



Informatica® PowerExchange
10.5.5

CDC Guide for i5/OS

Informatica PowerExchange CDC Guide for i5/OS

10.5.5

February 2024

© Copyright Informatica LLC 2008, 2024

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, PowerCenter, 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.

Subject to your opt-out rights, the software will automatically transmit to Informatica in the USA information about the computing and network environment in which the Software is deployed and the data usage and system statistics of the deployment. This transmission is deemed part of the Services under the Informatica privacy policy and Informatica will use and otherwise process this information in accordance with the Informatica privacy policy available at <https://www.informatica.com/in/privacy-policy.html>. You may disable usage collection in Administrator tool.

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

See patents at <https://www.informatica.com/legal/patents.html>.

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: 2024-02-09

Table of Contents

Preface	8
Informatica Resources.	8
Informatica Network.	8
Informatica Knowledge Base.	8
Informatica Documentation.	9
Informatica Product Availability Matrices.	9
Informatica Velocity.	9
Informatica Marketplace.	9
Informatica Global Customer Support.	9
Chapter 1: Change Data Capture Introduction.....	10
PowerExchange CDC Overview.	10
Planning Considerations.	11
PowerExchange CDC Components.	11
PowerExchange CDC Architecture.	12
PowerExchange Condense Processing.	13
Rollbacks or Backouts.	13
Multiple Changes to the Same Record.	14
Change Data Extraction.	14
Extraction Maps.	14
Extraction Process.	14
PowerExchange Integration with PowerCenter.	14
Chapter 2: PowerExchange Listener.....	16
PowerExchange Listener Overview.	16
PowerExchange Listener Message Log Files.	16
Threadsafe Processing Requirement.	17
Configuring the PowerExchange Listener.	17
Summary of DBMOVE Statements for CDC.	18
AS400EVENTMSGQ Statement.	18
AS400USRJRNCODE Statement.	19
AS400USRMSGQ Statement.	20
CAPI_CONNECTION Statements.	20
CAPI_CONN_NAME Statement.	31
CPX_DIR Statement.	32
Starting PowerExchange Listener.	32
Stopping the PowerExchange Listener.	33
Stopping a PowerExchange Listener Task for an Extraction.	33
Displaying Active PowerExchange Listener Tasks.	33

Chapter 3: PowerExchange Condense.....	35
PowerExchange Condense Overview.	35
PowerExchange Condense Tasks.	36
Controller Task.	37
Condense Task and Condense Cycles.	37
Command Handler Task.	38
Files That PowerExchange Condense Uses.	38
Synchronization of the Checkpoint File to the CDCT File.	40
PowerExchange Condense Message Log Files.	40
PowerExchange Condense Operational Modes.	41
PowerExchange Condense Configuration Parameters.	42
QPRINT File Size Limits.	47
Using Multiple Journals with PowerExchange Condense.	47
Starting PowerExchange Condense.	48
Managing PowerExchange Condense.	49
PowerExchange Condense Progress Messages and Tracing.	49
Displaying PowerExchange Condense Status.	49
Stopping PowerExchange Condense.	50
Performing a Fileswitch.	50
Determining If PowerExchange Condense Captured Changes.	51
Backing Up PowerExchange Condense Output Files.	51
Generating PowerExchange Condense Performance Statistics.	51
 Chapter 4: DB2 for i5/OS Change Data Capture.....	 52
DB2 for i5/OS Change Data Capture Overview.	52
Planning for DB2 Change Data Capture.	53
DB2 for i5/OS CDC Prerequisite.	53
DB2 Datatypes Supported for CDC.	53
DB2 for i5/OS CDC Considerations.	54
i5/OS Security Requirements.	55
Configuring DB2 Journaling for Change Data Capture.	56
Availability of Journal Receivers for DB2 CDC.	57
Using Remote Journals with PowerExchange.	59
Configuring PowerExchange for DB2 Change Data Capture.	61
PowerExchange Event Message Queue.	61
Configuring PowerExchange CDC without PowerExchange Condense.	62
Configuring PowerExchange CDC with PowerExchange Condense.	63
Managing DB2 Change Data Capture.	64
Stopping DB2 CDC.	64
Changing a DB2 Table Definition.	64
Determining the Journal Receiver Needed for Restart Processing.	65
Refreshing the CDC Interest List After Adding or Deleting Libraries.	66

Refreshing the PowerExchange Environment After an i5/OS Upgrade.	71
Generating SQL Statements to Re-create a Source or Target for Troubleshooting.	71
Chapter 5: Remote Logging of Data.	73
Remote Logging Overview.	73
Remote Logging of Data from Sources on IBM i or z/OS Systems.	73
Requirements for Capture Registrations.	76
Configuration Tasks for Remote Logging.	76
Customizing the PowerExchange Logger Configuration File for Logging Data from a Remote Source.	77
Customizing the dbmover File on the System to Which Data Is Logged.	78
Customizing the dbmover Configuration File on the PowerCenter Integration Service System.	79
Configuring Capture Registrations for the PowerExchange Logger.	79
Configuring PowerCenter Connection Attributes for Extracting Data from the Log Files.	80
Example of Remote Logging from a Db2 for i Data Source.	80
Chapter 6: Introduction to Change Data Extraction.	83
Change Data Extraction Overview.	83
Extraction Modes.	84
PowerExchange-Generated Columns in Extraction Maps.	84
Uses of BI and CI Fields in Extraction Maps.	90
Restart Tokens and the Restart Token File.	91
Multiple-Source Processing in CDC Sessions.	92
Commit Processing with PWXPC.	93
Tuning Options.	95
Chapter 7: Extracting Change Data.	96
Overview of Extracting Change Data.	96
Task Flow for Extracting Change Data.	97
Testing an Extraction Map.	98
Configuring PowerCenter CDC Sessions.	99
Changing Default Values for Session and Connection Attributes.	99
Configuring Application Connection Attributes.	100
Examples of Controlling Commit Processing.	109
Recovery and Restart Processing for CDC Sessions.	111
PowerCenter Recovery Tables for Relational Targets.	112
PowerCenter Recovery Files for Nonrelational Targets.	113
Application Names.	114
Restart Processing for CDC Sessions by Start Type.	114
Creating Restart Tokens for Extractions.	116
Displaying Restart Tokens.	117
Configuring the Restart Token File.	117
Restart Token File Statements.	118

Example Restart Token File.	121
Chapter 8: Monitoring CDC Sessions.	123
Monitoring Overview.	123
Monitoring CDC Sessions in PowerExchange.	123
Read Progress Messages.	124
Extraction Statistics Messages.	124
Multithreaded Processing Statistics.	125
PowerExchange Listener DISPLAY ACTIVE or LISTTASK Command.	126
PowerExchange Listener DISPLAYSTATS Command.	126
Monitoring CDC Sessions in PowerCenter.	128
Session Log Messages.	129
Performance Details in Workflow Monitor.	129
Viewing Performance Details in Workflow Monitor.	131
Chapter 9: Managing Change Data Extractions.	133
Starting PowerCenter CDC Sessions.	133
Cold Start Processing.	134
Warm Start Processing.	134
Recovery Processing.	135
Stopping PowerCenter CDC Sessions.	136
Stop Command Processing.	136
Terminating Conditions.	137
Changing PowerCenter CDC Sessions.	137
Examples of Adding Sources and Creating Restart Tokens.	138
Recovering PowerCenter CDC Sessions.	140
Example of Session Recovery.	140
Chapter 10: Tuning CDC Sessions.	142
Tuning Overview.	142
PowerExchange DBMOVE Statements for Tuning CDC Sessions.	143
PowerCenter Connection Attributes and Session Properties.	146
PowerCenter Connection Attributes for Tuning CDC Sessions	146
PowerCenter Session Properties for Tuning Buffer Memory.	149
CDC Offload Processing.	149
Rules and Guidelines for CDC Offload Processing.	150
Enabling Offload Processing for CDC Sessions.	150
Example of CDC Offload Processing with a DB2 for i5/OS Source.	151
Multithreaded Processing.	151
Rules and Guidelines for Multithreaded Processing.	151
Enabling Multithreaded Processing for CDC Sessions.	152

Appendix A: DTL_CAPXTIMESTAMP Time Stamps.....	153
Time Stamps That Are Reported in the DTL_CAPXTIMESTAMP Field by Data Source.	153
Index.....	155

Preface

Use the *Informatica® PowerExchange® CDC Guide for i5/OS* to learn about configuring and implementing PowerExchange change data capture (CDC) for DB2 for i sources on IBM i systems. Refer to this guide after you install the PowerExchange components required for CDC.

In this guide, the term *i5/OS* refers to the IBM i operating system and the term *DB2 for i5/OS* refers to the Db2 for i database.

Note: Certain settings of third-party systems may prevent Informatica PowerExchange connectors from (i) retrieving data from the source database and/or (ii) populating data in the target database.

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>.

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 <https://network.informatica.com> and select the Support option.

CHAPTER 1

Change Data Capture Introduction

This chapter includes the following topics:

- [PowerExchange CDC Overview, 10](#)
- [Planning Considerations, 11](#)
- [PowerExchange CDC Components, 11](#)
- [PowerExchange CDC Architecture, 12](#)
- [PowerExchange Condense Processing, 13](#)
- [Change Data Extraction, 14](#)
- [PowerExchange Integration with PowerCenter, 14](#)

PowerExchange CDC Overview

PowerExchange Change Data Capture (CDC) works in conjunction with PowerCenter® to capture changes to data in source tables and replicate those changes to target tables or files.

PowerExchange can capture change data from DB2 for i5/OS journal receivers in near real time. If you use the optional PowerExchange Condense feature, you can capture change data in continuous or batch mode to PowerExchange Condense condense files. PowerExchange then extracts change data from the condense files.

PowerExchange works with PowerCenter to extract the captured change data and apply it to one or more target tables or files. The targets can be on the same system as the source or on a different system.

Use PowerExchange CDC to keep targets synchronized with source tables. By replicating only the changes instead of bulk data, PowerExchange CDC helps make synchronization faster and less costly. This benefit is important to businesses that need to keep data accurate, complete, and up to date for decision-making and critical operations. When multiple changes are made to same field, PowerExchange CDC applies only the latest change.

You can use CDC to:

- Maintain a subset of the source data on a target, for example, to create a data mart for decision-making.
- Maintain an exact duplicate of the source.
- Update a data warehouse.
- Quickly apply changes to a target after source data is updated to meet specific business or application requirements.
- Apply changes to a target at specific intervals to meet a business requirement such as data cleansing or translation.

PowerExchange CDC provides a variety of options and parameters to help you create an optimal change data capture implementation for your business requirements and environment.

Planning Considerations

Plan your DB2 for i5/OS CDC environment before starting change capture.

When determining how to implement a change data capture environment, consider the following questions:

- Do you need to make changes available at specific intervals in batch mode or on a continual basis?
- What are the backup and recovery requirements?
- From which journal receivers do changes need to be captured? Where do these journal receivers reside?
- If you plan to use CDC to maintain a target data warehouse or data mart, what are the business requirements for delivering change data to the target?
- How will you materialize the target tables or files before starting CDC?

You can use PowerExchange bulk data movement for this purpose. For more information, see the *PowerExchange Bulk Data Movement Guide*.

PowerExchange CDC Components

Multiple PowerExchange components are involved in change data capture processing.

These components are:

PowerExchange Listener

The PowerExchange Listener coordinates activities across systems, initiates requests, or processes requests from third-party applications. You must run a PowerExchange Listener to move data across systems.

The PowerExchange Listener performs the following functions:

- Handles all tasks that use capture registrations and extraction maps.
- Communicates with the PowerExchange Navigator interface.
- Communicates with the PowerExchange Client for PowerCenter (PWXPC) to integrate with PowerCenter.
- Run processes to extract data for client requests.

PowerExchange Condense

PowerExchange Condense captures change data and stores the data in condense files. The condense files store only the successfully completed UOWs, in chronological order by end time. When an extraction session runs, the data is extracted from the condense files rather than from the change stream.

If you add or edit capture registrations, you must restart the PowerExchange Condense job to activate the additional or changed registrations.

PowerExchange Navigator

The PowerExchange Navigator is a Windows graphical user interface from which you create and maintain capture registrations and extraction maps. You can also define data maps if you need to do column-level processing, such as building expressions to populate user-defined columns.

Create a capture registration for each source table of interest for CDC. The PowerExchange Navigator automatically generates a corresponding extraction map.

A capture registration identifies:

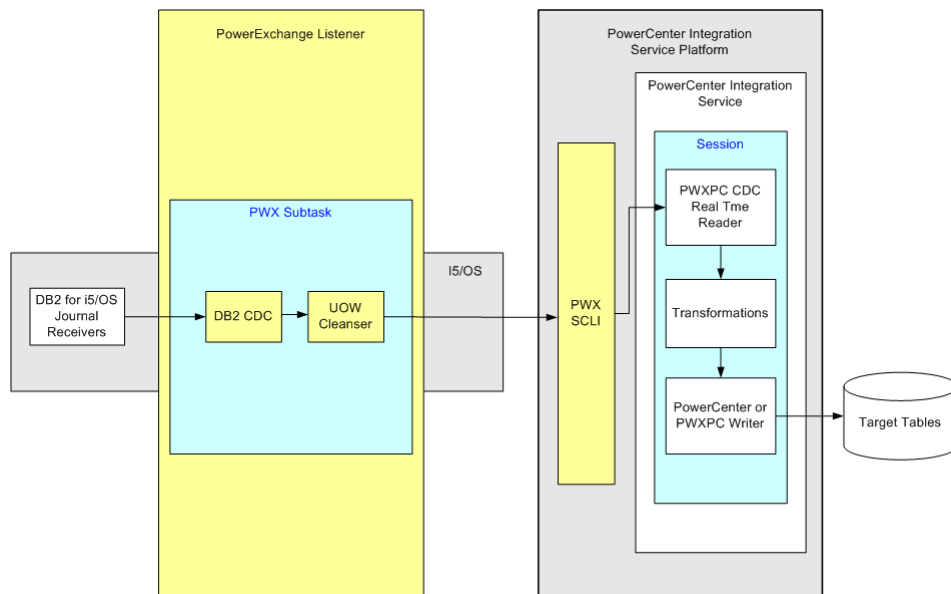
- The data source type, which is AS4 for DB2 for i5/OS
- The columns in a source table for which to capture changes
- Whether to use PowerExchange Condense full or partial processing of change data
- Whether the registration is active for use in CDC

PowerExchange CDC Architecture

The PowerExchange CDC architecture is sufficiently flexible to handle many change data replication scenarios.

You can use PowerExchange in conjunction with PowerCenter to replicate change data from multiple DB2 for i5/OS sources to multiple targets of different types in a single session.

The following figure shows a simple CDC configuration that uses real-time extraction mode to access change data directly from local DB2 journal receivers without PowerExchange Condense:



In this real-time configuration, PowerExchange CDC uses the CAPXRT access method to capture change data from DB2 for i5/OS journal receivers. The UOW Cleanser reconstructs intermingled changes read from the change stream into complete UOWs in chronological order based on end time.

When an extraction request runs, PowerCenter connects to the PowerExchange Call Level Interface (SCLI) to contact the PowerExchange Listener. The change data is passed to the SCLI and then to the PWXPC CDC Real Time reader. In this manner, the PowerCenter extraction session pulls the change data that PowerExchange captured. After the PWXPC reader reads the change data, PowerCenter uses the mapping

and workflow that you create to transform the data and load it to the target. With this configuration, you can replicate change data from multiple sources in the same database to multiple target tables in a single extraction process.

PowerExchange Condense Processing

PowerExchange Condense captures change data to condense files.

The condense files store only the successfully completed UOWs, in chronological order by end time. After PowerExchange Condense completes writing changes to a condense file and closes the file, extraction of change data can begin. The CDC session extracts data from the condense files rather than directly from the journal receivers.

You can run PowerExchange Condense in one of the following operational modes:

- Batch mode. A single Condense cycle runs and then shuts down.
- Continuous mode. A Condense job runs for a long period, perhaps on a 24-hour basis. In this mode, the condense subtask “sleeps” after each condense operation.

If you run PowerExchange Condense in continuous mode, a condense process is triggered when any of the following events occur:

- Initialization completes.
- A CONDENSE command or pwxcmd condense command is issued.
- A FILESWITCH command or a pwxcmd fileswitch command is issued.
- The wait period that is specified in the NO_WAIT_DATA parameter of the CAPTPARM member elapses.

Also, you can specify the type of condense processing that PowerExchange Condense uses by setting the **Condense** option in the capture registrations to one of the following values:

- **Full**. PowerExchange Condense creates condense files that contain only the latest changes.
- **Part**. PowerExchange Condense creates condense files that contain all successful changes for a source.

For extraction processing, you can use either batch extraction mode or continuous extraction mode.

PowerExchange Condense provides the following benefits:

- Does not record changes that were caused by backing out or rolling back transactions.
- If you select **Full** for the **Condense** option, records only the latest change for each record when multiple changes are made.

Rollbacks or Backouts

During normal application processing, the source data is updated with changes. These changes are captured as part of the standard data capture process. However, if at a later time, the transaction processing fails and the changes are backed out, a second set of changes are captured.

PowerExchange Condense can eliminate both the update and backout of changes from the data passed to the target. This feature ensures that only successful changes are available to update the target and reduce the time required to update the target and hence network traffic.

Multiple Changes to the Same Record

With change data being collected at regular intervals, it is possible that several changes are captured for the same data row or record. However, for the purpose of updating the target, only the latest update is required. PowerExchange uses only the latest update when you specify Full for the Condense parameter in a capture registration.

Change Data Extraction

The PowerExchange Listener handles requests to extract change data in real time from journal receivers or in batch extraction mode from condense files.

Extraction Maps

An extraction map is a definition of a change data extraction for a source table.

PowerExchange generates an extraction map when you create a capture registration. You can edit the generated extraction maps to add or remove columns. You can also create additional extraction maps for a capture registration, each for a subset of the columns that are registered for capture. In the PowerExchange Navigator, extraction maps are listed under the extraction group for the database instance and node location from which the data is extracted.

To define a data source in PowerCenter, you can import the extraction map. When you run a CDC session, PowerExchange uses the source definition for extracting change data.

Extraction Process

You can use PowerCenter to extract change data in real time or from condense files. The extraction process logs information at various points regarding the process status and the contents of various control tables. This enables the overall system to keep track of what has happened and what is required to be extracted at some later request. This is also used to form part of the audit trail.

The extraction process is based on an application name. An application name is created in either of the following situations:

- On first use of an extraction process in a PowerCenter task
- By using the DTLUAPPL utility

After an application name is used in an extraction process, it is preserved as part of the audit trail and additional extraction processes are sensitive to it.

PowerExchange Integration with PowerCenter

PowerCenter works in conjunction with the PowerExchange Client for PowerCenter (PWXPC) to extract the change data that PowerExchange captured and apply it to one or more targets.

The primary function of PWXPC is to integrate PowerExchange with PowerCenter so that PowerCenter can access PowerExchange-controlled data and write it to various targets. With PWXPC, CDC sessions can extract change data from both DB2 journals and PowerExchange Condense condense files.

PowerCenter provides transformation and data cleansing capabilities, which you can use in your CDC sessions.

For more information about PWXPC, see *PowerExchange Interfaces for PowerCenter*.

CHAPTER 2

PowerExchange Listener

This chapter includes the following topics:

- [PowerExchange Listener Overview, 16](#)
- [PowerExchange Listener Message Log Files, 16](#)
- [Threadsafe Processing Requirement, 17](#)
- [Configuring the PowerExchange Listener, 17](#)
- [Starting PowerExchange Listener, 32](#)
- [Stopping the PowerExchange Listener, 33](#)
- [Stopping a PowerExchange Listener Task for an Extraction, 33](#)
- [Displaying Active PowerExchange Listener Tasks, 33](#)

PowerExchange Listener Overview

In a change data capture (CDC) environment, the PowerExchange Listener provides the following services:

- Stores and manages capture registrations, extraction maps, and data maps for CDC data sources.
- Provides captured change data to PowerCenter when you run an extraction and load session.
- Provides captured change data or source data to the PowerExchange Navigator when you perform a database row test of an extraction map or a data map.

The PowerExchange Listener interacts with the following PowerExchange CDC components:

- PowerExchange Navigator
- Other PowerExchange Listeners

PowerExchange Listener Message Log Files

The files to which the PowerExchange Listener writes messages depends on whether you enable alternative logging.

If you do *not* enable alternative logging, the PowerExchange Listener writes messages to the DTLLOG member of the LOG file in the PowerExchange *datalib* library, by default. PowerExchange allocates this file if it does not exist. Because PowerExchange might write a lot of information to this file, Informatica

recommends that you routinely use the CLRPFM command to clear the file contents and prevent the file from becoming too large.

If you enable alternative logging by specifying the TRACING statement in the DBMOVER configuration member, the PowerExchange Listener writes messages to alternative message members in the *Plistener_port_number* file in the *datalib* library. The member names have the prefix that is specified in the PFX parameter of the TRACING statement. If you specify the APPEND parameter in the TRACING statement, whenever PowerExchange restarts, new messages are appended to the most recent message log member. You can control the size of an alternative log member by using the SIZE parameter in the TRACING statement. When an alternative log member becomes full, PowerExchange rotates to another alternative log member, which prevents out-of-space conditions. Also, PowerExchange buffers messages and writes them to the alternative log members on disk based on a specific flush interval. This type of buffered logging can reduce I/O activity on the log files. For more information about alternative logging, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“PowerExchange Condense Message Log Files” on page 40](#)
- [“PowerExchange Event Message Queue” on page 61](#)

Threadsafe Processing Requirement

PowerExchange uses asynchronous network communication for all send and receive data operations between a PowerExchange client and a PowerExchange Listener. With asynchronous communication, PowerExchange uses separate threads for network processing and data processing, so that network processing overlaps with data processing.

Because the PowerExchange Listener and any jobs that it spawns are multithreaded, the PowerExchange Listener must not be used to invoke IBM i operating system commands or APIs that are not threadsafe, such as RCLRSC.

Configuring the PowerExchange Listener

Before starting CDC on i5/OS, configure the PowerExchange Listener parameters in the DBMOVER member of the CFG file in the PowerExchange *datalib* library. This section describes the key parameters and statements that pertain to CDC processing.

The PowerExchange Listener uses these parameters and statements to perform the following functions during CDC:

- Connect to journals for source RDBMS databases and objects.
- Determine the library to use for extraction maps.

Summary of DBMOVER Statements for CDC

You must define certain statements in the DBMOVER configuration member for DB2 for i5/OS CDC.

The following table describes each of these key statements:

Statement	Description
AS400EVENTMSGQ	Specifies the message queue on the i5/OS system to which PowerExchange writes messages that do <i>not</i> require a user reply. These messages include those that are generated when you specify ALWCLRPFM=Y, ALWLIBRFSH=Y, or AS4JRNEXT=Y in the AS4J CAPI_CONNECTION statement.
AS400USRJRNCODE	Specifies unique, two-character journal entry type codes that identify add library and delete library operations in a DB2 for i5/OS journal. Use this statement to override the default codes for refreshing the CDC library interest list after adding or deleting libraries that are journaled to the same journal and that contain instances of the same source table.
AS400USRMSGQ	Specifies the message queue on the i5/OS system to which PowerExchange writes messages that require a user reply.
AS4J CAPI_CONNECTION, UOWC CAPI_CONNECTION	Defines a named set of parameters that the PowerExchange Consumer API (CAPI) uses to connect to the change stream and control extraction processing. A CAPI connection is specific to a data source type. For DB2 for i5/OS CDC, you must define AS4J and UOWC CAPI_CONNECTION statements. You can define up to eight CAPI_CONNECTION statements in a DBMOVER configuration member.
CAPI_CONN_NAME	Specifies the name of the CAPI_CONNECTION statement that PowerExchange uses by default for all data sources when you do not supply a CAPI connection override.
CPX_DIR	Specifies the name of the i5/OS library in which PowerExchange stores extraction maps for the DB2 for i5/OS CDC sources. When you run the CRTPWXENV command during installation, PowerExchange creates this library and includes its name in the DBMOVER configuration file. Default is CPXLIB.

AS400EVENTMSGQ Statement

The AS400EVENTMSGQ statement specifies the message queue on an i5/OS system to which PowerExchange writes messages that do *not* require a user reply.

Operating Systems: i5/OS

Data Sources: DB2 for i5/OS

Related Statements: CAPI_CONNECTION - AS4J, AS400USRMSGQ

Required: No

Syntax:

```
AS400EVENTMSGQ=library/queue_name
```

Parameters: Separate the following parameters by a slash (/):

library

The name of the library that contains the message queue.

queue_name

The message queue name.

Usage Notes: PowerExchange writes messages to the specified message queue when ALWCLRPFM=Y, ALWLIBRFSH=Y, or AS4JRNEXIT=Y is specified on the AS4J CAPI_CONNECTION statement, as follows:

- If ALWCLRPFM=Y, PowerExchange writes the DTL3002 message to the message queue if PowerExchange encounters a journal entry for a CLRPFM command. PowerExchange cannot replicate changes that a CLRPFM command makes.
- If ALWLIBRFSH=Y, PowerExchange writes message DTL3003I for an *ADDLIB (add library) event and message DTL3004I for a *DLTLIB (delete library) event to the message queue when you issue the SNDPWXJRNE command with an **Event Identifier** value. This command triggers refresh processing of the CDC library interest list for the added or deleted libraries.
- If AS4JRNEXIT=Y, PowerExchange writes the DTL3001 message for each journal receiver that a change data extraction processes. PowerExchange writes the message when it reads the first journal entry for the next journal receiver on the chain. If multiple change data extractions successfully process the same journal receiver, the message queue contains multiple DTL3001 messages for that journal receiver.

For more information about this message queue, see the *PowerExchange CDC Guide for i5/OS*.

AS400USRJRNCODE Statement

The AS400USRJRNCODE defines two-character journal entry type codes that identify SNDPWXJRNE add library and delete library entries in a DB2 for i5/OS user journal. Use this statement to override the default journal entry type codes that the SNDPWXJRNE command uses.

The SNDPWXJRNE command issues requests for refreshing the CDC library interest list for added or deleted libraries. The libraries must contain instances of the same source table, and the table instances must be journaled to the same user journal.

This statement applies to all of the DB2 for i5/OS instances that are defined in AS4J CAPI_CONNECTION statements with the ALWLIBRFSH=Y setting in the same DBMOVER member.

Operating Systems: i5/OS

Data Sources: DB2 for i5/OS

Related Statements: CAPI_CONNECTION - AS4J

Required: No

Syntax:

```
AS400USRJRNCODE=(ADDLIB=jrn_entry_type_code,DLTLIB=jrn_entry_type_code)
```

Use a comma (,) to separate the ADDLIB and DLTLIB parameters.

Parameters:

ADDLIB=journal_entry_type_code

Optional. Enter a unique two-character journal entry type code, such as U1, that the SNDPWXJRNE command writes to the journal for an SNDPWXJRNE add library (*ADDLIB) entry. Overrides the default code of I1.

DLTLIB=journal_entry_type_code

Optional. Enter a unique two-character journal entry type code, such as U2, that the SNDPWXJRNE command writes to the journal for a SNDPWXJRNE delete library (*DLTLIB) entry. Overrides the default code of I2.

Usage Notes:

- For PowerExchange to use this statement, you must set the ALWLIBRSH parameter in the AS4J CAPI_CONNECTION statement to Y. This setting enables you to use the SNDPWXJRNE command to refresh the CDC library interest list after you add or delete i5/OS libraries that contain instances of the same DB2 source table and that are journaled to the same user journal, without restarting real-time PowerCenter PWXPC CDC sessions.
- By default, the SNDPWXJRNE command uses I1 as the entry type code for an add library request and uses I2 for a delete library request. If these journal entry type codes are used by another application, you must specify unique entry type codes with this AS400USRJRNCODE statement.
- The journal entry codes that you specify in this statement, or the default journal entry codes, are not specific to the journals that are identified in the AS4J CAPI_CONNECTION statements in the same DBMOVER member. PowerCenter CDC sessions can override these journals at runtime.

For more information about event message queues, see the *PowerExchange CDC Guide for i5/OS*.

AS400USRMSGQ Statement

The AS400USRMSGQ statement specifies the message queue on an i5/OS system to which PowerExchange writes messages that require a user reply. These messages typically require a user reply of Y or N, or Yes or No.

Operating Systems: i5/OS

Data Sources: DB2 for i5/OS

Related Statements: CAPI_CONNECTION - AS4J, AS400EVENTMSGQ

Required: No

Syntax:

```
AS400USRMSGQ=library/queue_name
```

Parameters: Separate the following parameters by a forward slash (/):

library

The name of the library that contains the message queue for user reply messages.

queue_name

The message queue name.

If you do not specify this statement, PowerExchange writes messages that require a user reply to the system operator message queue, QSYSOPR, by default.

