the following format:

```
ENC:encrypted_password
```

Where "ENC:" is the prefix that the CDC Publisher uses to identify an encrypted password value. When you enter the encrypted password in the `Extract.pwxXmapPwd` or `Extract.pwxNodePwd` property, include the ENC: prefix, for example:

```
Extract.pwxXmapPwd=ENC:AC/YB0tjDJDR2DBdAUXgqDQ==
```

Do not use this type of encryption for the `Extract.pwxNodeEPwd` property value. For this property, use the PowerExchange Navigator Encrypt Password facility to encrypt the password. For more information, see the *PowerExchange Navigator User Guide*.

CHECKCUSTOMPATTERNS

Checks the content of the custom patterns for validity and optionally generates sample output. You must set the INSTANCE parameter so the utility can get the following configuration properties from the `cdcPublisherAvro.cfg` file:

- `Formatter.customPatternFileList`. This property identifies the full path and file name of the file that contains the paths to the table-specific custom pattern files for the Custom Pattern Formatter.
- `Formatter.customPatternDataFile`. This property specifies the custom pattern data file. The custom pattern data file contains data that will be consumed as input and used to generate results based on the defined custom pattern files. You can use the output to validate that the custom pattern files generate the result you expect.

If you do not provide the custom pattern file, the utility will not run. The custom pattern data file is optional. If you do not specify the `Formatter.customPatternDataFile` property, only the keywords of the custom patterns are checked. If the custom pattern data file is not present, PowerExchange CDC Publisher does not perform the data substitution.

For more information, see [“ Custom Pattern Syntax Validation” on page 109](#).

APPENDIX B

Avro Schema Formats

This appendix includes the following topics:

- [Types of Avro Schema Formats, 90](#)
- [Avro Fields That Define a Source Column, 91](#)
- [Avro Flat Schema Format, 91](#)
- [Avro Nested Schema Format, 92](#)
- [Avro Generic Schema Format, 92](#)
- [Avro Wrapper Schema Format, 92](#)

Types of Avro Schema Formats

PowerExchange CDC Publisher generates an Avro schema for each source object. The Avro schema defines the structure of the messages to be sent to Kafka topics. Consumer applications must have a copy of the Avro schema to consume the Avro-formatted message content.

In the PowerExchange CDC Publisher `cdcPublisherAvro.cfg` configuration file, you specify the Avro schema format type in the `Formatter.avroSchemaFormat` parameter. Options are:

- `avroFlatSchemaFormatV1` - The "flat" schema
- `avroNestedSchemaFormatV1` - The "nested" schema
- `avroGenericSchemaFormatV1` - The "generic" schema

Also, you can "wrap" a flat, nested, or generic schema in a "wrapper" schema by setting the `Formatter.avroWrapperSchemaFormat` property to `avroWrapperSchemaFormatV1` in the `cdcPublisherAvro.cfg` configuration file. The wrapper schema defines three fields for the following metadata: the change sequence number, the source `mapname_tablename` value from the extraction map name, and the change operation type. The wrapper schema also includes a field for the "wrapped" child schema, which is expressed as a large string.

A generic or wrapper schema can be used for multiple source objects. Consider using a generic or wrapper schema when you want to send messages with change data from multiple source objects to a single Kafka topic. The topic is identified in the `Connector.kafkaTopic` property.

A flat or nested schema pertains to a specific source table. It defines the columns in the source table. Consider using a flat or nested schema when you want to send change data from a specific source object to the topic that is generated for that source object. In this case, set the `Connector.kafkaTopic` property to `USE_TABLE_NAME`.

This appendix shows the general structure of the flat, nested, generic, and wrapper formats. It also describes the four Avro fields that are used to describe each source column or field in an Avro flat or nested schema.

Avro Fields That Define a Source Column

In an Avro flat or nested schema, the following four fields define each source column:

- Column definition:

```
{ "name" : "COL1", "type" : [ "null", "string" ], "doc" : "...", "default" : null }
```

- Column value is-present indicator:

```
{ "name" : "COL1_Present", "type" : "boolean", "doc" : "Contains true when the value of field COL1 is present in the change data stream", "default" : false }
```

- Column before image:

```
{ "name" : "COL1_BeforeImage", "type" : [ "null", "string" ], "doc" : "Contains the before image of field COL1", "default" : null }
```

- Column before image is-present indicator:

```
{ "name" : "COL1_BeforeImage_Present", "type" : "boolean", "doc" : "Contains true when the value of field COL1_OLD is present.", "default" : false }
```

For example, two columns named "COL1" and "COL2" have the following Avro definitions:

```
{ "name" : "COL1", "type" : [ "null", "string" ], "doc" : "...", "default" : null }
{ "name" : "COL1_Present", "type" : "boolean", "doc" : ... }
{ "name" : "COL1_BeforeImage", "type" : [ "null", "string" ], "doc" : ... }
{ "name" : "COL1_BeforeImage_Present", "type" : "boolean", "doc" : ... }

{ "name" : "COL2", "type" : [ "null", "string" ], "doc" : "...", "default" : null }
{ "name" : "COL2_Present", "type" : "boolean", "doc" : ... }
{ "name" : "COL2_BeforeImage", "type" : [ "null", "string" ], "doc" : ... }
{ "name" : "COL2_BeforeImage_Present", "type" : "boolean", "doc" : ... }
```

Avro Flat Schema Format

An Avro schema that uses the flat format contains all of the Avro column-definition fields in one Avro record. The schema is specific to the source object that contains the defined columns.

The following flat schema defines two columns in one record:

```
{ "type" : "record", "fields" : [ {
  { "name" : "COL1", "type" : [ "null", "string" ], "doc" : "...", "default" : null }
  { "name" : "COL1_Present", "type" : "boolean", "doc" : ... }
  { "name" : "COL1_BeforeImage", "type" : [ "null", "string" ], "doc" : ... }
  { "name" : "COL1_BeforeImage_Present", "type" : "boolean", "doc" : ... }
  { "name" : "COL2", "type" : [ "null", "string" ], "doc" : "...", "default" : null }
  { "name" : "COL2_Present", "type" : "boolean", "doc" : ... }
  { "name" : "COL2_BeforeImage", "type" : [ "null", "string" ], "doc" : ... }
  { "name" : "COL2_BeforeImage_Present", "type" : "boolean", "doc" : ... }
} ] }
```

Avro Nested Schema Format

An Avro schema that uses the nested format has a main Avro record that contains a separate nested record for each type of Avro column-definition field. The schema is specific to the source object that contains the defined columns.

The following nested schema defines two columns in four records:

```
{ "name" : "columns" , "type" : "record", "fields" : [ {
  { "name" : "COL1", "type" : [ "null", "string" ], "doc" : "...", "default" : null}
  { "name" : "COL2", "type" : [ "null", "string" ], "doc" : "...", "default" : null}}]
{ "name" : "isPresent" , "type" : "record", "fields" : [ {
  { "name" : "COL1_Present", "type" : "boolean", "doc" : ...}
  { "name" : "COL2_Present", "type" : "boolean", "doc" : ...} } ]
{ "name" : "beforeImageValues" , "type" : "record", "fields" : [ {
  { "name" : "COL1_BeforeImage", "type" : [ "null", "string" ], "doc" : ... }
  { "name" : "COL2_BeforeImage", "type" : [ "null", "string" ], "doc" : ... } ]}
{ "name" : "isPresentIndicatorsForBeforeImages" , "type" : "record", "fields" : [ {
  { "name" : "COL1_BeforeImage_Present", "type" : "boolean", "doc" : ...}
```

Avro Generic Schema Format

An Avro schema that uses the generic format can be used for any source table. The schema represents the source columns as an array of fields for data and metadata. Most of the columns have the type "string" to be generic across multiple source table types. The column name and type is included in every Avro message along with the data.

A generic schema has the following format:

```
{ "name" : "columns",
  "type" : [ "null", {
    "type" : "array",
    "items" : {
      "type" : "record",
      "name" : "column",
      "doc" : "A column for the table",
      "fields" : [
        { "name" : "name", "type" : [ "null", "string" ], "default" : null},
        { "name" : "value", "type" : [ "null", "string" ], "default" : null},
        { "name" : "isPresent", "type" : [ "null", "boolean" ], "default" : null},
        { "name" : "beforeImage", "type" : [ "null", "string" ], "default" : null},
        { "name" : "isPresentBeforeImage", "type" : [ "null", "boolean" ], "default" :
null}
      ]
    }
  ]
}
```

Avro Wrapper Schema Format

An Avro "wrapper" schema wraps a flat, nested, or generic schema. The wrapper, or parent, schema defines three fields for metadata and one field for the wrapped child schema, which is expressed as a large string. The wrapped child schema has the format that is specified in the `Formatter.avroSchemaFormat` property.

To a consumer application, all messages that use the wrapper schema appear to have the same four-field format. The metadata fields are for the change sequence number, the source `mapname_tablename` value from the extraction map name, and the change operation type.

A single wrapper schema can be used to send messages that contain change data from different source objects to the target messaging system, using different Avro schema formats. When the consumer application reads the messages, it parses the underlying data and puts it in the Avro format designated in the wrapped child schema field.

As compared with the Avro generic schema format, a wrapper schema results in smaller messages because it adds only the source *mapname_tablename* in each message instead of both the column name and type. Also, the wrapper schema maintains the column data types rather than using the generic "string" type for all columns with data.

To process the data in the messages based on this schema format, the consumer application must parse the messages to get the source *mapname_tablename* and then find the Avro flat, nested, or generic schema that matches that name value by using their own methods.

A wrapper schema has the following format:

```
{
  "type" : "record",
  "name" : "InfaAvroWrapperSchema",
  "fields" : [
    {"name" : "INFA_SEQUENCE","type" : [ "null", "string" ], "default" : null},
    {"name" : "INFA_TABLE_NAME", "type" : [ "null", "string" ],"default" : null},
    {"name" : "INFA_OP_TYPE","type" : [ "null", "string" ],"default" : null},
    {"name" : "ChildSchema","type" : [ "null", "string" ],"default" : null}
  ]
}
```

APPENDIX C

Custom Pattern Formats

This appendix includes the following topics:

- [Custom Pattern Usage Considerations, 94](#)
- [Syntax of Table-Specific Custom Pattern Files, 94](#)
- [Formatter Properties and Custom Patterns, 104](#)
- [Custom JSON Examples, 104](#)
- [Custom JSON in Avro Format Example, 108](#)
- [Custom Pattern Syntax Validation, 109](#)

Custom Pattern Usage Considerations

Before you use custom patterns, review the following considerations:

- The Custom Pattern Formatter generates character string output. Do not use the Custom Pattern Formatter if the consumer application requires or expects a different format.
- Do not use the Custom Pattern Formatter if the consumer application requires binary-encoded data. The Custom Pattern Formatter does not generate binary-encoded output. This guideline is also true for Data Engineering Streaming (DES) because it cannot consume binary-encoded messages. For more information, see ["If Informatica Data Engineering Streaming Will Consume Data from a Target" on page 66](#).
- Do not use the Custom Pattern Formatter with the Confluent Schema Registry. Confluent Schema Registry requires the `Formatter.avroEncodingType` property set to `none`. The Custom Pattern Formatter is only compatible with `Formatter.avroEncodingType` set to `json`.

Syntax of Table-Specific Custom Pattern Files

The custom pattern files that you define for source tables must use tags that comply with the syntax described in this section.

Section Tags

Patterns are characterized by change data capture (CDC) event types. The following table describes the tags to specify the sections that define formatting for specific event types and other formatting attributes:

Tag	Description
<COLUMN_ATTRIBUTES>	Specifies how to process individual table columns during event formatting. For more information, see "Column Attributes" on page 95 .
<COMMIT_EVENT>	Specifies how to format a commit event. Note: If you set <code>Formatter.generateCommitDML</code> to <code>none</code> in the <code>cdcPublisherAvro.cfg</code> file, no commit events are produced in the message output. For other values, the commit DML is generated based on the value provided. You must enable the <code>Formatter.generateCommitDML</code> property so messages for transaction commit operations are generated. For more information, see "Formatter Configuration Properties" on page 45 .
<DELETE_EVENT>	Specifies how to format a delete event.
<INSERT_EVENT>	Specifies how to format an insert event.
<SPECIAL_ATTRIBUTES>	Specifies the special processing to use for pattern processing. For more information, see "Special Attribute Tags" on page 101 .
<UPDATE_EVENT>	Specifies how to format an update event

For each of these section tags, include a corresponding end tag that starts with a forward slash(/) to indicate the end of the section:

```
<COMMIT_EVENT>
...
</COMMIT_EVENT>
```

To generate messages, define at least one event type. Each section in the file is independent. You do not need to define every event type. For examples, see ["Custom JSON Examples" on page 104](#).

Comments

You can place comments anywhere in a custom pattern file. The number sign (#) must occur in column 1. For example:

```
# This is a comment
```

Column Attributes

Column attributes define how PowerExchange CDC Publisher processes individual columns. You can define columns to have a unique tag, exclude columns from processing, wrap column values with one or more characters based on the column data type, change the data type description of a column, or changing the format of date and time columns using masking.

PowerExchange CDC Publisher applies column attributes in the following order:

1. <TAG> Tag columns so that the name of the column can be used as a tag.
2. <EXCLUDE> Exclude columns from PowerExchange CDC Publisher generated message output.
3. <DT_SWAP> Change the data type descriptions of columns.
4. <DT_MASK> Change the date or time format using masks.

5. <DT_QUOTES> Delimit the data type values with a quotation mark or other character.

The following table describes the tags that you can use within the Column Attributes section of a custom pattern file.

Tag	Description
<DT_MASK>	<p>With date and time masking, you can change the date, time, and timestamp formats. Use one of the following syntax options:</p> <p>Option 1:</p> <pre data-bbox="711 491 1263 537"><DT_MASK><column_name>xxxxxxxx</column_name> <format>MM/dd/yyyy</format></DT_MASK></pre> <p>Where <i>xxxxxxxx</i> is the name of the column to mask. <i>MM/dd/yyyy</i> is the date format you want to use. The format is specified by the <format> tag. The format you specify for the column must be appropriate for the column data type.</p> <p>Option 2:</p> <pre data-bbox="711 709 1338 756"><DT_MASK><data_type>date time timestamp</data_type> <format>MM/dd/yyyy</format><DT_MASK></pre> <p>Where the data type is appropriate for the mask. All columns that have the specified data type are masked.</p> <p>Notes:</p> <ul data-bbox="711 852 1422 932" style="list-style-type: none"> - If the data type of the column does not permit the format you provide, the column is not masked. - Column masks take precedence over data type masks. <p>You can enter any character string that the Java class <code>DateTimeFormatter</code> supports for formatting dates and times. For more information, see the Java documentation.</p> <p>The input date formats that you can use are: PowerExchange Change Data Capture formatted Date, Time, or Timestamp.</p>
<DT_QUOTES>	<p>You can wrap column values with one or more characters, based on the data type of the column. Specify this tag for each data type where a value requires a double-quotation mark character.</p> <p>The following syntax wraps the value of the column with quotation mark characters:</p> <pre data-bbox="711 1268 1422 1314"><DT_QUOTES><data_type>string</data_type><quote_character>" </quote_character></DT_QUOTES></pre> <p>Where <quote_character> specifies the character that wraps the column value.</p> <p>Note: CDC Publisher does not validate the wrap character. The wrap character can be any character, or characters, you choose.</p> <p>By default, a data type that is not defined with the <DT_QUOTES> tag does not have column values that are wrapped by the wrap character. If needed, you can explicitly specify <DT_QUOTES> with an empty quote value, for example:</p> <pre data-bbox="711 1570 1422 1617"><DT_QUOTES><data_type>string</data_type><quote_character> </quote_character></DT_QUOTES></pre> <p>Typically, the following data types are quoted:</p> <ul data-bbox="711 1663 841 1789" style="list-style-type: none"> - String - Date - Time - Timestamp - Binary

Tag	Description
	<p>You can use the <quote_null_values> tag to indicate whether null column values are quoted. Valid values are:</p> <ul style="list-style-type: none"> - true. Null values are quoted based on the column data type and <DT_QUOTE> settings. - false. Null column values are never quoted. Data type and other <DT_QUOTE> settings are ignored. <p>Default is false. For example:</p> <pre><DT_QUOTES><quote_null_values>>false</quote_null_values> </DT_QUOTES></pre>
<DT_SWAP>	<p>Use this tag to change the data type description of a specific column or to change one data type description to another data type description for all columns.</p> <p>To change the data type description of a specific column, use the following syntax:</p> <pre><DT_SWAP><column_name>xxxxxxxx</column_name> <old_data_type>yyyyyyyy</old_data_type> <new_data_type>zzzzzzzz</new_data_type></DT_SWAP></pre> <p>Where <i>xxxxxxxx</i> is the name of the column, and <i>yyyyyyyy</i> is the data type description to be changed to the new data type description <i>zzzzzzzz</i>. The data type description you specify for the column must be appropriate for the data value of the column. The data type is not edited and can be any value you choose.</p> <p>To change a data type description to another data type description for all columns of a specific data type, use the following syntax:</p> <pre><DT_SWAP><old_data_type>yyyyyyyy</old_data_type> <new_data_type>zzzzzzzz</new_data_type></DT_SWAP></pre>

Tag	Description
<EXCLUDE>	<p>To exclude columns from PowerExchange CDC Publisher generated message output, use the following syntax:</p> <pre><EXCLUDE><column_name>xxxxxxx</column_name></EXCLUDE></pre> <p>Where xxxxxxxx is the name of the column to exclude.</p> <p>Wildcard characters are permitted at the beginning and end of column names. For example:</p> <ul style="list-style-type: none"> - A column name of FieldX_* excludes all columns with names that begin with "FieldX_". - A column name of INFA* excludes all columns with names that start with "INFA". - A column name of DTL* excludes all PowerExchange-generated columns with names that start with "DTL_". - A column name of *_field_Present excludes all columns that contain a suffix of "_field_Present".
<TAG>	<p>To tag a column so that the name of the column can be used as a tag, use the following syntax:</p> <pre><TAG><column_name>xxxxxxx</column_name></TAG></pre> <p>Where xxxxxxxx is the column name to use for the new tag. The column name must be a valid column name that is defined in the source data. After you tag the column, you can use the column name in a custom pattern.</p> <p>Note: If the specified column name does not appear in the table schema, an exception occurs at run time.</p>

<DT_SWAP> Example

The following example changes the data type Column1 description from string to "Int32".

```
<DT_SWAP><column_name>Column1</column_name><old_data_type>string</old_data_type>
<new_data_type>int32</new_data_type></DT_SWAP>
```

<EXCLUDE> Examples

The following examples show several ways to exclude specific columns from the message output.

To exclude columns with names that begin with "INFA":

```
<EXCLUDE><column_name>INFA*</column_name></EXCLUDE>
```

To exclude columns that are named "Column1":

```
<EXCLUDE><column_name>Column1</column_name></EXCLUDE>
```

To exclude columns with names that end in "_BeforeImage":

```
<column_name>*_BeforeImage</column_name></EXCLUDE>
```

<TAG> Example

The following example allows the tag <Column1> to be used in other patterns as a tag:

```
<TAG><column_name>Column1</column_name></TAG>
```

Event Tags

The Event sections identify the types of source DML events to include in the message output. The following table describes the tags that you can include in an event section:

Tag	Description	Events
<column_name>	The column name	Insert, Update, and Delete
<column_type>	The source column type	Insert, Update, and Delete
<column_value>	The column value	Insert, Update, and Delete
<column_isPresent>	Whether the column is present	Update
<column_beforeImage_isPresent>	Whether the column before image present	Update
<column_beforeImage_value>	The value for the before image	Update
<event_type>	The event type	All
<sequence>	The ID assigned to the PowerExchange sequence number	All
<table_name>	The table name	Insert, Update, Delete, and Commit
<table_name_type>	The data type of the table with the specified table name	Insert, Update, Delete, and Commit
<transaction_id>	The transaction identifier	Insert, Update, and Delete

Special Attribute Tags

The Special Attributes section defines how pattern processing handles special requirements for delimiters and null handling. The following table describes the tags that you can define in this section:

Tag	Description
<DELIMITER>	Defines the delimiter that is used in a custom pattern. The delimiter can be any number of characters. Use this tag in a pattern where a delimiter is needed. To place newline characters in the output, specify NEWLINE. By default, no delimiter is defined.
<NULL_REPLACEMENT_VALUE>	Defines the text to display for a null column value. Use this tag to specify how null column values are handled. For example, use this tag to specify that null values should be handled in the same manner as Avro-formatted JSON null values in repeat patterns. Note: If you do not specify this tag and produce JSON output, the JSON output is valid but not formatted in the same format as the Avro Formatter generates. By default, no text null replacements are performed.
<REPEAT_DELIMITER>	<REPEAT> tags are defined in the custom patterns file. The <REPEAT_DELIMITER> tag defines the delimiter that is used in the pattern when the processor encounters the ending </REPEAT> tag. The delimiter can be any number of characters. To place newline characters in the result, specify NEWLINE. Default value is a comma (,).

Special Attributes Section Example

In the following example, a semicolon (;) is used as a delimiter where the Custom Pattern Formatter specifies the <DELIMITER> tag and uses a comma(,) to separate repeated sections in a custom pattern:

```
<SPECIAL_ATTRIBUTES>
  <DELIMITER>;</DELIMITER>
  <REPEAT_DELIMITER>,</REPEAT_DELIMITER>
</SPECIAL_ATTRIBUTES>
```

Null Replacement Value Example

In the following example, null values will be handled in the same manner as Avro-formatted JSON values in repeat patterns. The <NULL_REPLACEMENT_VALUE> of "null" will be displayed for null column values. The pattern repeats for a series of columns in the table:

```
<SPECIAL_ATTRIBUTES>
  <DELIMITER>,</DELIMITER>
  <REPEAT_DELIMITER>,</REPEAT_DELIMITER>
  <NULL_REPLACEMENT_VALUE>null</NULL_REPLACEMENT_VALUE>
</SPECIAL_ATTRIBUTES>
```

Special Pattern Tags

The following special pattern tags can be specified for an event type to process a series of columns.

Tag	Description
<BOOLEAN_VALUE>	When a column type is boolean, ignore the pattern content between the start and end <BOOLEAN_VALUE> tags and produce the boolean true or false value for the column.
<NULLABLE_VALUE>	When value of a column is null, ignore the pattern content between the start and end <NULLABLE_VALUE> tags and produce a null value. This null value is a value that matches the <NULL_REPLACEMENT_VALUE> attribute.
<REPEAT>	Repeat the contents between the start and end <REPEAT> tags for all columns in the object schema. For example, assume you have a table with 1,000 columns. It would be time-consuming to define patterns for each column. If you use the REPEAT tag, you can repeat the pattern between the start and end <REPEAT> tags for the 1,000 columns in the table. At the end of the repeat segment, the delimiter character defined in the Special Attributes section by the <REPEAT_DELIMITER> tag is placed where the </REPEAT> tag is found in the pattern.

<BOOLEAN_VALUE> Example

Assume the custom pattern has the following format:

```
"<column_name>":{"<column_type>":<column_value>}
```

The following values are generated for COL1="a" and COL1_Present="true":

```
"COL1":{"string":"a"}  
"COL1_Present":{"boolean":true}
```

To match JSON standards, you want the output to display the value of "true" for the boolean COL1_PRESENT column:

```
"COL1":{"string":"a"}  
"COL1_Present":true
```

To do so, use the <BOOLEAN_VALUE> tag in the custom pattern:

```
"<column_name>":<BOOLEAN_VALUE>{"<column_type>":<column_value>}</BOOLEAN_VALUE>
```

<NULLABLE_VALUE> Examples

Assume the custom pattern has the following format:

```
"<column_name>":{"<column_type>":<column_value>}
```

The following values are generated for COL1="a" and COL2=null:

```
"COL1":{"string":"a"}  
"COL2":{"string":null}
```

To match JSON standards, you want to generate the following values:

```
"COL1":{"string":"a"}  
"COL2":null
```

If you want null values to replace a section of the pattern, specify the `<NULLABLE_VALUE>` tag. For example:

```
<column_name>":<NULLABLE_VALUE>{"<column_type>":<column_value>}</NULLABLE_VALUE>
```

Boolean and Null Values Example

You can use both `<BOOLEAN_VALUE>` and `<NULLABLE_VALUE>` tags in the same custom pattern to generate different text for boolean and null values.

If you specify the following values:

```
COL1="a", COL1_Present="true", COL2=null
```

And specify the following custom pattern:

```
"<column_name>":<NULLABLE_VALUE><BOOLEAN_VALUE>{"<column_type>":<column_value>}</BOOLEAN_VALUE></NULLABLE_VALUE>
```

The following values are then generated in the output:

```
"COL1":{"string":"a"}
"COL1_Present":true
"COL2":null
```

Repeat Tag Examples

The following examples show the optimal placement of `<REPEAT>` tags within an event tag.

Insert Event with Repeat Tag Example

The following example shows an insert event that contains a `<REPEAT>` group of column attributes, which is repeated for each column in the table:

```
<INSERT_EVENT>
  <event_type>
  <table_name>
  <transaction_id>
  <sequence>
  < REPEAT >          <- repeats for each column in the table
    <column_name>
    <column_type>
    <column_value>
  </REPEAT>
</INSERT_EVENT>
```

Update Event with Repeat Tag Example

The following example shows an update event that contains a `<REPEAT>` group of column attributes, which is repeated for each column in the table:

```
<UPDATE_EVENT>
  <event_type>
  <table_name>
  <transaction_id>
  <sequence>
  <REPEAT>            <- repeats for each column in the table
    <column_name>
    <column_type>
    <column_value>
    <column_beforeImage_value>
    <column_beforeImage_isPresent>
  </REPEAT>
</UPDATE_EVENT>
```

Delete Event with Repeat Tag Example

The following example shows a delete event that contains a <REPEAT> group of column attributes, which is repeated for each column in the table:

```
<DELETE_EVENT>
  <event_type>
  <table_name>
  <transaction_id>
  <sequence>
  < REPEAT >          <- repeats for each column in the table
    <column_name>
    <column_type>
    <column_value>
  </REPEAT>
</DELETE_EVENT>
```

Formatter Properties and Custom Patterns

The standard Formatter properties that affect custom pattern processing have a global effect on all custom patterns. The Formatter processes Avro properties before custom patterns.

The following Formatter properties affect custom patterns:

- **Formatter.avroSchemaFormat.** The only valid value when using custom patterns is avroFlatSchemaFormatV1.
- **Formatter.generateCommit DML.** If you use the <COMMIT_EVENT> tag to generate output for commit operations, you must set this property to either ALL_TABLES or LAST_TABLE. For more information, see [“Formatter Configuration Properties” on page 45.](#)

Custom JSON Examples

These custom patterns show examples of a simple JSON view of a table for insert, update, and delete CDC events, patterns that format JSON for each event type, and a JSON example that uses multiple tag types.

Tip: After you create a JSON pattern, you can validate the JSON pattern using a JSON validator such as [jsonlint.com](#). Copy only the pattern into the validator and then test it. Do not copy the section tags. Replace the keywords with test data. Repeat the test for each pattern you want to validate.

Simple JSON Example

Company XYZ wants a simple JSON view of insert, update, and delete events on the My.Table table. The message format should contain only column names as keys with their associated values for each of the event types. The table columns are named AA_field, AB_field, and AC_field.

The <REPEAT_DELIMITER> tag places the delimiter comma between the repeating group of items in the message. The last repeated group has the delimiter removed.

The custom pattern file for My.Table contains the following specifications:

```
# Special attribute that will place a comma between all of the repeating group items
<SPECIAL_ATTRIBUTES>
  <REPEAT_DELIMITER>,</REPEAT_DELIMITER>
```



```

</SPECIAL_ATTRIBUTES>

# Exclude Informatica generated columns
<COLUMN_ATTRIBUTES>
  <EXCLUDE> <column_name>      INFA*                </column_name> </EXCLUDE>
  <EXCLUDE> <column_name>      DTL*                </column_name> </EXCLUDE>
  <EXCLUDE> <column_name>      *_Present            </column_name> </EXCLUDE>
  <EXCLUDE> <column_name>      *_BeforeImage         </column_name> </EXCLUDE>
  <EXCLUDE> <column_name>      *_BeforeImage_Present  </column_name> </EXCLUDE>
</COLUMN_ATTRIBUTES>

# For insert events
<INSERT_EVENT>
{"evnt": "<event_type>", <REPEAT>"<column_name>": "<column_value>"</REPEAT>}
</INSERT_EVENT>

#For update events
<UPDATE_EVENT>
{"evnt": "<event_type>", <REPEAT>"<column_name>": "<column_value>"</REPEAT>}
</UPDATE_EVENT>

#For delete events
<DELETE_EVENT>
{"evnt": "<event_type>", <REPEAT>"<column_name>": "<column_value>"</REPEAT>}
</DELETE_EVENT>

```

Insert Event Example

The following example shows a pattern that formats JSON for an insert event:

```

<INSERT_EVENT>
  {
    "Event" : "<event_type>" ,
    "table_name" : "<table_name>" ,
    "transaction" : "<transaction_id>" ,
    "sequence" : "<sequence>" ,
  },
  "column_Information" ; [
  < REPEAT >
    {
      "column_name" : "<column_name>" ,
      "column_type" : "<column_type>" ,
      "column_value" : "<column_value>"
    }
  </REPEAT>
  ] }
</INSERT_EVENT>

```

Update Event Example

The following example shows a pattern that formats JSON for an update event:

```

<UPDATE_EVENT>
  {
    "Event" : "<event_type>" ,
    "table_name" : "<table_name>" ,
    "transaction" : "<transaction_id>" ,
    "sequence" : "<sequence>" ,
    "column_information" : [
    <REPEAT>
      {
        "column_name" : "<column_name>" ,
        "column_type" : "<column_type>" ,
        "column_value" : "<column_value>" ,
        "before_image_value" : "<column_beforeImage_value>" ,
        "column_before_image_present" : "<column_beforeImage_isPresent>"
      }
    </REPEAT>
  ]
}

```

```
    ] }
  </UPDATE_EVENT>
```

Delete Event Example

The following example shows a pattern that formats JSON for a delete event:

```
<DELETE_EVENT>
  {
    "Event" : "<event_type>" ,
    "table_name" : "<table_name>" ,
    "transaction" : "<transaction_id>" ,
    "sequence" : "<sequence>" ,
    "column_information" : [
      < REPEAT >
        {
          "column_name" : "<column_name>" ,
          "column_type" : "<column_type>" ,
          "column_value" : <column_value>" ,
        }
      </REPEAT>
    ] }
  "</DELETE_EVENT>" ,
```

Commit Event Example

The following example shows a pattern that formats JSON for a commit event:

```
<COMMIT_EVENT>
  {
    "Event" : "<event_type>"
    "transaction" : "<transaction_id>"
  }
</COMMIT_EVENT>
```

JSON Example that Uses Multiple Tag Types

This example produces a JSON format for insert and delete events for an Oracle object. The result will be a condensed JSON string that contains the following information:

- The event types are of insert and delete only.
- The num1 and num2 columns of the object have a number data type.
- A column that contains the creation date for the change record in the format yyyyMMdd.

The table is defined by the following Oracle DDL:

```
CREATE TABLE "XXX"."NUMTBL3"
("num1" NUMBER(2) NOT NULL ENABLE,
 "num2" NUMBER(10),
 PRIMARY KEY ("num1") ...
```

In the PowerExchange CDC capture registrations, the table name is numtbl3_NUMTBL3.

In the custom pattern, which is defined for numtbl3_NUMTBL3, the following tags define column attribute processing:

- Use the <EXCLUDE> tag to exclude all columns except for DTL__CAPXTIMESTAMP, which is the source creation timestamp for the row.
- Use the <DT_SWAP> tag to define the following processing:
 - Convert the DTL__CAPXTIMESTAMP column, which is a "long" data type from PowerExchange CDC to a date format.
 - Swap the num2 column from a string data type to an int32 data type.
- Use the <DT_MASK> tag to convert DTL__CAPXTIMESTAMP into the format yyyyMMdd.

- Use the <DT_QUOTES> tag to define the following processing:
 - Explicitly state that any value for data type int32 will not have its value wrapped in double-quotation marks.
 - The data type string will have its value quoted with a double-quotation mark.
 - The data type date will have its value quoted with a double-quotation mark.
 - The data type binary will have its value quoted with a double-quotation mark. Even though there are no binary items produced, it is safe to include them because they are ignored in this case.

Note: The <data_value> tag is not quoted in the syntax of the insert and delete event.

```

<SPECIAL_ATTRIBUTES>
<DELIMITER>NEWLINE</DELIMITER>
<REPEAT_DELIMITER>, NEWLINE</REPEAT_DELIMITER>
</SPECIAL_ATTRIBUTES>

<COLUMN_ATTRIBUTES>
# exclude columns
<EXCLUDE><column_name>* Present</column_name></EXCLUDE>
<EXCLUDE><column_name>* BeforeImage</column_name></EXCLUDE>
<EXCLUDE><column_name>* BeforeImage Present</column_name></EXCLUDE>
<EXCLUDE><column_name>DTL_CAPXACTION</column_name></EXCLUDE>
<EXCLUDE><column_name>DTL_CAPXRESTART1</column_name></EXCLUDE>
<EXCLUDE><column_name>DTL_CAPXRESTART2</column_name></EXCLUDE>
<EXCLUDE><column_name>DTL_CAPXROWID</column_name></EXCLUDE>
<EXCLUDE><column_name>DTL_CAPXUOW</column_name></EXCLUDE>
<EXCLUDE><column_name>DTL_CAPXUSER</column_name></EXCLUDE>
<EXCLUDE><column_name>INFA_SEQUENCE</column_name></EXCLUDE>
<EXCLUDE><column_name>INFA_OP_TYPE</column_name></EXCLUDE>
<EXCLUDE><column_name>INFA_TABLE_NAME</column_name></EXCLUDE>
# swap data types
<DT_SWAP><column_name>DTL_CAPXTIMESTAMP</column_name><old_data_type>long</
old_data_type><new_data_type>date</new_data_type></DT_SWAP>
<DT_SWAP><column_name>num2</column_name><old_data_type>string</old_data_type><new_data_type>int32</
new_data_type></DT_SWAP>
<DT_SWAP><old_data_type>integer</old_data_type><new_data_type>int32</new_data_type></DT_SWAP>
# masks
<DT_MASK><column_name>DTL_CAPXTIMESTAMP</column_name><format>yyyyMMdd</format></DT_MASK>
# dont use quotes on these data types
<DT_QUOTES><data_type>int32</data_type><quote_character></quote_character></DT_QUOTES>
# use quotes on these data types
<DT_QUOTES><data_type>string</data_type><quote_character>"</quote_character></DT_QUOTES>
<DT_QUOTES><data_type>date</data_type><quote_character>"</quote_character></DT_QUOTES>
<DT_QUOTES><data_type>binary</data_type><quote_character>"</quote_character></DT_QUOTES>
</COLUMN_ATTRIBUTES>

<INSERT_EVENT>
{ "event" : "insert", "columns" : [ <delimiter>
<REPEAT> { "column" : "<column_name>", "<column_type>" : <column_value> } </REPEAT>
]
}
</INSERT_EVENT>

<DELETE_EVENT>
{
"event" : "delete", "columns" : [ <delimiter>
<REPEAT> { "column" : "<column_name>", "<column_type>" : <column_value> } </REPEAT>
]
}
</DELETE_EVENT>

```

The following example shows the output for an insert event that has columns num1 = 27 and column num2 = 270:

```

{ "event" : "insert", "columns" : [
{ "column" : "DTL_CAPXTIMESTAMP", "date" : "20200812" } ,
{ "column" : "num1", "int32" : 27 } ,
{ "column" : "num2", "int32" : 270 } ] ]

```

Custom JSON in Avro Format Example

You can use the Custom Pattern Formatter to customize your existing JSON or generate the same JSON output in Avro format.