CAPI_CONNECTION Statements

PowerExchange requires that you define CAPI_CONNECTION statements in the DBMOVER configuration file on any i5/OS system where PowerExchange captures or extracts change data. PowerExchange uses the parameters that you specify in the CAPI_CONNECTION statements to connect to the change stream and to customize capture and extraction processing.

For each data source type, you must define a specific type of CAPI_CONNECTION statement. For DB2 on i5/OS, define a AS4J CAPI_CONNECTION statement. You must also specify a UOWC CAPI_CONNECTION statement for the UOW Cleanser.

You can specify up to eight CAPI_CONNECTION statements in a dbmover.cfg file. You can identify one of the CAPI_CONNECTION statements for a specific data source type as the default. You can also specify overrides in various ways.

Note: PowerExchange does not require CAPI_CONNECTION statements in the dbmover.cfg on the PowerExchange Navigator machine. You can register tables for CDC and perform database row tests on that machine because the PowerExchange Navigator can communicate with the PowerExchange Listener on the source machine where CDC occurs.

CAPI_CONNECTION - AS4J Statement

The AS4J CAPI_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and control CDC processing for Db2 for i (i5/OS) CDC sources.

Operating Systems: IBM i (formerly i5/OS)

Data Sources: Db2 for i

Related Statements: CAPI_CONNECTION - UOWC, AS400EVENTMSGQ, AS400USRJRNCODE

Required: Yes for Db2 for i CDC

Syntax:

```
CAPI_CONNECTION=( [DLLTRACE=trace_id]
                  ,NAME=capi_connection_name
                  [,TRACE=trace_name]
                  ,TYPE=(AS4J
                        ,INST=instance_name
                        ,JOURNAL=(JRN=library/journal_name
                                  [,FROMRCVR=library/from_receiver]
                                  [,TORCVR=library/to_receiver])
                        [,ALWCLRPFM={N|Y}]
                        [,ALWLIBRFSH={N|Y}]
                        [,ALWNONRENT={N|Y}]
                        [,ALWPARTIAL={N|Y}]
                        [,AS4JRNEXT={N|Y}]
                        [,EOF={N|Y}]
                        [,LIBASUSER={N|Y|M|P}]
                        [,NOCCUOWSZ=uow_size]
                        [,POLWAIT={seconds|10}]
                        [,RTNBUFFSIZE={kilobytes|960}]
                        [,STOPIT=({CONT|TERM},{number_exceptions|5}]
                        [,UOWRSTANY={Y|N}]
                  )
                )
```

Parameters:

DLLTRACE=trace_id

Optional. User-defined name of the TRACE statement that activates internal DLL tracing for this CAPI. Specify this parameter only at the direction of Informatica Global Customer Support.

NAME=capi_connection_name

Required. Unique user-defined name for this CAPI_CONNECTION statement.

Maximum length is eight alphanumeric characters.

TRACE=trace_name

Optional. User-defined name of the TRACE statement that activates the common CAPI tracing. Specify this parameter only at the direction of Informatica Global Customer Support.

TYPE=(AS4J, ...)

Required. Type of CAPI_CONNECTION statement. For Db2 for i5/OS sources, this value must be AS4J.

INST=instance_name

Required. User-defined name for the source instance. This name must match the name that you specify in the **Collection Identifier** property of the registration group.

If you are running PowerExchange Condense, this name must also match the name specified in DBID parameter in the CAPTPARM member.

JOURNAL=(JRN=library/journal_name, [FROMRCVR=library/from_receiver], [TORCVR=library/to_receiver])

Required. Defines the name of the Db2 for i journal associated with the CAPI_CONNECTION instance for registrations. The journal contains data from the registered source tables that is available for extraction processing. JRN specifies the library name, which contains the journal that resides on the IBM i server. PowerExchange reads data by means of the journal, which has associated journal receivers.

You can optionally specify a range of journal receivers using the FROMRCVR and TORCVR parameters to override the journal receivers that are processed during extraction. You must specify both the FROMRCVR and TORCVR values.

Optionally, you can also override the journal specified in the JRN parameter by using one of the following methods:

- For CDC sessions that use real-time extraction and PWXPC connections, use the **Journal Name** on the PWX CDC application connection.

For real-time extraction CDC sessions that use ODBC connections, use the DTLJRNL parameter for the ODBC data source.

- For PowerExchange Condense, use the JRN statement in the CAPTPARM configuration member.
- For CAPXRT database row tests, use the **AS400 Journal/Library** property in the **CAPXRT Advanced Parameters** dialog box in the PowerExchange Navigator.

FROMRCVR=library/from_receiver

Optional. Specifies the beginning of a valid sequence of journal receivers associated with the journal specified in the JOURNAL JRN parameter. If you specify FROMRCVR, you must specify TORCVR.

TORCVR=library/to_receiver

Optional. Specifies the end of a valid sequence of journal receivers associated with the journal specified in the JOURNAL JRN parameter. If you specify TORCVR, you must specify FROMRCVR.

The following example specifies a range of journal receivers:

```
CAPI_CONNECTION=(NAME=DTLJPAS4,
TYPE=(AS4J, JOURNAL=(JRN=MYSHEMA/QSQJRN,
FROMRCVR=MYSHEMA/QSQJRN0010, TORCVR=MYSHEMA/QSQJRN0012), INST=PWXTEST1, EOF=N,
STOPIT=(CONT=5), LIBASUSER=N, AS4JRNEXT=Y, POLWAIT=10))
```

ALWCLRPFM= {N|Y}

Optional. Controls whether Db2 for i CDC processing stops or continues when PowerExchange encounters changes that result from an i5/OS Clear Physical File Member (CLRPFM) command or SQL TRUNCATE statement that is issued against a Db2 table registered for change data capture. PowerExchange cannot capture the deletions that result from a CLRPFM command or TRUNCATE statement.

Note: IBM i version 7.2 introduced support for SQL TRUNCATE operations on i5/OS tables. If you have an earlier IBM i version, this parameter pertains to CLRPFM commands only.

Options are:

- **N.** PowerExchange CDC processing stops when PowerExchange detects changes that are caused by a CLRPFM command or TRUNCATE statement.
- **Y.** PowerExchange ignores the CLRPFM command or TRUNCATE statement and continues CDC processing. The data integrity of the CDC target might be damaged. If you also specify the AS400EVENTMSGQ statement in the DBMOVER configuration file, PowerExchange issues the DTL3002 message to the specified message queue when PowerExchange encounters a journal entry for a CLRPFM command or TRUNCATE statement.

Attention: If you set this parameter to Y, the data integrity of the CDC targets might be damaged. Specify this parameter only at the direction of Informatica Global Customer Support. If you also specify a message queue in the AS400EVENTMSGQ parameter of the DBMOVER file, PowerExchange writes message DTL3002 to the queue if PowerExchange encounters a journal entry for a CLRPFM command or TRUNCATE statement.

Default is N.

ALWLIBRFSH={N|Y}

Optional. Controls whether you can refresh the CDC interest list after you add or delete libraries with instances of the same Db2 source table, without stopping and restarting real-time CDC sessions. This parameter is useful in the following situation: 1) you have instances of the same Db2 source table (or file) in multiple libraries, 2) the table instances are journaled to the same journal, and 3) you routinely add or delete libraries that contain an instance of the table.

Options are:

- **N.** Disables refresh processing of the CDC interest list for library additions or deletions. If you add libraries, PowerExchange does not start change data capture for those libraries until you stop and restart the CDC session. In this case, some changes might be lost. If you delete libraries, the CDC session ends abnormally.
- **Y.** Enables you to use the SNDPWXJRNE command to refresh the CDC interest list for library additions and deletions. The command sends journal entries for add library and delete library requests to the journal, using the PowerExchange default journal entry type codes or the unique codes that you define in the AS400USRJRNCODE statement. In PowerCenter, you must specify the asterisk (*) wildcard for the library name in the **Library/File Override** session property for the real-time session.

If you also specify a message queue in the AS400EVENTMSGQ parameter of the DBMOVER file, PowerExchange writes message DTL3003I for an *ADDLIB (add library) event and message DTL3004I for a *DLTLIB (delete library) event to the queue.

Default is N.

ALWNONRENT={N|Y}

Optional. Controls whether PowerExchange generates a next receiver (NR) journal entry when i5/OS does not provide one in the journal receiver. PowerExchange requires an NR entry to switch to the next journal receiver in the chain when the current one becomes full. However, i5/OS might not provide an NR entry under certain conditions, such as when a high level of transaction activity is occurring at the time a journal receiver switch is required. If i5/OS does not provide an NR entry, PowerExchange CDC processing ends unless you enable PowerExchange to generate an NR entry.

Options are:

- **N.** Do not generate an NR entry. If the i system does not provide an NR entry, CDC processing ends abnormally.

- **Y.** Generate an NR entry so that CDC processing can continue when the i system does not provide an NR entry. The generated NR entry is for PowerExchange internal use only and is not written to the actual journal receiver.

Default is N.

ALWPARTIAL={N|Y}

Optional. Controls whether PowerExchange processes journal receivers in partial status.

Options are:

- **N.** PowerExchange fails processing if a journal receiver is in partial status.
- **Y.** PowerExchange processes journal receivers in partial status.

Default is N.

Attention: If you specify Y for this parameter, you might compromise the data integrity of the change data being extracted because required changes might be unavailable. Specify this parameter only at the direction of Informatica Global Customer Support.

AS4JRNEXIT={N|Y}

Optional. Controls whether PowerExchange uses an exit program installed at the Delete Journal Receiver exit point, QIBM_QJO_DLT_JRNRCV, to prevent the deletion of journal receivers being processed for CDC.

Options are:

- **N.** PowerExchange does not lock journal receivers that it is processing.
- **Y.** PowerExchange locks any journal receiver that it is processing so that the journal receiver cannot be deleted from the i system. PowerExchange records the journal receivers that it is processing in a lock file called PWXJRNLCCKP in the CONDLIB library. When PowerExchange switches to the next journal receiver on the chain, PowerExchange removes the record for the previous journal receiver from the lock file.

If you also specify a message queue in the AS400EVENTMSGQ parameter of the DBMOVER file, PowerExchange writes the DTL3001 message to the specified queue for each journal receiver that a change data extraction successfully processes.

Default is N.

EOF={N|Y}

Optional. Controls whether PowerExchange stops change data extractions when the end-of-log (EOL) is reached.

Options are:

- **N.** Do not stop change data extractions when the EOL is reached.
- **Y.** Stop change data extractions when the EOL is reached.

Default is N.

Important: Because this parameter affects all users of the AS4J CAPL_CONNECTION statement, Informatica recommends that you use one of the following alternative methods to stop change data extractions at EOL:

- For CDC sessions that use real-time extraction mode, specify 0 for the **Idle Time** attribute of the PWX DB2i50S CDC Real Time application connection.

- For PowerExchange Condense, specify 1 for the COLL_END_LOG statement in the CAPTPARM configuration member.
- For CDC sessions that use ODBC connections, specify 0 for the WAITTIME parameter in the ODBC data source.

LIBASUSER={N|Y|M|P}

Optional. Controls the type of value that PowerExchange writes to the DTL__CAPXUSER field of each change record.

Options are:

- **N.** Writes the user ID of the user who made the change.
- **Y.** Writes the library name and file name of the file where the change was made.

If you set the ALWLIBRFSH parameter to Y, Informatica recommends that you specify LIBASUSER=Y. This setting enables you to identify the libraries that contain the table instances for which changes are extracted.

- **M.** Writes the library name, file name, and data member name of the file where the change was made.
- **P.** Writes the name of the program that made the change.

Default is N.

NOCCUOWSZ=*uow_size*

Optional. Size, in number of records, of the UOW that PowerExchange creates when reading change records from journal receivers that were created without commitment control.

Enter a number from 1 through 50000.

If you do not specify commitment control when making changes to Db2 for i tables, PowerExchange creates a UOW for each change record. Use this parameter to increase the size of the UOWs that PowerExchange creates. PowerExchange and PowerCenter process larger UOWs more efficiently. Larger UOWs also reduce commit activity on the targets of the change data.

Because this parameter affects all users of this CAPI_CONNECTION statement, Informatica recommends that you use the **Minimum Rows Per commit** attribute on the PWX DB2i5OS CDC Real Time application connection to specify UOW size values for CDC sessions.

PowerExchange ignores this parameter for change records created with commitment control.

POLWAIT={seconds|10}

Optional. The number of seconds that PowerExchange waits after reaching the end of a journal receiver before checking for change data in that journal receiver.

Enter a number from 0 through 10. Default is 10.

RTNBUFFSIZE={kilobytes|960}

Adjust the buffer size, in kilobytes, that is used to collect journal entries for CDC as a result of PowerExchange calls to the IBM QjoRetrieveJournalEntries API. You can use this parameter to tune performance based on your environment.

Valid values are 128 through 12288. Default is 960 KB, which is the IBM default size.

Note: If you also specify the RTNBUFFSIZE in the **PWX Overrides** field of the PowerCenter PWX DB2i5OS CDC application connection, the override takes precedence.

STOPIT=({CONT|TERM},number_exceptions|5)

Optional. Controls whether an extraction session stops after it encounters exceptions when processing change data from the journal. An exception can be one of the following:

- An after image change without a before image change
- An attempt to delete a record that has no data

For the first positional parameter, enter one of the following options:

- **CONT.** The extraction session continues to run after the specified number of exceptions.
- **TERM.** The extraction session ends after the specified number of exceptions.

Default is CONT.

For the second positional parameter, *number_exceptions*, enter the number of exceptions after which PowerExchange takes the action defined in the first positional parameter of the statement. The value 0 indicates that the extraction continues to run but does not report any exceptions. Valid values are 0 through 255. Default is 5.

UOWRSTANY={Y|N}

This parameter is deprecated. Y is the default value. When PowerExchange extraction sessions cold start, they skip any partial UOW for which the start-UOW point is on a previous journal receiver. The sessions begin reading change data with the next UOW in the current journal receiver.

If you omit this parameter, the default behavior is the same as if Y were specified. If you enter N, PowerExchange ignores the N value and uses Y.

CAPI_CONNECTION - UOWC Statement

The UOWC CAPI_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses for the UOW Cleanser.

In the change stream for some data sources, changes from multiple UOWs are intermingled. The UOW Cleanser reconstructs the intermingled changes read from the change stream into complete UOWs in chronological order based on end time.

Operating Systems: i5/OS, Linux, UNIX, Windows, and z/OS

Data Sources: Db2 for i5/OS and z/OS CDC sources

Related Statements: CAPI_CONNECTION - AS4J and CAPI_CONNECTION - LRAP

Required: Yes, for CDC for the specified sources

Syntax:

```
CAPI_CONNECTION=( [DLLTRACE=trace_id]
                  ,NAME=capi_connection_name
                  [,TRACE=trace_name]
                  ,TYPE=(UOWC
                        ,CAPINAME=source_capi_name
                        [,BLKSIZE=block_size]
                        [,CUOWS={number_of_concurrent_UOWs|34}]
                        [,DATACLASS=data_class]
                        [,LARGEOPS=number_of_operations]
                        [,MEMCACHE={cache_size|1024}]
                        [,MONITORINT={minutes|5}]
                        [,RSTRADV=seconds]
                        [,SPACEPRI={primary_space|50}]
                        [,SPACETYP={BLK|TRK|CYL}]
                        [,SPILLKEEP=number_of_spill_files]
                        [,STORCLASS=storage_class]
```

```

        [,TIMESTAMP={LOG|COMMIT}]
        [,UNIT=unit]
    )
)

```

Parameters:

DLLTRACE=trace_id

Optional. A user-defined name for the TRACE statement that activates internal DLL tracing for this CAPI. Specify this parameter only at the direction of Informatica Global Customer Support.

NAME=capi_connection_name

Required. A unique user-defined name for this CAPI_CONNECTION statement. Maximum length is eight alphanumeric characters.

TRACE=trace_name

Optional. A user-defined name for the TRACE statement that activates the common CAPI tracing. Specify this parameter only at the direction of Informatica Global Customer Support.

TYPE=(UOWC, ...)

Required. The type of CAPI_CONNECTION statement. For the UOW Cleanser, this value must be UOWC.

CAPINAME=capi_name

Required. The value of the NAME parameter in the related source-specific CAPI_CONNECTION statement, which can be one of the following statement types:

- AS4J CAPI_CONNECTION statement for Db2 for i (i5/OS) sources
- LRAP CAPI_CONNECTION statement for data sources on z/OS

BLKSIZE=block_size

Optional. The block size, in bytes, for the sequential UOW spill files that the UOW Cleanser creates when the memory cache cannot hold all changes for a UOW.

The following table shows valid values by CDC source type:

Data Source Type	Valid Values	Default Value
Db2 for i (i5/OS)	A number from 8 through 32760	32760
z/OS data sources	A number from 8 through 32760	18452

CUOWS=number_of_concurrent_UOWs

The expected number of concurrent UOWs that contain data of CDC interest. If you specified a large MEMCACHE value, have excessive spill file allocations, and notice a large number of concurrent UOWs reported in message PWX-10742 or PWX-10782, increase this parameter value to cause less memory cache to be used per UOW. By reducing memory cache use per UOW, more concurrent UOWs can be accommodated without allocating spill files. This parameter in no way restricts the maximum number of concurrent UOWs that the UOW Cleanser can process.

Valid values are 1 through 65535. Default is 34.

DATACLASS=*data_class*

Optional. On z/OS, the SMS data class that the UOW Cleanser uses when allocating the sequential UOW spill files. If you do not specify this parameter, the SMS ACS routines can assign the data class.

LARGEOPS=*number of operations*

Optional. Overrides the default value that PowerExchange uses to identify transactions as large transactions for reporting purposes. Enter the number of DML operations (inserts, updates, and deletes), in thousands, that a transaction must contain to be considered a large transaction.

PowerExchange issues status messages for large transactions that meet this criteria. If PowerExchange issues too many messages, you can increase this value to limit the number of messages.

Valid values are 1 through 2147483 (1000 through 2,147,483,000 operations). The default value is one half of the MEMCACHE parameter value rounded up to the nearest thousand. Based on the default MEMCACHE value of 1024 KB, the default LARGEOPS value is 1000 (1,000,000 operations).

MEMCACHE={*cache_size*|1024}

Optional. The maximum memory cache size, in kilobytes, that PowerExchange allocates to reconstruct complete UOWs.

Enter a number from 0 through 2147483647. Default is 1024 KB. If you enter 0, the memory cache size is limited only by the available memory on the system.

For each extraction session, PowerExchange keeps all changes for each UOW in the memory cache until it processes the end-UOW record. PowerExchange incrementally allocates memory cache up to the limit that this parameter specifies. If the memory cache is too small to hold all of the changes in a UOW, PowerExchange spills the changes to a sequential files on disk, called UOW spill files.

Each UOW spill file contains one UOW. A UOW might require multiple UOW spill files to hold all of the changes for that UOW. If the change stream contains multiple large UOWs and the memory cache is insufficient, PowerExchange might create numerous UOW spill files.

PowerExchange processes the change stream more efficiently if it does not need to use UOW spill files. A large number of UOW spill files can degrade extraction performance and cause disk space shortages.

Important: If the change stream contains small UOWs, the default value might be sufficient. However, Informatica recommends that you specify a larger value because the default value is often too small.

The location in which PowerExchange allocates the UOW spill files varies by operating system, as follows:

- For i5/OS, PowerExchange uses CRTPF command to create a physical file for UOW spill files.

PowerExchange names the UOW spill files using the C/C++ `tmpnam()` function.

- For Linux and UNIX, PowerExchange uses the current directory by default for UOW spill files. To use a different directory, specify the `TMPDIR` environment variable.

PowerExchange names the UOW spill file names using the prefix "dtlq" and the operating system function `tempnam`.

Note: The UOW spill files are temporary files that are deleted when PowerExchange closes them. These files are not visible in the directory while they are open.

- For Windows, PowerExchange uses the current directory by default for UOW spill files. To use a different directory, specify the TMP environment variable.

PowerExchange names the UOW spill file using the prefix "dtlq" and the Windows _tempnam function.

- For z/OS, PowerExchange uses dynamic allocation to allocate temporary data sets for the UOW spill files. Generally, SMS controls the location of temporary data sets. If you do not use SMS to control temporary data sets, the UNIT parameter controls the location for the UOW spill files.

Because PowerExchange allocates temporary data sets for the UOW spill files, z/OS assigns these files system-generated data set names, which begin with *SYSyddd.Thhmmss.RA000.jobname*.

- For z/OS, MEMCACHE cannot exceed 25600 in a listener task, or 1024000 in a netport job.

Important: If many spill files are being created, Informatica recommends that you specify a larger value in the range of 1024 to 25600.

Warning: PowerExchange allocates the cache size for each extraction operation. If you use a large MEMCACHE value and run many concurrent extraction sessions, memory constraints can occur.

MONITORINT=minutes

Optional. The time interval, in minutes, at which PowerExchange checks transaction activity for long outstanding transactions and large transactions. A long outstanding transaction is one that remains active for two monitoring intervals, and a large transaction is one that meets the LARGEOPS criteria. When this interval elapses, PowerExchange issues messages that identify the large transactions and long outstanding transactions and report their processing activity. PowerExchange also issues messages that identify the current position in the change stream. Valid values are 0 through 720. A value of 0 disables monitoring. Default is 5.

RSTRADV=seconds

The time interval, in seconds, that PowerExchange waits before advancing restart and sequence tokens for a registered data source during periods when UOWs do not include any changes of interest for the data source. When the wait interval expires, PowerExchange returns the next committed "empty UOW," which includes only updated restart information.

Enter a number from 0 through 86400. No default is provided.

PowerExchange resets the wait interval to 0 when one of the following events occur:

- PowerExchange completes processing a UOW that includes changes of interest.
- PowerExchange returns an empty UOW because the wait interval expired without PowerExchange receiving any changes of interest.

For example, if you specify 5, PowerExchange waits five seconds after it completes processing the last UOW or after the previous wait interval expires. Then PowerExchange returns the next committed empty UOW that includes the updated restart information and resets the wait interval to 0.

If you do not specify RSTRADV, PowerExchange does not advance restart and sequence tokens for a registered source during periods when PowerExchange receives no changes of interest. When PowerExchange warm starts, it reads all changes, including those not of interest for CDC, from the restart point.

For Db2 for i (i5/OS) sources, Informatica recommends that you use this parameter if the change records that PowerExchange reads from i5/OS journal receivers are created under commitment control. If the change records are created without commitment control, do not specify this parameter.

Attention: A value of 0 can degrade performance. In addition to the UOWs that contain changes for registered sources of interest, PowerExchange returns an empty UOW for every UOW that does not contain changes for the registered sources of interest.

SPACEPRI={primary_space|50}

Optional. On z/OS, the amount of primary space that the UOW Cleanser uses for allocating UOW spill files. The SPACETYP parameter indicates the type of space units.

Enter a number from 1 through 16777215. Default is 50 blocks.

The UOW Cleanser does not use secondary space. Instead, when a spill file becomes full, the UOW Cleanser allocates another spill file of the same size.

SMS ACS routines can override the UOW spill file size.

Note: On i5/OS, the UOW Cleanser allocates UOW spill files as physical files with SIZE(*NOMAX), which means that the maximum spill file size is controlled by the system maximum file size. On Linux, UNIX, and Windows, PowerExchange allocates UOW spill files as temporary files that are 2 GB in size.

SPACETYP={BLK|TRK|CYL}

Optional. On z/OS, the type of units in which the primary space for UOW Cleanser allocation of UOW spill files is expressed.

Options are:

- **BLK.** Blocks.
- **CYL.** Cylinders.
- **TRK.** Tracks.

Default is BLK.

SPILLKEEP=number_of_spill_files

Optional. The number of spill files that the UOW Cleanser retains for re-assignment. The UOW Cleanser retains spill files instead of deallocating them so that the files are available to be reassigned to new transactions. This feature is intended to prevent excessive file deallocation and allocation activity.

Valid values are 0 through 999. On z/OS and i5/OS, the default is 3. On Linux, UNIX, and Windows, the default is 0.

STORCLASS=storage_class

Optional. On z/OS, the SMS storage class name that the UOW Cleanser uses to allocate UOW spill files.

TIMESTAMP={LOG|COMMIT}

Optional. The type of timestamp that PowerExchange records in the generated DTL__CAPXTIMESTAMP column of each change record for a transaction. Specify this parameter only if you want to display the commit timestamp instead of the timestamp from the source logs or data sets.

Options are:

- **LOG.** The timestamp of a change on the source database, as recorded by the DBMS in the source database logs or data sets near the time when the change is made. For more information, see [Appendix A, "DTL__CAPXTIMESTAMP Time Stamps" on page 153.](#)

- **COMMIT.** The timestamp of the transaction commit on the source database. Specify this option if you use the timestamp to calculate latency.

Default is LOG.

UNIT=unit

Optional. On z/OS, the generic or esoteric unit name that the UOW Cleanser uses to allocate UOW spill files.

Sample CAPI_CONNECTION Statements

The sample DBMOVER configuration member in the CFG file contains the following entries:

```
/* CAPI_CONN_NAME=DTECAPU
/* CAPI_CONNECTION=(NAME=DTECAPU,
/*   TYPE=(UOWC,CAPINAME=DTLJPAS4))
/* CAPI_CONNECTION=(NAME=DTLJPAS4,
/*   TYPE=(AS4J,JOURNAL=(JRN=MYSHEMA/QSQJRN,INST=FRED002,EOF=N,
/*     STOPIT=(CONT=5),LIBASUSER=N,AS4JRNEXT=N,POLWAIT=10))
CPX_DIR=cpxlib
```

CAPI_CONN_NAME Statement

The CAPI_CONN_NAME statement specifies the name of the CAPI_CONNECTION statement that PowerExchange uses by default for all data source types when you do not supply a CAPI connection override.

Operating Systems: All

Related Statements: CAPI_CONNECTION and CAPI_SRC_DFLT

Required: No

Syntax:

```
CAPI_CONN_NAME=capi_connection_name
```

Value: For the *capi_connection_name* variable, enter the name of the CAPI_CONNECTION statement that PowerExchange uses when you do not specify a CAPI_CONNECTION override value.

Usage Notes:

- Optionally, you can use the CAPI_SRC_DFLT statement to provide a source-specific default connection. A source-specific default is useful if the DBMOVER configuration file contains many CAPI_CONNECTION statements.
- Informatica recommends that you always specify an override CAPI_CONNECTION statement for a PowerExchange operation, instead of relying on default connection definitions. You can define the following overrides, based on the type of PowerExchange operation:
 - For CDC sessions, specify the **CAPI Connection Name Override** attribute on the PWX CDC application connection.
 - For PowerExchange Condense, specify the CONN_OVR statement in CAPTPARM configuration file.
 - For the PowerExchange Logger for Linux, UNIX, and Windows, specify the CONN_OVR statement in pwxcl.cfg configuration file.
 - For the generation of restart tokens with the DTLUAPPL utility, specify the CONN_OVR statement in the DTLUAPPL control statement.
 - For CAPXRT database row tests in the PowerExchange Navigator, specify the **CAPI Connection Name** value in the **Advance** dialog box.

- For the generation of restart tokens during a database row test in the PowerExchange Navigator, specify the CONNAME parameter.
- For PowerExchange ODBC connections, specify the DTLCONN_OVR parameter in the odbc.ini file or the DTLCONNOVR SQL escape sequence override.

CPX_DIR Statement

The CPX_DIR statement specifies the name of the library on i5/OS in which PowerExchange stores extraction maps for DB2 for i5/OS CDC sources.

Operating Systems: i5/OS

Data Sources: DB2 for i5/OS CDC

Required: No

Syntax:

```
CPX_DIR=library
```

Value: For the *library* variable, enter the name of the i5/OS library in which PowerExchange stores extraction maps for DB2 for i5/OS CDC sources. Default is CPXLIB.

Usage Notes: When you run the CRTPWXENV command during PowerExchange installation, PowerExchange creates this library and includes its name in the DBMOVER configuration member. The name is based on the CPXLIB parameter on the CRTPWXENV command.

Starting PowerExchange Listener

To start the PowerExchange Listener on i5/OS, use the SBMJOB command to invoke the DTLLST program.

Note: In PowerExchange 9.0 and later, you cannot run the DTLLST program in interactive mode.

Before starting the PowerExchange Listener, verify that the following prerequisites are met:

- The QMLTTHDACN system value is set to 1 or 2. For more information about the QMLTTHDACN system value, see the IBM i Information Center for i5/OS.
- The JOBDD description includes the ALWMLTTHD(*YES) parameter to allow multiple threads.
- The JOBDD description includes the SPLFACN(*DETACH) parameter to prevent the system job table from exceeding its maximum size.