For example, the following custom pattern file generates JSON output in Avro format:

```
# The following pattern is an example of using the Custom Pattern Formatter
# to generate the same JSON output as the Avro Formatter.
#
#-----
# Special Attributes Section
#-----

<SPECIAL_ATTRIBUTES>
  <DELIMITER>,</DELIMITER>
  <REPEAT_DELIMITER>,</REPEAT_DELIMITER>
  # Handling null values like Avro does - this is the text displayed for a null column value.
  <NULL_REPLACEMENT_VALUE>null</NULL_REPLACEMENT_VALUE>
</SPECIAL_ATTRIBUTES>

#-----
# Column Attributes Section
#-----

<COLUMN_ATTRIBUTES>

# Tagging columns <TAG>

# Excluding columns <EXCLUDE>

# Data Type Swapping <DT_SWAP>

# Date/Time masking <DT_MASK>

# Data type value quoting <DT_QUOTES>
<DT_QUOTES> <data_type>string</data_type> <quote_character>"</quote_character> </DT_QUOTES>
<DT_QUOTES> <data_type>date</data_type> <quote_character>"</quote_character> </DT_QUOTES>
<DT_QUOTES> <data_type>time</data_type> <quote_character>"</quote_character> </DT_QUOTES>
<DT_QUOTES> <data_type>timestamp</data_type> <quote_character>"</quote_character> </DT_QUOTES>
<DT_QUOTES> <data_type>bytes</data_type> <quote_character>"</quote_character> </DT_QUOTES>

# Enabled or disable allowing quotes around null values. Just explicitly disabled here for example
that it can be changed
<DT_QUOTES> <quote_null_values>>false</quote_null_values> </DT_QUOTES>

</COLUMN_ATTRIBUTES>

#-----
# Insert Event Section
#-----
<INSERT_EVENT>
{
  <REPEAT>
  "<column_name>":<NULLABLE_VALUE><BOOLEAN_VALUE>{"<column_type>":<column_value>}</BOOLEAN_VALUE>
  </NULLABLE_VALUE>
  </REPEAT>
}
</INSERT_EVENT>

#-----
# Update Event Section
#-----
<UPDATE_EVENT>
{
  <REPEAT>
  "<column_name>":<NULLABLE_VALUE><BOOLEAN_VALUE>{"<column_type>":<column_value>}</BOOLEAN_VALUE>
  </NULLABLE_VALUE>
  </REPEAT>
}
</UPDATE_EVENT>

#-----
```

```

# Delete Event Section
#-----
<DELETE_EVENT>
{
<REPEAT>
"<column_name>":<NULLABLE_VALUE><BOOLEAN_VALUE>{"<column_type>":<column_value>}</BOOLEAN_VALUE>
</NULLABLE_VALUE>
</REPEAT>
}
</DELETE_EVENT>

#-----
# Commit Event Section (not used)
#-----
#<COMMIT_EVENT>
#</COMMIT_EVENT>

```

Custom Pattern Syntax Validation

You can use the PwxCDCAdmin utility to validate custom pattern syntax without sending data to the target.

To validate custom pattern syntax, specify the CHECKCUSTOMPATTERNS and INSTANCE parameters in the utility command. The utility uses the INSTANCE value to get the location of the custom pattern file. For more information, see ["Custom Pattern Formatter Configuration Properties" on page 51](#).

The validation process checks the custom pattern syntax. It does not validate the structure or format of the output. For example, if you define custom patterns to create JSON output, the validation process does not validate the output to ensure it meets JSON syntax requirements. The validation process only validates the custom pattern tags and parameters that it recognizes.

If you want to simulate the message output from CDC Publisher, create a custom pattern data file. This file is optional. The custom pattern data file contains table definitions and the data for each table definition. It should also include a default table definition. The validation process uses the default table definition if it cannot find a custom pattern file for a specific table in the custom pattern data file. The validation process checks input values against the patterns and generates output based on the patterns. Specify the location of the custom pattern data file using the Formatter.customPatternDataFile property in the cdcPublisherAvro.cfg file. For more information, see ["Custom Pattern Formatter Configuration Properties" on page 51](#).

Note: The processor generates Informatica and PowerExchange-generated columns to simulate and check the custom patterns column attributes section if these columns are included in the custom pattern file for a table.

Custom Pattern Syntax Validation Example

In this example, the following Custom Pattern Formatter configuration properties are set in the cdcPublisherAvro.cfg file:

```

Formatter.customPatternFileList=C:\\CDCPUB_Install\\Instance141\\config\\patterns.list
Formatter.customPatternDataFile=C:\\CDCPUB_Install\\Instance141\\config\\patternTestData.dat

```

The patternTestData.dat custom pattern data file contains the following definitions for two specific tables and the default table:

```

[
# It has 3 data columns, but 1 of them is commented.
# This is the default definition used if a table appears that does not have a specific definition
# in this file.
table name = DEFAULT,
column_name = "Column_1", column_type = "string", column_value = "This is column 1 data",
# column_name = "Column_2", column_type = "string", column_value = "This is column 2 data",
column_name = "Column_3", column_type = "integer", column_value = 22

```

```

]
# definition for tableA
[
  table_name = tableA,
  column_name = "AColumn_1",
  column_type = "string",
  column_value = "This is column 1 data for tableA",
  column_name = "AColumn_2",
  column_type = "string",
  column_value = "This is column 2 data for tableA"
]
# definition for tableB
[
  table_name = tableB,
  column_name = "B_1",
  column_type = "string",
  column_value = "This is B 1 data for tableB",
  column_name = "B_2",
  column_type = "integer",
  column_value = 999
]
]

```

The patterns.list Custom Pattern Formatter configuration file contains the following specifications:

```

<table>tableA</table><path>C:\CDCPUB_Install\Instance141\config\tableA.pattern</path>
<table>DEFAULT</table><path>C:\CDCPUB_Install\Instance141\config\generic.pattern</path>

```

The tableA.pattern custom pattern file contains the following specifications:

```

<SPECIAL_ATTRIBUTES>
  <DELIMITER>,</DELIMITER>
  <REPEAT_DELIMITER>,</REPEAT_DELIMITER>
</SPECIAL_ATTRIBUTES>

<COLUMN_ATTRIBUTES>
  <EXCLUDE><column_name> INFA* </column_name></EXCLUDE>
  <EXCLUDE><column_name> DTL* </column_name></EXCLUDE>
  <EXCLUDE><column_name> *_Present </column_name></EXCLUDE>
  <EXCLUDE><column_name> *_BeforeImage </column_name></EXCLUDE>
  <EXCLUDE><column_name> *_BeforeImage_Present </column_name></EXCLUDE>
</COLUMN_ATTRIBUTES>

# this is for insert events
<INSERT_EVENT>
{"evnt": "<event_type>",<REPEAT>"<column_name>": "<column_value>"</REPEAT>}
</INSERT_EVENT>

# this is for update events
<UPDATE_EVENT>
{"evnt": "<event_type>",<REPEAT>"<column_name>": "<column_value>"</REPEAT>}
</UPDATE_EVENT>

# this is for delete events
<DELETE_EVENT>
{"evnt": "<event_type>",<REPEAT>"<column_name>": "<column_value>"</REPEAT>}
</DELETE_EVENT>

```

The generic.pattern custom pattern file for the DEFAULT table pattern contains the following specifications:

```

<SPECIAL_ATTRIBUTES>
  <DELIMITER>,</DELIMITER>
  <REPEAT_DELIMITER>,</REPEAT_DELIMITER>
</SPECIAL_ATTRIBUTES>

<COLUMN_ATTRIBUTES>
  <EXCLUDE><column_name> INFA* </column_name></EXCLUDE>
  <EXCLUDE><column_name> DTL* </column_name></EXCLUDE>
  <EXCLUDE><column_name> *_Present </column_name></EXCLUDE>
  <EXCLUDE><column_name> *_BeforeImage </column_name></EXCLUDE>
  <EXCLUDE><column_name> *_BeforeImage_Present </column_name></EXCLUDE>
</COLUMN_ATTRIBUTES>

```

```

# this is for insert events
<INSERT_EVENT>
{"evnt":"<event_type>", "<transaction_id>", "<sequence>"
<REPEAT>"<column_name>":"<column_value>"</REPEAT>}
</INSERT_EVENT>

# this is for update events
<UPDATE_EVENT>
{"evnt":"<event_type>", "<transaction_id>" : "<sequence>",
<REPEAT>"<column_name>":"<column_value>"</REPEAT>}
</UPDATE_EVENT>

# this is for delete events
<DELETE_EVENT>
{"evnt":"<event_type>", "<transaction_id>" : "<sequence>",
<REPEAT>"<column_name>":"<column_value>"</REPEAT>}
</DELETE_EVENT>

<COMMIT_EVENT>
{ "COMMIT" : "sequence" : "<sequence>", "transaction" : "<transaction_id>" }
</COMMIT_EVENT>

```

Example Output

The PwxCDCAdmin utility generated the following example output based on two table-specific custom patterns and the DEFAULT custom pattern:

```

-----
Checking the custom patterns...
Table tableA file C:\CDCPUB_Install\Instance141\config\tableA.cfg is processed
Table DEFAULT file C:\CDCPUB_Install\Instance141\config\generic.cfg is processed
-----
Custom Pattern Tables List:
  tableA
  DEFAULT
-----
Custom Pattern Data file in use is C:/CDCPUB_Install/Instance141/config/
patternTestData.dat
Start Custom Data Patterns Processing...
-----
Custom Pattern Data Tables List:

Table: tableB
  Column Name: DTL__CAPXACTION
    Column Type: string
    Column Value: dtlaction
  Column Name: DTL__CAPXRESTART1
    Column Type: string
    Column Value: restart1ABCDEF
  Column Name: DTL__CAPXRESTART2
    Column Type: string
    Column Value: restart2GHIJK
  Column Name: DTL__CAPXUSER
    Column Type: string
    Column Value: capxUser
  Column Name: DTL__CAPXUOW
    Column Type: string
    Column Value: UOW 1
  Column Name: DTL__CAPXTIMESTAMP
    Column Type: timestamp
    Column Value: 202009251013370005050000
  Column Name: DTL__CAPXROWID
    Column Type: integer
    Column Value: 3
  Column Name: B_1
    Column Type: string

```

Column Value: This is B 1 data for tableB
Column Name: B_2
Column Type: integer
Column Value: 999

Table: tableA

Column Name: DTL__CAPXACTION
Column Type: string
Column Value: dtlaction
Column Name: DTL__CAPXRESTART1
Column Type: string
Column Value: restart1ABCDEF
Column Name: DTL__CAPXRESTART2
Column Type: string
Column Value: restart2GHIJK
Column Name: DTL__CAPXUSER
Column Type: string
Column Value: capxUser
Column Name: DTL__CAPXUOW
Column Type: string
Column Value: UOW_1
Column Name: DTL__CAPXTIMESTAMP
Column Type: timestamp
Column Value: 202009251013370005050000
Column Name: DTL__CAPXROWID
Column Type: integer
Column Value: 2
Column Name: AColumn_1
Column Type: string
Column Value: This is column 1 data for tableA
Column Name: AColumn_2
Column Type: string
Column Value: This is column 2 data for tableA

Table: DEFAULT

Column Name: DTL__CAPXACTION
Column Type: string
Column Value: dtlaction
Column Name: DTL__CAPXRESTART1
Column Type: string
Column Value: restart1ABCDEF
Column Name: DTL__CAPXRESTART2
Column Type: string
Column Value: restart2GHIJK
Column Name: DTL__CAPXUSER
Column Type: string
Column Value: capxUser
Column Name: DTL__CAPXUOW
Column Type: string
Column Value: UOW_1
Column Name: DTL__CAPXTIMESTAMP
Column Type: timestamp
Column Value: 202009251013370005040000
Column Name: DTL__CAPXROWID
Column Type: integer
Column Value: 1
Column Name: Column_1
Column Type: string
Column Value: This is column 1 data
Column Name: Column_3
Column Type: integer
Column Value: 22

Table tableB will be processed using the DEFAULT custom pattern.

Table tableB sample INSERT_EVENT:


```
{"evnt":"INSERT_EVENT" , "table_name" : "tableB", "trandid" : "102", "sequence" :
1,2,SomeObject,3,4,5" "B_1":"This is B 1 data for tableB","B_2":"999"}
```

```
Table tableB sample UPDATE_EVENT:
{"evnt":"UPDATE_EVENT", "table_name" : "tableB", "trandid" : "102", "sequence" :
1,2,SomeObject,3,4,5", "B_1":"This is B 1 data for tableB","B_2":"999"}
```

```
Table tableB sample DELETE_EVENT:
{"evnt":"DELETE_EVENT", "table_name" : "tableB", "trandid" : "102", "sequence" :
"1,2,SomeObject,3,4,5", "B_1":"This is B 1 data for tableB","B_2":"999"}
```

```
Table tableB sample COMMIT_EVENT:
{ "COMMIT" : "sequence" : "1,2,SomeObject,3,4,5", "trandid" : : "102" }
```

```
-----
Table tableA will be processed using a defined custom pattern.
-----
```

```
Table tableA sample INSERT_EVENT:
{"evnt":"INSERT_EVENT" , "AColumn_1":"This is column 1 data for
tableA","AColumn_2":"This is column 2 data for tableA"}
```

```
Table tableA sample UPDATE_EVENT:
{"evnt":"UPDATE_EVENT", "AColumn_1":"This is column 1 data for
tableA","AColumn_2":"This is column 2 data for tableA"}
```

```
Table tableA sample DELETE_EVENT:
{"evnt":"DELETE_EVENT", "AColumn_1":"This is column 1 data for tableA","AColumn_2":"This
is column 2 data for tableA"}
```

```
-----
Custom pattern validation succeeded
```

Custom Pattern Data File

Use the table definition syntax and parameters to specify data for the custom pattern data file.

Table Definition Syntax

To define tables in the custom pattern data file, use the following table definition syntax:

```
# a specific table definition
[
  table_name = tableName1,

  column_name = columnName1,
  column_type = columnType1,
  column_value = columnValue1,
  ...
  column_name = columnNameN,
  column_type = columnTypeN,
  column_value = columnValueN
]
```

The `table_name`, `column_name`, `column_type`, and `column_value` entries can be optionally enclosed in quotation marks.

Include a default table definition that can be used if a definition for a specific table is not in the file. To define a default table definition, use the following syntax:

```
# a DEFAULT table definition. Will be used if no specific table definition is found for
# a given file in the custom pattern file.
[
  table_name = DEFAULT,
  column_name = columnName1,
  column_type = columnType1,
```

```

        column_value = columnValue1,
    ...
        column_name = columnNameN,
        column_type = columnTypeN,
        column_value = columnValueN
    ]

```

Begin a comment line with the number (#) character.

Parameters

Use the following parameters for the custom pattern data file content:

table_name

Required. The fully-qualified table name of the source table. This parameter is required for each table that requires its own data definition. The validation process uses a table name of DEFAULT if a custom pattern file for a specific table is not found in the custom pattern data file.

column_name

Required. The column name to use in the output. These column names are used by the Custom Pattern Formatter to replace the <column_name> references in the custom pattern file. The number of columns that are defined determine the number of columns that appear in the substitution within the custom pattern.

column_type

Required. The column data type. This value is used to substitute column_type tags in the custom pattern. Valid values are:

boolean

For example, provide a value of true or false:

```
column_name = MY_STRING_Present, column_type = boolean, column_value = "true"
```

date

For example, provide a value of 8 or 14 characters that matches one of the following masks: "YYYYMMDD" or "YYYYMMDDHHMMSS":

```
column_name = MY_DATE, column_type = date, column_value = "20191231"
```

integer

For example, provide any integer value:

```
column_name = MY_INT, column_type = integer, column_value = 123456
```

string

For example, provide any string value:

```
column_name = MY_STRING, column_type = string, column_value = "any quoted value"
```

time

For example, provide a value of six characters that matches the mask: "HHMMSS":

```
column_name = MY_TIME, column_type = time, column_value = "172859"
```

timestamp

For example, provide a value of 24 characters that matches the mask: "YYYYMMDDHHMMSSffffffff":

```
column_name = MY_TIMESTAMP, column_type = timestamp, column_value =
"201912311728590000000000"
```

Note: You must include all 10 fractional seconds.

To represent null column values, use the keyword "null" as the value with or without quotation marks. For example:

```
column_name = MY_STRING_BeforeImage, column_type = string, column_value = "null"
```

Note: Timestamp values must be 14 bytes in length without fractional seconds or 24 bytes in length with fractional seconds.

column_value

Required. The column data value. This value should correspond with the specified column type.

include_default_metadata_columns

Optional. Controls whether or not default values for metadata columns are included. Valid values are:

- **true.** Default values are provided for metadata columns that have the INFA* and DTL_* prefixes. You do not need to provide the column name, type, or value.
- **false.** Default values are not provided for these metadata columns. You must specify the column name, type, and value for the metadata columns you want to test in the custom pattern data file.

Note: If you set `include_default_metadata_columns = true` and also provide values in the custom data pattern file for the same default metadata columns, unpredictable results might occur.

Default is true.

In the following example, this parameter is set to true:

```
#example definition for tableA
[include_default_metadata_columns = true,
 table_name = tableA,
 ...
```

With this setting, the following example metadata values are generated:

```
Column Name: DTL__CAPXACTION
Column Type: string
Column Value: dtlaction
Column Name: DTL__CAPXRESTART1
Column Type: string
Column Value: restart1ABCDEF
Column Name: DTL__CAPXRESTART2
Column Type: string
Column Value: restart2GHIJK
```

In the following example, this parameter is set to false and you must provide the metadata column information:

```
[include_default_metadata_columns = false,
 table_name = tableA,
 column_name = DTL__CAPXUSER, column_type = string, column_value = "USERID",
 column_name = "AColumn_1", column_type = "string", column_value = "This is column 1 data for tableA",
 column_name = "AColumn_2", column_type = "string", column_value = "This is column 2 data for tableA"
]
```

APPENDIX D

Message Reference

This appendix includes the following topics:

- [Message Reference Overview, 116](#)
- [CDCPUB_10001 to CDCPUB_16001, 117](#)
- [CPPSVC_00100 to CPPSVC_00103, 149](#)
- [LOGCFG_10001 to LOGCFG_10006, 150](#)
- [PWXLIC_00110 to PWXLIC-00999, 151](#)
- [PWXRDR-01000 to PWXRDR-01301, 152](#)

Message Reference Overview

This appendix describes the messages that are issued by PowerExchange CDC Publisher.

The message prefixes indicates the message categories. Messages in each of the following categories are described:

Message Prefix	Message Category
CDCPUB	CDC Publisher common messages
CPPSVC	Diagnostic messages for a C++ service that is involved in CDC Publisher extraction of change data from PowerExchange
LOGCFG	Message-logging configuration messages
PWXLIC	Product licensing messages
PWXRDR	PWX Reader messages Note: The PWX Reader is part of the CDC Publisher Extractor process.

CDCPUB_10001 to CDCPUB_16001

CDCPUB_10001 to CDCPUB_16001

CDCPUB_10001	The process [process_name] could not perform the state change or action [action]. Reason: [reason].
Explanation:	The specified internal process encountered an error while trying to stop, start, or perform another requested action. This message includes the reason for the error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10002	Generating dump output. Reason: [reason].
Explanation:	This error message introduces dump information that was produced by the CDC Publisher for the specified reason.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10003	DUMP: [text].
Explanation:	This error message contains one or more lines of detailed debug information, in text format, from a CDC Publisher dump operation.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10004	HEXDUMP: [hexadecimal_dump_information].
Explanation:	This error message contains one or more lines of detailed debug information, in hexadecimal format, from a CDC Publisher dump operation.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10005	Generated dump complete.
Explanation:	This error message identifies the end of the detailed debug information from a CDC Publisher dump operation.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10006	The process [current_process_name] was unable to start a thread for [failed_process_name]. The maximum number of threads might have been reached. Error: [error_description].
Explanation:	The specified process encountered an error while trying to start a new operating system thread. This message includes the thread ID or the name of the subprocess associated with the thread, and a description of the error. This error can be caused by an operating system resource limitation or a failure to remove unused threads.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_10007	The process [current_process_name] was unable to start a thread for [failed_process_name] because of an exception. Error: [error_description].
Explanation:	The specified process encountered an error while trying to start a new operating system thread. The request for a new thread generated an exception. This problem can occur because of an operating system resource limitation or because the CDC Publisher cannot remove unused threads.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10008	The process [current_process_name] could not process the request named [failed_request_name]. The process current state [state] does not support the request. Error: [error_description].
Explanation:	An internal process was unable to stop, start, or perform another requested action because its current state does not support the request type.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10011	The process [process_name] with thread ID [thread_ID] stopped running because of a fatal error.
Explanation:	The internal process or its operating system thread stopped unexpectedly because of a fatal error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10012	The process [process_name] with thread ID [thread_ID] encountered an error. Error code: [error_code]. Error description: [description].
Explanation:	The specified internal process or its operating system thread stopped unexpectedly because of an error. This message provides details about the error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10013	The process [current_process_name] could not find the directory [directory_name] requested in the configuration parameter [parameter_name].
Explanation:	The specified configuration parameter points to a directory on the local file system, but the specified process cannot find that directory.
User Response:	To correct the error, verify that the configuration parameter points to an existing directory, or create the directory if it does not exist.
CDCPUB_10014	The process [current_process_name] could not write to the file or directory [file_or_directory_name] requested in the configuration parameter [parameter_name].
Explanation:	The specified configuration parameter points to a file or directory that does not exist or is not writable. As a result, the process cannot write to the file or directory.
User Response:	To correct the error, verify that the configuration parameter points to an existing file or directory for which write permissions have been granted.

CDCPUB_10015	The process <i>[extractor_process_name]</i> has requested a shutdown of its helper thread <i>[thread_name]</i>.
Explanation:	This informational message reports that the CDC Publisher Extractor process requested its helper thread to shut down.
User Response:	No response is required.
CDCPUB_10016	The process <i>[extractor_process_name]</i> was unable to shut down its helper thread <i>[thread_name]</i>.
Explanation:	The CDC Publisher Extractor process encountered an error while trying to shut down its internal helper thread.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10017	The process <i>[extractor_process_name]</i> is waiting for its helper thread <i>[thread_name]</i> to shut down. Current wait duration: <i>[milliseconds]</i>.
Explanation:	The CDC Publisher Extractor process requested that its internal helper thread shut down. This informational message reports the number of milliseconds that the process has been waiting for the helper thread to complete shutdown processing.
User Response:	No response is required.
CDCPUB_10018	The helper <i>[helper_process_name]</i> with thread ID <i>[thread_ID]</i> is starting.
Explanation:	This informational message reports that the specified internal helper process is starting on the specified thread.
User Response:	No response is required.
CDCPUB_10019	The helper <i>[helper_process_name]</i> with thread ID <i>[thread_ID]</i> has shut down successfully.
Explanation:	This informational message reports that the specified internal helper process has shut down successfully.
User Response:	No response is required.
CDCPUB_10020	The helper <i>[helper_process_name]</i> with thread ID <i>[thread_ID]</i> stopped running because of a fatal error.
Explanation:	The specified internal helper process stopped because of a fatal error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10021	The process <i>[helper_process_name]</i> with thread ID <i>[thread_ID]</i> encountered an error. Error code: <i>[error_code]</i>. Error description: <i>[description]</i>.
Explanation:	The specified internal helper process stopped because of an error. This message provides information about the error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10022	Stack Trace <i>[stack_trace_information]</i>.
Explanation:	This message reports stack trace information, which is typically for debugging a preceding error.

User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10023	The process [process_name] result for a test connection request was successful.
Explanation:	The specified internal process successfully tested connectivity to an outside element.
User Response:	No response is required.
CDCPUB_10024	The process [process_name] result for a test connection request was not successful.
Explanation:	This informational message reports that the specified internal process tested connectivity to an outside element, but the test was unsuccessful.
User Response:	No response is required.
CDCPUB_10025	The process [process_name] with thread ID [thread_ID] encountered an error.
Explanation:	The internal process encountered an error. Typically, the details of the error are provided in accompanying error messages.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10026	The process [process_name] with thread ID [thread_ID] encountered an error. Error code: [error_code]. Description: [error_description].
Explanation:	The specified internal process encountered an error. This message provides information about the error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10027	One of the subordinate threads or processes for this process is no longer running.
Explanation:	An internal process detected that its helper or subordinate process is not running.
User Response:	Review the accompanying error messages from the helper or subordinate process. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10028	The process [current_process_name] could not read or find the file [file_name].
Explanation:	The specified internal process could not find or does not have access to the specified file.
User Response:	Verify that the file exists and that read permissions have been granted on the file. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10029	The process [current_process_name] found the directory name [directory_name] when expecting a file name.
Explanation:	The internal process expected a file name but found the specified directory name instead.
User Response:	Verify that the configuration properties that specify files and directories are set correctly. Also, verify that you have not inadvertently specified a directory name instead of a file name.
CDCPUB_10030	The process [process_name] with thread ID [thread_ID] encountered a non-fatal error. Error description: [description].
Explanation:	The specified process encountered a non-fatal error. The process should be able to ignore the error and continue running. The process will attempt to process the condition that caused the error again.

User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10031	The process [process_name] with thread ID [thread_ID] will restart from the default position in the change stream. An explicit location is not required.
Explanation:	This informational message reports that the specified process is restarting from the default location in the change data stream. The process cannot find or identify an explicit or logical restart point. The default location is typically the start or end of the change stream.
User Response:	If you do want the process to restart from the default location, reconfigure the environment to restart from the location you want to use. If necessary, contact Informatica Global Customer Support.
CDCPUB_10032	The process [process_name] with thread ID [thread_ID] will restart from the beginning or earliest available position in the change stream.
Explanation:	This informational message reports that the specified process is restarting from the beginning of the change data stream or from the earliest available position in the change stream.
User Response:	If you do not want to start from the beginning or earliest available position in the stream, reconfigure the environment to restart from a specific location. If necessary, contact Informatica Global Customer Support.
CDCPUB_10033	The process [process_name] with thread ID [thread_ID] will restart from the end or latest available position in the stream.
Explanation:	This informational message indicates that the specified process is restarting from the end of the change data stream or from the latest available position in the change stream.
User Response:	If you do not want to restart from the end of or latest available position in the change stream, reconfigure the environment to restart from a specific location. If necessary, contact Informatica Global Customer Support.
CDCPUB_10034	The process [process_name] with thread ID [thread_ID] will restart from the stream position identified by hex value [0xvalue].
Explanation:	This informational message indicates that the specified process is restarting from the change data stream position that is represented by the hexadecimal value.
User Response:	If you do not want to restart from this position, you can reconfigure the environment to restart from a specific location. If necessary, contact Informatica Global Customer Support.
CDCPUB_10035	The process [process_name] was unable to stop processing on the path [path_name].
Explanation:	The specified process could not stop processing data on the specified logical internal path. A path is the combination of internal processes responsible for moving data from a source to a target.
User Response:	Typically, the process that owns the path retries the stop request.
CDCPUB_10036	The process [process_name] is waiting for path [path_name] to stop processing data. Current wait duration: [milliseconds].
Explanation:	This informational message reports that the process has been waiting the specified number of milliseconds for data processing on the specified logical internal path to stop. A path is the combination of internal processes responsible for moving data from a source to a target.

User Response:	No response is required.
CDCPUB_10037	The process [process_name] was unable to shut down processing on the path [path_name].
Explanation:	The process failed to shut down processing on the specified logical internal path. A path is the combination of internal processes responsible for moving data from a source to a target.
User Response:	The process that owns the path typically retries the shutdown. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10038	The process [process_name] is waiting for the path [path_name] to shut down. Current wait duration: milliseconds.
Explanation:	The process has been waiting the specified number of milliseconds for processing on the logical internal path to shut down. Processing is still active. A path is the combination of internal processes responsible for moving data from a source to a target.
User Response:	No response is required.
CDCPUB_10039	The process [process_name] with thread ID [thread_ID] is configured to use a pre-existing PowerExchange environment.
Explanation:	This informational message indicates that the specified process is using an existing Informatica PowerExchange environment as the source of change data. You do not need to create a new PowerExchange environment as the source.
User Response:	No response is required.
CDCPUB_10040	The process [process_name] could not interpret the restart configuration value [configuration_parameter_value]. The value is expected to start with one of the following prefixes: [prefixes].
Explanation:	The process tried to use the specified restart configuration parameter value to determine the location in the change data stream from which to begin consuming data. However, the process could not interpret that value. The configuration value does not have a prefix or initial set of characters that match any of the expected characters.
User Response:	Review the Informatica CDC Publisher documentation for the proper format for the configuration parameter value. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10041	The process [process_name] could not find or read the file [file_name] that is requested in the configuration parameter [parameter_name].
Explanation:	The internal process could not find or does not have read permissions on the specified file. This message identifies the configuration parameter that specifies the file name.
User Response:	Verify that the file exists and that read permissions have been granted on the file. Alternatively, edit the configuration parameter value to point to the correct file. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10042	The process [process_name] encountered an error while performing [action]. Error: [error_description].
Explanation:	The specified internal process encountered an error while performing the specified action or step. The message includes a description of the error.
User Response:	To diagnose and correct the error, analyze this message and accompanying log messages. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_10043	The process [process_name] encountered an error while processing the configuration [parameter_name] with the value [parameter_value]. Error: [error_description].
Explanation:	The specified internal process encountered an error while processing configuration information. This message reports the name and value of the configuration parameter that was being processed when the error occurred.
User Response:	If the configuration parameter value was entered incorrectly, correct the value. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10044	The process [process_name] connected successfully to [entity].
Explanation:	This informational message reports that the specified process successfully connected to an entity.
User Response:	No response is required.
CDCPUB_10045	The process [process_name] was unable to connect to [entity] because of the error [error_description].
Explanation:	The specified internal process was unable to connect to an external entity. This message provides the name or label of the external entity and a description of the connectivity error.
User Response:	Verify that the external entity is available and that the configuration properties related to accessing the entity are correct. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10046	The process [process_name] was unable to create the directory [directory_name] requested in the configuration parameter [parameter_name].
Explanation:	An internal process was unable to create the directory that is named in the specified configuration parameter. This message identifies the configuration parameter name and the directory name.
User Response:	Perform one of the following actions: <ul style="list-style-type: none"> • Correct the configuration value. • Change the file system permissions to allow the process to create the directory. • Create the directory manually, eliminating the need for the process to create it. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10047	The process [process_name] was unable to create the directory [directory_name] requested in the configuration parameter [parameter_name]. Error: [errorDescription_or_message].
Explanation:	The internal process was unable to create the directory that is identified in the specified configuration parameter. This message includes the directory name, parameter name, and error description or message.
User Response:	Perform one of the following actions: <ul style="list-style-type: none"> • Correct the configuration value if it is incorrect. • Change the file system permissions to allow the process to create the directory. • Create the directory manually, eliminating the need for the process to create it. If the problem persists, Contact Informatica Global Customer Support for assistance.

CDCPUB_10048	CDC Publisher internal branch <i>[branch_identifier]</i> build number <i>[build_number]</i> dated <i>[date]</i>.
Explanation:	This informational message reports the CDC Publisher internal branch identifier, build number, and build date and time.
User Response:	No response is required.
CDCPUB_10050	The environment <i>[environment_item]</i> has the value <i>[value]</i>.
Explanation:	This informational message reports environment variables, library paths, or other information about the environment where the CDC Publisher process is running.
User Response:	No response is required.
CDCPUB_10051	The process <i>[process_name]</i> successfully loaded the external library <i>[library_name]</i>.
Explanation:	This informational message reports that the process successfully loaded the specified external library.
User Response:	No response is required.
CDCPUB_10052	The process <i>[process_name]</i> was unable to load the external library <i>[library_name]</i>. Error: <i>[errorDescription_or_message]</i>.
Explanation:	The specified internal process was unable to load the specified external library. This message provides the error message or description.
User Response:	Perform one of the following actions: <ul style="list-style-type: none"> • Add or download the library if it is missing. • Correct environment settings to include the library in the system path. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10053	The configuration parameter <i>[parameter_name]</i> has the value <i>[value]</i> which is outside the allowable range of <i>[lowest_value]</i> to <i>[highest_value]</i>.
Explanation:	The specified configuration parameter is set to a value that is not within the range of allowable values that is identified in this message.
User Response:	To correct the error, change the configuration parameter value to a value that is within the range of allowable values.
CDCPUB_10054	The configuration parameter <i>[parameter_name]</i> has the invalid value <i>[value]</i>. Allowable values are <i>[allowable_values]</i>.
Explanation:	The specified configuration parameter is set to a value that is not allowed or recognized.
User Response:	To correct the error, change the configuration parameter value to one of the allowed values that are identified in this message.
CDCPUB_10055	The configuration parameter <i>[parameter_name]</i> has a null or empty value. This parameter requires a valid value.
Explanation:	The specified configuration parameter has a null or empty value. It must be set to a valid value.
User Response:	To correct the error, enter a valid value for the configuration parameter.