To enter the SBMJOB command, use the following syntax:

```
SBMJOB CMD(CALL PGM(dtllib/DTLLST) PARM('node_name' '[CONFIG=library/  
file(myconfig_member)]' '[LICENSE=library/file(mylicense_key_member)]') JOB(job_name)  
JOBDD(datalib/DTLLIST) PRTDEV(*JOBDD) OUTQ(*JOBDD) CURLIB(*CRTDFT) INLLIBL(*JOBDD)  
SPLFACN(*JOBDD)
```

This syntax contains the following variables:

- *dtllib* is the name of the PowerExchange software library that was entered at installation.
- *node_name* is the PowerExchange Listener node name that was specified in the LISTENER statement of the *datalib*/CFG(DBMOVER) configuration member.
- *job_name* is the name of the PowerExchange Listener job or started task.
- *datalib* is the user-specified name for the PowerExchange data library that was entered at installation.

You can enter the SBMJOB command at the command line.

Alternatively, you can execute the SBMJOB command by using an automated scheduler, a CL program, or a REXX procedure. For example, include the SBMJOB command in a REXX member named STARTLST and then use the following statement to start the PowerExchange Listener:

```
STREXPRC SRCMBR(STARTLST) SRCFILE(data lib/REXX)
```

Note: You cannot use the pwxcmd program to start the PowerExchange Listener.

Stopping the PowerExchange Listener

On i5/OS, use the following SNDLSTCMD command to stop the PowerExchange Listener:

```
SNDLSTCMD data lib CLOSE [CLOSEOPT(FORCE)]
```

The *data lib* variable is the user-specified name for the PowerExchange data library that was entered at installation.

Alternatively, use the pwxcmd program to issue a close or closeforce command from a Linux, UNIX, or Windows system to a PowerExchange Listener running on an i5/OS system.

Stopping a PowerExchange Listener Task for an Extraction

On i5/OS, use the following SNDLSTCMD command syntax to stop a PowerExchange Listener task for an application name:

```
SNDLSTCMD LSTMSGQLIB(data lib) LSTCMD(STOPTASK) STOPTASK(application_name)
```

Where:

- *data lib* is the user-specified name for the PowerExchange data library that was entered at installation.
- *application_name* is the name for the active extraction process that you want to stop. This name is included in a PWX-00712 message of the DISPLAY ACTIVE command output.

Alternatively, issue the pwxcmd stoptask command from a Linux, UNIX, or Windows system to a PowerExchange Listener running on an i5/OS system.

Displaying Active PowerExchange Listener Tasks

Use the DISPLAY ACTIVE command to display information about each active PowerExchange Listener task that is running on i5/OS.

Use the following command syntax to issue the command on i5/OS:

```
SNDLSTCMD LSTMSGQ(DATALIB/LISTENER) LSTCMD(DISPLAY) DISPLAYOPT(ACTIVE)
```

The *data lib* variable is the user-specified name for the PowerExchange data library that was entered at installation.

Alternatively, issue the pwxcmd listtask command from a Linux, UNIX, or Windows system to a PowerExchange Listener running on the i5/OS system.

The DISPLAY ACTIVE or pwxcmd listtask command displays the TCP/IP address, port number, application name, access type, and status.

CHAPTER 3

PowerExchange Condense

This chapter includes the following topics:

- [PowerExchange Condense Overview, 35](#)
- [PowerExchange Condense Tasks, 36](#)
- [Files That PowerExchange Condense Uses, 38](#)
- [PowerExchange Condense Message Log Files, 40](#)
- [PowerExchange Condense Operational Modes, 41](#)
- [PowerExchange Condense Configuration Parameters, 42](#)
- [QPRINT File Size Limits, 47](#)
- [Using Multiple Journals with PowerExchange Condense, 47](#)
- [Starting PowerExchange Condense, 48](#)
- [Managing PowerExchange Condense, 49](#)

PowerExchange Condense Overview

PowerExchange Condense captures change data from DB2 for i5/OS databases and writes that data to condense files. You can make the change data in the condense files available for extraction processing at user-defined intervals.

When you create a capture registration for a data source, you must set the **Condense** option to **Full** or **Part** to indicate the type of condense processing.

If you select **Full**, PowerExchange Condense uses full condense processing. Full condense processing does not maintain transactional integrity, but it can decrease the amount of change records that are processed and extracted. During a full condense cycle, PowerExchange Condense records only the last change made to a row in a condense file. If multiple changes are made to the same row, later changes supersede earlier changes, leaving only the latest change. Use the following criteria to determine if you want to use full condense processing:

- For full condense processing, tables must have a primary index or a unique key that is defined through DDS. PowerExchange Condense creates keys in the condense files based on the key columns in the source tables to uniquely identify rows.
- If a source file includes multiple data members, do not use full condense processing unless the primary key for the file is unique across all data members. Otherwise, unpredictable results might occur.
- If changes for different tables are written to the same full condense file, the record size is determined by the largest change record.

- If a change is recorded for a row in a full condense file and then a condense file switch occurs while additional changes for that row are pending, you might get a change record for the row in both condense files. This situation depends on the file switch parameters and the level of change activity in your environment.

If you select the **Part** option, PowerExchange Condense uses partial condense processing. Partial condense processing maintains transactional integrity because PowerExchange Condense does not eliminate any change. During a partial condense cycle, PowerExchange Condense writes all of the changes for successfully completed UOWs to condense files, in chronological order by UOW end time. If a DB2 for i5/OS table is updated outside of commit control, PowerExchange Condense writes changes in the order that each update occurred. To save disk space, PowerExchange Condense organizes records in partial condense files into 32 KB blocks. The records might appear incorrect if you access them through the i5/OS DSPPFM command.

Tip: Instead of using PowerExchange Condense partial condense processing, you can use the PowerExchange Logger for Linux, UNIX, or Windows to offload condense processing from the i5/OS system. For more information about remote logging, see the *PowerExchange CDC Guide for Linux, UNIX, and Windows*.

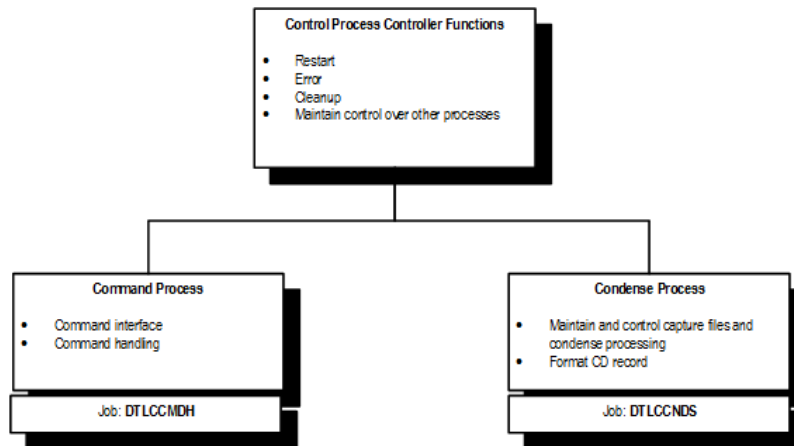
PowerExchange Condense Tasks

When you start PowerExchange Condense, the main Controller task initializes and then starts the subtasks.

The following subtasks run under the Controller:

- Command Handler, DTLCCMDH
- Condense, DTLCNDS

The following image shows the PowerExchange Condense tasks and their functions:

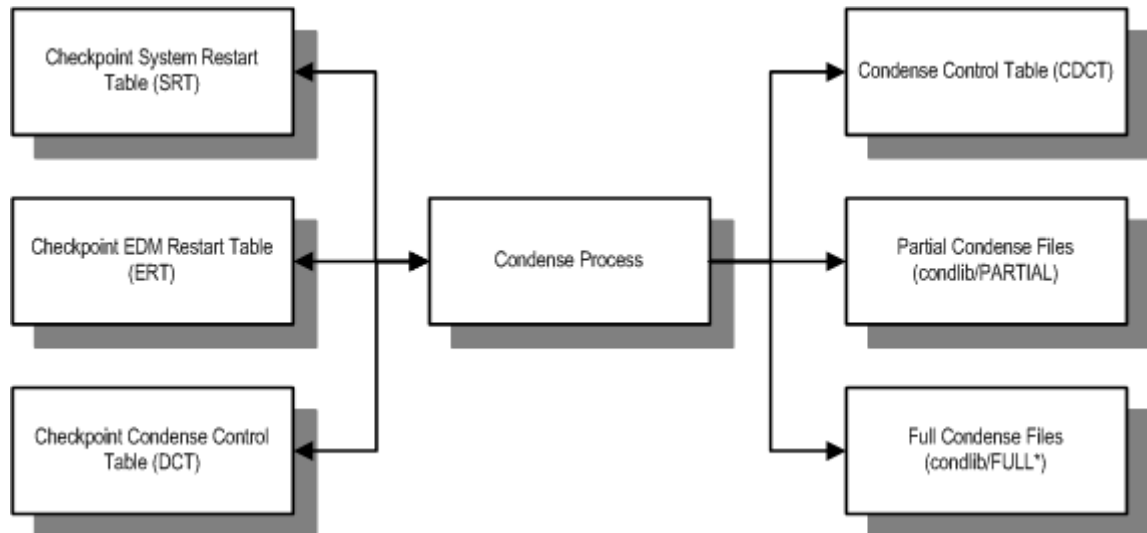


The following table describes the main functions of each PowerExchange Condense task and subtask:

Task	Description
Controller	Manages startup, error handling, cleanup, and the running of subtasks.
Command Handler	Handles the processing of commands and the command interface.
Condense	Periodically performs condense processing.

The communication among the tasks is handled by using an internal process event mechanism.

The following image shows the tables and files that a Condense process uses and generates during a condense cycle:



Controller Task

The Controller task controls shared memory. It is the first program to gain control during PowerExchange Condense startup. The Controller acquires various global memory structures and initializes them.

The Controller reads the Input Parameter File and formats the parameters into an internal format in shared memory. The Controller also rebuilds the address space from checkpoint files, if any exist. Otherwise, the Controller determines if a cold start is needed by sending a message that requires a confirmation response to a configurable message queue or to QSYSOPR if no queue was configured.

Then the Controller submits the jobs for the Command Handler and Condense subtasks (DTLCCMDH and DTLCCNDS). The Controller synchronizes the initialization of these tasks to ensure that no task can proceed beyond its initialization phase until all tasks are ready to proceed.

PowerExchange Condense writes progress messages to the message log.

After all tasks have shut down, the Controller shuts down and PowerExchange Condense ends. A return code of 0 is issued for a normal shutdown, and a return code of 8 is issued for any other circumstance.

Condense Task and Condense Cycles

The Condense task performs a condense cycle at user-defined intervals to process DB2 for i5/OS changes.

The following events trigger a condense cycle:

- PowerExchange Condense initialization.
- A CONDENSE command is issued to PowerExchange Condense from the i5/OS system, or a pwxcmd condense command is issued to PowerExchange Condense from a Linux, UNIX, or Windows system.

During the condense cycle, the Condense task reads the data written to the journal receivers for the registered tables of interest since the last condense cycle. PowerExchange Condense determines the point in the change stream from which to begin reading data based on the sequence and restart tokens.

PowerExchange Condense stores the change data in full or partial condense files.

When one of the following events occur, the condense cycle ends:

- PowerExchange Condense reaches the end of log and the wait interval that is specified in the NO_DATA_WAIT2 parameter in the CFGCOND(CAPTPARM) member elapses without any data having been received.
- If PowerExchange Condense runs in continuous mode, the interval between condense cycles that you specify in the NO_WAIT_DATA parameter of the CAPTPARM member elapses. This event causes the current condense cycle to end and another condense cycle to begin.
- A condense file switch occurs based on the file switch parameters in the CAPTPARM member.

When the condense cycle ends, PowerExchange Condense closes the current condense file, takes a checkpoint, and issues a message that identifies the point in the change stream to which it processed changes. PowerExchange Condense uses the number of checkpoint files that you specify in the CHKPT_NUM parameter of the CAPTPARM member on a rotational basis.

PowerExchange Condense also stores information about the closed condense file in the Condense Control Table (CDCT). The CDCT includes an entry for each closed condense file, including the registration tag names, UOW start and end times, condense file type, and number of records. During extraction processing, the PowerExchange Listener reads the CDCT file to determine the closed condense file from which to extract change data.

You can define the retention period for the CDCT file and the condense files in the COND_CDCT_RET_P parameter in the CAPTPARM member. PowerExchange Condense deletes files older than this retention period at initialization, during a file switch, or at shutdown.

If a PowerExchange Condense job fails, an error message is written to the i5/OS console.

Command Handler Task

The Command Handler accepts commands from users. It syntactically validates the commands and arguments and then notifies the Controller that one or more commands have been entered. Depending on the command, the Command Handler might also execute the command.

Files That PowerExchange Condense Uses

PowerExchange Condense uses multiple files to perform its processing.

Some files are allocated at installation, and other files are created dynamically by PowerExchange Condense. Also, some files are re-used within a process, whereas other files are allocated, written to, and passed to other change data capture processes.

The following table lists the files that PowerExchange Condense uses:

File Name	Library	Description
CCT	<i>datalib</i>	Contains information about capture registrations. This file is used by the PowerExchange Listener and PowerExchange Condense. The PowerExchange Listener job opens the file in read/write mode, whereas PowerExchange Condense only reads the file.
CDCT	<i>datalib</i>	Contains information about the condense files for extraction processing. PowerExchange Condense opens the file in read/write mode to update CDCT information. After a warm start, PowerExchange Condense synchronizes the CDCT file with the current checkpoint file. The PowerExchange Listener job reads the CDCT information on behalf of data extraction processes to find the condense files to process.
CDEP	<i>datalib</i>	Contains information about each extraction process that has run to provide an audit trail. The PowerExchange Listener uses this file to provide output to extraction processes and input to application groups in PowerExchange Navigator.
CFG	<i>datalib</i>	Contains the DBMOVER member with the PowerExchange configuration statements. Some of these statements are applicable to PowerExchange Condense.
CFGCOND	<i>condlib</i>	Contains the CAPTPARM member with the PowerExchange Condense parameters.
CHKPTVnn	<i>condlib</i>	CHKPTVnn checkpoint files are dynamically created and deleted by PowerExchange Condense based on parameters in the CAPTPARM member. The primary purpose of the checkpoint files is to provide the restart and sequence tokens, which indicate the log position from which to resume reading uncommitted UOWs and the the position at which to release data records that have not yet been processed. The secondary purpose of the checkpoint files is to reconstruct the CDCT file if you need to resume processing from an older point in time. Informatica recommends the use of nine checkpoint files so that you are likely to have enough checkpoint files from which to reconstruct the CDCT file.
DTLMSG	<i>dtllib</i>	Contains prototype strings for PowerExchange messages.
Full condense file	<i>condlib</i>	A condense file that is produced at <i>condlib</i> /FULL* if you select Full condense processing.
LICENSE	<i>dtllib</i>	Contains the PowerExchange license key for PowerExchange and the PowerExchange options that you purchase.
LOG	<i>datalib</i>	Contains the PowerExchange message log member, DTLLOG, that includes messages about PowerExchange Condense and PowerExchange Listener processing when alternate logging is <i>not</i> enabled. Because PowerExchange writes a lot of information to this file, you should perform routine maintenance on it to ensure that it does not become too large. Informatica recommends that you use the CLRPFM command to clear the file. Note: If you enable alternate logging by specifying the TRACING statement in the DBMOVER member, PowerExchange creates message members in the file named JOBjob_number in the <i>datalib</i> library. The member names have the prefix that is specified in the TRACING statement.
Partial condense file	<i>condlib</i>	A condense file that is produced at <i>condlib</i> /PARTIAL after each file switch if you select Part condense processing.

Synchronization of the Checkpoint File to the CDCT File

During a warm start of PowerExchange Condense, PowerExchange checks whether the records in the current checkpoint file that track the condense files match the records in the CDCT file with the same key fields. If the records in these files do not match, PowerExchange Condense synchronizes the CDCT file with the checkpoint file. Synchronization occurs during the PowerExchange Condense initialization phase only.

During synchronization, if a record is absent from the CDCT file and is present in the checkpoint file, the record from the checkpoint file is added to the CDCT file. However, if a record is absent from the CDCT file and the only corresponding record in the checkpoint file is a compressed RET record, PowerExchange cannot add the missing record to the CDCT file.

Note: RET records are compressed records that contain only fields for the source instance, registration tag, condense file name, and retention information. They are used to delete CDCT records that reach the expiration threshold that is specified in the COND_CDCT_RET_P parameter in the CAPTPARM configuration member.

For each missing CDCT record that the synchronization process detects, PowerExchange issues warning message PWX-06446 and optionally issues the operator reply message PWX-06449. You must reply Y or N to the operator reply message to indicate whether to continue or end PowerExchange Condense processing. If PowerExchange Condense continues, some change data will not be extracted because the extraction processes will not be able to find all of the condense files based on the CDCT records. If you do not need to extract change data for the registrations and time stamps reported in the PWX-06446 messages for missing CDCT records, you can suppress the operator reply messages by setting the OPER_WTOR_ENABLED parameter to N in the CAPTPARM configuration member.

Under normal circumstances, the checkpoint file and CDCT file match. However, if the synchronization process detects that the records in these files differ, the records in the checkpoint file take precedence until a subsequent PowerExchange Condense warm start resynchronizes the files. In the following atypical situations, resynchronization might be necessary:

- You want to use an older checkpoint file to restart PowerExchange Condense.
- You restored the CDCT file from an older CDCT backup file and now want to bring the CDCT up to date.
- You deleted all records for a particular instance from the CDCT file.

PowerExchange Condense Message Log Files

The files to which PowerExchange Condense writes messages depends on whether you enable alternative logging.

If you do *not* enable alternative logging, PowerExchange Condense writes messages to the DTLLOG member of the LOG file in the PowerExchange *datalib* library, by default. PowerExchange allocates this file if it does not exist. Because PowerExchange might write a lot of information to this file, Informatica recommends that you routinely use the CLRPFM command to clear the file contents and prevent the file from becoming too large.

If you enable alternative logging by specifying the TRACING statement in the DBMOVER configuration member, PowerExchange Condense writes messages to alternative message members in the JOB_{job_number} file in the *datalib* library. The member names have the prefix that is specified in the PFX parameter of the TRACING statement. If you specify the APPEND parameter in the TRACING statement, whenever PowerExchange restarts, new messages are appended to the most recent message log member. You can control the size of an alternative log member by using the SIZE parameter in the TRACING statement. When an alternative log member becomes full, PowerExchange rotates to another alternative log member, which prevents out-of-space conditions. Also, PowerExchange buffers messages and writes them to the alternative

log members on disk based on a specific flush interval. This type of buffered logging can reduce I/O activity on the log files. For more information about alternative logging, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“PowerExchange Event Message Queue” on page 61](#)
- [“PowerExchange Listener Message Log Files” on page 16](#)

PowerExchange Condense Operational Modes

A PowerExchange Condense task can run in batch or continuous mode. To set the operational mode, use the COLL_END_LOG parameter in the PowerExchange Condense configuration member.

To determine which mode to use, review the following information:

batch mode

A single condense cycle runs and then the condense job shuts down. For example, you might want to use this mode if you insert a single condense process after update jobs in an automated schedule.

continuous mode

The condense job runs for a long period. PowerExchange Condense “sleeps” after each condense cycle. The next condense cycle is triggered by one of the following events:

- The wait period defined by the NO_DATA_WAIT parameter elapses.
- A CONDENSE command is manually entered on the i5/OS system, or a pwxcmd condense command is issued from a Linux, UNIX, or Windows system to the PowerExchange Condense process on the i5/OS system.
- A FILESWITCH command is manually entered on the i5/OS system, or a pwxcmd fileswitch command is issued from a Linux, UNIX, or Windows system to the PowerExchange Condense process on the i5/OS system.

During continuous processing, condense files can be read by a change extraction process only after a file switch occurs. File switch processing closes the open condense files that contain data and opens a new set of condense files to accept change data. Extractions can process only the closed condense files. A file switch occurs when one of the following events occurs:

- The file switch criteria defined by the FILE_SWITCH_CRIT and FILE_SWITCH_VAL parameters are met.
- A FILESWITCH command or pwxcmd fileswitch command is issued.

Note: The file switch does not occur if the current condense file contains no data. The next file switch attempt occurs when the criteria defined by the FILE_SWITCH_CRIT and FILE_SWITCH_VAL parameters are met. If the current condense file still contains no data, this cycle continues at the set intervals until the file contains data.

In continuous mode, the PowerExchange Condense process does not shut down. To shut down PowerExchange Condense, issue the PowerExchange Condense SHUTDOWN command from the command line on the i5/OS system. Alternatively, issue the pwxcmd shutdown command from a Linux, UNIX, or Windows system to the PowerExchange Condense process running on the i5/OS system.

RELATED TOPICS:

- [“PowerExchange Condense Configuration Parameters” on page 42](#)

PowerExchange Condense Configuration Parameters

Configure PowerExchange Condense parameters in the CAPTPARM configuration member.

PowerExchange provides a sample CAPTPARM member in the following libraries and files:

- *condlib*/CFGCOND
- *datalib*/CFG

In the command for starting PowerExchange Condense, the PARM parameter specifies the location of the customized CAPTPARM member.

Parameter descriptions:

CAPT_IMAGE={AI|BA}

The type of data image that PowerExchange Condense captures to condense files. PowerExchange Condense can capture after images only or both before images and after images of the data.

The capture image type must be consistent with the image type delivered to the target during extraction processing.

Valid values:

- **AI.** After images only. If you enter AI, the following limitations apply:
 - You cannot extract before images to the target.
 - You cannot use DTL_BI columns in extraction maps.
 - If you add DTL_CI columns to extraction maps, any Insert or Delete operations result in Null values in these columns.
- **BA.** Both before and after images.

Default is AI.

Informatica recommends that you specify BA so that you have the flexibility to use either AI or BA for the PowerCenter **Image Type** connection attribute for extraction processing.

CHKPT_BASENAME =library/base_file_name

The library and base file name that PowerExchange Condense uses for generating checkpoint files. At installation, this value is set to the following value:

```
CONDLIB/CHKPT
```

To create the specific checkpoint file names, PowerExchange Condense appends *Vn* to the base file name, where *n* is a number from 0 to 99. For example:

```
CONDLIB/CHKPTV9
```

CHKPT_NUM={number|3}

The number of PowerExchange Condense checkpoint files.

Minimum value is 3. Informatica recommends nine checkpoint files so that enough checkpoint files are likely to be available for reconstructing the CDCT file, if necessary.

COLL_END_LOG={0|1}

Indicates whether PowerExchange Condense runs in continuous mode or batch mode.

Valid values:

- **0.** Continuous mode. After each Condense run, PowerExchange Condense waits for the number of minutes that is specified in the NO_DATA_WAIT parameter and then starts another condense cycle. Use this setting if you require 24 x 7 condense processing.
- **1.** Batch mode. PowerExchange Condense shuts down after a single Condense run. For example, use batch mode if Condense is scheduled to run after a specific batch update job and then shut down.

Default is 0.

COND_CDCT_RET_P={days|60}

The number of days to retain condense files, CDCT records, and the checkpoint file records that point to condense files. After the retention period elapses for these files and records, PowerExchange Condense deletes them the next time a PowerExchange Condense startup, checkpoint, fileswitch, or shutdown operation runs.

Enter a number of days greater than 0. This period must be long enough for extraction sessions to complete processing the condense files before the files are deleted. Default is 60.

COND_DIR=directory

The name of the directory for condense files.

CONDENSE_SHUTDOWN_TIMEOUT={seconds|600}

The maximum number of seconds that PowerExchange Condense (DTLCAON) waits after receiving a SHUTDOWN or pwxcmd shutdown command before it stops the shutdown process and fails.

Enter a number from 0 through 2147483647. Default is 600 seconds (10 minutes).

Set this value based on your environment. You might need to use a value greater than the default value if you have a large number of tables for PowerExchange Condense to process.

CONDENSENAME=service_name

The user-defined name of the command-handling service for a PowerExchange Condense process to which you issue pwxcmd commands.

This service name must match the service name that is specified in the associated SVCNODE statement in the DBMOVER configuration file. The SVCNODE statement specifies the TCP/IP port on which this service listens for pwxcmd commands.

Enter a string of up to 12 characters for the service name. This is the maximum length of the service name that you can specify in the SVCNODE statement. No default is provided.

DB_TYPE=AS4

The source database type. This value must be **AS4** for DB2 for i5/OS databases.

DBID=instance_name

The name of the DB2 for i5/OS instance. When used in conjunction with DB_TYPE value, it defines criteria for selecting capture registrations recorded in the CCT file.

This value must match the instance name that is specified for the registration group that contains the capture registrations of interest.

FILE_SWITCH_CRIT={M|R}

The type of units to use for the FILE_SWITCH_VAL value. These parameters provide criteria for determining when PowerExchange Condense does an automatic file switch.

Valid values:

- **M.** Minutes.
- **R.** Records.

Default is M.

FILE_SWITCH_VAL={number|30}

The number of FILE_SWITCH_CRIT units at which PowerExchange Condense does an automatic file switch.

Enter a number greater than 0. Default is 30.

For example, if this value is 100000 and FILE_SWITCH_CRIT=R, PowerExchange Condense does a file switch after every 100,000 records.

If FILE_SWITCH_CRIT value is M but the condense file does not contains change data when the FILE_SWITCH_VAL interval is met, a file switch does not occur.

JRNL=library/journal_name

The fully qualified library name and journal name to use instead of the journal that is specified in the PowerExchange capture registration. For example:

```
JRNL=STQA/NEWJOURNAL
```

KEY_CHANGE_ALW={N|Y}

Determines how PowerExchange Condense responds if a key in a source table is changed.

By default, PowerExchange Condense expects keys to not be changed after registration of the source table. Use this parameter if you want to change this behavior.

Valid values:

- **N.** PowerExchange Condense fails if a key is changed.
- **Y.** PowerExchange Condense ignores the change to the key and continues processing.

Default is N.

NO_DATA_WAIT={minutes|60}

The number of minutes that PowerExchange Condense must wait before starting the next condense cycle, when running in continuous mode.

Enter a number greater than 0. Default is 60.

If the FILE_SWITCH_CRIT parameter is set to M for minutes and the FILE_SWITCH_VAL value is less than the NO_DATA_WAIT value, PowerExchange Condense uses the FILE_SWITCH_VAL value instead.

If COLL_END_LOG is set to 1 to run PowerExchange Condense in batch mode, this parameter is ignored.

NO_DATA_WAIT2={seconds|600}

The number of seconds that PowerExchange Condense waits after it reaches the end-of-log to receive more data from the journal. If this wait period elapses without PowerExchange Condense having received more data, the condense cycle stops.

Enter a number greater than 0. Default is 600 seconds (10 minutes). Informatica recommends 60 seconds (1 minute).

The optimal value for this parameter varies based on work load:

- If the parameter value is too low, PowerExchange Condense might incorrectly report that no data exists. A delay can occur if a large UOW that contains thousands of rows is started.
- If the parameter value is too high, an excessive period of apparent inactivity might occur before control returns to the Command Handler and you can enter commands.

OBJLOC=(minutes,message_queue_name)

The number of minutes that PowerExchange Condense waits after an object is locked before sending a message to the specified i5/OS message queue. The message prompts the user to indicate whether PowerExchange Condense should retry object access.

Enter a number from 0 through 10 followed by a comma and then the message queue name. For example:

```
OBJLOC=(0,*LIBL/QSYSOPR)
```

OPER_WTOR_ENABLED={N|Y}

Controls whether PowerExchange sends PWX-06449 messages that require a user reply when PowerExchange Condense detects that a record is missing from the CDCT file after synchronization of the checkpoint file to the CDCT file. This message requires a user reply of Y or N.

Synchronization occurs at PowerExchange Condense initialization, after a warm start. During synchronization, if any record in the checkpoint file does not match a record in the CDCT file based on the key fields, the checkpoint file record is not added to the CDCT file. The CDCT file is then missing a record that points to a condense file from which to extract change data for a registered source object and time stamp. For each record that is missing from the CDCT file, PowerExchange issues the following PWX-06446 warning message followed by the optional PWX-06449 user reply message:

```
PWX-06446 Checkpoint to CDCT synchronization not done for time stamp "time_stamp"  
tag "registration_tag" number record_count reason.  
PWX-06449 There are missing resources, these can not be recovered. Continue? (Y/N)
```

You must respond Y or N to each PWX-06449 message to indicate whether to continue or end PowerExchange Condense processing. Use this parameter to suppress these user reply messages if you want PowerExchange Condense to continue without interruption and if the loss of some change data during extraction processing is tolerable for the reported time stamps and registrations.