CDCPUB_10056	The CDC Publisher could not find the required configuration parameter <i>[parameter_name]</i>.
Explanation:	The CDC Publisher could not find the specified configuration parameter, which is required.
User Response:	Verify that the configuration parameter is appropriate for your environment and then add it to your configuration. If the parameter is not appropriate, the CDC Publisher might still expect it because another related parameter is incorrectly set. In this case, review the other configuration parameter settings to determine if any of them are inadvertently requiring the specified parameter to be configured. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_10057	One or more configuration parameters in the logical group named <i>[group_name]</i> have a value that is in error. Error: <i>[description_or_message]</i>.
Explanation:	The specified logical group contains one or more configuration parameters that have invalid values. The message includes the error description or message.
User Response:	Review the information in this message and accompanying messages to determine the configuration parameters that have the invalid values and the corrective actions to take.
CDCPUB_10058	The process <i>[process_name]</i> found the file <i>[file_name]</i> when expecting a directory name.
Explanation:	The internal process found the specified file name but expected to find a directory name.
User Response:	Verify that the configuration properties that specify files and directories are set correctly. Make sure that you have not inadvertently specified a file name instead of a directory name.
CDCPUB_10059	The configuration property <i>[property_name]</i> has the value <i>[value]</i>
Explanation:	This informational message reports a configuration property value that is in use by the CDC Publisher process.
User Response:	No response is required.
CDCPUB_10060	The server is listening on port <i>[port_number]</i>.
Explanation:	This informational message reports the port number that the CDC Publisher server is using.
User Response:	No response is required.
CDCPUB_10061	The configuration parameter <i>[parameter_name]</i> has been deprecated. Use alternate parameter <i>[parameter_description]</i>.
Explanation:	This informational message reports that the specified configuration parameter has been deprecated and identifies the parameter that replaces it, if available.
User Response:	No response is required.
CDCPUB_10062	The CDC Publisher detected that the value of the configuration parameter <i>[parameter_name]</i> is not valid because it is not a string representation of a hexadecimal value.
Explanation:	The CDC Publisher detected that the specified configuration parameter has an invalid value. The value should be a string representation of a hexadecimal value.
User Response:	For the specified configuration parameter, enter a valid hexadecimal string that consists of an even number of characters, which can include the alphanumeric characters 0-9 and A-F. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_10063	Executing <i>product</i> release [<i>release</i>] build number [<i>build_number</i>] dated [<i>date</i>].
Explanation:	This informational message reports the product release, build number, and build date for the product that is running.
User Response:	No response is required.
CDCPUB_10064	The server is listening on host [<i>host_IP_address</i>] and port [<i>port_number</i>].
Explanation:	This informational message identifies the host IP address and port number that the CDC Publisher server is using.
User Response:	No response is required.
CDCPUB_10065	No valid license found.
Explanation:	The CDC Publisher found no license for the product or one of its features.
User Response:	Check the system log for additional messages that provide details about the license error. If you need assistance, contact Informatica Global Customer Support.
CDCPUB_10066	TRACE: [<i>trace_information</i>]
Explanation:	This error message provides trace-level information about processing at the time of an error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, provide this trace information and any related messages to Informatica Global Customer Support for further analysis.
CDCPUB_10067	File [<i>file_name</i>] :<i>line</i>
Explanation:	This informational message reports the contents of the specified file. One message is produced for each line in the file.
User Response:	No response is required.
CDCPUB_10068	The configuration property [<i>property_name</i>] in group [<i>group_name</i>] is not recognized. Please verify that the property name is correct.
Explanation:	The specified configuration parameter is not recognized.
User Response:	Verify that the configuration property is spelled correctly and in the correct group of properties.
CDCPUB_11001	The CDC Publisher requested that a checkpoint be written to the target. Request timeout interval is [<i>number_of_milliseconds</i>] milliseconds.
Explanation:	The CDC Publisher requested its main process to write a checkpoint to the target. The checkpoint can be used to validate that all change data has been consumed up to the point in time when the checkpoint was taken. The request will time out with an error if it is not successful before the timeout interval elapses.
User Response:	No response is required.
CDCPUB_11002	The CDC Publisher request to force a checkpoint to the target completed successfully after [<i>number_of_milliseconds</i>] milliseconds.
Explanation:	This informational message indicates that a CDC Publisher request to force a checkpoint to the target completed successfully. The request ran for the reported number of milliseconds.
User Response:	No response is required.

CDCPUB_11003	The main CDC Publisher process was unable to force a checkpoint to the target. Error: [description_or_message].
Explanation:	The CDC Publisher main process was unable to complete the request to force a checkpoint and validate that all change data has been sent to the target. This error can occur if the target system is unavailable or if the time required to send all data to the target exceeds the amount of time the caller can wait for completion.
User Response:	This message does not indicate that data loss occurred. Data that has not yet reached the target will be sent to the target the next time the process restarts. If this message is issued repeatedly, it might indicate a slow system or a timeout configuration that is too restrictive. Analyze the accompanying log messages to diagnose and correct the error. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_11004	The process [process_name] has the following statistics:
Explanation:	This informational message introduces CDC Publisher statistics, which are presented in subsequent messages.
User Response:	No response is required.
CDCPUB_11005	stat_owner : stat_name : stat_value
Explanation:	This informational message reports, for a given statistic, the process that owns or created the statistic and the statistic name and value.
User Response:	No response is required.
CDCPUB_11006	The CDC Publisher request to force a checkpoint succeeded. Checkpoint label [owner_name] and sequence value [sequence_value].
Explanation:	This informational message reports that a CDC Publisher request to force a checkpoint succeeded. The message includes the sequence value from the checkpoint and the owner of the sequence value. A sequence value indicates the position in the change data stream up to which change data has been processed and applied to the target.
User Response:	No response is required.
CDCPUB_11007	The CDC Publisher request to force a checkpoint succeeded. Checkpoint sequence value [sequence_value].
Explanation:	This informational message reports that a CDC Publisher request to force a checkpoint succeeded. The message includes the sequence value from the checkpoint. A sequence value indicates the position in the change data stream up to which change data has been processed and applied to the target.
User Response:	No response is required.
CDCPUB_11008	Other checkpoint information [checkpoint_information].
Explanation:	This informational message provides additional information about a checkpoint. To determine the checkpoint label (owner name) and sequence in the change stream, see the preceding checkpoint messages.
User Response:	No response is required.

CDCPUB_11011	The process <i>[process_name]</i> is waiting for verification that all work has been sent to the target. Current wait duration: <i>[milliseconds]</i>.
Explanation:	The CDC Publisher process is waiting for its request to verify that all data has been sent to the target to complete. This message reports the current wait time and is issued periodically when the verification takes longer than a few seconds.
User Response:	No response is required.
CDCPUB_11012	Memory <i>:[details]</i>.
Explanation:	This informational message reports a memory consumption attribute or statistic from the CDC Publisher. Multiple instances of this message can occur, each reporting a different element.
User Response:	No response is required.
CDCPUB_11013	Processing a request to clear the cache for Formatter <i>[formatter_name]</i>.
Explanation:	This informational message reports that a request to clear the cache for the specified Formatter is being processed. The message includes the logical name of the Formatter.
User Response:	No response is required.
CDCPUB_11014	Processing a request to clear the cache of Formatter <i>[formatter_name]</i> for table <i>[table_name]</i>.
Explanation:	This informational message reports that a request to clear a table entry from the cache of the specified Formatter is being processed. The message includes the logical name of the Formatter and the name of the table.
User Response:	No response is required.
CDCPUB_11015	Processing a request to print the cache entries for Formatter <i>[formatter_name]</i>.
Explanation:	This informational message reports that a request to print the contents of the cache for the specified Formatter is being processed. The message includes the logical name of the Formatter.
User Response:	No response is required.
CDCPUB-11016	Processing a request to print the a table entry in the cache of Formatter <i>[formatter_name]</i> for table <i>[table_name]</i>.
Explanation:	This informational message reports that a request to print a table entry that is in the cache of the specified Formatter is being processed. The message includes the logical name of the Formatter and the name of the table.
User Response:	No response is required.
CDCPUB_11017	Processing a request to load or reload the cache of the Formatter <i>[formatter_name]</i>.
Explanation:	This informational message reports that a request to load or reload cache entries for the specified Formatter is being processed. Only new or missing entries will be created. Typically, existing cache entries are not overwritten. To replace an entry, you must clear it from cache first.
User Response:	No response is required.

CDCPUB_11018	The request to load or reload the formatter cache for Formatter [formatter_name] failed. The load process ended prematurely.
Explanation:	A request to load or reload the cache for the specified Formatter failed. Additional messages that describe the error in detail should have preceded this message.
User Response:	For more information about the error, see the preceding messages. After you diagnose the cause of the error, try to correct the problem and restart the process. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_11019	The request to load or reload the cache for Formatter [formatter_name] is waiting for extraction maps. Current wait duration: [milliseconds].
Explanation:	This informational message reports the number of milliseconds that the specified Formatter has been waiting to receive the extraction maps that were read from PowerExchange so that it can load or reload its cache.
User Response:	No response is required.
CDCPUB_11020	The request to load or reload the cache for Formatter [formatter_name] failed. The process failed to complete within a reasonable amount of time.
Explanation:	A request to load the cache of the specified Formatter did not finish within the maximum time allowed. As a result, the process was terminated before the cache was completely loaded.
User Response:	If the cache entries that you need were loaded, you can ignore this error. Otherwise, run the request to load the cache again. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_11021	A request to perform JVM garbage collection processing was executed by [process_name] with thread ID [thread_ID].
Explanation:	This informational message reports that JVM garbage collection processing was successfully executed by an internal process on the specified thread.
User Response:	No response is required.
CDCPUB_11022	The process [process_name] found configuration property [property_name] in both file [file_name1] and file [file_name2].
Explanation:	This informational message reports that the specified process found multiple configuration files with the same configuration property.
User Response:	Informatica recommends that a configuration property be specified in only one configuration file in your environment.
CDCPUB_11023	The process [process_name] found configuration property [property_name] in file [file_name1] and file [file_name2] with conflicting values.
Explanation:	The process found that two configuration files specify the same configuration property but with different values. The process cannot determine the correct value to use.
User Response:	Edit the configuration files to ensure that the property is specified in only one of the configuration files with the correct value. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_11024	The process [process_name] found no configuration files that have suffix [file_suffix] and contain header text [file_header] in the directory [directory_name].
Explanation:	The process found no configuration files that have the required file-name suffix and header content in the specified directory. As a result, no configuration information is available.
User Response:	Verify that the requested directory name is correct and that this directory contains configuration files that have the expected file-name suffix and header content. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_11025	The process [process_name] is reading configuration properties file [file_name].
Explanation:	This informational message reports the name of the configuration file that the specified process is reading.
User Response:	No response is required.
CDCPUB_11026	Processing to load or reload the cache of the Formatter [formatter_name] completed. number_of_items items were loaded.
Explanation:	This informational message reports that a request to load or reload cache entries for the specified Formatter successfully completed. The specified number of items were loaded.
User Response:	No response is required.
CDCPUB_11027	The requested instance directory [directory_name] was not found.
Explanation:	The specified instance directory name was not found or is not a directory.
User Response:	Verify that the correct instance directory name was provided. Make sure that you have not inadvertently specified a file name instead of a directory name.
CDCPUB_11028	Using instance directory [directory_name].
Explanation:	This informational message reports the instance directory that the product is being used for configuration, logging, reporting, and other files.
User Response:	No response is required.
CDCPUB_13001	Internal Logic Error: The Capture Consumer on thread [thread_name] could not find a cached schema for the object ID [object_ID] when processing a DML event at the change stream sequence position [sequence_position].
Explanation:	This error message is generated by the internal Capture Consumer process that is responsible for consuming or capturing change data. The process caches information about the schema or definition of the object for which change data is being captured or consumed. However, the process could not retrieve the object schema or definition from the cache because of an internal logic error.
User Response:	Restart the process to re-create the cached entry. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13002	The Capture Consumer [process_name] with the thread ID [thread_ID] is starting.
Explanation:	This informational message reports that the internal Capture Consumer process is starting.
User Response:	No response is required.

CDCPUB_13003	The Capture Consumer <i>process_name</i> with the thread ID <i>[thread_ID]</i> shut down successfully.
Explanation:	This informational message reports that the internal Capture Consumer process shut down successfully.
User Response:	No response is required.
CDCPUB_13004	The Capture Consumer <i>[process_name]</i> with the thread ID <i>[thread_ID]</i> stopped running because of a fatal error.
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data stopped running because of a fatal error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13005	The Capture Consumer <i>[process_name]</i> with the thread ID <i>[thread_ID]</i> ended because of an error. Error code: <i>[error_code]</i>. Error description: <i>[error_description]</i>.
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data ended because of a fatal error. This message provides the error code and description.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13006	The Capture Consumer <i>[process_name]</i> with thread ID <i>[thread_ID]</i> encountered an I/O error. Resetting the connection to the Change Data Capture stream. The last stream position was <i>[position]</i>.
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data has reset its connection to the entity that creates the change data capture stream. This message might be issued after a brief network outage or on a periodic basis to ensure that an active connection exists. The message reports the last position in the change stream that was processed prior to the error.
User Response:	If this message is issued repeatedly, check if a network outage exists and if the entity responsible for generating the change data capture stream is operating.
CDCPUB_13007	The Capture Consumer <i>[process_name]</i> with the thread ID <i>[thread_ID]</i> is waiting for the change data capture stream to be re-established. Current wait time: <i>[hh:mm:ss]</i>.
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data is in a wait state because the entity responsible for generating the change stream is unavailable. This informational message reports the amount of time, in <i>hours:minutes:seconds</i> , that this process has been waiting.
User Response:	No response is required.
CDCPUB_13008	The Capture Consumer <i>[process_name]</i> with the thread ID <i>[thread_ID]</i> is waiting for data from the change data capture stream. Current wait duration: <i>[hh:mm:ss]</i>. Current message count <i>[number]</i>.
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data is in a wait state because it is not receiving change data from the entity that is responsible for generating the change stream. This informational message reports the amount of time, in <i>hours:minutes:seconds</i> , that this process has been waiting and the number of messages that have been processed to the current point in time.

User Response:	No response is required.
CDCPUB_13009	The Capture Consumer [process_name] with the thread ID [thread_ID] is waiting to write data from the change data capture stream because the internal InputStream is at capacity. Current wait duration: [hh:mm:ss].
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data is in a wait state because the internal buffer that holds the change data is full. This informational message reports the amount of time, in <i>hours:minutes:seconds</i> , that this process has been waiting. Typically, the wait time is a short. As change data is processed, items are removed from the buffer, allowing this process to continue.
User Response:	If this message is issued repeatedly, the change data is probably arriving at a rate faster than the rate at which it is being processed.
CDCPUB_13010	The Capture Consumer [process_name] with the thread ID [thread_ID] is shutting down.
Explanation:	This informational message reports that the internal Capture Consumer process that is responsible for consuming or capturing change data is shutting down.
User Response:	No response is required. If necessary, contact Informatica Global Customer Support.
CDCPUB_13011	The Capture Filter [process_name] with the thread ID [thread_ID] is starting.
Explanation:	This informational message reports that the specified internal Capture Filter process, which filters the change data to be captured or consumed, is starting.
User Response:	No response is required.
CDCPUB_13012	The Capture Filter [process_name] with the thread ID [thread_ID] shut down successfully.
Explanation:	This informational message reports that the internal Capture Filter process that is responsible for filtering the change data to be captured or consumed has successfully shut down.
User Response:	No response is required.
CDCPUB_13013	The Capture Filter [process_name] with the thread ID [thread_ID] stopped running because of a fatal error.
Explanation:	The internal Capture Filter process that is responsible for filtering change data to be captured or consumed has stopped because of a fatal error.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13014	The Capture Filter [process_name] with the thread ID [thread_ID] stopped because of a fatal error. Error code: [error_code]. Error description: [error_description].
Explanation:	The internal Capture Filter process that is responsible for filtering change data stopped because of a fatal error. This message provides the error code and a description of the error.
User Response:	To diagnose and correct the error, analyze the error information in this message and any accompanying log messages. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_13015	The process [process_name] with the thread ID [thread_ID] is waiting for incoming data. Current wait duration: [hh:mm:ss].
Explanation:	The internal process that is responsible for change data capture is sleeping because it has no input data to process. This situation is typical for an idle system. This informational message reports the amount of time, in <i>hours:minutes:seconds</i> , that the process has been waiting.
User Response:	No response is required.
CDCPUB_13016	The CDC Publisher Extractor [process_name] requested a shutdown of its helper thread [thread_name].
Explanation:	This informational message reports that the CDC Publisher Extractor process has requested a helper or subordinate thread to shut down.
User Response:	No response is required.
CDCPUB_13017	The CDC Publisher Extractor [process_name] was unable to shut down its helper thread [thread_name].
Explanation:	This informational message reports that the CDC Publisher Extractor process was unable to shut down the specified helper or subordinate thread.
User Response:	No response is required.
CDCPUB_13018	The CDC Publisher Extractor [process_name] is waiting for its helper thread [thread_name] to shut down. Current wait duration: [hh:mm:ss].
Explanation:	The CDC Publisher Extractor process is waiting for a helper or subordinate thread to shut down. This informational message reports the amount of time, in <i>hours:minutes:seconds</i> , that this process has been waiting.
User Response:	No response is required.
CDCPUB_13019	The Capture Consumer [process_name] with the thread ID [thread_ID] is waiting for the PowerExchange Capture Service to shut down. Current wait duration: [hh:mm:ss].
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data is waiting for the PowerExchange Capture Service interface to shut down. This informational message reports the amount of time, in <i>hours:minutes:seconds</i> , that this process has been waiting for the PowerExchange Capture Service to shut down.
User Response:	No response is required.
CDCPUB_13020	The Capture Consumer [process_name] with the thread ID [thread_ID] has issued the command [command] to the PowerExchange Capture Service.
Explanation:	This informational message reports that the internal Capture Consumer process that is responsible for consuming or capturing change data issued the specified command to the PowerExchange Capture Service interface.
User Response:	No response is required.