Valid values:

- **Y.** When PowerExchange Condense detects missing records in the CDCT file after checkpoint-to-CDCT synchronization, PWX-06449 messages are displayed as user reply messages and written to the PowerExchange message log that is specified in the AS400USRMSGQ statement in the DBMOVER member of the CFG configuration file. You must reply Y or N to each of the user reply messages to indicate whether PowerExchange Condense processing should continue without the CDCT records or stop.
 - If you reply Y, PowerExchange Condense continues processing. In this case, extraction processes will not be able to find some condense files based on the CDCT file for data extraction. Skipping some change data might be acceptable if the condense files contain old data or if the CDC workflow already processed these condense files. Use the time stamps reported in the PWX-06446 messages to determine if the data is old. If you need to extract the data, you must cold start the CDC session from earlier point in time.
 - If you reply N, PowerExchange Condense stops.
- **N.** When PowerExchange Condense detects missing records in the CDCT file, PWX-06449 messages are suppressed. PowerExchange Condense processing continues uninterrupted without the PWX-06449 user reply messages.

Default is Y.

REG=(value)

Identifies the capture registrations that PowerExchange Condense will use to process source data. For example:

```
REG=(myreg)
```

Note: For i5/OS sources, you can specify the capture registrations in either the CAPTPARM member or in the default DBMOVER configuration file. However, do not specify them in both locations.

If you do not specify registrations, condense processing is as before, everything in the instance which is labelled for condense.

RESTART_TOKEN=token and SEQUENCE_TOKEN=token

The restart tokens that PowerExchange Condense uses to determine the start point in the change data stream when you cold start PowerExchange Condense.

Valid values:

- Specific non-zero token values that are valid in both parameters. PowerExchange Condense starts from the point in the journal receivers that the specific token values define.
- Only zeroes for the entire token values in both parameters. PowerExchange Condense starts from the beginning of the oldest receiver in the current chain of receivers.
- No parameter values. PowerExchange Condense starts from the beginning of the current chain of journal receivers.

Note: The PowerExchange sample 80-column configuration file lets you enter 65 characters for the token values. However, the sequence token is a 72-byte hexadecimal value. If you use the sample configuration file, you can optionally use the SEQUENCE_TOKN2 and RESTART_TOKN2 parameters to split the token values at any point you specify. PowerExchange concatenates the two parts for processing.

SIGNALLING={N|Y}

Indicates whether PowerExchange Condense attempts to handle abnormal termination conditions.

Valid values:

- **Y.** PowerExchange Condense takes automatic action when certain types of abnormal termination conditions occur, such as memory corruption. The system attempts to close down in an orderly manner.
- **N.** No automatic trapping of system errors occurs. Instead, the operating system uses its default error handling. Usually, the operating system reports the offending program line and dump memory.

VERBOSE={N|Y}

Indicates whether PowerExchange Condense issues verbose or terse messages for frequent PowerExchange Condense activities such as cleanup processing, checkpoints, condense cycles, and file switch processing.

Valid values:

- **Y.** Verbose messaging.
- **N.** Terse messaging

Default is Y.

QPRINT File Size Limits

To perform batch processing, PowerExchange overrides the normal QPRINT output file size limitations so that no maximum record size limitation exists. This override prevents QPRINT from ever having the message wait (*MSGW) status.

Using Multiple Journals with PowerExchange Condense

You can configure the PowerExchange change data capture environment to capture changes from more than one journal. To do this, you need to establish a PowerExchange Condense, or CONDLIB, library for each journal that you want to condense.

If you have multiple journals, or you might have them in the future, copy the supplied CONDLIB as a template before you customize or use it. Each CONDLIB library has its own CAPTPARM member that specifies the journal to be condensed.

A single PowerExchange Listener can be used for multiple PowerExchange Condense jobs and extractions from multiple journals.

Within each CONDLIB library, you must customize the CAPTPARM member to point to the appropriate journal and CONDLIB library.

For example, to run PowerExchange Condense for the first journal, configure the following parameters in *condlib*/CFGCOND(CAPTPARM):

```
DBID=instance1 (This uses the journal in DBMOVE.)  
CHKPT_BASENAME=condlib/CHKPT  
COND_DIR=condlib
```

To run PowerExchange Condense for a second journal, configure the following parameters in a second *condlib* called *condlib2*/CFGCOND(CAPTPARM):

```
DBID=instance2  
JRNL=library/journal2 (The journal override.)  
REG=(reg1)  
REG=(reg2)  
REG=(reg3,DB=library/file)  
CHKPT_BASENAME=condlib2/CHKPT  
COND_DIR=condlib2
```

To run PowerExchange Condense for a third journal, configure the following parameters in a second *condlib* called *condlib3*/CFGCOND(CAPTPARM):

```
DBID=instance3  
JRNL=library/journal3 (The journal override.)  
CHKPT_BASENAME=condlib3/CHKPT  
COND_DIR=condlib3
```

Capture registrations can be specified in either the CAPTPARM member or the DBMOVE configuration member but not in both members. If no capture registrations are specified, all capture registrations that specify a Condense Type other than None are included.

Starting PowerExchange Condense

Use the SBMJOB command to start the PowerExchange Condense DTLCACON program.

Before starting PowerExchange Condense, complete the following tasks:

- Configure the DBMOVER and CAPTPARM configuration members.
- Verify that the capture registrations for the DB-TYPE and DBID that you want to use for the current run were added to the CCT file. If necessary, you can disable or delete registrations in the PowerExchange Navigator.
- Verify that the checkpoint files are in the correct state. If you need to cold start PowerExchange Condense, verify that no checkpoint files exist for the CHKPT_BASENAME mask in the CAPTPARM member.
- For a cold start, configure the start point. The start point depends on the RESTART_TOKEN and SEQUENCE_TOKEN parameter settings in the CAPTPARM member:
 - If the RESTART_TOKEN and SEQUENCE_TOKEN parameters are not defined, PowerExchange Condense starts from the current position in the journal.
 - If the RESTART_TOKEN and SEQUENCE_TOKEN parameters are set to zero, PowerExchange Condense starts from the beginning of the journal. In this case, PowerExchange Condense might reprocess data from previous runs.
 - If the RESTART_TOKEN and SEQUENCE_TOKEN parameters specify valid non-zero values, PowerExchange Condense starts from the point that is defined by the token values.

In the SBMJOB command, you can include the optional CONFIG and LICENSE parameters to specify an override configuration file or license key file that you want to use instead of the default DBMOVER configuration file or license.key file. The override files must have a file name or path that is different from that of the default files. These override files take precedence over any override configuration and license key files that you specified in the optional PWX_CONFIG and PWX_LICENSE environment variables.

To start PowerExchange Condense on the same subsystem as the PowerExchange Listener, use the following command syntax:

```
SBMJOB CMD(CALL PGM(DTLCACON) PARM('[CS=library/file(mycondense_config_member)]'
 '[CONFIG=library/file(myconfig_member)]' '[LICENSE=library/file(mylicensekey_member)]')
 JOB(job_name) JOBD(datalib/DTLLIST) JOBQ(*JOBQ) PRTDEV(*JOBQ) OUTQ(*JOBQ)
 CURLIB(datalib) INLLIBL(*JOBQ) SPLFACN(*JOBQ))
```

This syntax includes the following variables:

- *jobname* is the job name.
- *datalib* is the user-specified name for the PowerExchange data library that was entered at installation.

Notes:

- In the CS parameter, you can specify *condlib*/CFGCOND(CAPTPARM), where *condlib* is the name of the library that contains the PowerExchange Condense files. To run a job for each journal that PowerExchange Condense processes, specify the appropriate *condlib* library name.
- To prevent the system job table from exceeding its maximum size, the job description specifies SPLFACN(*DETACH). Because this is not the system default, including the SPLFACN(*JOBQ) parameter ensures that the parameter is specified correctly.

In the syntax, you can specify one or more of the following optional parameters:

CS

Specifies the library, file name, and member name for any condense configuration member that you created and want to use instead of the default *condlib*/CFGCOND(CAPTPARM) configuration member.

CONFIG

Specifies the library, file name, and member name of any override dbmover configuration member that you created and want to use instead of the default *datalib/CFG(DBMOVER)* configuration member.

LICENSE

Specifies the library, file name, and member name for any override license key member that you created and want to use instead of the default license key member.

Note: In these parameters, the full path is required only if the member does not reside at the default location.

After the SBMJOB command is submitted, the DTLCACON program submits the following jobs:

```
Command Handler (DTLCCMDH)
Condense (DTLCCNDS)
```

Note: You cannot use the pwxcmd program to start PowerExchange Condense.

Managing PowerExchange Condense

You can issue commands to display the status of the PowerExchange Condense controller and subtasks, or to perform a file switch. You can also configure PowerExchange to write PowerExchange Condense performance statistics to the PowerExchange log file.

PowerExchange Condense Progress Messages and Tracing

In the case of a cold start with no checkpoint files, the following message, which requires a YES/NO (Y/N) response, is issued:

```
No checkpoint files, cold-start from specified restart point? (Y/N)
```

PowerExchange Condense issues messages during startup that indicate the progress of the start or restart operation. Usually, the last of these messages is:

```
PWX-06111 Controller - All tasks initialization complete.
```

This message appears just before PowerExchange Condense goes into its main loop waiting for events. Alternatively error messages are issued in the event of startup problems.

When shutting down, PowerExchange Condense issues messages PWX-06065 and PWX-06039, which indicate that collection has ended with a specified highest portion of the log reached, and that the capture system is shutting down.

Throughout the rest of a capture, the controller task issues little or no output, unless tracing is switched on. The trace literal is "CONTROLLER." There are a variety of trace levels reflecting importance and severity of information.

Displaying PowerExchange Condense Status

To display the current status of the Condense Controller and each subtask for a PowerExchange Condense job, issue the DISPLAY STATUS command.

Use the following syntax to issue the DISPLAY STATUS command:

```
SNDPWXCMD CMDHDLRLIB(condlib) DTLCMD(DISPLAY) DISPLAYOPT(STATUS)
```

The *condlib* variable is the name of the library that contains the condense files, which you specified in the COND_DIR parameter of the CAPTPARM configuration file.

Alternatively, issue a `pwxcmd displaystatus` command from a Linux, UNIX, or Windows system to a PowerExchange Condense process running on an i5/OS system.

Stopping PowerExchange Condense

You can use the following commands to shut down PowerExchange Condense:

SHUTDOWN

The SHUTDOWN command causes a shutdown event to be passed to the other subtasks and the Controller. The condense subtask closes any open condense files, writes the CDCT records, and takes a checkpoint that contains the latest restart tokens. All of the other subtasks shut down. Each of these subtasks report when shutdown is complete. Finally, the Controller shuts down, ending the Condense job.

Use the following syntax to issue the SHUTDOWN command:

```
SNDPWXCMD CMDHDLRLIB(condlib) DTLCMD(SHUTDOWN)
```

Where *condlib* is the name of the library that contains the PowerExchange Condense files.

Alternatively, issue a `pwxcmd shutdown` command from a Linux, UNIX, or Windows system to a PowerExchange Condense process running on an i5/OS system.

Restriction: PowerExchange Condense does not accept SHUTDOWN or `pwxcmd shutdown` commands while a condense process is in progress. Condense processing is complete when the last message for PowerExchange Condense is displayed in the DTLLOG file, as follows:

```
PWX-06421 Condense: 02/09/24 15:44:50 Starting wait on commands for 300 Seconds
```

Access the DTLLOG file by issuing the following command:

```
DSPPFM condlib/LOG (DTLLOG)
```

Where *condlib* is the name of the library that contains the PowerExchange Condense files.

SHUTCOND

The SHUTCOND command performs the same processing as the SHUTDOWN command, except it performs a final condense operation before passing the shutdown event to the other subtasks.

Use the following syntax to issue the SHUTCOND command:

```
SNDPWXCMD CMDHDLRLIB(condlib) DTLCMD(SHUTDCOND)
```

Where *condlib* is the name of the library that contains the PowerExchange Condense files.

Alternatively, issue a `pwxcmd shutcond` command from a Linux, UNIX, or Windows system to a PowerExchange Condense process running on an i5/OS system.

Performing a Fileswitch

To close the current file or files and start new ones, issue the FILESWITCH command.

Use the following syntax to issue the FILESWITCH command:

```
SNDPWXCMD CMDHDLRLIB(condlib) DTLCMD(FILESWITCH)
```

Where *condlib* is the name of the library that contains the PowerExchange Condense files.

Alternatively, issue a `pwxcmd fileswitch` command from on a Linux, UNIX, or Windows system to a PowerExchange Condense process running on an i5/OS system.

Determining If PowerExchange Condense Captured Changes

To determine if PowerExchange Condense captured committed changes for registered tables of interest, look for the following message in the PowerExchange message log:

```
PWX-09967 CAPI i/f: End of log for time 10/05/20 14:18:18 reached
```

This message indicates that PowerExchange Condense read all of the changes that were available at the start the condense cycle. Look for this message if a condense file does not receive change data within the time period you expect. Delays can occur for various reasons.

Backing Up PowerExchange Condense Output Files

Periodically, back up PowerExchange Condense checkpoint files, CDCT file, and condense files. If the existing files become damaged or deleted, you can then use the backups to restore the files.

Informatica recommends that you back up the checkpoint files followed by the CDCT file and then the condense files. Back up the files during a period of low activity.

The CDCT file must be backed up in coordination with the checkpoint files. For every $(2n-1)$ condense cycles completed, where n is the number of checkpoint files that you use, you must back up the CDCT at least once. If you do not back up the CDCT file in coordination with the checkpoint files and file corruption occurs, the CDCT file and the condense files to which the CDCT file points might no longer be synchronized.

For example, if you use eight checkpoint files and perform a file switch every 20 minutes, back up the CDCT file at least every $((2 * 8) - 1) * 20 = 300$ minutes. Back up the checkpoint files before they are overwritten by a later condense cycle.

The frequency with which you back up the condense files is at your discretion.

Generating PowerExchange Condense Performance Statistics

If you want to generate performance statistics for PowerExchange Condense, set `VERBOSE=Y` in the configuration file. The performance statistics are written to the PowerExchange log file.

The following types of statistics are reported:

- **Condense timing - total.** Number of CPU seconds that PowerExchange Condense took to get data records from CAPI and write the records to the condense files. This timing indicates the amount of time that it took PowerExchange Condense to process the data records.
- **Condense timing - data.** Number of CPU seconds that PowerExchange Condense took for all read and write processing.
- **CAPI read timing - data.** Number of CPU seconds that CAPI took to read data records from the change stream.
- **CAPI read timing - total.** Number of CPU seconds that CAPI took for all read operations.
- **Quantity of data processed.** Total number of records, updates, UOWs, and bytes that PowerExchange Condense processed.

Note: These timings do not include the time for PowerExchange Condense initialization and for repositioning after a restart.

The non-data timings indicate the amount of processing required to reach the restart point in the change stream after PowerExchange Condense issues the first read call.

CHAPTER 4

DB2 for i5/OS Change Data Capture

This chapter includes the following topics:

- [DB2 for i5/OS Change Data Capture Overview, 52](#)
- [Planning for DB2 Change Data Capture, 53](#)
- [Configuring DB2 Journaling for Change Data Capture, 56](#)
- [Configuring PowerExchange for DB2 Change Data Capture, 61](#)
- [Managing DB2 Change Data Capture, 64](#)
- [Refreshing the PowerExchange Environment After an i5/OS Upgrade, 71](#)
- [Generating SQL Statements to Re-create a Source or Target for Troubleshooting, 71](#)

DB2 for i5/OS Change Data Capture Overview

PowerExchange reads change data from the DB2 journal receivers for the database that contains the source tables. For CDC to work, source tables must be configured to use journaling.

If you configure PowerExchange Condense, PowerExchange Condense captures change data from the DB2 journal receivers and writes that data to condense files. The change data is then extracted from the condense files. Benefits of using PowerExchange Condense include faster restart of change data capture and no prolonged retention of journal receivers.

With either CDC strategy, you must define a capture registration for each source table. In the capture registration, you can select a subset of columns for which to capture data. PowerExchange generates a corresponding extraction map.

If a source table contains columns in which you store data that is inconsistent with the column datatype, you can create a data map to manipulate that data with expressions. For example, if you store packed data in a CHAR column, you can create a data map to manipulate that data to prepare it for loading to a target. Then, you merge the data map with an extraction map.

PowerExchange works in conjunction with PowerCenter to extract change data and load it to one or more targets. In PowerCenter, you must create a mapping, workflow, session, and connection. When you are ready to replicate change data, you just verify that the capture registration is active and start the workflow.

RELATED TOPICS:

- [“PowerExchange CDC Components” on page 11](#)
- [“PowerExchange CDC Architecture” on page 12](#)
- [“PowerExchange Condense Processing” on page 13](#)
- [“Change Data Extraction” on page 14](#)
- [“PowerExchange Integration with PowerCenter” on page 14](#)

Planning for DB2 Change Data Capture

Before you configure DB2 change data capture, verify that the following prerequisites and user authority requirements are met. Also, review the restrictions so that you can properly configure CDC.

DB2 for i5/OS CDC Prerequisite

To register DB2 tables for capture with PowerExchange, you must configure journaling for the database physical files and specify that both before and after images are saved.

DB2 Datatypes Supported for CDC

PowerExchange supports most DB2 for i5/OS datatypes for CDC.

The following table identifies the DB2 for i5/OS source datatypes that PowerExchange supports and does not support for CDC:

DB2 Datatype	Supported for CDC?
BIGINT	Yes
BINARY	Yes
BLOB	No
BOOLEAN	Yes, as of Db2 for i version V7R5M0
CHAR	Yes
CLOB	No
DATALINK	No
DATE	Yes
DBCLOB	No
DECFLOAT	No
DECIMAL	Yes
DISTINCT (user-defined)	No

DB2 Datatype	Supported for CDC?
DOUBLE	Yes
FLOAT (floating-point)	Yes
GRAPHIC	Yes
INTEGER	Yes
LONG VARCHAR	Yes
LONG VARGHAPHIC	Yes
NUMERIC	Yes
REAL	Yes
ROWID	No
SMALLINT	Yes
TIME	Yes
TIMESTAMP	Yes
VARBINARY	Yes
VARCHAR	Yes
VARGRAPHIC	Yes

DB2 for i5/OS CDC Considerations

The following considerations and restrictions apply to DB2 CDC processing:

- You can register only tables that are associated with physical files for change data capture.
- The maximum length of a DB2 row or column for which PowerExchange can capture changes is 128,000 bytes.
- PowerExchange cannot process journals for registered tables that include minimized journal entries. Verify that you do not use minimized journal entries. Use the CRTJRN or CHGJRN command on the source i5/OS system to set MINENTDTA to *NONE.
- PowerExchange cannot capture changes that result from a Clear Physical File Member (CLRPFM) command or SQL TRUNCATE operation on DB2 for i5/OS tables that are registered for change data capture. PowerExchange CDC processing treats a CLRPFM command and TRUNCATE operation in the same way. i5/OS records only a single journal entry for all delete operations that result from a CLRPFM command or TRUNCATE operation. However, a single journal entry is not sufficient for PowerExchange to properly replicate CLRPFM or TRUNCATE deletions to CDC targets. By default, when PowerExchange encounters a journal entry for a CLRPFM command or TRUNCATE operation, CDC processing stops.

Note: DB2 for i5/OS 7.2 introduced support for SQL TRUNCATE operations on DB2 tables.
- If you use SQL fast delete processing, SQL DELETE operations are not written to the DB2 journals. To use DB2 for i5/OS CDC, you must prevent SQL fast delete processing by setting a query option in the QAQQINI

file. By default, the SQL_FAST_DELETE_ROW_COUNT query option specifies *DEFAULT. You must change this default by setting the SQL_FAST_DELETE_ROW_COUNT parameter to *NONE. For more information about this parameter and the query option file, see the IBM DB2 for i5/OS documentation.

- For IBM i5/OS servers with a QPWDVLV password level setting of 3, PowerExchange supports mixed-case passwords and long passwords of up to 30 characters.
- PowerExchange CDC does not honor Row and Column Access Control (RCAC) rules because of a DB2 limitation. When PowerExchange tries to process journal entries for tables that have RCAC rules applied, the columns or rows are not masked or filtered based on the rules. DB2 for i5/OS 7.2 introduced RCAC rules to help database administrators control the visibility of sensitive DB2 data.

i5/OS Security Requirements

PowerExchange has some i5/OS security requirements. To install and run PowerExchange, ensure that these security requirements are met.

The following table describes the PowerExchange security requirements:

File Name	Library	PowerExchange Listener and PowerExchange Condense
AUTHSKLST	<i>datalib</i>	*USE Note: Used for PowerExchange task administration and created with *EXCLUDE authority during the installation process. Grant this authority level only to user profiles that will be used to list or stop PowerExchange subtasks.
AUTHSKSTP	<i>datalib</i>	*USE Note: Used for PowerExchange task administration and created with *EXCLUDE authority during the installation process. Grant this authority level only to user profiles that will be used to list or stop PowerExchange subtasks.
CCT	<i>datalib</i>	*CHANGE
CDCT	<i>datalib</i>	*CHANGE
CDEP	<i>datalib</i>	*CHANGE
CFG	<i>datalib</i>	*USE
CFGCOND	<i>condlib</i>	*USE
DTLMSG	<i>dtllib</i>	*USE
LICENSE	<i>dtllib</i>	*USE
LOG	<i>datalib</i>	*CHANGE

Also, PowerExchange dynamically creates other files, such as those for capture registrations and data maps. Access to these files depends on the following settings:

- System security value QCRTAUT.
- The value you specify with the library-created authority parameter on the CRTLIB command for the library in which the objects are stored.

The default authority to these objects for users other than the object owner is set accordingly. This system security value can be one of the following values:

- *EXCLUDE
- *USE
- *CHANGE
- *ALL

Journal and File Security

You must grant the appropriate level of authority for accessing journal and file objects that PowerExchange uses to the user ID that runs the PowerExchange Listener and PowerExchange Condense.

The following table lists these objects and their authority requirements:

Object	Authority
Journal	*USE *OBJEXIST
Journal Library	*USE
Journal Receiver	*USE
Journal Receiver Library	*USE
File	*USE
File Library	*USE

Configuring DB2 Journaling for Change Data Capture

To register tables for capture, you must configure tables and physical files to write before and after images to journals. PowerExchange reads change data from journals and requires both before and after images to properly process change data.

You can capture changes from local or remote journals.

Use the following DB2 for i5/OS command to start journaling for a physical file for which you want to capture change data:

```
STRJRNPF FILE(library/file) JRN(jlib/jfile) IMAGES(*BOTH) OMTJRNE(*OPNCLO)
```

The following table describes the parameters in this command:

Parameter	Description
library/file	Library and physical file for which changes are written to the journal.
jlib/jfile	The library and journal to which changes are written.

Parameter	Description
IMAGES(*BOTH)	Before and after images are written to the journal.
OMTJRNE(*OPNCLO)	Open and close entries are omitted from the journal.

RELATED TOPICS:

- [“Using Remote Journals with PowerExchange” on page 59](#)

Availability of Journal Receivers for DB2 CDC

PowerExchange extracts change data for DB2 tables registered for capture by reading journal entries from journal receivers. On busy systems, journal receivers can use a lot of disk storage space and need to be deleted at regular intervals.

Ensure that the journal receivers that are required for change data extraction are available to PowerExchange for read processing. Do not delete a journal receiver until the change data has been read from it and the restart tokens for the CDC sessions or PowerExchange Condense jobs no longer reference the receiver.

To control the deletion of journal receivers, you can use the CRTJRN (Create Journal) or CHGJRN (Change Journal) command to set the MNGRCV (Manage Receivers) and DLTRCV (Delete Receivers) attributes for the journals that are associated with the journal receivers. Informatica recommends the following settings:

- MNGRCV(*SYSTEM)
- DLTRCV(*NO)

Note: If you specify DLTRCV(*YES), the journal receivers might be deleted during an IPL of the i5/OS server. In this case, PowerExchange CDC processing loses the entry that corresponds to the restart token.

Also, PowerExchange provides the following configuration options in the DBMOVER member in the CFG file to lock a journal receiver that it is processing and to notify you when that processing is complete:

- **AS4JRNEXIT.** Use this journal exit parameter on the AS4J CAPI_CONNECTION statement to prevent deletion of journal receivers from which PowerExchange is reading change data for DB2 tables registered for capture.
- **AS400EVENTMSGQ.** Use this parameter to specify a message queue to which to write the following messages related to journal processing:
 - If you specify AS4JRNEXIT=Y on the AS4J CAPI_CONNECTION statement, PowerExchange writes message DTL3001 to the message queue when PowerExchange finishes extracting change data from a journal receiver.
 - If you specify ALWCLRPFM=Y on the AS4J CAPI_CONNECTION statement, PowerExchange writes message DTL3002 to the message queue if a journal entry for a CLRPFM command is encountered. PowerExchange cannot replicate changes that result from a CLRPFM command.

RELATED TOPICS:

- [“Determining the Journal Receiver Needed for Restart Processing” on page 65](#)

PowerExchange Journal Exit

To use the PowerExchange journal exit, specify Y for the AS4JRNEXIT parameter on the AS4J CAPI_CONNECTION statement.

When you install PowerExchange and run the CRTPWXENV command, you must provide the following information:

- In the JRNEXTSEQ parameter, an exit program number for the DTLRCVRX program
- In the CONDLIB parameter, a library name for the CONDLIB library

PowerExchange installs the DTLRCVRX exit program in the i5/OS system at the Delete Journal Receiver exit point, QIBM_QJO_DLT_JRNRCV, and uses this program to prevent the deletion of the journal receivers that are currently being processed. Use the following command to display the exit programs installed for the QIBM_QJO_DLT_JRNRCV exit point:

```
WRKREGINF EXITPNT(QIBM_QJO_DLT_JRNRCV)
```

The DTLRCVRX exit program, by default, runs under the profile that you use to perform the PowerExchange installation. You can specify a different user profile in the PGMDTA command parameter of the ADDEXITPGM command.

For example:

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRCV) FORMAT(DRCV0100) PGMNBR(5705) PGM(CONDLIB/  
DTLRCVRX) THDSAFE(*UNKNOWN) MLTTHDACN(*SYSVAL) TEXT('PowerExchange DLTJRNRCV Exit  
Point') REPLACE(*YES) CRTEXITPNT(*NO) PGMDTA(*JOB 10 MYUSRPRF)
```

The underlined values should be the same as those used in the CRTPWXENV command. You can view the current values by displaying exit programs for the exit point QIBM_QJO_DLT_JRNRCV, which is available through the WRKREGINF command.

If you do not specify a value for the PGMDTA parameter, the exit program runs under the user profile QUSER and authority issues might cause run-time errors. The errors should not prevent the deletion of the journal receiver. If the value is changed, use a suitable user profile that has the requisite authority to PowerExchange objects.

PowerExchange also creates a physical file in the CONDLIB library called PWXJRNLCCKP, which is used to record the lock records for journal receivers. PowerExchange associates a journal called PWXJRN and journal receivers called PWXJRNnnnn with the PWXJRNLCCKP file to journal changes made to that file.

If you use the PowerExchange journal exit, PowerExchange performs the following processing:

- When PowerExchange starts reading a journal receiver, it writes a lock record for that journal receiver in the PWXJRNLCCKP file.
- When PowerExchange switches to the next journal receiver on the chain, PowerExchange writes a lock record for the next journal receiver and deletes the lock record for the previous journal receiver from the lock file.
- PowerExchange writes lock records for each change data extraction that is processing a journal receiver. As a result, there might be multiple records for a single journal receiver in the PWXJRNLCCKP file.

Each lock record in the PWXJRNLCCKP file includes information about the journal receiver that is locked, such as the library and journal receiver file names, the job name and number of the task that is processing the journal receiver, and the timestamp for when the journal receiver was locked.

When one of the following events occur, PowerExchange releases the lock on the journal receiver by deleting the lock record for the journal receiver from the PWXJRNLCCKP file:

- PowerExchange reads journal entries that indicate a change of journal receiver has occurred.
- The PowerExchange Condense job or the real-time extraction mode operation that is reading the change data from the journal receiver ends.
- The PowerExchange Listener or PowerExchange Condense job that obtained the locks ends normally.

If the PowerExchange job that obtains the lock on the journal receiver abnormally terminates, you must manually remove the records for that job from the PWXJRNLCCKP file.

Note: PowerExchange only locks journal receivers that it is currently processing. Because i5/OS prevents deletion of journal receivers that are attached or more current in the receiver chain from those being processed, PowerExchange does not need to lock more current journal receivers in the journal receiver chain.

Using Remote Journals with PowerExchange

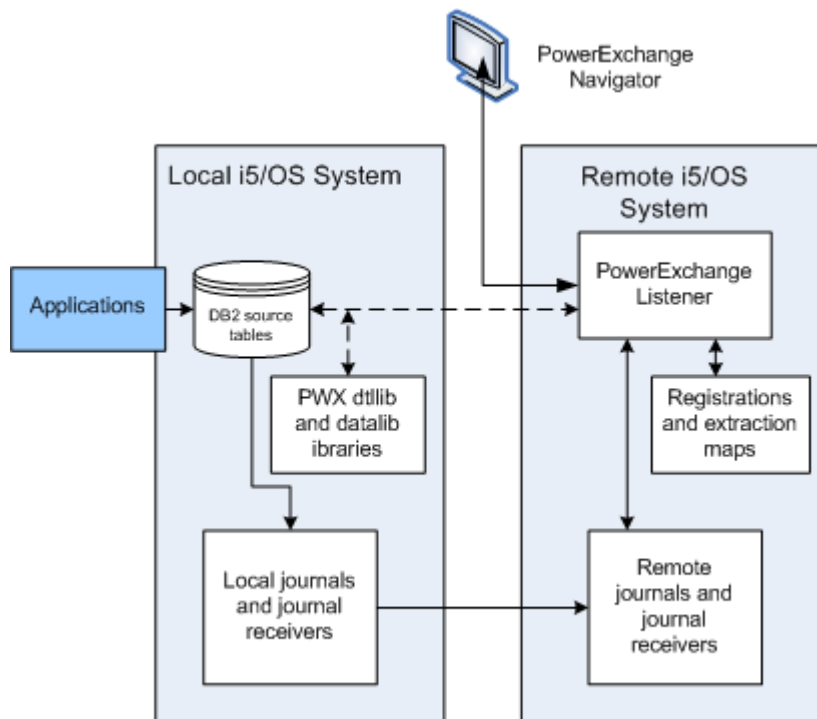
PowerExchange can capture change data from remote journal receivers that reside on a remote system or in an independent disk pool. The remote journal receivers get change entries from the local journal receivers on the local system where the DB2 source tables reside.

You can run PowerExchange on the remote system where the remote journal receivers reside and not on the local system where the source tables reside. PowerExchange can use distributed data management (DDM) to access metadata for the DB2 source tables on the local system for registration and database row tests, provided that you specify values other than the default values for the RMTRDBDIRE and RMTSYSNAME parameters in the CRTPWXENV command or DBMOVER configuration member.

If you have copies of the DB2 source tables on the remote system, PowerExchange can access metadata for the tables without using DDM. In this case, you must accept the default values for the RMTRDBDIRE and RMTSYSNAME parameters. Also, PowerExchange does not create *dtllib* and *datalib* libraries on the local system during installation.

You might want to use remote journals for change capture if your local production system has space constraints or high overhead. By running PowerExchange CDC processing on a remote system with higher availability, you can avoid degrading performance of the production system.

The following figure shows a simple PowerExchange CDC configuration in which PowerExchange reads change entries from remote journal receivers and the source tables reside only on the local system:



In this configuration, the local i5/OS system contains the following items:

- The journaled DB2 source tables that applications update.

- The i5/OS local journals and journal receivers that contain change entries for the DB2 source tables.
- The PowerExchange *dtllib* and *datalib* libraries on the local system. These libraries are created when you run the CRTPWXENV command on the remote system during PowerExchange installation. The local *dtllib* library contains a SQL package that enables the PowerExchange Listener on the remote system to access the local DB2 database. The local *datalib* library contains logical files, or SQL views, that enable PowerExchange to access metadata for DB2 tables of interest.

The remote system contains the following items:

- The i5/OS remote journals and journal receivers
- The PowerExchange for DB2 for i5/OS installation including:
 - *dtllib*, *datalib*, *condlib*, and *cpplib* libraries
 - PowerExchange Listener
 - DBMOVER configuration member
 - PowerExchange Condense, if used, and its condense files, checkpoint files, CDCT file, and CFGCOND(CAPTPARM) configuration member
- PowerExchange capture registrations and extraction maps for the DB2 source tables

When an application updates a journaled DB2 source table on the local system, i5/OS writes the entries for the changes to the local journals. Each local journal is associated with at least one remote journal. i5/OS replicates journal entries from the local journals to the associated remote journals. PowerExchange gets the journal entries from the remote journal receivers for the source tables and columns of interest, as identified by the capture registrations. If you use PowerExchange Condense, PowerExchange Condense reads the changes from the remote journal receivers and writes the changes to condense files on the remote system.

When you register DB2 source tables or perform database row tests from the PowerExchange Navigator, the PowerExchange Listener on the remote system connects to the local system to read metadata for the source tables. The PowerExchange Listener uses the SQL package and views in the *dtllib* and *datalib* libraries on the local system to access the metadata. PowerExchange stores the capture registrations and corresponding extraction maps in the *datalib* and *cpplib* libraries, respectively, on the remote system.

Note: If you have copies of the source tables on the remote system and accept the default values for the RMTRDBDIRE and RMTSYSNAME parameters, the CRTPWXENV command does not create the *dtllib* and *datalib* libraries during installation.

Considerations for Using Remote Journals

The following considerations pertain to using remote journals and journal receivers:

- When PowerExchange runs on the same system as the DB2 source tables, PowerExchange can validate that a registered source table is journaled to a specific journal. However, when PowerExchange runs on a remote system, PowerExchange cannot validate the association between a source table and a remote journal. In this case, you should verify that a source table is correctly associated with a remote journal. On the local system where the table resides, enter the WRKJRNA command and select the option to view remote journal information.
- When you create capture registrations for the DB2 source tables or perform database row tests in the PowerExchange Navigator, select the node name for the remote system in the **Location** field.
- You can install and run the PowerExchange Listener on the local system as well as on the remote system to access metadata for the DB2 source tables when you create capture registrations or perform database row tests. However, running the PowerExchange Listener on the local system is not necessary because the PowerExchange Listener on the remote system can access metadata for the source tables on the local system.

- PowerExchange supports i5/OS library redirection so that you can use names for the remote journals and journal libraries that are different from those of the associated local journals and journal receivers, if you prefer.
- The remote journals can be in a cascade configuration in which one remote journal writes to another remote journal, or in a broadcast configuration in which a local journal writes to multiple remote journals.

Customizing the DBMOVER Configuration Member for Remote Journal Use

You must define some DBMOVER statements to enable PowerExchange to capture change data from remote journal receivers and to read metadata from the DB2 source tables.

On the remote i5/OS system that contains the remote journal receivers, configure the following DBMOVER statements in addition to those that are normally required for CDC:

- In the JOURNAL parameter in the AS4J CAPI_CONNECTION statement, enter the *JRN=library/journal_name* subparameter to identify the remote journal to override. The journal contains journal entries for the registered source tables.
- Optional. Edit the RMTSYSNAME and RMTRDBDIRE statements to override the host name and database name that the CRTPWENV command sets during installation:
 - In the RMTSYSNAME statement, enter the host name of the system where the Db2 source tables reside.
 - In the RMTRDBDIRE statement, enter the name of the Db2 database that contain the Db2 source tables.

On the Windows system where the PowerExchange Navigator runs, define a NODE statement in the dbmover.cfg file that points to the remote system where the remote journal receivers reside. This statement enables the PowerExchange client on Windows to connect to the PowerExchange Listener on the remote system. You do not need to specify an AS4J CAPI_CONNECTION statement in the dbmover.cfg file on the Windows system.

Configuring PowerExchange for DB2 Change Data Capture

The specific tasks that you perform to configure PowerExchange for CDC depend on whether you want to use PowerExchange Condense.

PowerExchange Event Message Queue

You can specify an event message queue to which PowerExchange writes certain i5/OS-specific messages in the AS400EVENTMSGQ statement in the DBMOVER configuration member.

To write messages to this queue, you must specify one or more of the following parameters in the AS4J CAPI_CONNECTION statement:

- AS4JRNEXIT=Y
- If ALWLIBRFSH=Y
- If ALWCLRPFM=Y

If you specify AS4JRNEXT=Y, PowerExchange writes message DTL3001 to the queue after an extraction finishes processing of a journal receiver and before it starts reading the first journal entry in the next journal receiver on the chain:

```
DTL3001 The application extraction_application finished processing entries from receiver
journal_receiver in library library.
```

PowerExchange writes the DTL3001 message for each PowerExchange Condense job that runs in continuous extraction mode or each change data extraction that runs in real-time extraction mode. Therefore, you might see multiple DTL3001 messages on the event message queue for the same journal receiver. Because PowerExchange writes DTL3001 messages only for extractions that successfully process a journal receiver, you can use these messages to determine when it is safe to delete a journal receiver. However, before you delete a journal receiver, consider the following guidelines:

- Do not delete a journal receiver until all real-time change data extractions finish extracting change data from the receiver.
- PowerExchange might need previous journal receivers to restart a change data extraction. Do not delete a journal receiver until the restart tokens for all of the CDC sessions or PowerExchange Condense jobs that process the receiver no longer reference the receiver.
- PowerExchange does not remove messages from the event message queue. After you delete a journal receiver, remove the messages about that receiver from the message queue.

If you specify ALWLIBRFSH=Y, PowerExchange writes message DTL3003 for an add library event or message DTL3004 for a delete library event to the queue after you issue the SNDPWXJRNE command with an **Event Identifier** value:

```
DTL3003 PowerExchange has processed add event event_identifier.
DTL3004 PowerExchange has processed delete event event_identifier.
```

This SNDPWXJRNE command triggers refresh processing of the CDC library interest list for an add library event (*ADDLIB) or delete library event (*DLTLIB), with minimal interruption to real-time extraction processing and PowerCenter workflows. Use this command if you have multiple instances of a DB2 for i5/OS source table that are journaled to the same user journal and you routinely add or delete libraries that contain a table instance.

If you specify ALWCLRPFM=Y, PowerExchange writes message DTL3002 to the queue if PowerExchange encounters a journal entry for a CLRPFM command:

```
DTL3002 The application application ignored operation operation on file file_name in
library library to continue PowerExchange CDC processing.
```

PowerExchange does not replicate CLRPFM operations on i5/OS files.

RELATED TOPICS:

- [“PowerExchange Condense Message Log Files” on page 40](#)
- [“PowerExchange Listener Message Log Files” on page 16](#)

Configuring PowerExchange CDC without PowerExchange Condense

If you do not plan to use PowerExchange Condense, use this procedure to configure PowerExchange CDC.

1. Configure the DBMOVER configuration file.

Include the following statements:

- CPX_DIR

- AS4J CAPI_CONNECTION
- UOWC CAPI_CONNECTION

Optionally, specify the AS400EVENTMSGQ and AS400USRJRNCODE statements.

For more information about these DBMOVER statements, see [“Summary of DBMOVER Statements for CDC” on page 18.](#)

2. In the PowerExchange Navigator, create a capture registration for each source table.

In the **Condense** list, select **Part** or **Full**, unless you have a specific reason not to do so. This practice prevents having to change the capture registrations later if you decide to use PowerExchange Condense. For more information about creating capture registrations, see the *PowerExchange Navigator User Guide*.

The PowerExchange Navigator generates a corresponding extraction map. Optionally, you can also create a data map if you want to perform field-level processing.

3. For each capture registration, set the **Status** field to **Active**.

Usually, this step is done after materializing the targets.

Next, configure extraction sessions to run in real-time extraction mode.

Configuring PowerExchange CDC with PowerExchange Condense

If you plan to use PowerExchange Condense, use this procedure to configure PowerExchange CDC.

1. Configure the DBMOVER configuration file.

Include the following statements:

- CPX_DIR
- AS4J CAPI_CONNECTION
- UOWC CAPI_CONNECTION

Optionally, specify the AS400EVENTMSGQ and AS400USRJRNCODE statements.

For more information about these DBMOVER statements, see [“Summary of DBMOVER Statements for CDC” on page 18.](#)

2. Configure the PowerExchange Condense CAPTPARM configuration member.

For parameter descriptions, see [“PowerExchange Condense Configuration Parameters” on page 42.](#)

3. In the PowerExchange Navigator, create a capture registration for each DB2 source table.

In the **Condense** list, select **Part** or **Full**. For more information about creating capture registrations, see the *PowerExchange Navigator User Guide*.

When you complete a registration, the PowerExchange Navigator generates a corresponding extraction map.

Tip: If capture registrations already exist for the source tables, delete these registrations and their corresponding extraction maps. Then register the tables again.

4. For each capture registration, set the **Status** field to **Active**.

5. Start PowerExchange Condense.

Next, configure the change data extractions to run in batch extraction mode.

RELATED TOPICS:

- [“Starting PowerExchange Condense” on page 48](#)

Managing DB2 Change Data Capture

Occasionally, you might need to stop change data capture for a source table or update the table metadata.

Stopping DB2 CDC

You might need to stop change data capture for DB2 sources to perform troubleshooting or routine maintenance tasks or to remove a source permanently from CDC processing.

To stop change data capture, use one of the following methods:

- To stop change capture for a registered source table, open the capture registration for the table, and change the **Status** option from **Active** to **History**.

Warning: After you set the status to **History**, you cannot activate the registration again. This status change permanently stops change data capture based on the capture registration. If you need to capture change data for the table again, you must create another capture registration.

- If you use PowerExchange Condense, use the SHUTCOND or SHUTDOWN command to stop the PowerExchange Condense job. This action stops the logging of changes in the condense files for all sources of interest. When you restart PowerExchange Condense, it resumes from the current end-of-log (EOL) or from where it left off in the change stream, based on whether you cold start or warm start it.
- For CDC sessions that use real-time extraction mode, stop the PowerCenter workflow to stop the replication of changes for all sources of interest. When you restart the workflow, it resumes from the current EOL or from where it left off in the change stream based on whether you cold start or warm start it.

Note: If you do not stop updates to the source prior to stopping CDC, you might lose change data. However, if you stop and then warm start PowerExchange Condense or a PowerCenter workflow, PowerExchange can resume processing changes from where it left off and no changes are lost.

RELATED TOPICS:

- [“Stopping PowerExchange Condense” on page 50](#)

Changing a DB2 Table Definition

Occasionally, you might need to change the definition of a DB2 source table that is registered for change data capture.

If your changes affect columns from which change data is captured, use this procedure to enable PowerExchange to switch to the updated table definition, while preserving access to previously captured data.

Perform this procedure whenever you add, alter, or drop columns for which change data is captured. You do not need to perform this procedure if you are selectively capturing change data for a subset of columns and none of the selected columns are affected by the metadata changes.

Tip: If you no longer need to capture change data from a column in a table, you can remove the column from the extraction map without changing the capture registration. Change data for that column is still captured but is not extracted.

To change a DB2 source table definition:

1. Stop all transactions, applications, and other activity that update the table.
2. Verify that any change data that was captured under the previous table definition has completed extraction processing. Then stop all workflows that extract change data for the table.
3. In the PowerExchange Navigator, open the original capture registration and set its status to **History**.
4. Use DDL to make the table changes.
5. In the PowerExchange Navigator, create a capture registration that reflects the metadata changes and set its status to **Active**.

PowerExchange uses the newly activated capture registration for change data capture.

6. If necessary, change the target table definition to reflect the source table metadata changes.
7. In PowerCenter Designer, import the altered source and target tables. Edit the mapping if necessary.
8. Rematerialize the target tables, if necessary. After materialization completes, create restart tokens.
9. Enable transactions, applications, and other activity to update the table again.
10. Restart extraction processing.

Determining the Journal Receiver Needed for Restart Processing

You can determine the journal receiver that is required for restarting a CDC session by examining the restart token values. The restart token portion of the token pair includes the journal entry sequence number.

To restart CDC sessions at the proper location in the journal receiver chain, PowerExchange records information about the last journal entry that the session successfully processed in restart tokens.

PowerExchange and PWXPC display the restart token values in messages such as PWX-04565, PWX-09959, PWXPC_10081, PWXPC_10082, PWXPC_12102, PWXPC_12103, and PWXPC_12128. Also, PWXPC writes the starting and ending restart tokens for a CDC session to files in the restart token file directory that is specified for the PWX CDC application connection.

The restart tokens consist of a pair of token values, each being an alphanumeric string:

- The first token value is called the *sequence token*. The sequence token has a length of either 72 or 80 characters, depending on whether it is displayed by PowerExchange or PWXPC. This token indicates the location of the last change record that the CDC session successfully processed.
- The second token is called the *restart token*. The restart token has a length of 52 characters. This token indicates the location in the change stream from which PowerExchange starts reading change data when you restart a CDC session.

When you restart a CDC session, PowerExchange uses the journal entry sequence number that is recorded in the restart token value to start reading journal entries from the journal receivers again. After PowerExchange reaches the journal entry sequence number that is recorded in the sequence token, PowerExchange begins passing change records to the CDC session.

In the restart token, the journal entry sequence number is a hexadecimal value in positions 31 through 50. For example, the termination restart token file for a CDC session contains the following information:

```
<!-- Restart Tokens for the Table: as4rr001_AS4_RRTB_SRC_001 -->
d3kjm850as.as4rr001_AS4_RRTB_SRC_001=513172FF043AA000000000000000000838F2513172FF043AA0000000
00000000000838F200000000
d3kjm850as.as4rr001_AS4_RRTB_SRC_001=D2D1D4F8F5F0C1E2513172FF043AA0000000000000000838F0
<!-- Restart Tokens for the Table: as4rr002_AS4_RRTB_SRC_002 -->
d3kjm850as.as4rr002_AS4_RRTB_SRC_002=513172FF043AA000000000000000000838F2513172FF043AA0000000
00000000000838F200000000
d3kjm850as.as4rr002_AS4_RRTB_SRC_002=D2D1D4F8F5F0C1E2513172FF043AA0000000000000000838F0
<!-- Restart Tokens for the Table: as4rr003_AS4_RRTB_SRC_003 -->
d3kjm850as.as4rr003_AS4_RRTB_SRC_003=513172FF043AA000000000000000000838F2513172FF043AA0000000
```

```
00000000000838F200000000
d3kjm850as.as4rr003_AS4_RRTB_SRC_003=D2D1D4F8F5F0C1E2513172FF043AA000000000000000000000838F0
```

In this file, the restart token for each source has the following value:

```
Column Position
----0----1----1----2----2----3----3----4----4----5----5
----5----0----5----0----5----0----5----0----5----0----5
D2D1D4F8F5F0C1E2513172FF043AA000000000000000000000838F0
```

In this example restart token, the journal entry sequence number is 000000000000000838. After converting 838 from hexadecimal to decimal, the journal entry sequence number that PowerExchange uses to start reading change data is 2104.

Important: Do not try to create a restart token from a journal entry sequence number. If you do so, the restart token will not be valid.

To determine the journal receiver that PowerExchange will use for restarting the CDC session, use the i5/OS DSPJRNRCVA command. This command displays information for journal receivers. The command output includes the beginning and ending journal entry sequence numbers for a journal receiver, in the **First sequence number** and **Last sequence number** fields. To find the journal receiver that contains the journal entry sequence value from the restart token, you might need to display many of the journal receivers from the journal receiver chain.

Refreshing the CDC Interest List After Adding or Deleting Libraries

If you have multiple instances of a DB2 for i5/OS source table that are journaled to the same user journal and you routinely add or delete libraries that contain a table instance, you can refresh the CDC library interest list with minimal interruption to real-time extraction processing and PowerCenter workflows.

The source table instances must be journaled to the same local user journal, and this journal must be one that PowerExchange is currently using.

This feature is intended for users who add or delete a large number of libraries on a routine basis. Without the refresh feature, when you delete a library from which data is captured, PowerExchange real-time extraction processing stops, causing the PowerCenter workflow to end abnormally. When you add a library, the CDC session does not start processing change data for the new library until you stop and restart the PowerExchange Listener and PowerCenter workflow. Use the refresh feature to avoid these downtime conditions and associated delays.

To enable refresh processing, you must configure the following PowerCenter and PowerExchange parameters:

- In PowerCenter, enter the asterisk (*) wildcard for the library name in the **Library/File Override** session property for the real-time CDC session. This library name override enables the session to read changes for all of the table (file) instances across multiple libraries, including any library you add. You can specify the table just once in the mapping.
- In PowerExchange, set the ALWLIBRFSH parameter to Y in the AS4J CAPI_CONNECTION statement of the DBMOVER member. This setting enables you to issue refresh requests with the SNDPWXJRNE command.

Other recommended DBMOVER parameters are the AS400USRJRNCODE statement, the AS400EVENTMSGQ statement, and the LIBASUSER parameter in the AS4J CAPI_CONNECTION statement.

After you add or delete libraries, use the PowerExchange SNDPWXJRNE command to issue a refresh request from the i5/OS console. PowerExchange writes messages to the PowerExchange message log file in the *data1ib* library or optionally to the AS400EVENTMSGQ event message queue for audit and monitoring purposes.

Considerations for Library Refresh Requests

Consider the following requirements and restrictions that pertain to refreshing the CDC library interest list for added or deleted DB2 for i5/OS libraries:

- The library refresh feature does not support remote journals. All of the libraries in the request must be journaled to the same local user journal.
- You must use PowerExchange Client for PowerCenter (PWXPC) connections to PowerCenter. The refresh feature does not support ODBC connections.
- The library refresh feature does not support PowerExchange Condense or remote logging of data to a PowerExchange Logger for Linux, UNIX, and Windows instance.
- A refresh request for added or deleted libraries applies to all current PowerExchange capture processes that use the user journal with the entry for the add library or delete library operation.
- Usually, an *ADDLIB or *DLTLIB refresh request does not significantly degrade capture processing and does not slow or stop PowerCenter workflow processing. However, if PowerExchange is behind in reading the journal, the refresh request might not be processed in a timely manner and could disrupt other planned activity.
- PowerExchange validates the library names in the refresh request to ensure that they exist and are not repeated in the same request. However, PowerExchange does not validate the journaled table (file) instances in these libraries.
- Ensure that the journal entry type codes that PowerExchange uses to mark an add library (*ADDLIB) request or a delete library (*DLTLIB) request are not used by any other application. If the default values I1 and I2 are used by another application, add the AS400USRJRNCODE statement in the DBMOVER member to define unique journal entry type codes for refresh operations.
- Refresh requests do not advance the restart point for extraction processing. If a data extraction stops between the time the refresh request is issued and the arrival of the next UOW of interest, or if the restart tokens are advanced based on the RSTRADV parameter in the UOWC CAPI_CONNECTION statement, the refresh transaction occurs beyond the last restart point and will be processed again when the extraction restarts. This behavior can be a problem on test systems in particular.
- If you issue a delete library (*DLTLIB) request while a CDC real-time session is running, PowerExchange refreshes the CDC interest list and blocks the deleted libraries from capture processing for the current CDC session run only. The next time you run the CDC session, the library changes that were in the refresh request are no longer honored. The new CDC session processes all of the libraries in its current CDC interest list.
- For an add library (*ADDLIB) request, PowerExchange writes audit message PWX-06815 for the add library request twice. The first time, the PWX-06815 message includes an asterisk (*) next to the library names to alert you that these libraries are included in a new *ADDLIB refresh request and will be added to CDC processing. The second time, the PWX-06815 message reports only the requested library names.
- If you delete multiple i5/OS libraries that contain DB2 for i5/OS table instances that have the same structure and use the same journal and then add back these same libraries, the SNDPWXJRNE request for the add-library operation does not update the CDC interest list and the warning message PWX-06817 is issued.

Configuring the DBMOVER Member for Library Refresh Processing

Configure the DBMOVER statements and parameters that pertain to DB2 for i5/OS library refresh processing.

You must configure the following parameter:

- **ALWLIBRFSH=Y in the AS4J CAPI_CONNECTION statement.** Enables PowerExchange to refresh the library interest list when you issue the SNDPWXJRNE command for an add library (*ADDLIB) or delete library (*DLTLIB) operation.

Informatica recommends that you also configure the following statements and parameters:

- **LIBASUSER=Y parameter in the AS4J CAPI_CONNECTION statement.** Populates the generated DTL__CAPXUSER column with the library name and file name where a change was made. This setting can help you determine the library from which a change was captured.
- **AS400EVENTMSGQ statement.** Specifies an event message queue on the i5/OS system to which PowerExchange sends DTL3003 and DTL3004 messages for an *ADDLIB or *DLTLIB event. You can use these messages to verify that the refresh request was processed. You must also specify the **Event Identifier** value in the SNDPWXJRNE command for the refresh request.
- **AS400USRJRNCODE statement.** Defines two-character journal entry type codes that identify *ADDLIB and *DLTLIB operations in the user journal. Use this statement to override the default codes that are used for refreshing the CDC library interest list after adding or deleting libraries that are journaled to the same journal and that contain instances of the same source table.

For more information about these statements and parameters, see [“Configuring the PowerExchange Listener” on page 17](#) and the *PowerExchange Reference Manual*.

Issuing a Library Refresh Request from the i5/OS Console

After you add or delete libraries, issue a SNDPWXJRNE refresh request to update the CDC library interest list.

Check with your i5/OS system security administrator to verify that you have the proper authority to issue the SNDPWXJRNE command. The required security level is the same as for other PowerExchange-supplied commands for i5/OS. If you do not have the proper authority, the operating system will issue an error message.

Also verify that you configured the required DBMOVER parameters and specified the library name override of the asterisk (*) wildcard in the PowerCenter **Library/File Override** session property for the CDC sessions.

1. In the i5/OS console, enter SNDPWXJRNE at the command line.

The **Send PWX Journal Entry (SNDPWXJRNE)** panel appears.

2. Complete the following fields:

Journal and Library Name

Required. The qualified name of the user journal to which the registered table (file) is journaled.

- For **Journal**, enter the name of the user journal to contain the journal entry for the *ADDLIB or *DLTLIB request.
- For **Library Name**, specify the library name where the journal resides, or enter *LIBL to search all libraries in the library list until the first match is found.

Request

Required. One of the following options:

- ***ADDLIB** for an add library request. Refreshes the CDC library interest list for current capture processes that read journal entries for the registered table (file) from the same journal. The refresh is triggered when the capture processes encounter the journal entry for the refresh operation.
- ***DLTLIB** for a delete library request. Removes the names of the deleted libraries from the CDC library interest list for current capture processes.

Request Library Name

Required. The names of the libraries that you added or deleted. You can enter up to 80 library names.

For an *ADDLIB request, the CDC library interest list will contain only the libraries that have a journaled table instance when the refresh request is issued.

Event Identifier

Optional. An event identifier such as ADDLIB #1 that helps identify when an interest list refresh request has been processed for an add library or delete library event. If you define the AS400EVENTMSGQ statement in the DBMOVER configuration member, messages DTL3003 and DTL3004, which contain this identifier, are written to the event message queue.

3. Press Enter.

PowerExchange performs the following processing:

- Validates that 1) the journal exists in the specified library, 2) the request type is *ADDLIB or *DLTLIB, and 3) the libraries specified in the **Request Library Name** field exist and are not repeated in the same request. Also, if you specified an **Event Identifier** value, the command verifies that the AS400EVENTMSGQ statement is defined in the DBMOVER configuration member.
- If you are *not* using the default journal entry type codes for the refresh request, retrieves the codes that you defined in the AS400USRJRNCODE statement in the DBMOVER member.
- Sends the journal entry to the journal for processing by the CAPI.
- Sends audit messages for the refresh request to the PowerExchange message log. Also, if you specified the AS400EVENTMSGQ statement, sends an event message to the event message queue to mark the add library or delete library event.

Monitoring Refresh Processing of the CDC Library Interest List

Use key messages to determine the status of refreshing the CDC library interest list for added or deleted libraries.

- The following messages, which are written to the AS400EVENTMSGQ event message queue, indicate that a SNDPWXJRNE refresh request was successfully processed:

```
DTL3003I PowerExchange has processed add event event_identifier
DTL3004I PowerExchange has processed delete event event_identifier
```

Note: For these messages to be issued, you must have specified an **Event Identifier** value for the SNDPWXJRNE command.

- The following messages, which are written to the PowerExchange message log file, provide the standard message output for a refresh request:

```
PWX-06813 Rebuilding CDC library interest list after processing user journal entry
with type code code for a SNDPWXJRNE request at journal time timestamp with sequence
sequence_number.
PWX-06814 Rebuilding CDC library interest list to action libraries
```

```

PWX-06815 library1_name use_count library2_name use_count library3_name use_count
library4_name use_count.
PWX-06815 library5_name use_count library6_name use_count library7_name use_count
library8_name use_count.
PWX-06815 ...
PWX-06781 Override for Table (library/file) collection using registration
registration_name (DB2 for i5/OS CDC)
PWX-06118 Registration loaded: DBName: database_name RegName: name Creator: creator
Table: table_name
PWX-06816 Rebuild of CDC library interest list is complete after processing user
journal entry with type code code for file_count sources

```

- **PWX-06813.** Indicates the start of rebuilding the CDC library interest list for a specific SNDPWXJRNE refresh request.
- **PWX-06814.** Indicates whether the CDC library interest list is being rebuilt for an add library or a delete library operation.
- **PWX-06815.** Lists the added or deleted libraries for which you are refreshing the CDC library interest list. Each message lists up to four libraries. If more than four libraries were added or deleted, this message is issued for each set of four libraries in the SNDPWXJRNE refresh request, as needed.
- **PWX-06781.** Indicates that a library name override is in effect, as required for refresh processing. The override must be an asterisk (*) wildcard and specified in the PowerCenter **Library/File Override** session property.
- **PWX-06816.** Indicates the completion of rebuilding the CDC library interest list.

Note: The PowerExchange message log file is either the DTLLOG member of the the LOG file in the *datalib* library if alternate logging is not enabled, or the JOBnnnnnn file in the *datalib* library, where *nnnnnn* is the i5/OS job number for the PowerExchange Listener task, if alternate logging is enabled.

Troubleshooting a Refresh Request

If you encounter a problem while using Data Replication, review the following list of previously reported issues and their solutions before contacting Informatica Global Customer Support.

An incorrect library name was specified in a SNDPWXJRNE *DLTLIB request. The library exists but is not a deleted library.

The *DLTLIB request blocks capture processing for the incorrect library. In this case, you must complete the following steps:

1. End the CDC sessions that process the table in the incorrect library in the SNDPWXJRNE request.
2. Rematerialize the targets.
3. Restart the CDC sessions.
4. Ensure that the correct library exists and its table is journaled.
5. Submit the SNDPWXJRNE request again with the correct library name.

Error message PWX-06812 was issued.