CDCPUB_13021	The Capture Consumer [process_name] with the thread ID [thread_ID] stopped waiting for the PowerExchange Capture Service to shut down after wait time [hh:mm:ss].
Explanation:	This informational message reports that the internal Capture Consumer process that is responsible for consuming or capturing change data stopped waiting for the PowerExchange Capture Service interface to shut down after the specified amount of time, in <i>hours:minutes:seconds</i> , elapsed. The Capture Consumer process will shut down.
User Response:	No response is required.
CDCPUB_13022	The Capture Consumer [process_name] with the thread ID [thread_ID] is resetting the connection to the change data capture stream because of a lack of activity. The last position in the change stream was [current_position].
Explanation:	The internal Capture Consumer process that is responsible for consuming or capturing change data reset its connection to the change data capture provider after a period of no change activity. This informational message reports the current position or sequence in the change stream. The message might be issued periodically when the capture process is idle. The connection is reset to verify that the change data capture provider is still active.
User Response:	No response is required.
CDCPUB_13023	The Capture Consumer [process_name] with the thread ID [thread_ID] is starting the PowerExchange Capture Service.
Explanation:	This informational message reports that internal Capture Consumer process that is responsible for consuming or capturing change data is starting the PowerExchange Capture Service interface.
User Response:	No response is required.
CDCPUB_13024	The process [process_name] could not interpret the configured restart value [restart_value] correctly. The value is not a numeric value or hex string.
Explanation:	A configured setting identifies the location in the change data stream where the process should resume consuming change data. However, this error message indicates that the configured restart value is not a valid numeric value or hexadecimal string.
User Response:	Review the Informatica product documentation to determine the proper format for specifying a restart point. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13025	The process [process_name] with the thread ID [thread_ID] is waiting until more space is available on its internal event queue. Current wait duration: [hh:mm:ss].
Explanation:	This informational message reports that the internal process that is responsible for processing change data is idle because it does not have adequate space available to process more data. The process has waited the specified amount of time, in <i>hours:minutes:seconds</i> , for more space. This condition typically occurs when the process cannot find an available helper thread for processing event data, because all helper threads are busy.
User Response:	No response is required.

CDCPUB_13026	The process [process_name] with the thread ID [thread_ID] is waiting until more space is available on its output queue. Current wait duration: [hh:mm:ss].
Explanation:	This informational message reports that the internal process that is responsible for processing change data is in a wait state because it does not have adequate space available to write results to an output queue. The process has waited the specified amount of time, in <i>hours:minutes:seconds</i> , to write results. This condition typically occurs when a downstream process is not consuming events as quickly as the process provides them.
User Response:	No response is required.
CDCPUB_13027	The process [process_name] with the thread ID [thread_ID] exceeded the maximum wait time for writing data from the capture stream because the internal InputStream is at capacity. Current wait duration: [seconds_elapsed] seconds. Maximum wait time allowed: maximum_seconds seconds.
Explanation:	The specified process has reached the maximum amount of time, in seconds, to wait on a full input buffer before treating the full buffer condition as an error. The full buffer condition typically occurs when other processes are unable to consume data from the buffer.
User Response:	Review the system log to determine why downstream processes are not processing data. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13028	The Capture Consumer process [process_name] with the thread ID [thread_ID] failed to start a change data stream. Return code: [return_code].
Explanation:	The specified process failed to start a change data stream. The message provides the non-zero return code for the error.
User Response:	To diagnose the error, review the accompanying messages in the system log. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13029	The Capture Consumer process [process_name] with the thread ID [thread_ID] found that the underlying change data stream stopped. Return code: [return_code].
Explanation:	The specified process detected that the change data stream stopped unexpectedly. The message includes the non-zero return code for the error.
User Response:	To diagnose the error, review the accompanying messages in the system log. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_13030	The Capture Consumer process [process_name] with the thread ID [thread_ID] stopped the change data stream because of a shutdown request. Return code: [return_code].
Explanation:	After the specified process received a shutdown request, it stopped the change data stream. The stream ended with specified return code.
User Response:	No response is required.
CDCPUB_13031	The Capture Consumer process [process_name] with the thread ID [thread_ID] successfully shut down the change data stream but the stream ended with the error: [error_description].
Explanation:	The specified process successfully shut down the change data stream, even though the reported error occurred during the process. Because the error did not prevent the shutdown from completing successfully, you can ignore the error.
User Response:	Ignore the error. If it occurs again repeatedly, contact Informatica Global Customer Support.

CDCPUB_13032	The Extractor [<i>logical_extractor_name</i>] processed the request [<i>input_request</i>] to generate an internal restart value. Generated restart value: [<i>restart_value</i>].
Explanation:	This informational message reports the internal restart value that Extractor process generated in response to the specified request.
User Response:	No response is required. However, if the result did not match the input request or if you do not expect a restart, contact Informatica Global Customer Support.
CDCPUB_13033	No tables have been selected for change data extraction. Verify that the configured schema and any include or exclude lists are correct for the PowerExchange source environment.
Explanation:	The configuration does not match any tables from which PowerExchange captures change data.
User Response:	Correct the configuration so that it matches at least one table from which PowerExchange captures change data.
CDCPUB_14001	The Formatter [<i>process_name</i>] is waiting for its helper thread [<i>thread_name</i>] to shut down. Current wait duration: [<i>hh:mm:ss</i>].
Explanation:	The internal Formatter process that is responsible for formatting change data is waiting for its helper or subordinate thread to shut down. This informational message reports the elapsed wait duration, in <i>hours:minutes:seconds</i> , to the current point in time.
User Response:	No response is required.
CDCPUB_14002	Internal logic error: The formatter helper thread [<i>thread_name</i>] could not find a schema cached for the object ID [<i>object_ID</i>] when processing a DML event at the change stream sequence location [<i>sequence_location</i>].
Explanation:	This error message is generated by the internal Formatter process that is responsible for formatting change data. The process caches information about the schema of an object for which it consumes change data. This message indicates that an internal logic error prevented the Formatter process from retrieving the schema for the specified object from an internal cache when processing a DML change operation. The message includes the object ID and the current sequence location within the change stream of the DML operation.
User Response:	To re-create the cached entry, restart the Formatter process. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_14003	The Formatter helper [<i>process_name</i>] encountered an error and ended while formatting the event at the change stream location [<i>sequence_location</i>]. Error description: [<i>error_description</i>].
Explanation:	The internal Formatter process that is responsible for formatting change data encountered an error and ended unexpectedly. This message includes a brief description of the error and the current sequence location within the change data stream.
User Response:	To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_14004	The Formatter [<i>process_name</i>] encountered an error while attempting to create a schema cache. Error description: [<i>error_description</i>].
Explanation:	The internal Formatter process that is responsible for formatting change data was unable to create an internal cache to hold schema information.

User Response: To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14005 The Formatter [*process_name*] encountered an error while attempting to create an Avro schema cache. Error description: [*error_description*].

Explanation: The internal Formatter process that is responsible for formatting change data into Avro messages was unable to create an internal cache to hold Avro schema information.

User Response: To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14006 The table [*table_name*] contains characters that are not valid for an Avro schema. Avro field names must begin with a letter and include only the digits 0-9, letters A-Z and a-z, and underscore (_) character. Consider changing the current replacement character [*replacement_character*].

Explanation: The table name is used to create an Avro field name. Avro field names must begin with an alphabetic character A-Z or a-z and can contain the digits 0-9, letters A-Z and a-z, and the underscore (_) character. If a replacement character is defined, it will replace any non-conforming characters in the table name before the name is passed to Avro.

User Response: Verify that you defined a replacement character that can be used to replace any special characters in the table name. If you suspect that the current replacement character is not supported by Avro, try changing the replacement character. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14007 The column [*column_name*] in table [*table_name*] contains characters that are not valid for an Avro schema. Avro field names must begin with a letter and include only the digits 0-9, letters A-Z and a-z, and underscore (_) character. Consider changing the current replacement character [*replacement_character*].

Explanation: The column name is used to create an Avro field name. Avro field names must begin with an alphabetic character A-Z or a-z and can contain the digits 0-9, letters A-Z and a-z, and the underscore (_) character. If a replacement value is defined, it will replace any non-conforming characters in the column name before the name is passed to Avro.

User Response: Verify that you defined a replacement character that can be used to replace any special characters in the column name. If you suspect that the current replacement character is not supported by Avro, try changing the replacement character. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14008 The process [*process_name*] failed to find an entry for the table [*table_name*]. The process will retry the lookup by using the following technique: [*retry_technique*].

Explanation: The process could not find an entry for specified table name. The process is retrying the table lookup by using a different technique.

User Response: No response is required.

CDCPUB_14009 All attempts by process [*process_name*] to find any entry for [*table_name*] failed.

Explanation: The process received a request to find the internal entities or entries for the specified table name, but all attempts to find these items failed. Either the table name was entered incorrectly or the system is not configured to process this table.

- User Response: Verify that you provided the correct table name. Also consider changing the table name to avoid any filtering or modifications that the system is configured to perform on a source table name. If the problem persists, contact Informatica Global Customer Support.
- CDCPUB_14010** **When generating a wrapped Avro schema, the process [process_name] found an unsupported Avro configuration. To generate a wrapped Avro schema, the outer Avro message must support or be the same as the inner Avro message encoding. Current outer encoding [outer_encoding] does not support the current inner encoding [inner_encoding].**
- Explanation: The encoding settings for messages that use a wrapped Avro schema are not valid. The encoding of the outer "wrapper" message must be binary, or the encoding of the inner "wrapped" message must match the encoding of the outer message.
- User Response: Correct one or both of the Avro message encoding settings so that they match or the outer message encoding is binary. If the problem persists, contact Informatica Global Customer Support.
- CDCPUB_14011** **When generating a wrapped Avro schema, the process [process_name] determined that the configured encoding of the inner 'wrapped' message is not supported. Inner message encoding [requested_inner_encoding] was replaced with [valid_encoding].**
- Explanation: When generating a wrapped Avro schema, the process determined that the encoding of the inner "wrapped" Avro message is not supported. As a result, the process replaced the requested invalid encoding with the specified valid encoding.
- User Response: No response is required
- CDCPUB_14012** **The Formatter [process_name] found that the configuration property [property_name] contains no value. Enter a name for the timestamp column that is added to formatted messages.**
- Explanation: The internal Formatter process, which is responsible for formatting change data into messages, detected that the specified configuration property is missing or does not contain a value. The property is required because you enabled the formatter to add a timestamp column in the Avro schema and formatted message output, which specifies the date and time at which the Publisher processed incoming change records. The property specifies the name of the added timestamp column or field.
- User Response: Define the missing configuration property in the cdcPublisherAvro.cfg file, or disable adding a timestamp column to the Avro schema and formatted message output by setting the Formatter.formatterAddTimestampColumn property to false. If the problem persists, contact Informatica Global Customer Support.
- CDCPUB_14013** **The Formatter [process_name] found that the configuration property [property_name] contains no value. Enter the date and time format of the timestamp column that is added to formatted messages.**
- Explanation: The internal Formatter process, which is responsible for formatting change data into messages, detected that the specified configuration property does not contain a value. The property should specify the date and time format of the timestamp column that is added in the Avro schema and formatted message output for each record when Formatter.formatterAddTimestampColumn=true is specified.

User Response: Specify a value for the configuration property. You can enter any character string that the Java class SimpleDateFormat supports for formatting dates and times. Alternatively, set `Formatter.formatterAddTimestampColumn=false` to disable adding a timestamp column to the formatted message output. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14014 The Formatter [*process_name*] found that the configuration property [*property_name*] contains a value that is not valid. Error: [*errorText*].

Explanation: The internal Formatter process, which is responsible for formatting change data into messages, detected that the specified configuration property value is not valid. The property specifies an invalid date and time format for the timestamp column or field that is added in the Avro schema and formatted message output for each record when `Formatter.formatterAddTimestampColumn=true` is specified.

User Response: Specify a valid value for the configuration property. You can enter any character string that the Java class SimpleDateFormat supports for formatting dates and times. Alternatively, set `Formatter.formatterAddTimestampColumn=false` to disable adding a timestamp column to the formatted message output. The property value must comply with the Java class SimpleDateFormat. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14015 The Formatter [*process_name*] found that the configuration property [*property_name*] contains no value. Enter a time zone option for the timestamp column that is added to formatted messages.

Explanation: The internal Formatter process, which is responsible for formatting change data into messages, found that the specified property does not contain a value. The property should specify the time zone of the timestamp values that appear in the timestamp column that is added in the Avro schema and formatted message output for each record when `Formatter.formatterAddTimestampColumn=true` is specified.

User Response: Specify a time zone option in the configuration property. Valid values are **local** and **UTC**. Alternatively, set `Formatter.formatterAddTimestampColumn=false` to disable adding a timestamp column to the formatted message output. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14016 The Formatter [*process_name*] found that the configuration property [*property_name*] contains a value that is not valid.

Explanation: The internal Formatter process, which is responsible for formatting change data into messages, detected that the specified configuration property value is not valid. The property specifies an invalid time zone option for the timestamp column that is added in the Avro schema and formatted message output for each record when `Formatter.formatterAddTimestampColumn=true` is specified.

User Response: Specify a valid value for the configuration property. Valid values are **local** and **UTC**. Alternatively, set `Formatter.formatterAddTimestampColumn=false` to disable adding a timestamp column to the formatted message output. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14017 The Formatter [*process_name*] failed to generate timestamp values for the timestamp column that is added to formatted messages. Error: [*errorText*]

Explanation: The internal Formatter process, which is responsible for formatting change data into messages, encountered an error while generating a timestamp value for the timestamp column that is added in the Avro schema and formatted message output for each record when `Formatter.formatterAddTimestampColumn=true` is specified.

User Response: Review the reported error text. If an incorrect configuration property value is responsible for the error, correct the property value. Alternatively, set `Formatter.formatterAddTimestampColumn=false` to disable adding a timestamp column to the formatted message output. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_14018 The process `[Formatter_process_name]` generated key value `[value]` for table `[schema.table_name]` at change stream location `[sequence_location]`.

Explanation: The internal Formatter process generated the specified message key value for a source table at the reported sequence location in the change stream. This message is issued if you specified the `trace=true` parameter in the rule-configuration file that is used for generating message keys.

System Action: No response is required.

CDCPUB_14019 Schema table `[schema.table_name]` column `[column_name]`.

Explanation: This message identifies a column in the schema of a source table. The message is issued for each column in the schema when the internal Formatter process is using a rule-configuration file to generate message keys. The Formatter compares the column names reported in these messages to the column names in the rule definitions in the rule-configuration file. The Formatter found that a column in a rule definition does not match any of the reported schema columns.

User Response: Correct the column names in the rule definitions for the source table. Look for mistyped column names. You can use the CDCPUB_14019 messages to find any column name in error.

CDCPUB_15001 The Connector `[process_name]` is waiting for its helper thread `[thread_name]` to shut down. Current wait duration: `[hh:mm:ss]`.

Explanation: The internal Connector process, which is responsible for connecting to and sending messages to an external entity, is waiting for a helper or subordinate thread to shut down. This informational message includes the amount of time, in `hours:minutes:seconds`, that this process has been waiting.

User Response: No response is required.

CDCPUB_15002 The Connector helper `[process_name]` encountered an error when sending the event at the change stream location `[sequence_location]`. Error description: `[error_description]`.

Explanation: The internal Connector process, which is responsible for connecting to and sending messages to an external entity, encountered an error and ended when processing the event at the specified change stream location.

User Response: To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_15003 The Connector `[process_name]` encountered an error while trying to connect to the target. Error description: `[error_description]`.

Explanation: The internal Connector process was unable to establish a connection to an external entity. This message includes a brief description of the error.

User Response: Verify that the entity to which the process is attempting to connect is available. If the problem persists, Contact Informatica Global Customer Support.

CDCPUB_15004	The Connector [process_name] will write output to the file [file_name].
Explanation:	This informational message reports that the internal Connector process is configured to write message output to the specified file.
User Response:	No response is required.
CDCPUB_15005	The Connector helper [helper_process_name] with the thread ID [thread_ID] is waiting for a request to send data to the target. Current wait duration: [hh:mm:ss].
Explanation:	The internal Connector helper process, which is responsible for sending data to the target, is waiting for permission to proceed. This message reports the wait duration, in <i>hours:minutes:seconds</i> , to the current point in time. In some configurations, the order in which messages are sent to the target entity is strictly enforced. This informational message might be issued when a helper process is ready to send its message but must wait for its turn. If this message is issued frequently or consistently, the number of helpers configured might be greater than the number needed for the current volume of data. When too many threads exist, they might wait on one another.
User Response:	No response is required.
CDCPUB_15006	The process [process_name] with the thread ID [thread_ID] could not find any Connectors defined in the current configuration. No paths will be created to send change data.
Explanation:	No Connectors are defined in the current configuration. As a result, the specified process cannot send change data to any external entity on the logical path. A logical path is the combination of internal Extractor, Formatter, and Connector processes that are responsible for moving captured source data to a target entity.
User Response:	Typically, this informational message is issued if the configuration is incomplete or contains an error. If you need assistance, contact Informatica Global Customer Support.
CDCPUB_15007	The Process [process_name] with the thread ID [thread_ID] determined that the Connector [connector_name] references the CDC Publisher Extractor name [logical_extractor_name], but that name is not in the configuration parameter [parameter_name].
Explanation:	The process determined that the specified Connector references an Extractor name that is not defined in the Connector configuration. This error message reports the Connector name, the Extractor name that cannot be found, and the Connector configuration parameter for specifying the Extractor name.
User Response:	In the Connector configuration parameter, correct the Extractor name or add the name of an existing Extractor process.
CDCPUB_15008	The process [process_name] with the thread ID [thread_ID] found that the Connector [connector_name] references the filter name [filter_name], but that Filter name is not found in the configuration parameter [parameter_name].
Explanation:	The process determined that the specified Connector references a filter name that is not defined in the Connector configuration. This error message includes the Connector name, the name of the filter that cannot be found, and the name of the Connector configuration parameter for specifying the filter name.
User Response:	In the Connector configuration parameter, correct the correct filter name or add the name of an existing filter.

CDCPUB_15009	The process [process_name] with the thread ID [thread_ID] found that the Connector [connector_name] references the Formatter name [formatter_name], but that Formatter name is not found in the configuration parameter [parameter_name].
Explanation:	The process determined that the specified Connector references a Formatter name that is not defined in the Connector configuration. This error message includes the Connector name, the name of the Formatter that was not found, and the name of the Connector configuration parameter for specifying the Formatter name.
User Response:	In the Connector configuration parameter, correct the Formatter name or add the name of an existing Formatter.
CDCPUB_15010	The process [process_name] with the thread ID [thread_ID] found no Connectors defined in the current configuration. The process expected to see the Connector names listed in the configuration parameter [parameter_name].
Explanation:	The process could not find any Connector names in the specified configuration parameter. The parameter should list all Connectors. As a result, no data will be sent to any external entity.
User Response:	This error message is typically produced if the configuration is incomplete or contains an error. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_15011	The process [process_name] with the thread ID [thread_ID] found that the Connector named [connector_name] has incomplete or inaccurate configuration information. The CDC Publisher will not use this Connector.
Explanation:	This informational message reports that the process found an inaccurate or incomplete definition of the specified Connector in the configuration. The process will ignore this Connector. If complete and accurate information is defined for other Connectors in the configuration, one of these Connectors can be used.
User Response:	If the configuration is complete, check if the configuration contains an error. If you need assistance, contact Informatica Global Customer Support.
CDCPUB_15012	The process [process_name] with the thread ID thread_ID did not find the logical name [logical_name] in the configuration parameter [parameter_name].
Explanation:	When examining configurations, the internal process expected to find the specified configuration parameter with the logical name of the entity for which configuration information is being processed. However, the configuration parameter was not found or did not contain the expected logical name.
User Response:	Add the missing configuration parameter with the logical name to the configuration. If the problem persists, contact Informatica Global Customer Support for assistance.
CDCPUB_15013	The process [process_name] with the thread ID [thread_ID] examined configurations for the logical name [logical_name_of_entity] but did not find this name in the configuration parameter [parameter_name].
Explanation:	The internal process examined configurations for the specified logical name of an entity but did not find the logical name in the specified configuration parameter as expected.
User Response:	Specify the logical name in the configuration parameter. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_15014	The Connector <i>[process_name]</i> will write checkpoint information to the file <i>[file_name]</i>.
Explanation:	The internal helper process that is responsible for sending messages to an external entity keeps a record of its progress, called a checkpoint. This informational message reports that the checkpoint information will be written to the specified file.
User Response:	No response is required.
CDCPUB_15015	The Connector <i>[process_name]</i> will read parameter information from the file <i>[file_name]</i>.
Explanation:	The specified internal helper process that is responsible for sending messages to an external entity normally receives configuration properties from a parent entity or process. The informational message indicates that the process is getting some configuration properties directly from the specified file. This behavior is normal for configurations for which connectivity parameters can be in a file, such as configurations with Kafka targets.
User Response:	No response is required.
CDCPUB_15016	The process <i>[process_name]</i> was unable to validate the Kafka partition ID <i>[partition_ID]</i> for the topic <i>[topic_name]</i>. Error: <i>[error_description]</i>.
Explanation:	The internal process that is responsible for Kafka handling found that the specified Kafka partition ID that is assigned to a message is not valid for the configured Kafka topic.
User Response:	Change the Kafka topic definition to support the partition ID, or change the partition configuration to match the existing Kafka topic definition. Also, to diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_15017	The process <i>[process_name]</i> was processing an event with these details when an error occurred: <i>[event_information]</i>.
Explanation:	The internal process that is responsible for Kafka communication encountered an error. This message reports the contents of the event or message that was sent to Kafka.
User Response:	This message typically accompanies additional messages that describe the error. To diagnose and correct the error, analyze the accompanying log messages. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_15018	The process <i>[process_name]</i> encountered an error while sending a message to Kafka. Error: <i>[error_description]</i>.
Explanation:	The internal process that is responsible for Kafka communication encountered the specified error while sending data to Kafka.
User Response:	After you review the error description, determine if you need to check the availability or configuration of the target Kafka environment or make changes to the CDC Publisher configuration. Also, analyze the accompanying log messages for additional diagnostic information. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_15019	The process <i>[process_name]</i> received an <i>[Kafka_exception_type]</i> exception while sending a message to Kafka for the event <i>[event_information]</i>. Error: <i>[error_description]</i>.
Explanation:	The internal process that is responsible for sending messages to Kafka encountered the specified error. This message includes the exception and event information that was returned from Kafka.

User Response: After you review the error information, determine if you need to check the configuration of the target Kafka environment or make changes to the CDC Publisher configuration. Also, analyze the accompanying log messages for additional diagnostic information. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_15020 **The process [process_name] is trying to resend the previous message to Kafka because of an error. Error: [error_description].**

Explanation: This informational message reports that the internal process that is responsible for sending messages to Kafka encountered the specified error. The process will retry sending the message until either the send operation is successful or the maximum number of retries is exceeded.

User Response: No response is required.

CDCPUB_15021 **The process [process_name] was unable to send the previous message to Kafka after [number_of_attempts] attempts because of the error: [error_description].**

Explanation: The internal process that is responsible for Kafka communication encountered the specified error while trying to resend a message with data to Kafka. After the maximum number of retries was exceeded, the process ended. This message includes the error description and number of retry attempts.

User Response: After you review the error description, determine if the error is related to the availability or configuration of the Kafka target environment. Edit the configuration if necessary. Also, analyze the accompanying log messages to diagnose the error. If the problem persists, contact Informatica Global Customer Support.

CDCPUB_15022 **The process [process_name] received the following Kafka producer properties.**

Explanation: This informational message introduces the list of Kafka properties that the internal process that is responsible for Kafka communication uses.

User Response: No response is required.

CDCPUB_15023 **Kafka property key [property_name] with value [property_value].**

Explanation: The internal process that is responsible for Kafka communication is using the specified Kafka property value. This informational message is repeated for each Kafka property.

User Response: No response is required.

CDCPUB_15024 **The process [process_name] is replacing the existing Kafka property [property_name] and value [property_value] with the new property [new_property_name] and value [new_property_value].**

Explanation: The internal process that is responsible for Kafka communication is replacing an incompatible Kafka property value. The property is required to connect to Kafka. Normally, the properties that are used to connect to Kafka are provided in a file, such as a Kafka producer properties file. If a Kafka property value is incompatible, the process replaces it. This informational message identifies the existing property name and value that is being replaced and the new property name and value.

User Response: No response is required.

CDCPUB_15025	The process [process_name] is adding the Kafka property [property_name] with the value [property_value] to support the Guaranteed Delivery configuration setting.
Explanation:	The internal process that is responsible for Kafka communication is adding a Kafka property value to the properties list that it uses. Normally, the properties that are used to connect to Kafka are provided in a file, such as the Kafka Producer properties file. If this process is configured to use Guaranteed Delivery of messages to Kafka, the process might add Kafka properties to meet the Guaranteed Delivery requirement. This informational message identifies a Kafka property and value that is being added to support Guaranteed Delivery.
User Response:	No response is required.
CDCPUB_15026	The process [process_name] is changing the Kafka property [property_name] with the value [property_value] to the new value [new_property_value] to support Guaranteed Delivery.
Explanation:	The internal process that is responsible for Kafka communication is changing the value of a Kafka property to support the Guaranteed Delivery setting in the configuration. Normally, the properties that are used to connect to Kafka are provided in a file, such as the Kafka Producer properties file. If this process has been configured for Guaranteed Delivery of messages to Kafka, the process might adjust the input Kafka properties to meet the Guaranteed Delivery requirement. This informational message identifies a Kafka property value that is being changed to another value to support Guaranteed Delivery.
User Response:	No response is required.
CDCPUB_15027	The process [process_name] is adding the Kafka property [property_name] with the default value of [property_value].
Explanation:	The internal process that is responsible for Kafka communication is adding the specified Kafka property with the default value to the properties list that it uses. Normally, the properties that are used to connect to Kafka are provided in a file, such as a Kafka Producer properties file. If a required Kafka property is missing, the process will add it. This message indicates the Kafka property is being added and its default value.
User Response:	If the default value is not appropriate, add the configuration parameter with another value to the Producer properties file.
CDCPUB_15028	An attempt to connect to Kafka was unsuccessful because of a Class Not Found exception. A third-party library is probably missing from the generated Java classpath. Error: [error_description].
Explanation:	The internal process that is responsible for Kafka communication tested a connection to Kafka but the test failed because of a Class Not Found exception. This error is not fatal. This informational message is issued only when the process tests connectivity to or availability of the Kafka instance and the test fails.
User Response:	Typically, this message indicates that not all of the Kafka client library files have been added to the generated classpath. Review the product documentation to determine how to properly add the Kafka client libraries to the environment. Then ensure that all of the Kafka libraries have been included.

CDCPUB_15029	A test connection to Kafka was unsuccessful because of a TimeOut exception. The Kafka broker might be down or unreachable, or the Kafka Producer properties file might be incorrect. Error description: [error_description].
Explanation:	The internal process that is responsible for Kafka communication tested a connection to Kafka but the test failed because of a TimeOut exception. This error is not fatal. This informational message is issued only when the process tests connectivity to or availability of the Kafka instance. The message might indicate that Kafka is not available or that a time-out property in the Kafka Producer properties file is set too low for the environment.
User Response:	No response is required.
CDCPUB_15030	A test connection to Kafka was unsuccessful. The Kafka broker might be down or unreachable, or the Kafka Producer properties file might be incorrect. Error description: [error_Description].
Explanation:	The internal process that is responsible for Kafka communication tested a connection to Kafka but the test failed. This error is not fatal. This informational message is issued only when the process tests connectivity to or availability of the Kafka instance and the test fails. The Kafka broker might be down or unreachable, or the Kafka Producer properties file might contain an error.
User Response:	Verify that the Kafka properties are correctly specified and that the Kafka instance is reachable in the current environment. If the problem persists, contact Informatica Global Customer Support for assistance.
CDCPUB_15031	The process [process_name] requested partition information from Kafka for the topic [topic_name], but the request failed with Kafka error: [error_description].
Explanation:	The internal process that is responsible for Kafka communication encountered the specified error while requesting partition information from Kafka for a topic.
User Response:	After you review the error description, determine if the error is related to the availability or configuration of the Kafka target environment or if the topic or its partitions are not available on the Kafka server specified in the Kafka Producer properties file. Also, analyze the accompanying log messages to diagnose the error. If the problem persists, contact Informatica Global Customer Support.
CDCPUB_15032	The CDC Publisher did not update any Kafka topics since the last reset.
Explanation:	This informational message reports that the CDC Publisher did not update any Kafka topics during the current run or interval. Either no data was available to process, or the data is not newer than the last record that the CDC Publisher sent to Kafka.
User Response:	No response is required.
CDCPUB_15033	The CDC Publisher updated the following Kafka topics:
Explanation:	This informational message introduces a list of the updated Kafka topics.
User Response:	No response is required.

CDCPUB_15034	The CDC Publisher sent <i>[count]</i> messages to the Kafka topic <i>[topic_name]</i> partition <i>[partition_ID]</i>. The last message was written to the topic partition offset <i>[offset]</i> with the key <i>[Kafka_key_value]</i> and includes the source table <i>[table_name]</i> operation <i>[dml_operation]</i> from the stream sequence <i>[sequence_value]</i>.
Explanation:	This informational message reports the number of messages that the CDC Publisher sent to the specified Kafka topic partition. This message also includes detailed information about the last message written to the topic, including its offset in the topic partition, key value, and the source table operation it contains.
User Response:	No response is required.
CDCPUB_15035	An event of type <i>[event_type]</i> for the table <i>[table_name]</i> at the stream sequence <i>[event_sequence_value]</i> recorded the following detailed latency statistics:
Explanation:	This informational message introduces latency information for the event that is associated with the specified table and located at the specified sequence position in the change stream.
User Response:	No response is required.
CDCPUB_15036	The process <i>[process_name]</i> performed <i>[action]</i> at UNIX Epoch time <i>[time]</i> milliseconds. Duration: <i>[duration_of_action]</i> milliseconds. Additional information: <i>[information]</i>.
Explanation:	<p>This informational message reports the following information about a process that affected the Change Data Capture event that is identified in message CDCPUB-15035.</p> <ul style="list-style-type: none"> • The action the process was performing on the event. • The UNIX Epoch time at which the process performed the action on the event. Epoch time is the number of milliseconds between the current time and midnight January 1, 1970 UTC. • The duration, in milliseconds, that has elapsed since the prior action. • Additional information about the process or environment at the time of the action. <p>This information is typically used to determine which processes require the most time to perform actions on an event.</p>
User Response:	No response is required.
CDCPUB_15037	The total latency for this Event was <i>[number_of_milliseconds]</i> milliseconds.
Explanation:	This informational message provides the total time, in milliseconds, that the CDC Publisher took to process the Change Data Capture event identified in message CDCPUB-15035.
User Response:	No response is required.
CDCPUB_15038	The Connector <i>[process_name]</i> will start sending messages to a target entity from the checkpoint <i>[checkpoint_value]</i>.
Explanation:	The Connector internal helper process that is responsible for sending messages to an external target entity keeps a record of its progress, called a checkpoint. This informational message reports the initial checkpoint value from which the specified Connector process will start sending messages.
User Response:	No response is required.

CDCPUB_15039	The Connector process [process_name] with the thread ID [thread_ID] has processed [number_of_operations] operations.
Explanation:	This informational message reports the number of change operations that the Connector process received. The message can be issued periodically to indicate the streaming activity.
User Response:	No response is required. You can configure the frequency at which this message is issued. If you need assistance, contact Informatica Global Customer Support.
CDCPUB_15040	The CDC Publisher wrote [count] messages to file [file_name].
Explanation:	This informational message reports the number of messages that the CDC Publisher wrote to the specified output file since the last RESET operation.
User Response:	No response is required.
CDCPUB_15041	The process [process_name] determined Kafka serialization settings are expecting message objects of type [object_type].
Explanation:	The process determined that the Kafka target is expecting message objects of the specified type. Kafka can use different serializers to support message objects of different types, including strings and byte arrays. The internal process that is responsible for Kafka communication determined the specified type is appropriate for the current environment.
User Response:	No response is required. If you do not want to use the specified message object type, check the value.serializer setting in your Kafka producer.properties file, which is used to make this determination. Alternatively, configure the process to not automatically detect the message object type based on the producer.properties file.
CDCPUB_15042	The process [process_name] will create and send message objects of type [object_type] to Kafka.
Explanation:	The internal process that is responsible for Kafka communication is configured to deliver message objects of the specified type to Kafka.
User Response:	No response is required. If you do not want to use the specified message object type, edit the appropriate configuration property to use a different Kafka object type.
CDCPUB_15043	The process [process_name] has loaded [count] topic definitions from the configuration for this environment.
Explanation:	The internal process that is responsible for communication with the target has been configured to use the specified number of Kafka topics.
User Response:	No response is required. If the number of topics is different from what you expected, check the topic configuration options to verify that all topic definitions have been configured.
CDCPUB-15051	The process [process_name] with the thread ID [thread_ID] is waiting for a response on its internal notification queue. Current wait duration: [hh:mm:ss].
Explanation:	This informational message reports that the internal process that is responsible for processing change data is idle because it is waiting to notify another process that work is available to do. The message reports the wait time up to the current point in time. This condition typically occurs when the process cannot find an available helper thread because all helper threads are busy.
User Response:	No response is required.

CDCPUB_16001 **The command `[command_name]` was unsuccessful. Error description: `[error_description]`.**

Explanation: This error message is typically issued by simple command-line tools to indicate that a command failed to run. The message includes a description of the error.

User Response: Review the system log that the command-line tool produces for more messages related to the problem.

CPPSVC_00100 to CPPSVC_00103

CPPSVC_00100 **No localized message factory provided. Message code `[message_code]`. Params: `[parameters]`.**

Explanation: An internal logic error occurred during CDC Publisher processing. A C++ service attempted to generate a localized message that is associated with the specified message code and parameters, but no localized message factory exists. This message is generated in place of the localized message.

User Response: Contact Informatica Global Customer Support.

CPPSVC_00101 **The localized message with code `[message_code]` was not found. Params: `[parameters]`.**

Explanation: An internal logic error occurred during CDC Publisher processing. A C++ service attempted to generate a localized message but did not find a localized message that is associated with the specified message code and parameters. This message is generated in place of the localized message.

User Response: Contact Informatica Global Customer Support.

CPPSVC_00102 **The process `[process_name]` with thread ID `[thread_ID]` encountered error code `[error_code]` with description `[description]`.**

Explanation: During CDC Publisher processing, a C++ service encountered an internal error related to a Java exception. This message provides the process name, associated thread number, and error information returned by Java. The error code is always 0.

User Response: Contact Informatica Global Customer Support.

CPPSVC_00103 **Stack trace `[stack]`.**

Explanation: During CDC Publisher processing, the stack trace associated with a C++ service encountered an unexpected error.

User Response: Contact Informatica Global Customer Support.

CPPSVC_00104 **The request to restart from the end of the PowerExchange CDC Stream identified the following position as the current end of the log `:position`**

Explanation: This informational message identifies the restart position that has been calculated when a restart from the end of the PowerExchange change data capture stream has been requested.

User Response: If you do not want to restart from this position, you can reconfigure the environment to restart from a specific location. If necessary, contact Informatica Global Customer Support.

LOGCFG_10001 to LOGCFG_10006

LOGCFG_10001	The Informatica logging component could not create the GMD appender.
Explanation:	The Informatica logging component failed to create the appender that is responsible for guaranteed message delivery.
User Response:	Analyze the accompanying log messages to diagnose the error and then try to correct the error. If the problem persists, contact Informatica Global Customer Support.
LOGCFG_10002	The Informatica custom Log4j configuration does not include the required appender [appender_name].
Explanation:	The Informatica logging component could not find the specified appender defined in the Log4j configuration file. This appender is required.
User Response:	Add the missing configuration statements for the appender to the Log4j configuration file. If the problem persists, contact Informatica Global Customer Support.
LOGCFG_10003	The Informatica logging component [component_name] references the appender [appender_name] but this appender is not in the Log4j configuration file. The reference is ignored.
Explanation:	The Informatica logging component could not find the specified appender defined in the Log4j configuration file.
User Response:	In the Log4j configuration file, add or correct the configuration statements for the missing appender or remove the AppenderRef reference for the appender from the Logger specification. If the problem persists, contact Informatica Global Customer Support.
LOGCFG_10004	The Informatica Logger name [logger_name] in the logging configuration conflicts with the reserved name [reserved_name]. The Logger specification is ignored.
Explanation:	The Informatica logging component determined that the Logger name that is specified in the logging configuration file matches a reserved name. Because the Logger cannot have a reserved name, the Logger specification is ignored.
User Response:	In the logging configuration file, change the Logger name to a name that is not reserved. If the problem persists, contact Informatica Global Customer Support.
LOGCFG_10005	The Log4j configuration file [file_name] could not be read.
Explanation:	The Informatica logging component could not read the specified Log4j configuration file.
User Response:	Verify that the file exists and is readable. If the problem persists, contact Informatica Global Customer Support.
LOGCFG_10006	Could not create the Log4j context. Result was null. Does a configuration error exist?
Explanation:	The Informatica logging component failed to create the Log4j context object.
User Response:	Verify that the Log4j configuration properties are valid. If the problem persists, contact Informatica Global Customer Support.

PWXLIC_00110 to PWXLIC-00999

PWXLIC_00110	Key error. License key [key_value] has a check digit error. Expected check digit [expected_check_digit] but got calculated check digit [calculated_check_digit].
Explanation:	The license key is not valid. The check digit is incorrect.
User Response:	Enter a valid license key.
PWXLIC_00111	Key error. License key length must be 44 or 64 characters. Key [key_value] has length [number_of_characters].
Explanation:	The license key length is not valid. The length must be either 44 or 64 characters.
User Response:	Specify a license key value that has a valid length.
PWXLIC_00112	Key error. Hyphen characters in license key [key_value] are not in expected locations.
Explanation:	The license key is not valid because it contains hyphens in locations that are not valid within the hexadecimal key values.
User Response:	Specify a valid license key that includes hyphens in locations that are valid within hexadecimal key values.
PWXLIC_00113	PowerExchange licensing check found I/O error related to file [key_file_name] operation [operation]. Error description:[error_description].
Explanation:	An I/O error occurred while PowerExchange was trying to perform a licensing check on the specified license key file. The message reports the file operation in progress when the error occurred and a description of the error. Operation types include open, read, and close.
User Response:	Verify that the file name is correct.
PWXLIC_00115	Key error. License key [key_value] expired on date [expiration_date]. Current date is [current_date].
Explanation:	The license key expired on the specified date.
User Response:	Specify a new license key.
PWXLIC_00116	Key error. License key [key_value] has [number_of_days] days before expiration.
Explanation:	The license key will expire in the specified number of days.
User Response:	Specify a new license key before the current key expires.
PWXLIC_00118	Key error. License key [key_value] does not match the platform [platform_type].
Explanation:	The specified license key was issued for a platform other than the platform on which the product will run.
User Response:	Get a license key that covers the specified platform.
PWXLIC_00121	Not licensed for feature [feature].
Explanation:	Your license key does not cover the specified feature.
User Response:	If you need to use the feature, get a license key that allows its use.

PWXLIC_00122	Key error. License key [key_value] has a vendor code mismatch. Key vendor code [key_vendor_code] does not match the software vendor code [software_vendor_code].
Explanation:	The license key is not valid because it has been issued for a vendor other than the vendor for which the software has been supplied.
User Response:	Verify that you are using the correct license key, which covers the vendor of the supplied software. If you need to get a new key, contact Informatica Global Customer Support.
PWXLIC_00800	Key error. License key [key_value] is not a valid hexadecimal value because it contains the character [character].
Explanation:	The license key is not valid because it contains the specified non-hexadecimal character. The license key must be a hexadecimal value.
User Response:	Specify a license key that is a valid hexadecimal value. Ensure the key value does not contain the specified non-hexadecimal character.
PWXLIC_00801	Key error. The 64-character license key [key_value] contains an indicator that identifies the key as invalid.
Explanation:	The 64-character license key is not valid, as indicated by a bit in the key value.
User Response:	Specify a valid long license key.
PWXLIC_00802	License environment variable [variable_name] is missing or has no value.
Explanation:	The environment variable that points to the PowerExchange license key file is not present or does not contain a value.
User Response:	Specify the correct path to the license key file in the environment variable.
PWXLIC_00999	PowerExchange license checking encountered the internal logic error [error_description].
Explanation:	An internal logic error occurred during PowerExchange license checking. This message includes a description of the error.
User Response:	Contact Informatica Global Customer Support.

PWXRDR-01000 to PWXRDR-01301

PWXRDR_01000	The PWX Reader has initialized. Version [version].
Explanation:	The PWX Reader Service has initialized. This message includes the PWX Reader version.
User Response:	No response is required.
PWXRDR_01001	The PWX Reader is terminating.
Explanation:	The PWX Reader service is terminating. This message does not indicate an error condition.
User Response:	No response is required.
PWXRDR_01002	Required PWX Reader configuration key [key_name] is missing.
Explanation:	The specified PWX Reader configuration key name is required but is missing.

User Response:	Enter the configuration key name.
PWXRDR_01003	PWX Reader configuration key <i>[key_name]</i> has an invalid value <i>[value]</i>.
Explanation:	An invalid value is specified for the specified PWX Reader configuration key.
User Response:	Correct the configuration key value.
PWXRDR_01004	PWX Reader configuration key <i>[key_name]</i> with value <i>[value]</i> conflicts with configuration key <i>[key_name]</i> with value <i>[value]</i>.
Explanation:	The PWX Reader configuration keys that are identified in this message have conflicting values.
User Response:	Correct one or both of the configuration key values so that they are compatible.
PWXRDR_01005	PWX Reader detected that no tables are selected for change data extraction.
Explanation:	The PWX Reader configuration does not match any tables from which PowerExchange captures changes. This situation might be caused by user-specified filtering criteria or an invalid extraction map schema name.
User Response:	Correct the PWX Reader configuration so that it matches at least one table from which PowerExchange captures changes.
PWXRDR_01006	PWX Reader termination has waited <i>[seconds]</i> for the following methods to complete: <i>[methods]</i>.
Explanation:	The PWX Reader has waited the specified number of seconds to terminate. It cannot terminate until the specified methods complete processing.
User Response:	No response is required.
PWXRDR_01007	PWX Reader configuration key <i>[old_key_name]</i> is deprecated. Use the new configuration key <i>[new_key_name]</i>.
Explanation:	The specified PWX Reader configuration key has been deprecated. Use the new configuration key that this message supplies.
User Response:	Enter a value for the new key in the configuration.
PWXRDR_01100	Connecting to PowerExchange location <i>[location]</i> to collect metadata.
Explanation:	The PWX Reader service is connecting to the specified PowerExchange location to collect metadata about the source objects for which to extract change data.
User Response:	No response is required.
PWXRDR_01101	Connecting to the PowerExchange location <i>[location]</i> for change data extraction.
Explanation:	The PWX Reader service is connecting to the specified PowerExchange location to extract change data.
User Response:	No response required.
PWXRDR_01102	Connection is using the CAPI_CONNECTION <i>[capi_connection_name]</i>.
Explanation:	The PWX Reader service is configured to use the specified CAPI_CONNECTION statement to extract change data from PowerExchange. The CAPI_CONNECTION statement is defined in the PowerExchange DBMOVER configuration file on the PowerExchange Listener system.

User Response: No response is required.

PWXRDR_01104 Connection is using the PowerExchange extraction map schema name [*schema_name*].

Explanation: The PWX Reader service is configured to use the specified schema name portion of extraction map name to identify the extraction maps to use for reading change data from source tables. PowerExchange generates this schema name when creating extraction maps.

User Response: No response is required.

PWXRDR_01105 Connecting to PowerExchange location [*location*] to get current restart position.

Explanation: The PWX Reader service is connecting to the specified PowerExchange location to determine the current restart position for extraction processing.

User Response: No response is required.

PWXRDR_01106 Restart position [*hex_restart_position*].

Explanation: This message reports the restart position of PWX Reader extraction processing in the PowerExchange change stream. The restart position is expressed as a hexadecimal restart token.

User Response: No response is required.

PWXRDR_01107 Connection is using the PowerExchange SCLI connect string [*connect_string*].

Explanation: The PWX Reader service is using the specified SCLI connect string to connect to PowerExchange.

User Response: No response is required.

PWXRDR_01110 Extractions maps selected for [*count*] source tables:

Explanation: After collecting metadata about the source tables, the PWX Reader identified the extraction maps to use for extracting change data for the specified number of source tables. The extraction map names are listed in subsequent PWXRDR-1111 messages.

User Response: No response is required.

PWXRDR_01111 Extraction map [*extraction_map_name*].

Explanation: The PWX Reader service of the PowerExchange CDC Publisher is extracting change data for the source table that is associated with the specified extraction map. The *extraction map name* has the format *xninstance.mapname.tablename*.

User Response: No response is required.

PWXRDR_01112 Extraction map [*extraction_map_name*] primary key columns [*column_name1,column_name2,...*]

Explanation: The PWX Reader service is extracting change data for the source table that is associated with the specified extraction map. This message identifies the primary key columns for the table.

User Response: No response is required.

- PWXRDR_01113** **Not using extraction map [extraction_map_name] for change data extraction because it contains no qualifying columns.**
- Explanation: The PWX Reader service will not extract change data based on the specified extraction map because the extraction map contains no columns that qualify for extraction processing.
- User Response: No response is required.
- PWXRDR_01114** **Not using extraction map [extraction_map_name] for change data extraction because column filtering has excluded all columns in the map.**
- Explanation: The PWX Reader service will not extract change data based on the specified extraction map because column filtering has excluded all columns in the map from extraction processing.
- User Response: No response is required.
- PWXRDR_01115** **Primary key column [column_name] in extraction map [extraction_map_name] is not being extracted because of filtering criteria.**
- Explanation: Data is not being extracted from the specified primary key column based on your configuration.
- User Response: No response is required.
- PWXRDR_01200** **Beginning of PowerExchange informational messages.**
- Explanation: This message introduces a set of informational messages that PowerExchange is returning to the PWX Reader.
- User Response: No response is required.
- PWXRDR_01201** **[info_message_text]**
- Explanation: The PWX Reader received the specified informational message from PowerExchange.
- User Response: No response is required.
- PWXRDR_01202** **End of PowerExchange informational messages.**
- Explanation: This message indicates the end of a set of informational messages that PowerExchange returned to the PWX Reader service.
- User Response: No response is required.
- PWXRDR_01300** **Command [command] received.**
- Explanation: The PWX Reader service received the specified command.
- User Response: No response is required.
- PWXRDR_01301** **Command [command] not recognized.**
- Explanation: The PWX Reader service did not recognize the specified command as a valid command.
- User Response: No response is required.

INDEX

A

- administering PowerExchange CDC Publisher
 - overview of administrative tasks [75](#)
- and custom patterns
 - and the Avro Formatter [104](#)
- Apache Kafka targets
 - data-streaming considerations [38](#)
 - overview [37](#)
 - preparing the target environment [37](#)
 - using Confluent Schema Registry [39](#)
- Avro Formatter
 - and custom patterns [104](#)
- Avro schemas
 - reporting generated schemas for source tables [76](#)

B

- boolean values
 - examples [102](#)

C

- checkpoint files [29](#)
- column attributes
 - custom patterns [95](#)
- command reference
 - commands for the command-line utilities [82](#)
 - PwxCDCInfo command [84](#)
 - PwxCDCPublisher command [83](#)
- command-line utilities
 - commands reference [82](#)
 - PwxCDCInfo command syntax [84](#)
 - PwxCDCPublisher command syntax [83](#)
- configuration files
 - cdcPublisherAvro.cfg [45](#)
 - cdcPublisherCommon.cfg [45](#)
 - cdcPublisherKafka.cfg [51](#)
 - cdcPublisherPowerExchange.cfg [56](#)
 - customizing the sample files [44](#)
- configuration properties
 - Connector properties for Kafka and MapR targets [51](#)
 - Custom Pattern Formatter [51](#)
 - properties for Avro schema format and message encoding [45](#)
 - properties for extracting data from PowerExchange [56](#)
 - properties for sending changes to multiple partitions [62](#)
 - properties for the PowerExchange CDC Publisher host and port [45](#)
- Configuring PowerExchange CDC Publisher
 - overview of configuration tasks [43](#)
- Confluent Schema Registry
 - use in a Kafka target environment [39](#)
- custom pattern
 - JSON output [108](#)
- custom pattern file syntax [94](#)

- custom pattern files [65](#)
- Custom Pattern Formatter
 - configuration properties [51](#)
 - configuring [65](#)
- custom patterns
 - column attributes [95](#)
 - event tags [100](#)
 - section tags [95](#)
 - special attributes [101](#)
 - special pattern tags [102](#)
 - usage considerations [94](#)
 - validating [109](#)

D

- data streaming
 - before starting PwxCDCPublisher streaming [68](#)
 - operational considerations [69](#)

E

- encrypting a password [79](#)
- environment variables
 - configuring [23](#)
- event tags
 - custom patterns [100](#)
- examples
 - boolean values [102](#)
 - Custom Formatter Patterns configuration file [65](#)
 - custom pattern files [65](#)
 - JSON [104](#)
 - null values [102](#)
 - repeat tags [103](#)
- extraction maps
 - filtering extraction maps and source objects [59](#)

G

- guaranteed delivery [29](#)

I

- Informatica Data Engineering Streaming
 - integration considerations [66](#)
- installation
 - installing on Linux or Windows [21](#)
 - overview [19](#)

J

- JSON examples
 - custom patterns [104](#)
- JSON output
 - Custom Pattern Formatter [108](#)

K

- Kafka headers
 - for checkpoint [30](#)
- Kafka targets
 - preparing the target environment [37](#)
 - using Confluent Schema Registry [39](#)
- Kerberos authentication
 - configuring for Kafka server communication [40](#)

L

- Log4j 2 message logging
 - configuring for the PowerExchange CDC Publisher [61](#)

M

- MapR Streams targets
 - overview [40](#)
- MapR targets
 - data-streaming considerations [42](#)
 - preparing the target environment [40](#)
- message keys
 - generating based on rules [62](#)
- message logging
 - configuring Log4j 2 logging [61](#)
- messages for data delivery
 - message content and format [26](#)
- monitoring PowerExchange CDC Publisher
 - overview [71](#)
- multiple-partitioned topics
 - sending changes to multiple partitions [62](#)

N

- null values
 - examples [102](#)

P

- password encryption
 - using the PwxCDCAdmin utility [79](#)
- PowerExchange CDC Publisher utilities
 - overview [17](#)
- PowerExchange environment
 - alternative configurations with CDC Publisher [13](#)
 - configuration tasks [34](#)
 - configuring connectivity to PowerExchange [35](#)
- product overview [9](#)
- PwxCDCAdmin utility
 - adding a source table [78](#)
 - clearing Avro schemas [76](#)

- PwxCDCAdmin utility (*continued*)
 - encrypting a password [79](#)
 - performing administrative tasks [75](#)
 - regenerating Avro schemas [76](#)
 - reporting Avro schemas for source tables [76](#)
 - shutting down the CDC Publisher [80](#)
- PwxCDCInfo utility
 - generating statistics for topics and processes [73](#)
 - reporting the status of CDC Publisher processes [72](#)
- PwxCDCPublisher utility
 - operational considerations [69](#)
 - prerequisites to starting [68](#)
 - restarting change stream processing [80](#)
- PwxCDCPublisher utility
 - starting a CDC Publisher process [68](#)

R

- repeat tags
 - examples [103](#)
- repeating group tags [103](#)
- restart
 - apply processing restart [31](#)
 - CDC Publisher restart in change stream [31](#)
 - restarting the PwxCDCPublisher utility in the input change stream [80](#)

S

- section tags [95](#)
- shutting down the CDC Publisher
 - with the PwxCDCAdmin utility [80](#)
- source tables
 - adding a table to a change stream [78](#)
 - handling changes to source table definitions [76](#)
- special attributes [101](#)
- special pattern tags [102](#)
- starting a CDC Publisher process
 - with the PwxCDCPublisher utility [68](#)
- statistics
 - generating with the PwxCDCInfo utility [73](#)
- status reporting
 - for CDC Publisher processes [72](#)
- syntax
 - custom pattern file [94](#)

T

- target message output
 - customizing [64](#)

U

- usage considerations [94](#)
- utilities
 - commands reference [82](#)
 - overview of PowerExchange CDC Publisher utilities [17](#)
 - PwxCDCInfo command syntax [84](#)
 - PwxCDCPublisher command syntax [83](#)