```

PWX-06812 Extraction encountered user journal entry with type code code during
processing with journal time timestamp and sequence sequence_number but entry is not in
expected format.

```

This error might be caused by using journal entry type codes for SNDPWXJRNE requests that are not unique. The entry type codes might be in use by another application. Use the AS400USRJRNCODE statement in the DBMOVER member to define other journal entry type codes for SNDPWXJRNE add library and delete library requests.

Refreshing the PowerExchange Environment After an i5/OS Upgrade

If you upgrade the i5/OS operating system on the DB2 system where PowerExchange is installed or on a remote DB2 system from which PowerExchange accesses data, you must run a few commands to refresh the PowerExchange environment for any changes to the system metadata. Run the commands after you complete the i5/OS upgrade and before you re-submit the PowerExchange Listener job or PowerExchange Condense job and resume bulk data movement or CDC processing.

You must run the commands under one of the following types of user profiles:

- The IBM-supplied user profile of QSECOFR
- A user profile defined with the USRCLS of *SECOFR and SPCAUT of *USRCLS
- A user profile defined with at least the following SPCAUT values: *SECADM, *ALLOBJ and *JOBCTL

1. Issue the following ADDLIB command:

```
ADDLIB LIB(dtllib) POSITION(*FIRST)
```

The *dtllib* variable is the PowerExchange software library.

2. To refresh the PowerExchange environment, issue one of the following CRTDTLENV commands with the same parameters that you used for the CRTPWXENV command during the last full PowerExchange installation:

- If the DB2 for i5/OS source database is on the same i5/OS server as the PowerExchange installation, use:

```
CRTDTLENV DTLLIB(dtllib) DATALIB(datalib) RMRDBDIRE(*LOCAL) OSLEVEL(*LOCAL)
```

- If the DB2 for i5/OS source database is on a server other than the one where PowerExchange is installed, use:

```
CRTDTLENV DTLLIB(dtllib) DATALIB(datalib) RMRDBDIRE(database_name)  
RMTSYSNAME(host_name) RMTOSLEVEL(os_level)
```

For descriptions of the parameters, see Chapter 4 in the *PowerExchange Installation and Upgrade Guide*.

3. If you changed the ownership of objects such the *dtllib*, *datalib*, *condlib*, and *cpxlib* libraries during the last full PowerExchange installation, run the commands for changing object ownership again.

For more information, see Chapter 4 in the *PowerExchange Installation and Upgrade Guide*.

Generating SQL Statements to Re-create a Source or Target for Troubleshooting

PowerExchange provides a command that generates SQL statements for re-creating SQL source or target objects that are used in a PowerExchange environment. The command is intended to produce SQL statements that Informatica Global Customer Support can use for troubleshooting bulk data movement or CDC problems.

To generate the SQL statements for an i5/OS SQL object, enter the RTVSQLSTMT command from the i5/OS system where PowerExchange is installed. The PowerExchange *dtllib* library must be in the current library list for the i5/OS job. The RTVSQLSTMT command prompts you for a series of parameters that control what SQL statements are generated. The RTVSQLSTMT command validates your entries to reduce the risk of errors.

Important: At any point, you can display Help information in the i5/OS Console to see field descriptions, command examples, and the authorities that are required to run the command.

The command can generate DDL statements for re-creating many types of objects, including aliases, constraints, indexes, procedures, schemas (libraries or collections), tables, triggers, and views. The command can also generate GRANT statements that provide the authority required to use the objects. Options are available to control the types of SQL statements to generate. For example, you can control whether to generate DROP statements that precede the CREATE statements for the selected object types. Also, the command can generate SQL statements even for some objects that were not created with SQL. For example, if you used data description specifications (DDS) to define physical and logical database files, the command can generate equivalent SQL CREATE statements to re-create those files.

The following sample command shows all of the required and optional parameters:

```
RTVSQLSTMT SQLOBJNAM(SQL_OBJECT_NAME) SQLOBJLIB(OBJECT_LIB) SQLTYPE(*TYPE)
SRCFILE(SOURCE_LIB/QSQLSRC) SRCMBR(SQLOBJNAM) RPLSRCMBR(*NO) CRTDROP(*NO) CRTCOMMENT(*NO)
CRTHEADER(*NO) CRTRIGGER(*NO) CRTCONSTR(*NO) CRTRENAME(*NO) CRTGRANT(*NO) GENCCSID(*NO)
CRTORRPL(*NO) CRTBFSTM(*NO) ACTROWCOL(*NO) CRTMSKPRM(*NO) CRTQUALSTM(*NO) CRTADDINDX(*NO)
CRTWINDX(*NO)
```

Note: The parameter names are highlighted in boldface. For descriptions of these parameters and all of their options, see the Help.

After you run the command, the command writes the SQL statements to the source file member that you specified. If a member that has the same source member name already exists, you can configure the RPLSRCMBR parameter to indicate whether to replace it.

Send the source file member that contains the SQL statements to Informatica Global Customer Support. Support personnel can then run the SQL statements on another machine to re-create the environment in which your error occurred.

For information about the validation messages that PowerExchange generates when you use the interface, see messages DTL5001 through DTL5008 in the *PowerExchange Message Reference Volume 3*.

CHAPTER 5

Remote Logging of Data

This chapter includes the following topics:

- [Remote Logging Overview, 73](#)
- [Requirements for Capture Registrations, 76](#)
- [Configuration Tasks for Remote Logging, 76](#)
- [Example of Remote Logging from a Db2 for i Data Source, 80](#)

Remote Logging Overview

You can log change data from any supported data source to PowerExchange Logger for Linux, UNIX, and Windows log files on another system.

You can log change data from data sources on an IBM i (i5/OS) or z/OS system to PowerExchange Logger log files on a Linux, UNIX, or Windows system. The PowerExchange Logger for Linux, UNIX, and Windows reads change data from PowerExchange on the source and logs the data to its log files. CDC sessions that run in continuous extraction mode can then extract the change data from the PowerExchange Logger log files instead of from the source.

The benefits of logging or relogging change data off of the source system depend on the source type and CDC environment. You can use remote logging to reduce resource consumption on the source system, move some resource-intensive CDC processing to the remote system, and reduce the network overhead of data transfer.

RELATED TOPICS:

- [“Requirements for Capture Registrations” on page 76](#)
- [“Configuration Tasks for Remote Logging” on page 76](#)
- [“Customizing the PowerExchange Logger Configuration File for Logging Data from a Remote Source” on page 77](#)
- [“Customizing the dbmover File on the System to Which Data Is Logged” on page 78](#)
- [“Customizing the dbmover Configuration File on the PowerCenter Integration Service System” on page 79](#)
- [“Example of Remote Logging from a Db2 for i Data Source” on page 80](#)

Remote Logging of Data from Sources on IBM i or z/OS Systems

You can use the PowerExchange Logger for Linux, UNIX, and Windows to extract change data for data sources on IBM i (i5/OS) and z/OS and relog that data to a less costly Linux, UNIX, or Windows system.

Multiple PowerCenter CDC sessions can then retrieve the change data from the local PowerExchange Logger for Linux, UNIX, and Windows log files.

For sources on IBM i and z/OS, the remote logging of data to a Linux, UNIX, or Windows system has the following benefits:

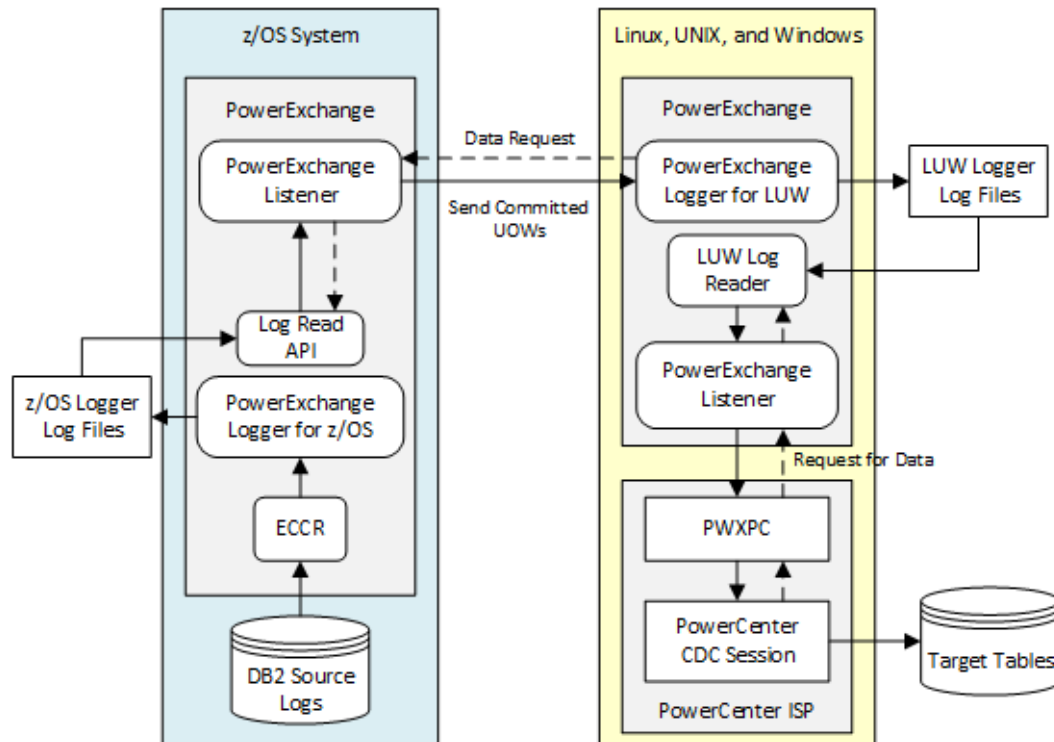
- Moves resource-intensive, column-level processing and UOW Cleanser processing off of the IBM i or z/OS system onto the Linux, UNIX, or Windows system where the PowerExchange Logger for Linux, UNIX, and Windows runs.
- Extracts change data from the Db2 for i (i5/OS) journal receivers or PowerExchange Logger for z/OS log files on z/OS in a single pass and transmits that data over the network to the PowerExchange Logger for Linux, UNIX, and Windows. The data is then available locally for PowerCenter CDC sessions to process. This single-pass processing reduces network traffic and avoids the overhead of multiple data extraction reads.
- Reduces costly CPU usage, disk space, and CDC processing time on the IBM i or z/OS source system.

To configure this remote logging scenario, you must specify the `CAPTURE_NODE` statement in the PowerExchange Logger for Linux, UNIX, and Windows configuration file, `pwxcl.cfg`, on the system where the Logger for Linux, UNIX, and Windows runs. The `CAPTURE_NODE` statement specifies the node name of the PowerExchange Listener that runs on the source system. When you create the registration group in the PowerExchange Navigator, enter the node name of the PowerExchange Listener that runs on the source system in the **Location** field. In PowerCenter, configure a PWX CDC Real Time connection for the PowerCenter CDC sessions that process change data from the source. In the connection attributes, set the **Location** attribute to the node name of the PowerExchange Listener that runs on the system where the PowerExchange Logger log files reside and set the **Mapping Location** attribute to the node name of the PowerExchange Listener that runs on the source system where the extraction maps reside.

Note: When the PowerExchange Logger for Linux, UNIX, and Windows runs on the PowerCenter Integration Service Platform (ISP) machine, you can use a Local connection rather than run a PowerExchange Listener on this machine. However, Informatica recommends that you run a PowerExchange Listener on the PowerCenter ISP machine so that you can issue commands to display information about the active PowerExchange Listener tasks, print PowerExchange Listener monitoring statistics, and stop the PowerExchange Listener task, if necessary.

For example, you can configure the PowerExchange Logger for Linux, UNIX, and Windows to extract Db2 for z/OS change data from PowerExchange Logger for z/OS logs files on a z/OS system and then relog that data

to PowerExchange Logger log files on the PowerCenter ISP machine. The following image shows this remote logging configuration:



In this scenario, set the PowerExchange Logger `CAPTURE_NODE` statement to point to the node name of the PowerExchange Listener on the z/OS system with the Db2 logs. Set the PowerCenter **Location** connection attribute to the node name of the PowerExchange Listener on the PowerCenter ISP machine where the PowerExchange Logger for Linux, UNIX, and Windows runs. Set the **Map Location** connection attribute to point to the node name of the PowerExchange Listener on the z/OS system.

The PowerExchange Logger for Linux, UNIX, and Windows sends a request for change data to the PowerExchange Listener on z/OS. This PowerExchange Listener contacts the Log Read API (LRAPI) to read captured change data from the PowerExchange Logger for z/OS log files. The PowerExchange Listener on z/OS transmits the change data in a single stream over the network to the PowerExchange Logger for Linux, UNIX, and Windows. The UOW Cleanser runs on the Powercenter ISP machine to cleanse the data, and then the PowerExchange Logger for Linux, UNIX, and Windows relogs the data in its local log files. When a Powercenter CDC session runs and requests change data for the tables of CDC interest, the PowerExchange Client for PowerCenter (PWXPC) requests change data from the PowerExchange Listener on the system with the Logger for LUW log files. The PowerExchange Listener contacts the local PowerExchange Logger Log Reader to read change data from the Logger log files. PWXPC makes the data available to the PowerCenter CDC session. Multiple PowerCenter CDC sessions can extract change data from the local PowerExchange Logger log files.

Note: When you run the PowerExchange Logger for Linux, UNIX, and Windows with a journal receiver override, you must specify the restart and sequence tokens in the `pwxccl.CFG` file for the override receivers.

Requirements for Capture Registrations

For the PowerExchange Logger for Linux, UNIX, and Windows to log change data from a remote source, verify that the capture registrations are compatible with the following requirements:

- To use the PowerExchange Logger for Linux, UNIX, and Windows, you must configure capture registrations for partial condense processing. In the PowerExchange Navigator, select **Part** in the **Condense** list for each registration. If you have remote i5/OS or z/OS data sources with capture registrations that specify **Full** for the **Condense** option, the PowerExchange Logger for Linux, UNIX, and Windows ignores these registrations. The PowerExchange Logger also ignores any capture registration that specify **None** for the **Condense** option.
- A PowerExchange Logger for Linux, UNIX, and Windows process must be able to read all of the capture registrations that it uses from a single CCT file on the source system.
- For the remote data sources, you cannot use capture registrations that were created from data maps that use any of the following features:
 - User access methods
 - User-defined fields that invoke programs by using the CALLPROG function
 - Record-level exits

Configuration Tasks for Remote Logging

To log change data to remote PowerExchange Logger for Linux, UNIX, and Windows log files and have PowerCenter CDC sessions extract data from those log files, complete the following configuration tasks:

1. Install PowerExchange on the system where the PowerExchange Logger log files will be located.
2. Customize the pwxcl.cfg configuration file on the system with the PowerExchange Logger log files.
3. Customize the dbmover configuration file on the system with the PowerExchange Logger log files. Copy the source-specific CAPI_CONNECTION statements from the source system to the dbmover file on the system with the PowerExchange Logger log files.

Note: Each PowerExchange Logger must have a unique pwxcl.cfg configuration file and a unique dbmover configuration file.

4. Configure a dbmover configuration file for the PowerExchange Listener on the system with the PowerExchange Logger log files.
You can use the same dbmover file for the PowerExchange Logger and the PowerExchange Listener. If you use different dbmover files, both files must specify the same CAPT_PATH value.
If the PowerExchange Logger log files are on the PowerCenter Integration Service machine, you can use a local connection instead of the PowerExchange Listener for change data extractions.
5. If you are not using a "local" connection, start the PowerExchange Listener on the system with the PowerExchange Logger log files.
6. Start the PowerExchange Logger on the system with the PowerExchange Logger log files.
7. Customize the dbmover configuration file on the PowerCenter Integration Service machine.
8. Configure capture registrations for PowerExchange Logger use.
9. Configure PWX CDC Real Time connection attributes for the CDC session to extract change data from the PowerExchange Logger log files.

Customizing the PowerExchange Logger Configuration File for Logging Data from a Remote Source

For the PowerExchange Logger for Linux, UNIX, and Windows to log data from a remote source, you must customize the PowerExchange Logger configuration file on the system with the PowerExchange Logger log files.

PowerExchange provides a sample configuration file, named `pwxcl`, in the PowerExchange installation directory. You can copy this file and customize the copy.

For a complete list of PowerExchange Logger configuration parameters, see the PowerExchange Logger for Linux, UNIX, and Windows chapter in the *PowerExchange CDC Guide for Linux, UNIX, and Windows*.

The following table describes the parameters that are used for remote logging:

Parameter	Description
CAPTURE_NODE	Required for remote logging. The node name that the PowerExchange Logger uses to retrieve capture registrations and change data from the source system. This name must be defined in a NODE statement in the dbmover configuration file on the system where the PowerExchange Logger runs. The PowerExchange Logger uses this node name to connect to the PowerExchange Listener on the source system. This name should correspond to the node name in the LISTENER statement on the source system.
CAPTURE_NODE_EPWD or CAPTURE_NODE_PWD	Optional. An encrypted password (EPWD) or clear text password (PWD) that is associated with the user ID specified in the CAPTURE_NODE_UID parameter. If you specify CAPTURE_NODE_UID, you must specify either CAPTURE_NODE_EPWD or CAPTURE_NODE_PWD. However, do not specify both CAPTURE_NODE_EPWD and CAPTURE_NODE_PWD.
CAPTURE_NODE_UID	A user ID that controls PowerExchange Logger read access to capture registrations and change data on the remote node that is specified in the CAPTURE_NODE parameter. Whether this parameter is required depends on the operating system of the remote node and the SECURITY setting in the DBMOVER configuration file for the PowerExchange Listener on that node. If CAPTURE_NODE specifies an i5/OS node that has a SECURITY setting of 0, do not specify this parameter. PowerExchange uses the user ID under which the PowerExchange Listener job runs to control access to capture registrations and change data. If CAPTURE_NODE specifies an i5/OS node that has a SECURITY setting of 1, you must enter a valid operating system user ID for this parameter. Otherwise, error message PWX-00231 is issued, indicating a signon failure. However, PowerExchange uses the user ID under which the PowerExchange Listener job runs to control access to capture registrations and change data. If CAPTURE_NODE specifies an i5/OS node that has a SECURITY setting of 2, you must enter a valid operating system user ID for this parameter. Otherwise, error message PWX-00231 is issued, indicating a signon failure. PowerExchange uses this user ID to control access to capture registrations and change data. If the specified user ID does not have the authority that is required to read capture registrations or change data, access fails.
CONDENSENAME	Optional. A name for the command-handling service for a PowerExchange Logger for Linux, UNIX, and Windows process to which <code>pwxcmd</code> commands are issued. This service name must match the service name in the associated SVCNODE statement in the dbmover configuration file.

Parameter	Description
CONN_OVR	Recommended. The name of the override CAPI_CONNECTION statement to use for the PowerExchange Logger. If you do not enter CONN_OVR, the PowerExchange Logger uses the default CAPI_CONNECTION in the dbmover configuration file, if specified. For DB2 for i5/OS data sources, enter the name of the UOW Cleanser (UOWC) CAPI_CONNECTION statement. Informatica recommends that you specify CONN_OVR because it is the only type of override that the PowerExchange Logger can use.
DB_TYPE	Required. The source database type. For DB2 for i5/OS data sources, you must enter AS4 .
DBID	Required. A source identifier, sometimes called the <i>instance</i> name, that is defined in capture registrations. When used with DB_TYPE, it defines selection criteria for capture registrations in the CCT file. This value must match the instance name that is displayed in the Resource Inspector of the PowerExchange Navigator for the registration group that contains the capture registrations. For DB2 for i5/OS, enter the Instance name that is displayed for the registration group. This name should match the INST parameter value in the AS4J CAPI_CONNECTION statement in the DBMOVER member of the CFG file configuration file.
EXT_CAPT_MASK	Required. An existing directory path and a unique prefix to be used for generating the PowerExchange Logger log files.

Customizing the dbmover File on the System to Which Data Is Logged

For the PowerExchange Logger for Linux, UNIX, and Windows to log data from a remote DB2 for i5/OS source, you must customize the dbmover configuration file on the system where the PowerExchange Logger log files will reside.

PowerExchange provides a sample dbmover file in the PowerExchange installation directory. You can copy this file and customize the copy. For a complete list of all dbmover configuration statements, see the *PowerExchange Reference Manual*.

The following table describes the dbmover statements that are used for remote logging:

Statement	Description
CAPT_PATH	Required. The path to the directory on the Linux, UNIX, or Windows system where the PowerExchange Logger CDCT file resides. The PowerExchange Logger stores information about its log files in the CDCT file. Each PowerExchange Logger that captures change data requires its own CDCT file.
CAPX CAPI_CONNECTION	Required. Parameters that the Consumer API (CAPI) uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files. The DFLTINST parameter value in this statement must match the DBID value in the PowerExchange Logger configuration file, pwxcl.
LOGPATH	Optional. A unique path and directory for PowerExchange message log files on the Linux, UNIX, or Windows system where the PowerExchange Logger logs data in its log files.

Statement	Description
NODE	Required. Information that PowerExchange uses to connect to the PowerExchange Listener on the source system from which change data is captured. This information includes a unique user-defined node name, the TCP/IP host name, and the port number. The node name that you enter in this statement must match the CAPTURE_NODE parameter value in the PowerExchange Logger configuration file.
Source-specific CAPI_CONNECTION	Required. A named set of parameters that the CAPI uses to connect to the change stream for a source type and control CDC processing. Copy the source-specific CAPI_CONNECTION statements from the DBMOVER configuration file on the source system. For DB2 for i5/OS sources, copy the AS4J and UOWC CAPI_CONNECTION statements.
SVCNODE	Optional. The TCP/IP port on which a command-handling service for a PowerExchange process, such as a PowerExchange Logger for Linux, UNIX, and Windows process, listens for pwxcmd commands.
TRACING	Optional. Enables PowerExchange alternative logging and specifies attributes for the alternative log files. PowerExchange uses the alternative log files instead of the default PowerExchange message log file to store messages.

Customizing the dbmover Configuration File on the PowerCenter Integration Service System

If you log change data in PowerExchange Logger for Linux, UNIX, and Windows log files on a system other than the source system, customize the dbmover configuration file on the PowerCenter Integration Service system, where the CDC sessions run, to identify the source and PowerExchange Logger nodes.

Add NODE statements for the PowerExchange Listeners that run on the following systems:

- The source system where the capture registrations reside and from which the PowerExchange Logger for Linux, UNIX, and Windows reads change data.
- The remote system where the PowerExchange Logger logs change data in its log files.

Configuring Capture Registrations for the PowerExchange Logger

For the PowerExchange Logger for Linux, UNIX, and Windows to extract change data from a remote source, the capture registrations for the source tables must specify **Part** for the **Condense** option.

Note: This requirement is not specific to remote logging. It also applies to PowerExchange Logger for Linux, UNIX, and Windows use on a source system.

If the capture registrations do not specify **Part** for the **Condense** option, you can edit the **Condense** setting. This change does not increment the registration version. You can continue to use the same registration and extraction map.

Tip: Do not add DTL_BI or DTL_CI columns to the extraction maps if you set the CAPT_IMAGE parameter to AI in the pwxcl.cfg configuration file. With the AI setting, the PowerExchange Logger stores after images only. Consequently, you cannot use before images of the data in extraction processing. Also, CDC sessions that reference any CI fields fail.

Configuring PowerCenter Connection Attributes for Extracting Data from the Log Files

For CDC sessions to extract change data from PowerExchange Logger for Linux, UNIX, and Windows log files on a system other than the source system, you must configure certain attributes on the PWX CDC Real Time connection.

The following table describes these connection attributes:

Connection Attribute	Value
Location	Enter the node name for the PowerExchange Listener that runs on the system where the PowerExchange Logger log files reside. If the log files are on the PowerCenter Integration Service machine, you can enter "local."
Map Location	Enter the node name for the location where the PowerExchange Listener on the source system stores the extraction maps. Usually, this node is the source system node.
Map Location User and Map Location Password	Enter a user ID and password that can access the extraction maps. If the PowerExchange Listener runs on a source system with PowerExchange security enabled, the user ID and password depends on the SECURITY statement settings in the DBMOVER configuration file. If the first parameter in the SECURITY statement is 2 and you are extracting z/OS data from the log files, enter a valid z/OS user ID and password in these fields. Also ensure that these z/OS user credentials have the following permissions: <ul style="list-style-type: none">- READ access to the PowerExchange data set that is defined in the DTLCAMAP DD statement of the PowerExchange Listener JCL- READ access to CAPX.CND.* resource profiles in the FACILITY class, which are managed by your z/OS security product
CAPI Connection Name Override	Enter the name of the CAPX CAPI_CONNECTION statement that is used by the PowerExchange Listener on the system where the PowerExchange Logger for Linux, UNIX, and Windows log files reside.

For more information about PWX CDC Real Time application connections, see *PowerExchange Interfaces for PowerCenter*.

Example of Remote Logging from a Db2 for i Data Source

In this example, you use a PowerExchange Logger for Linux, UNIX, and Windows instance on a UNIX system to capture change data from Db2 for i journal receivers on i5/OS. The system where the PowerExchange Logger runs is separate from the PowerCenter Integration Service system where you run CDC sessions.

You need the PowerExchange Logger to capture change data for registered tables from Db2 journals in the Db2 instance PROD2 and then log that data to its log files on the remote UNIX system. To do so, you must customize a PowerExchange Logger configuration file on the UNIX system and the dbmover configuration files on both the i5/OS and UNIX systems. Also, for the PowerCenter CDC sessions to extract change data from the PowerExchange Logger log files on UNIX, you must add NODE statements for the source and PowerExchange Logger systems to the dbmover configuration file on the Integration Service system and configure some PWXPC connection attributes.

First install PowerExchange on all three systems. You must run a PowerExchange Listener on the source system and on the PowerExchange Logger system. A PowerExchange Listener is not required on the PowerCenter Integration Service system.

1. On the i5/OS source system, ensure that the DBMOVER member in the *datalib*/CFG library includes the following CAPI_CONNECTION statements:

```
LISTENER=(i5OS1,TCPIP,2480)
/* UOW Cleanser CAPI Connection
CAPI_CONNECTION=(NAME=i5UOWC,TYPE=(UOWC,CAPINAME=i5_AS4J,RSTRADV=600,MEMCACHE=20480))
/* DB2 for i5/OS CAPI Connection
CAPI_CONNECTION=(NAME=i5_AS4J,TYPE=(AS4J,JOURNAL=(JRN=PRODDATA/PRODJRN),
INST=PROD2,EOF=N,STOPT=(CONT=5),LIBASUSER=Y))
```

Note: In the AS4J CAPI_CONNECTION statement, the INST parameter value must match the **Instance** name that is displayed for the registration group in the PowerExchange Navigator.

2. On the UNIX system with the PowerExchange Logger log files, ensure that the dbmover configuration file includes the following statements:

```
/*
/* dbmover
/*
LISTENER=(unix2,TCPIP,2480)
NODE=(unix1,TCPIP,prod2,2480)
...
LOGPATH=/pwx/logs/i5oscond
CAPT_XTRA=/pwx/capture/i5oscond/camaps
CAPT_PATH=/pwx/capture/i5oscond
/*
/* Source-specific CAPI Connection
CAPI_CONNECTION=(NAME=i5UOWC,TYPE=(UOWC,CAPINAME=i5_AS4J,RSTRADV=600,MEMCACHE=20480))
CAPI_CONNECTION=(NAME=i5_AS4J,TYPE=(AS4J,JOURNAL=(JRN=PRODDATA/
PRODJRN),INST=PROD2,EOF=N,STOPT=(CONT=5),LIBASUSER=Y))
/*
/* CAPX CAPI Connection for continuous extraction
CAPI_CONNECTION=(NAME=CAPXPROD,TYPE=(CAPX,DFLTINST=PROD2,FILEWAIT=60,RSTRADV=600))
```

Note:

- In the CAPX CAPI_CONNECTION, the DFLTINST value is the name that is displayed in the **Instance** field for the registration group in the PowerExchange Navigator.
- You can use the JOURNAL parameter to specify journal overrides for specific extraction operations where you need to process changes from a range of journal receivers outside of default processing.

3. On the UNIX system with the PowerExchange Logger system log files, customize the PowerExchange Logger configuration file, pwxcl.cfg. For this example, include the following statements:

```
/*
/* pwxcl
/*
DBID=PROD2
DB_TYPE=AS4
CONN_OVR=i5UOWC
CAPTURE_NODE=i5OS1
CAPTURE_NODE_UID=db2user
CAPTURE_NODE_EPWD=encrypted_password
PROMPT=Y
EXT_CAPT_MASK=/pwx/capture/i5oscond/condense
COND_CDCT_RET_P=50
LOGGER_DELETES_EXPIRED_CDCT_RECORDS=Y
COLL_END_LOG=0
NO_DATA_WAIT=0
NO_DATA_WAIT2=10
FILE_SWITCH_VAL=20000
FILE_SWITCH_CRIT=R
CAPT_IMAGE=BA
```

Note: The CAPTURE_NODE parameter points to the source system node where the PowerExchange Listener processes capture requests.

4. Start the PowerExchange Listener and PowerExchange Logger for Linux, UNIX, and Windows on the UNIX system. Verify that the PowerExchange Listener is also running on the I5/OS system.
5. On the PowerCenter Integration Service system, add the following NODE statements to the dbmover file:
 - A NODE statement that points to the PowerExchange Listener on the source system
 - A NODE statement that points to the UNIX system with the PowerExchange Logger log files

This example uses the following NODE statements in the dbmover file on the PowerCenter Integration Service system:

```
NODE=(i5OS1,TCPIP,i5OS1,2480)
NODE=(unix2,TCPIP,prod2,2480)
```

6. Create a PowerCenter mapping, session, and workflow.
7. Configure a PWX DB2i5OS CDC Real Time application connection for CDC sessions that extract change data from the PowerExchange Logger log files on the UNIX system.

For this example, set the following connection attributes:

- For the **Location** attribute, enter unix2 to point to the node where the PowerExchange Logger for Linux, UNIX, and Windows log files reside. CDC sessions will read data from this location.
 - For the **Map Location** attribute, enter i5OS1 to point to the location of the extraction maps, which is the source system node.
 - For the **Map Location User** attribute, enter a valid user ID for the map location.
 - For the **Map Location Password** attribute, enter the password for the map location user.
 - For the **CAPI Connection Name** attribute, enter CAPXPROD to indicate the CAPX CAPI_CONNECTION statement to use.
8. Cold start the CDC session.

The session begins extracting change data from the PowerExchange Logger log files on the UNIX system.

CHAPTER 6

Introduction to Change Data Extraction

This chapter includes the following topics:

- [Change Data Extraction Overview, 83](#)
- [Extraction Modes, 84](#)
- [PowerExchange-Generated Columns in Extraction Maps, 84](#)
- [Uses of BI and CI Fields in Extraction Maps, 90](#)
- [Restart Tokens and the Restart Token File, 91](#)
- [Multiple-Source Processing in CDC Sessions, 92](#)
- [Commit Processing with PWXPC, 93](#)
- [Tuning Options, 95](#)

Change Data Extraction Overview

PowerExchange works in conjunction with PWXPC and PowerCenter to extract captured change data and write it to one or more targets. Learn key concepts about extraction processing so that you can configure CDC sessions for efficient extraction of data and proper restart and recovery.

To extract change data that PowerExchange captured, import the metadata for the capture source into PowerCenter Designer. Use one of the following methods:

- For relational data sources, import either the extraction maps from PowerExchange or the source metadata from the database. If you import source metadata, you might need to modify the source definition in Designer to add PowerExchange-defined CDC columns or to remove any columns that are not included in the extraction map. If you import extraction maps, you do not need to manually add or remove these columns from the PowerCenter source definition.
- For nonrelational data sources, import the extraction maps from PowerExchange.

After you import the metadata, you can use the source definitions in PowerCenter to create mappings, sessions, and workflows for extracting change data from PowerExchange.

Extraction Modes

You can extract the change data that PowerExchange captured in near real time or as a batch process.

You indicate the extraction mode by setting the PowerCenter connection type and certain PowerExchange CDC configuration parameters. Some extraction modes are available only if you use PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows.

Based on your extraction requirements, use one of the following extractions modes:

Real-time extraction mode

Continuously extracts change data in near real time from the change stream. Extraction processing continues until the CDC session stops or is interrupted.

To implement this mode, configure a PWX CDC Real Time application connection in PowerCenter for your data source type.

Batch extraction mode

Extracts change data from PowerExchange Condense condense files on IBM i (i5/OS) or z/OS, or from PowerExchange Logger for Linux, UNIX, and Windows log files. Data is extracted only from the files that are closed at the time the CDC session runs. The CDC session ends after it completes processing the files.

To implement this mode, configure the following items:

- In the PowerExchange Navigator, set the **Condense** option to **Part** or **Full** in the capture registrations.
- In PowerCenter, configure a PWX CDC Change application connection for your data source type.

Continuous extraction mode.

Continuously extracts change data from open and closed PowerExchange Logger for Linux, UNIX, and Windows log files in near real time.

For data sources on IBM i (i5/OS) or z/OS, this extraction mode is available only if you log data to a remote PowerExchange Logger for Linux, UNIX, and Windows on another system.

To implement this mode, configure the following items:

- In the PowerExchange Navigator, set the **Condense** option to **Part** in the capture registrations.
- In PowerCenter, configure a PWX CDC Real Time application connection for your data source type.
- Configure a CAPX CAPI_CONNECTION statement in the DBMOVER configuration file.
- If you remote logging of data from z/OS or i5/OS data sources to a PowerExchange Logger for Linux, UNIX, and Windows, configure the remote PowerExchange Logger to log change data from the source system.

PowerExchange-Generated Columns in Extraction Maps

Besides the table columns that are defined in capture registrations, extraction maps include columns that PowerExchange generates.

These PowerExchange-generated columns contain CDC-related information, such as the type of SQL change and time stamp.

When you import an extraction map in Designer, PWXPC includes the PowerExchange-generated columns in the source definition.

When you run a database row test on an extraction map, the PowerExchange Navigator displays the PowerExchange-generated columns in the results. By default, the PowerExchange Navigator hides these columns from view when you open the extraction map. To display these columns, open the extraction map, right-click anywhere within the **Extract Definition** window, and select **Show Auto Generated Columns**.

Note: By default, all columns are selected in extraction maps except the DTL__columnname_CNT, DTL__columnname_IND, and DTL__CI_columnname columns. To add these columns, you must edit the extraction map.

The following table describes the columns that PowerExchange generates for each change record:

Column	Description	Datatype	Length
DTL__CAPXRESTART1	<p>Provides a binary value that represents the position of the end of the UOW for that change record followed by the position of the change record itself.</p> <p>The length of a sequence token varies by data source type, except on z/OS where sequence tokens for all data source types have the same length.</p> <p>The value of DTL__CAPXRESTART1 is also known as the <i>sequence token</i>, which when combined with the <i>restart token</i> comprises the restart token pair.</p> <p>A sequence token for a change record is a strictly ascending and repeatable value.</p>	VARBIN	255
DTL__CAPXRESTART2	<p>Provides a binary value that represents a position in the change stream that can be used to reconstruct the UOW state for the change record, with the following exceptions:</p> <ul style="list-style-type: none"> - Microsoft SQL Server CDC. A binary value that contains the DBID of the distribution database and the name of the distribution server. - Change data extracted from full condense files on z/OS or i5/OS. A binary value that contains the instance name from the registration group of the capture registration. <p>The length of a restart token varies by data source type. On z/OS, restart tokens for all data source types have the same length, except for change data extracted from full condense files.</p> <p>The value of DTL__CAPXRESTART2 is also known as the <i>restart token</i>, which when combined with the <i>sequence token</i> comprises the restart token pair.</p>	VARBIN	255
DTL__CAPXROWID	<p>For PowerExchange Express CDC for Oracle, provides the physical rowid value. PowerExchange can include rowid values in change records for Oracle tables only if the tables do not have row movement enabled.</p> <p>To enable the capture of rowid values, you must configure the OPTIONS ROWID=Y statement in the Express CDC configuration file.</p> <p>The rowid is useful for processing rows in unkeyed tables during CDC extraction sessions.</p>	CHAR	18

Column	Description	Datatype	Length
DTL__CAPXRRN	For Db2 for i sources only, provides the relative record number up to 2,147,483,647. DTL__CAPXRNN is deprecated but still supported in existing extraction maps created before PowerExchange 10.5.	NUM32	4
DTL__CAPXERRN	For Db2 for i sources only, provides the relative record number, including extended values up to the system maximum value of 4,294,967,288.	DTLNUM64U	8
DTL__CAPXUOW	Provides a binary value that represents the position in the change stream of the start of the UOW for the change record.	VARBIN	255
DTL__CAPXUSER	Provides the user ID of the user who made the change to the data source, with the following exceptions: <ul style="list-style-type: none"> - For Adabas 8.3 CDC sources, this value is the Security User-id (SECUID) of the user if the Adabas File Definition includes the system field SY=SECUID. - For Datacom table-based CDC sources, this value is the MUF name. - For Db2 for i (i5/OS) CDC sources, this value depends on the LIBASUSER parameter in the AS4J CAPI_CONNECTION statement. If LIBASUSER=Y, this value is the library name and file name of the file where the change was made. If LIBASUSER=M, this value is the library name, file name, and data member name of the file where the change was made. If LIBASUSER=N, this value is the user ID of the user who made the change. If LIBASUSER=P, this value is the name of the program that made the change. - For Db2 for z/OS CDC sources, this value depends on the UIDFMT parameter in the LRAP CAPI_CONNECTION. Depending on the parameter setting, this value can be a Db2 connection identifier, correlation identifier, connection type, plan name, user ID, or all of these values in the format UID:PLAN:CORR:CONN:CTYPE. If you do not specify the UIDFMT parameter, this value is the user ID of the user who made the change. - For IDMS CDC sources, this value is the value that the user program puts in the program name field of the application subschema control block. Usually, this value is the user program name. - For IMS synchronous CDC sources, this value depends on the UIDFMTIMS parameter in the LRAP CAPI_CONNECTION statement. Depending on the parameter setting, this value can be a user ID, a PSB name, or both values in the format <i>userid:psbname</i>. If you do not specify the UIDFMTIMS parameter, the user ID is used by default. - For Microsoft SQL Server CDC sources, this value depends on the UIDFMT parameter in the MSQL CAPI_CONNECTION statement. If UIDFMT=DBNAME, this value is the SQL Server publication database name. If UIDFMT=NONE, this value is a null. - For Oracle CDC sources, this value is a user ID that PowerExchange gets from Oracle, if available. Otherwise, this value is null. 	VARCHAR	255

Column	Description	Datatype	Length
DTL__CAPXTIMESTAMP	<p>Provides the time stamp that the source DBMS records for the database change record.</p> <p>This value can be either the time stamp that the source DBMS writes to the change record in the database logs or the time stamp of the transaction commit on the source database.</p> <p>The type of time stamp depends on the source type and certain parameters:</p> <ul style="list-style-type: none"> - For Db2 sources on Linux, UNIX, or Windows, the transaction commit time stamp. - For Microsoft SQL Server sources, the time at which the change was written to the distribution database. - For MySQL sources, the time at which MySQL recorded the change event in the binary log. - For Oracle sources, the timestamp type is controlled by the TIME_STAMP_MODE parameter in the OPTIONS statement of the Express CDC configuration file. - For all sources that require a UOWC CAPI_CONNECTION statement, the time stamp type is controlled by the TIMESTAMP parameter in the UOWC CAPI_CONNECTION statement in the DBMOVER file. <p>For more detailed information about time stamps for each source type, see Appendix A, "DTL__CAPXTIMESTAMP Time Stamps" on page 153.</p> <p>The time stamp format is:</p> <p>YYYYMMDDhhmmssnnnnnn</p> <p>Where:</p> <ul style="list-style-type: none"> - YYYY is the four-digit year. - MM is the month. - DD is the day. - hhmmssnnnnnn is hours, minutes, seconds, and microseconds. <p>Note: Db2 on Linux, UNIX, or Windows and Oracle do not support microseconds in the time stamp.</p>	CHAR	20

Column	Description	Datatype	Length
DTL__CAPXACTION	<p>Indicates the type of change record that PowerExchange passed to the target during extraction processing. This indicator corresponds to the type of SQL change operation on the source database.</p> <p>Valid values:</p> <ul style="list-style-type: none"> - I. Insert. - D. Delete. - U. After image of an UPDATE. - T. Before image of an UPDATE. (ODBC connections only) <p>If you specify an Image Type of BA on the connection for a CDC session, PowerExchange generates a delete record followed by an insert record for a source update. In the delete record, the DTL__CAPXACTION column contains the value D. In the insert record, the DTL__CAPXACTION column contains the value I.</p> <p>If you specify an Image Type of AI on the connection for a CDC session, PowerExchange generates one record for an update. In this record, the DTL__CAPXACTION column contains the U value.</p> <p>If you use an ODBC connection to write change data to a staging table and either set the ODBC driver CAPXIMAGETYPE parameter to TU or enter the SQL escape sequence DTLIMTYPE=TU in PowerCenter, this column can contain a value of T or U. For each source update, PowerExchange delivers two records to the staging table: one for the before image and another for the after image. In the before image record, the DTL__CAPXACTION column contains the T value. In the after image record, The DTL__CAPXACTION column contains the U value.</p>	CHAR	1
DTL__CAPXCASDELIND	<p>For Db2 for z/OS sources only, indicates whether or not Db2 deleted the row because the table specifies the ON DELETE CASCADE clause. Valid values:</p> <ul style="list-style-type: none"> - Y. Indicates that Db2 deleted the row because of a cascade delete rule. - N. Indicates that Db2 did not delete the row because of a cascade delete rule. 	CHAR	1
DTL__BI_< i>columnname	Provides the before image of a column that an UPDATE operation changed.	Datatype of the source column	Length of the source column
DTL__CI_< i>columnname	<p>Indicates whether or not an UPDATE operation changed the column value. Valid values:</p> <ul style="list-style-type: none"> - Y. The column value was changed by an UPDATE operation. - N. The column was not changed by an UPDATE operation. - <i>null</i>. The column was changed by an INSERT or DELETE operation. It was not changed by an UPDATE. <p>Note: By default, the change indicator column is not included in extraction maps. To add it, you must edit an extraction map and select this auto-generated column.</p>	CHAR	1

Column	Description	Datatype	Length
DTL__ST_lob_columnname	<p>For a Db2 for z/OS LOB column, indicates whether or not the column contains all of the LOB data. The ECCR provides incomplete LOB data if the data is not stored fully inline in the base table space or exceeds 32 KB in size. Valid values:</p> <ul style="list-style-type: none"> - C. The source column contains all of the LOB data. The ECCR was able to capture all of the LOB data because the data is stored fully inline in the base table space and does not exceed 32 KB. - I. The source column contains incomplete LOB data. The ECCR was unable to capture all of the LOB data because the data is stored in an auxiliary table space, or the data is stored fully inline but exceeds 32 KB in size. - <i>null.</i> The column contains null values only. <p>If the LOB data is not stored fully inline in the base table, include the DTL__ST_lob_columnname column. When this column specifies I to indicate incomplete LOB data, you can use PowerCenter transformations to retrieve all the current LOB data for the associated source column.</p> <p>For an Oracle LOB column, DTL__ST_lob_columnname indicates whether or not the associated source column contains the LOB data. PowerExchange Express CDC for Oracle provides complete LOB data if the data is fully stored in the row. Valid values:</p> <ul style="list-style-type: none"> - C. The source column contains all the LOB data. PowerExchange Express CDC captured all of the LOB data because the data is stored fully inline in the row. - I. The source column was changed but it does not contain LOB data. PowerExchange Express CDC was not able to capture the LOB data because the data is not stored inline in the row. - <i>null.</i> The source column contains null values only. <p>If LOB data is not stored inline in the base table, include the DTL__ST_lob_columnname column. When this column specifies I to indicate incomplete LOB data, you can use PowerCenter transformations to retrieve all the current LOB data for the associated source column.</p> <p>Note: This column is included in extraction maps by default. To remove it, open the extraction map in the PowerExchange Navigator and clear this automatically generated column.</p>	CHAR	1
DTL__columnname_CNT	<p>A binary count that PowerExchange generates for a variable length column of the type VARCHAR and VARBIN. The count is used to determine the length of the column during change data extraction processing.</p> <p>Note: By default, the binary count column is not included in extraction maps. To add it, you must edit an extraction map and select this auto-generated column.</p>	NUM32U	0
DTL__columnname_IND	<p>Indicates whether or not a nullable column contains a null. PowerExchange generates this column only for nullable columns.</p> <p>Note: By default, the null indicator column is not included in extraction maps. To add it, you must edit an extraction map and select this auto-generated column.</p>	BIN	1

RELATED TOPICS:

- [“Uses of BI and CI Fields in Extraction Maps” on page 90](#)

Uses of BI and CI Fields in Extraction Maps

PowerExchange captures both before images and after images of data for all SQL UPDATE operations on source columns. To access before image data to process change data in some way during CDC sessions, add before image (BI) and change indicator (CI) fields to extraction maps.

For example, you can use the BI and CI fields for the following purposes:

- To filter captured data for extraction and apply processing.
- To update primary keys on the target based on whether primary keys on the source changed.

Case 1. Filtering Change Data for Extraction and Apply Processing

If you add CI fields for one or more data columns in an extraction map, PowerExchange compares before and after images of the data captured for these columns. If an UPDATE occurred, PowerExchange sets the generated `DTL__CI_column_name` value to Y.

You can use a `DTL__CI_column_name` in WHERE clause filters for CDC sessions to filter the change stream during extraction processing. In PowerCenter, define the filters in the **Filter Override** attribute of the session properties. By using these filters, you can reduce the amount of data that PowerCenter processes.

During extraction processing, PWXPC creates SQL SELECT statements that include the WHERE clause filters. PWXPC passes these statements to PowerExchange. PowerExchange selects and returns the data that matches the WHERE conditions. PWXPC then makes this data available to the CDC sessions. Additional manipulation of the data might occur in PowerCenter, based on how you define the mappings.

To filter change data for extraction and apply processing:

1. In the PowerExchange Navigator, edit the extraction map that you plan to import as the source definition for the CDC session. For each column that you want to filter on, add a CI field.
PowerExchange generates CI fields that have names in the format `DTL__CI_column_name`.
For more information about adding CI fields to extraction maps, see the *PowerExchange Navigator User Guide*.
2. In PowerCenter, define WHERE clause filters in the **Filter Override** attribute of the CDC session properties.
For the filters, enter `DTL__CI_column_name` conditions. For example, enter `DTL__CI_ACCOUNT='Y'`, where 'Y' indicates an Update occurred.
For more information about filter overrides on CDC sessions, see *PowerExchange Interfaces for PowerCenter*.

When the CDC session runs, PWXPC provides only the change data that matches the WHERE filter to PowerCenter for extraction and apply processing.

Note: Using many filters with CI fields might noticeably increase CPU overhead.

Case 2. Updating Primary Key Fields on the Target

If the target primary key does not match the source primary key, or if the source database allows updates to primary key fields, CDC sessions cannot apply updates to target keys based on after image data only.

To prevent this problem, you can select the **BA** option for the **Image Type** attribute on PWX CDC application connections. This option causes PWXPC to generate two transactions for each source UPDATE: a DELETE

followed by an INSERT. The DELETE deletes the old row based on the before image. The INSERT inserts a row based on the after image.

Alternatively, to avoid the overhead of generating two transactions for every source UPDATE, select the **AI** option for the **Image Type** attribute. Also use CI and BI columns in combination with a PowerCenter Flexible Target Key Custom transformation. With this configuration, PowerCenter generates an INSERT or UPDATE transaction only when a source UPDATE results in changes to primary key fields on the target. Complete the following steps to implement this solution.

To update primary key fields on the target using BI and CI fields:

1. In the PowerExchange Navigator, edit the extraction map that you plan to import as the source definition for the CDC session. Add both BI and CI fields for one or more of the primary key columns on the source.
2. Verify that the **Image Type** attribute on the PWX CDC application connection for the CDC session is **AI**. This setting causes PWXPC to pass Updates to the CDC session as Updates. Because you added BI and CI fields for key columns in the extraction map, Update rows for these columns include both before and after images.
3. In PowerCenter, define a Flexible Target Key Custom transformation. The transformation uses the DTL__CI indicator for the source key columns to detect when Updates to primary key columns on the target are needed.
4. Add the transformation to the mapping for the CDC session.

For more information about Flexible Target Key Custom transformations, see *PowerExchange Interfaces for PowerCenter*.

Restart Tokens and the Restart Token File

PowerExchange uses a pair of token values, called a restart token pair, to determine where to begin extracting change data in the change stream for each source in a CDC session. A restart token pair matches the position of a specific change record in the change stream.

You can specify restart token pairs in the restart token file. PWXPC also stores restart tokens for CDC sessions that have run in a state table or file. The token values in the restart token file override those in the state table or file.

Specify restart tokens in the restart token file in the following situations:

- For a new CDC session, specify restart token pairs for the sources in the session. You can define a unique restart token pair for each source, or use the special override statement to specify a restart token pair that pertains to all or multiple data sources. The restart tokens should represent the point-in-time in the change stream when you materialized the corresponding targets.
- If you add a data source to a CDC session, specify a restart token pair for that source.
- If you need to override token values for one or more data sources in a CDC session, use override statements in the restart token file.

A restart token pair is composed of the following token types:

Sequence token

A binary value that represents, for each change record that is read, the change stream position of the end of the UOW followed by the position of the change record. A sequence token is a strictly ascending and repeatable value.

Restart token

A binary value that represents, for each change record that is read, a change stream position that PowerExchange can use to reconstruct the UOW state for the change record.

In some cases, the restart token might contain the position of the oldest open UOW. An open UOW is a UOW for which PowerExchange has read the beginning of the UOW from the change stream but has not yet read the commit record, or end-UOW.

When a CDC session runs, PWXPC reads the token values for each source from the state table or file and also reads the restart token file. PowerExchange uses the appropriate restart token values to determine the point from which to start reading change data from the change stream for each source in the CDC session. After determining the start point, PowerExchange starts reading and passing change data to PWXPC. PWXPC uses the sequence token for a source to determine the point at which to start providing the change data for the source.

RELATED TOPICS:

- [“Creating Restart Tokens for Extractions” on page 116](#)
- [“Configuring the Restart Token File” on page 117](#)
- [“Displaying Restart Tokens” on page 117](#)

Multiple-Source Processing in CDC Sessions

When you use PWX CDC application connections to extract change data, PowerExchange reads the change stream in a single pass for all source definitions in the mapping. The sources must be of the same type and use the same change stream.

To create source definitions in Designer, import source metadata in one of the following ways:

- Import a PowerExchange extraction map by using the **Import from PowerExchange** dialog box.
- Import table definitions from a relational database by using the **Import from PowerExchange** dialog box or the **Import from Database** dialog box.

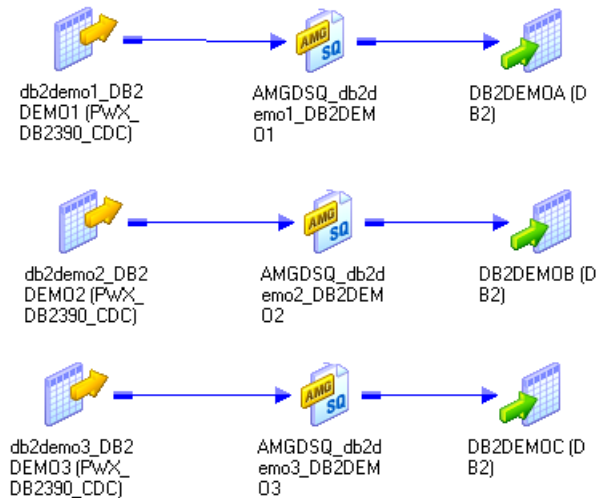
Informatica recommends that you import extraction maps. It makes creating mappings and sessions easier for the following reasons:

- The source definition contains the extraction map name. You do not need to provide this name when you configure the session.
- The source definition contains the PowerExchange-generated CDC columns, such as the DTL__CAPX columns. You do not need to add these columns to the source definition.

During change data extraction, PowerExchange processes all DB2 for i5/OS source definitions in a mapping in a single pass if the sources use the same change stream.

The following figure shows an example mapping in PowerCenter Designer with three DB2 sources:

Mapping Designer



If you include this mapping in a session that uses a PWX DB2i50S CDC application connection, PowerExchange reads the change stream and extracts changes for all three source tables in a single pass. PowerExchange extracts change data in chronological order, based on when the UOWs completed. PowerExchange passes the change data to PWXPC, and PWXPC provides the changes to the appropriate source qualifier.

If you create a workflow that contains multiple CDC sessions, PowerExchange uses a connection for each session, even if the sessions extract change data from the same change stream, such as PowerExchange Logger for z/OS log files.

Note: Because the example mapping uses source definitions created from extraction maps, it cannot be used for bulk data movement operations. However, mappings that use source definitions created from database relational metadata can be used for either change data extraction or bulk data movement.

Commit Processing with PWXPC

The PowerCenter Integration Service, in conjunction with PWXPC, commits data to the target based on the **Commit Type** session property and the commitment control attributes specified on PWX CDC Change or Real Time application connections.

By default, the **Commit Type** session property specifies **Target** for target-based commit processing. However, the PowerCenter Integration Service always uses source-based commit processing for CDC sessions. Change the commit type to **Source**. If you retain the default value and run a CDC session, the PowerCenter Integration Service automatically uses source-based commit processing and writes message WRT_8226 in the session log. You do not need to set the **Commit Interval** session property because PWXPC ignores it.

To control when commits occur, configure commitment control attributes on the PWX CDC Change and Real Time application connections.

The following table describes these connection attributes:

Connection Attribute	PWX Real Time or Change Connections	Description
Maximum Rows Per commit	Both	Maximum number of change records that PWXPC processes before it flushes the data buffer to commit the change data to the targets. If necessary, PWXPC continues to process change records across UOW boundaries until this maximum rows limit is met. PWXPC does not wait for a UOW boundary to commit the change data. Default is 0, which causes PWXPC to not use this maximum rows limit.
Minimum Rows Per commit	Real Time	Minimum number of change records that PowerExchange reads from the change stream before it passes any commit records in the change stream to PWXPC. Before reaching this minimum, PowerExchange skips commit records and passes only the change records to PWXPC. Default is 0, which causes PowerExchange to not use this minimum rows limit.
Real-time Flush Latency in milli-seconds	Real Time	Number of milliseconds that must elapse before PWXPC flushes the data buffer to commit change data to the targets. When this latency period expires, PWXPC continues to read the changes in the current UOW until it reaches the end of the UOW. Then, PWXPC flushes the data buffer to commit the change data to the targets. Default is 0, which causes PWXPC to use 2,000 milliseconds.
UOW Count	Both	Number of UOWs that PWXPC must process before flushing the data buffer to commit the change data to the targets. Default is 1.

PWXPC flushes the data buffer to commit change data to the targets when one of the following thresholds is met, whichever one is first:

- **Maximum Rows Per commit**
- **Real-Time Flush Latency in milli-seconds**
- **UOW Count**

If you specify **Minimum Rows Per commit**, this threshold must also be met before a commit occur.

After PWXPC commits the change data, it resets the UOW count, the maximum and minimum rows per commit, and the real-time flush latency timer. PWXPC continues to read change data. Whenever one of the commitment control thresholds is met, PWXPC commits change data to the targets. Commit processing continues until the CDC session is stopped, ends, or terminates abnormally. When the PWXPC CDC reader ends normally, PWXPC issues a final commit to flush all complete, buffered UOWs and their final restart tokens to the targets. Prior to ending, the PWXPC CDC reader writes the following message to the session log:

```
PWXPC 12075 [INFO] [CDCRestart] Session complete. Next session will restart at: Restart 1 [restart1_token] : Restart 2 [restart2_token]
```

RELATED TOPICS:

- [“Commitment Control Attributes” on page 106](#)
- [“Examples of Controlling Commit Processing” on page 109](#)

Tuning Options

PowerExchange provides flexible tuning options that you can use to reduce CPU usage on a source system that has constrained CPU resources. These options can also potentially improve throughput for CDC sessions.

The tuning options move some extraction processing to another machine such as the PowerCenter Integration Service machine. If the machine to which processing is offloaded has sufficient resources, the performance of CDC sessions might improve.

The following tuning options can help you take maximum advantage of the system resources that are available and maximize throughput for CDC sessions:

- **Offload processing.** Use offload processing to transfer column-level extraction processing from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine. Also, if the data source type requires use of the UOW Cleanser (UOWC), offloading transfers UOWC processing to the Integration Service machine. Use offloading to help increase throughput when resources available for the PowerExchange Listener are constrained on the source system.
- **Remote logging of change data.** Configure a PowerExchange Logger for Linux, UNIX, and Windows instance on a system other than the source system. The PowerExchange Logger reads change data from the source and writes the data to its local log files. CDC sessions extract the change data from the PowerExchange Logger log files. This configuration moves resource-intensive, column-level processing from the source system to the PowerExchange Logger system. Use remote logging to help improve throughput for CDC sessions when resources on the source system are constrained.
- **Multithreading.** Enable the use of multiple worker threads for resource-intensive, column-level extraction processing. You can use multithreading on the source system to process data from Linux, UNIX, or Windows data sources, or on another system where the extraction processing runs. Enable multithreading only if extractions appear to be CPU bound. You can use multithreading with the offloading feature or remote logging.

CHAPTER 7

Extracting Change Data

This chapter includes the following topics:

- [Overview of Extracting Change Data, 96](#)
- [Task Flow for Extracting Change Data, 97](#)
- [Testing an Extraction Map, 98](#)
- [Configuring PowerCenter CDC Sessions, 99](#)
- [Recovery and Restart Processing for CDC Sessions, 111](#)
- [Creating Restart Tokens for Extractions, 116](#)
- [Displaying Restart Tokens, 117](#)
- [Configuring the Restart Token File, 117](#)

Overview of Extracting Change Data

Use PowerExchange in conjunction with PWXPC and PowerCenter to extract captured change data and write the data to one or more targets.

To extract the change data that PowerExchange captures, in Designer, import metadata for the CDC sources and targets and create a mapping. Then, in Workflow Manager, create an application connection, a session, and a workflow. You can create multiple mappings, sessions, and workflows based on the same source and target definitions, if appropriate.

For relational data sources, you can import the metadata from either database definitions or PowerExchange extraction maps. For nonrelational sources, you must import the metadata from PowerExchange extraction maps.

Tip: Informatica recommends that you import the metadata from PowerExchange extraction maps. When you use extraction maps, the source definitions contain all of the PowerExchange-generated CDC columns, including any before image (BI) and change indicator (CI) columns you added. Also, you do not need to specify the extraction map name for each source in the session properties because PWXPC can derive the extraction map name from the source definition.

Before starting a CDC session for the first time, create restart tokens to define the extraction start point in the change stream. You might also need to create restart tokens to resume extraction processing in a recovery scenario.

Optionally, you can configure event table processing to stop a CDC session that uses real-time extraction mode based on user-defined events.

Also, you can use the following tuning options to help take maximum advantage of the available system resources and maximize throughput for CDC sessions:

- **Offload processing.** Use offload processing to transfer column-level extraction processing from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine.
- **Remote logging of change data.** Configure a PowerExchange Logger for Linux, UNIX, and Windows instance on a system other than the source system. The PowerExchange Logger reads change data from the source and logs it in the PowerExchange Logger log files on the other system. CDC sessions can then extract change data from the PowerExchange Logger log files.
- **Multithreading.** Enable the use of multiple worker threads to use multithreading for resource-intensive, column-level extraction processing. You can use multithreading on the source system if you are processing data from Linux, UNIX, or Windows data sources, or on another system where the extraction processing runs.

Task Flow for Extracting Change Data

Use this task flow to identify the tasks that you need to complete to configure and start extraction processing. You do these tasks in the PowerExchange Navigator, PowerCenter Designer, and PowerCenter Workflow Manager.

Before you begin, complete configuration of the data source and PowerExchange, and create capture registrations in the PowerExchange Navigator.

1. Edit the extraction map if necessary.

You can make the following changes:

- Deselect any column for which you do not want to extract change data. PowerExchange still captures change data for these columns.
 - Add change indicator (CI) and before image (BI) columns.
2. To test the extraction map, do a database row test on the extraction map in the PowerExchange Navigator.
 3. In Designer, import metadata for the sources and targets.
 4. In Designer, configure a mapping to extract and process change data.
 5. In Workflow Manager, configure a connection and session.
 6. Create restart tokens for the CDC session.
 7. Configure the restart token file.
 8. If you want to stop extraction processing based on user-defined events, implement event table processing.
 9. To offload column-level extraction processing and UOW Cleanser processing from the source system to the PowerCenter Integration Service machine, configure offload processing. You can also use offload processing to offload change data to a remote PowerExchange Logger for Linux, UNIX, and Windows process on another machine.

If you configure offload processing for real-time extractions, you can also configure multithreaded processing to help improve throughput.
 10. Start the CDC session.

Testing an Extraction Map

In the PowerExchange Navigator, perform a database row test to verify that PowerExchange can retrieve change data from a registered source based on an extraction map.

A database row test enables you to:

- Preview change data that PowerExchange captured for the registered data source.
- Preview change data that either PowerExchange Condense on i5/OS or z/OS or the PowerExchange Logger for Linux, UNIX, and Windows captured for registered source.
- Verify that the extraction map properly maps the captured change data.

1. In the PowerExchange Navigator, open the extraction group and the extraction map.
2. Select the extraction map and click **File > Database Row Test**.
3. In the **Database Row Test** dialog box, enter information in the following fields:

DB Type

An option that indicates the extraction mode:

- **CAPXRT**. Real-time extraction mode or continuous extraction mode.
- **CAPX**. Batch extraction mode.

Location

Node name for the location of the system on which the captured change data resides. This name must be defined in a NODE statement in the dbmover.cfg configuration file on the Windows machine from which you run the database row test.

UserID and Password

Optional. A user ID and password that provides access to the source data.

Fetch

To preview data, select **Data**.

Application

An application name. For a row test, an application name is not required. However, you must enter at least one character in this field. PowerExchange does not retain this value.

SQL Statement

A SQL SELECT statement that PowerExchange generates for the fields in the extraction map. You can edit this statement, if necessary.

In the statement, a table is identified as follows:

Schema.RegName_TableName

Where:

- *Schema* is a schema name for the extraction map.
- *RegName* is the name of the capture registration that corresponds to the extraction map.
- *TableName* is the table name of the data source.

Note: If you enter **CAPX** in the **DB Type** field, you can extract change data only after PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows closes at least one condense file or log file. Otherwise, PowerExchange does not display change data and writes message PWX-04520 to the PowerExchange message log. PowerExchange also writes this message if no change data for the source has been captured, condensed, or logged.

4. Click **Advanced**.

5. Complete the fields in the **CAPX Advanced Parameters** dialog box or **CAPXRT Advanced Parameters** dialog box.
 - If you use continuous extraction mode, enter the CAPX CAPI_CONNECTION name in the **CAPI Connection Name** field.
 - If you offload change data to PowerExchange Logger for Linux, UNIX, and Windows log files on a system that is remote from the source, enter the location of the extraction maps in the **Location** field.
6. Click **OK**.
7. Click **Go**.

The database row test returns each change from the extraction start point, by column. The results include the PowerExchange-generated CDC columns, which provide information such as the change type, timestamp, and user ID.

Configuring PowerCenter CDC Sessions

After you import metadata for CDC data sources and targets into PowerCenter, you can create a mapping, connection, and a CDC session for extracting change data. You must configure many session and connection attributes.

Changing Default Values for Session and Connection Attributes

Some PowerCenter session and application connection attributes have default values that are appropriate only for bulk data movement operations. You must edit these attributes for CDC sessions.

The following table describes the session and connection attributes that you need to set for CDC, including the recommended values:

Attribute Name	Attribute Location	Recommended Value for CDC	Description
Commit Type	Properties Tab for the session	Source	Default value is Target . If you accept the default, the PowerCenter Integration Service automatically overrides the default to use source-based commit processing. However, you should change this attribute to Source so that you can disable the Commit On End Of File attribute.
Commit On End Of File	Properties Tab for the session	Disabled	By default, this attribute is enabled. If you accept the default, the PowerCenter Integration Service commits the change data in the buffer to the targets when the session ends. The final commit occurs after the PWXPC CDC reader has committed all complete UOWs in the buffer, along with their restart tokens, to the targets. This timing can cause the restart tokens and target data to be out of sync. The final restart tokens might represent a point in the change stream that is earlier than final change data that the PowerCenter Integration Service commits to the targets. As a result, duplicate data might occur when the CDC session restarts. To prevent potential duplicate data, disable this attribute.

Attribute Name	Attribute Location	Recommended Value for CDC	Description
Recovery Strategy	Properties Tab for the session	Resume from last checkpoint	Default value is Fail task and continue workflow . To properly restart CDC session, PowerExchange CDC and PWXPC require that this option is set to Resume from last checkpoint .
Stop on errors	Config Object Tab for the session	1	Default value is 0. By default, the PowerCenter Integration Service does not consider errors when writing to targets as fatal. The following types of error are non-fatal: <ul style="list-style-type: none"> - Key constraint violations - Loading nulls into a not null field - Database trigger responses If write errors occur, change data loss might occur because PWXPC has advanced the restart tokens values. To maintain target data and restart token integrity, set this option to 1.
Application Name	Application Connection	Enter a unique name for each CDC session.	Default is the first 20 characters of the workFlow name. Attention: Because the default might not result in a unique name, enter a unique name.
RestartToken File Folder	Application Connection	Default value	The default is \$PMRootDir/Restart. This default is acceptable for CDC.
RestartToken File Name	Application Connection	Enter a unique name for each CDC session.	If you enter an Application Name value, the default is that application name. If you do not enter an Application Name value, the default is the workflow name. Attention: Because a default might not result in a unique name, enter a unique restart token file name.
Number of Runs to Keep RestartToken File	Application Connection	1 or greater	Default is 0. PWXPC keeps only one backup copy of the restart token initialization and termination files. Enter a value greater than 0 to make history available for recovery purposes.

Configuring Application Connection Attributes

To extract change data, you must configure certain application connection attributes. For a complete list of all PWX CDC application connection attributes, see *PowerExchange Interfaces for PowerCenter*.

RELATED TOPICS:

- [“Image Type” on page 101](#)
- [“Event Table Processing” on page 104](#)
- [“CAPI Connection Name Override” on page 101](#)
- [“Idle Time” on page 102](#)
- [“Restart Control Attributes” on page 103](#)
- [“Flush Latency” on page 105](#)
- [“Target Latency ” on page 105](#)

Image Type

Use the **Image Type** attribute to indicate how PWXPC passes captured Updates to CDC sessions that extract and apply the updates to the target.

Enter one of the following options for this attribute:

- **AI**. Process Updates as Update operations. PWXPC passes each Update as a single Update record. An Update record includes after images of the data only, unless you add before image (BI) and change indicator (CI) fields to the extraction map that you import for the source definition for the CDC session.
- **BA**. Process Updates as Deletes followed by Inserts. PWXPC passes each Update as a Delete record followed by an Insert record. The Delete record contains the before image of the data, and the Insert record contains the after image.

Default is **BA**.

If you use **BA**, PWXPC generates, for each captured Update operation, a Delete record that contains the before image of the data and an Insert record that contains the after image. If you also define BI and CI fields for some columns in the extraction map that you import for the source definition, PWXPC populates the BI and CI fields with data in both the generated Delete and Insert records. However, for any Insert and Delete operations captured from the source, the BI and CI fields in the generated Delete and Insert records contain Null values.

If you specify **AI**, you can still use before images of the data, if available, in extraction processing. PWXPC can embed before-image data and after-image data in the same Update row. To embed before-image data, you must complete the following configuration tasks:

- In the PowerExchange Navigator, add BI and CI fields to the extraction map that you plan to import for the source definition in PowerCenter.
- If you use batch or continuous extraction mode, enter BA for the CAPT_IMAGE parameter in the PowerExchange Condense or PowerExchange Logger for Linux, UNIX, and Windows configuration file. This setting causes both before and after images to be stored in the PowerExchange Logger log files or PowerExchange Condense condense files. When CDC sessions run, they extract data from these files.

Informatica recommends that you use the **AI** setting if you want to process before images of data. CDC sessions can process a single Update record more efficiently than separate Delete and Insert records to get the before image data.

For example, embed before-image data and after-image data in the same Update row to handle changes to primary keys. Relational databases that allow changes to primary keys, such as Db2 for z/OS, treat these Updates as equivalent to deleting the row and readding it with a new key value. To enable PowerExchange to detect primary key changes, include BI and CI fields for the primary key columns in the extraction map for the source definition. Then, in PowerCenter, define a Flexible Target Key Custom transformation to apply the changes to the target as a Delete followed by an Insert. Include the transformation in the mapping for the CDC session. If a target relational database does not allow changes to primary keys, updates to primary keys fail.

Note: To use a Flexible Target Key Custom transformation, you must set the **Image Type** attribute to **AI** and configure BI and CI fields in the PowerExchange extraction map for the source.

For more information about adding BI and CI columns, see the *PowerExchange Navigator User Guide*.

CAPI Connection Name Override

If you define multiple CAPI_CONNECTION statements in the DBMOVER configuration file, you can use the **CAPI Connection Name Override** connection attribute to select one of the statements for a CDC session.

PowerExchange allows a maximum of eight CAPI_CONNECTION statements in the DBMOVER configuration file. You might want to use multiple CAPI_CONNECTION statements to extract changes for multiple source

types with a single PowerExchange Listener on a single machine. For example, you can extract changes for Oracle and Db2 sources through a single PowerExchange Listener by specifying multiple CAPI_CONNECTION statements.

If you use CDC offload processing, you must define the CAPI_CONNECTION statements in the dbmover.cfg file on the PowerCenter Integration Service machine. If you do not use CDC offload processing, you must define the CAPI_CONNECTION statements on the system where the change data resides.

To specify the CAPI_CONNECTION statement to use for a specific CDC session, enter the name of the CAPI_CONNECTION statement in the **CAPI Connection Name Override** connection attribute. By using the override instead of a default CAPI_CONNECTION statement, you clearly indicate which statement to use for a session.

Idle Time

Use the **Idle Time** connection attribute to indicate whether a CDC session that uses real-time or continuous extraction mode runs continuously or shuts down after it reaches the end-of-log (EOL).

You can specify that PowerExchange wait for a certain period without change activity before shutting down.

Enter one of the following values:

- -1. The CDC session runs continuously. PowerExchange returns an end-of-file (EOF) only when you manually stop the CDC session.
- 0. After reaching the EOL, PowerExchange returns an EOF and the CDC session ends.
If you want a CDC session to end periodically on an active system that is rarely idle, enter 0.
- *n*. After reaching the EOL, PowerExchange waits the specified number of seconds, *n*. If PowerExchange receives no change data of interest during this time interval, PowerExchange sends an EOF to the PowerCenter Integration Service and the CDC session ends successfully.
If you enter a low value, such as 1, the CDC session might end before PowerExchange has read all available data in the change stream.

Default is -1.

PowerExchange determines the EOL by using the current end of the change stream at the point that PowerExchange started to read the change stream. PowerExchange uses the concept of EOL because the change stream is usually not static. The actual EOL is continually moving forward. After PowerExchange reaches the EOL, it writes message PWX-09967 in the PowerExchange message log.

Often, CDC sessions that run in real-time or continuous extraction mode use the default value of -1. You can manually stop a long-running CDC session by using the PowerCenter Workflow Monitor, pmcmd commands, or the PowerExchange STOPTASK command.

If you set the **Idle Time** attribute to 0, when PowerExchange reaches the EOL, it returns an EOF to PWXPC. PWXPC and the PowerCenter Integration Service then perform the following processing:

1. PWXPC flushes all buffered UOWs and ending restart tokens to the targets.
2. The CDC reader ends.
3. After the PowerCenter Integration Service finishes writing the flushed data to the targets, the writer ends.
4. After any post-session commands and tasks run, the CDC session ends.

If you set the **Idle Time** attribute to a positive number, the following processing occurs:

1. PowerExchange reads the change stream until it reaches EOL and then the **Idle Time** wait interval begins.

2. If more data is in the change stream after the EOL, PowerExchange continues to read the change stream, looking for change data of interest to the CDC session, as follows:
 - If the idle time expires before PowerExchange reads a change record of interest for the CDC session, PowerExchange stops reading the change stream.
 - If PowerExchange reads a change record of interest to the CDC session, PowerExchange restarts the timer, passes the change data to PWXPC, and continues to read the change stream. This processing continues until the idle time expires.
3. After the idle time expires, PowerExchange passes an EOF to PWXPC.
4. PWXPC and the PowerCenter Integration Service perform the same processing as when the **Idle Time** value is 0 and the CDC session ends.

When a CDC session ends because the idle time elapsed or a PowerExchange STOPTASK command was issued, PWXPC writes the following message in the session log:

```
[PWXPC_10072] [INFO] [CDCDispatcher] session ended after waiting for [idle_time] seconds. Idle Time limit is reached
```

If you stop a continual CDC session with the PowerExchange STOPTASK command, PWXPC substitutes 86400 for the *idle_time* variable in the PWXPC_10072 message.

Note: If you specify both the **Reader Time Limit** and **Idle Time** attributes, the PowerCenter Integration Service stops reading data from the source when one of these attribute conditions is met, whichever one is first. Because the reader time limit does not result in normal termination of a CDC session, Informatica recommends that you use only the idle time limit.

Restart Control Attributes

Use PWXPC restart control attributes to identify restart information to use for a CDC session. The restart information determines the point from which PowerExchange starts reading change data for the session.

Specify restart control attributes in the following situations:

- When you create CDC session.
- When you add a source to an existing CDC session and need to specify restart information for that source.
- When you want to override some restart information that is in the state table or file for a CDC session.

The following table describes the restart control attributes that you can enter on a PWX CDC application connection:

Connection Attribute	Description
Application Name	A unique application name for the CDC session. The application name is case sensitive and cannot exceed 20 characters. Default is the first 20 characters of the workflow name. Because the default might not result in a unique name, Informatica recommends that you enter a unique name.
RestartToken File Folder	Directory name on the PowerCenter Integration Service machine that contains the restart token override file. Default is \$PMRootDir/Restart.
RestartToken File Name	The unique file name of the restart token file. This file is in the directory that is specified in the RestartToken File Folder attribute. PWXPC uses the contents of this file, if any, in conjunction with the state table or state file to determine the restart point for the CDC session. Default is the Application Name value, or if you do not specify the application name, default is the workflow name.

Attention: The values for the **Application Name** and **RestartToken File Name** attributes must be unique for each CDC session. If either one of these values is not unique, unpredictable results might occur, including session failure and potential data loss.

Event Table Processing

Use event table processing to stop the extraction of changes based on user-defined events, such as an end-of-day event.

For example, to stop an extraction process every night, after all changes for the day are processed, write a change to the event table at midnight. This change triggers PowerExchange to stop reading change data and shut down the extraction process after the current UOW completes.

Use the following rules and guidelines:

- You can use event table processing only with real-time or continuous extraction modes.
- You must create the event table and define the applications that can update the table.
- You must register the event table for change data capture from the PowerExchange Navigator.
- A CDC session monitors a single event table. Each user-defined event requires its own event table and a separate extraction process.
- The event table and all of the source tables in the CDC session must be of the same source type.

Implementing Event Table Processing

Use this procedure to implement event table processing. With event table processing, you can stop change data extraction processing based on user-defined events.

1. Create an event table.

The event table must be of the same source type and on the same machine as the change data to be extracted. For example, if you extract Db2 change data on z/OS, the event table must be a Db2 table in the same Db2 subsystem as the Db2 source tables for the extraction.

2. In the PowerExchange Navigator, create a capture registration for the event table.

When you create the capture registration, the PowerExchange Navigator generates a corresponding extraction map.

3. In PowerCenter, create a CDC connection and session.

In the **Event Table** attribute on the PWX CDC Real Time application connection, enter the name of the extraction map associated with the capture registration that you created.

4. Define applications that write an update to the event table whenever the defined event occurs.

PowerExchange reads the update and places an end-of-file (EOF) in the change stream. PWXPC processes the EOF, passes it to the PowerCenter Integration Service, and then shuts down the PowerExchange reader. The PowerCenter Integration Service completes writing all of the data that is in the pipeline to the targets and then ends the CDC session.

Flush Latency

PowerExchange reads change data into a buffer on the source system, or into a buffer on the PowerCenter Integration Service machine if you use offload processing. The PowerExchange Consumer API (CAPI) periodically flushes the buffer to transfer the change data to PWXPC on the PowerCenter Integration Service machine.

The CAPI flushes the buffer to PWXPC when the one of the following events occurs:

- The buffer becomes full.
- The CAPI timeout value that is specified by the **PowerExchange Latency in seconds** attribute on the PWX CDC Real Time connection expires.
- A commit point occurs.

To specify the flush latency for CDC sessions that run in real-time or continuous extraction mode, set the **PWX Latency in seconds** attribute on the PWX CDC Real Time application connection. This attribute specifies the maximum time that PowerExchange waits for more change data before flushing data to PWXPC. This attribute applies to PowerExchange on the source system, or to the PowerExchange client on the PowerCenter Integration Service machine if you use offload processing.

For CDC sessions that use batch extraction mode, PowerExchange always uses 2 seconds for the flush latency.

PowerExchange writes message PWX-09957 to the PowerExchange message log to identify the CAPI timeout value based on the **PWX Latency in seconds** attribute. If you select **Retrieve PWX Log Entries** on the application connection, PWXPC also writes this message in the session log.

After PowerExchange flushes the change data, PWXPC provides the data to the source qualifiers in the CDC session for further processing. Then the PowerCenter Integration Service commits the data to the targets.

Note: The **PWX Latency in seconds** value also affects how fast a CDC session responds to a stop command from Workflow Monitor or pmcmd program. Before PWXPC can process a stop request, it must wait for PowerExchange to return control to it. Use the default value of 2 seconds for the **PWX Latency in seconds** attribute to avoid unacceptable delays in stop command processing.

Target Latency

Target latency is the total time for applying change data to the targets.

This total includes the time that PWXPC takes to extract change data from the change stream and the time that PowerCenter Integration Service takes to apply that change data to the targets. If extraction and apply processing occurs quickly, target latency is low.

The values for the commitment control attributes affect target latency. When you set the commitment control attributes, balance target latency requirements with resource consumption on the PowerCenter Integration Service machine and the target databases.

Lower target latency values result in higher resource use. The increased resource use occurs because the PowerCenter Integration Service must flush the change data more frequently. Also, the target databases must process more commit requests.

The following table describes the default values for the commitment control attributes, which provide the lowest latency:

Attribute	Default
Maximum Rows Per commit	0, which disables this attribute
Minimum Rows Per commit	0, which disables this attribute
Real-time Flush Latency in milli-seconds	0, which is equivalent to 2000 milliseconds or 2 seconds
UOW Count	1

These values decrease target latency because PWXPC commits changes after each UOW or on UOW boundaries. However, these values can have the following drawbacks:

- Highest resource consumption on the source system, PowerCenter Integration Service machine, and target databases
- Decreased throughput for the CDC sessions because PWXPC flushes change data too frequently for the PowerCenter Integration Service or target databases to handle this processing

To lower resource consumption and potentially increase throughput for CDC sessions, enter a value greater than the default value for one of the following attributes:

- **Minimum Rows Per commit**
- **UOW Count**
- **Real-time Flush Latency in milli-seconds**

Then disable the other attributes.

Commitment Control Attributes

PWXPC, in conjunction with PowerExchange and the PowerCenter Integration Service, controls the timing of commit processing for CDC sessions based on commitment control attributes on PWX CDC connections.

Commit processing is not controlled by a single commitment control attribute. When setting these attributes, try to balance performance and resource consumption with latency requirements.

The **Maximum Rows Per commit**, **Real-Time Flush Latency in milli-seconds**, and **UOW Count** attributes control the timing of real-time flushes of change data to the targets. The **Minimum Rows Per commit** attribute controls if a commit can occur.

Set one or more of the following commitment control attributes on PWX CDC connections:

Maximum Rows Per commit

Maximum number of change records in a source UOW that PWXPC processes before flushing the data buffer to commit the change data to the targets.

Use this attribute to have PWXPC commit change data to the targets without waiting for the UOW boundary, or end-UOW, to be met. This type of commit is called a *subpacket commit*. By using subpacket commits for large UOWs, you can minimize use of storage on the PowerCenter Integration Service machine and locking contention on the target databases.

Attention: Because PWXPC can commit the change data to the targets between UOW boundaries, relational integrity (RI) might be compromised. Do not use this connection attribute if you have targets in the CDC session with RI constraints.

After the maximum rows limit is met, PWXPC flushes the change data from the buffer on the PowerCenter Integration Service machine and commits the data to the targets. PWXPC also writes message PWXPC_12128 to the session log. After commit processing completes, the RDBMS releases locks on the target databases and PowerCenter Integration Service can reuse the buffer space for additional change records.

The maximum rows limit is cumulative across all sources in the CDC session. PWXPC issues a real-time flush when the limit is met, regardless of the number of sources with changes.

PWXPC resets the maximum rows limit when a real-time flush occurs. The flush can occur because of the maximum rows limit, UOW count limit, or real-time flush latency timer.

If PWXPC reaches a UOW boundary and the maximum row limit has not been met, PWXPC continues to process change records across UOW boundaries.

Use a maximum rows limit if you have extremely large UOWs in the change stream that might cause the following problems:

- Locking issues on the target database
- Resource issues on the PowerCenter Integration Service node

For example, you have a large UOW with 10,000 updates for a single source, and you set the **Maximum Rows per Commit** attribute to 1000. In this case, PWXPC issues a subpacket commit after each 1,000 change records.

Or, you might have a UOW that contains updates for more than one source. For example, the UOW contains 900 updates for source 1, 100 updates for source 2, and then 500 more updates for source 1. If you set the **Maximum Rows per Commit** attribute to 1000, PWXPC issues a subpacket commit after reading 1,000 change records, or after processing the updates for source 2.

Default is 0, which causes PWXPC to not use this maximum rows limit. If you specify 0 or do not enter a value for the maximum rows limit, commits occur only on UOW boundaries.

If you specify a low maximum rows limit, the CDC session uses more system resources on the PowerCenter Integration Service machine and target systems. This increased resource use occurs because PWXPC flushes data to the targets more frequently.

Note: The **Maximum Rows Per commit** attribute is a count of the records within a UOW. The **UOW Count** attribute is a count of complete UOWs.

Minimum Rows Per commit

Minimum number of change records that PowerExchange must pass to PWXPC before passing a commit record. Until the minimum rows limit is met, PowerExchange discards any commit records that it reads from the change stream and passes only change records to PWXPC. After the minimum rows limit is met, PowerExchange passes the next commit record it encounters to PWXPC and then resets the minimum rows counter.

If the change stream has many small UOWs, you can set the **Minimum Rows Per commit** attribute to create larger UOWs of a more uniform size. Online transactions that run in transaction control systems such as CICS and IMS often commit after only a few changes, which results in many, small UOWs in the

change stream. PowerExchange and PWXPC process a few large UOWs more efficiently than many small UOWs. By using the minimum rows limit to increase the size of UOWs, you can improve CDC processing efficiency.

The minimum rows limit does not impact the relational integrity of the change data because PowerExchange does not create additional commit points in the change data. PowerExchange skips some of the original commit records in the change stream.

Default is 0, which causes PowerExchange to not use this minimum rows limit.

If you enter a minimum rows limit, PowerExchange changes the number of change records in a UOW to match or exceed this limit.

Note: PWXPC does not commit change data to the targets based on the minimum rows limit. PWXPC commits change data to the targets based on the **Maximum Rows Per commit, Real-Time Flush Latency in milli-seconds**, and **UOW Count** attributes.

Real-Time Flush Latency in milli-seconds

For real-time or continuous extraction mode, the number of milliseconds that must elapse before PWXPC flushes the data buffer to commit change data to the targets. After the flush latency interval expires and PWXPC reaches a UOW boundary, PWXPC issues a real-time flush to commit change data and restart tokens to the targets. PWXPC also writes message PWXPC_10082 in the session log.

PWXPC resets the flush latency interval when a real-time flush occurs. The flush can occur because of the maximum rows limit, UOW count limit, or real-time flush latency timer.

Valid values for the real-time flush latency are:

- -1. Disables data flushes based on time.
- 0 through 2000. Sets the interval to 2000 milliseconds, or 2 seconds.
- 2000 through 86400. Sets the interval to the specified number of milliseconds.

Default is 0.

If you set the flush latency interval value is 0 or greater, PWXPC flushes the change data for all complete UOWs after the interval expires and the next UOW boundary occurs. The lower you set the flush latency interval, the faster PWXPC commits change data to the targets. If you require a low latency for applying changes to the targets, enter a low value for the flush latency interval.

However, if you specify a low flush latency interval, the CDC session might consume more system resources on the PowerCenter Integration Service and target systems. This increased consumption occurs because PWXPC commits change data to the targets more frequently.

UOW Count

Number of complete UOWs that PWXPC reads from the change stream before flushing change data to the targets. When PWXPC reads change data from PowerExchange and provides it to the source qualifier in the CDC session, the count of the UOWs begins.

After the UOW count limit is met, PWXPC issues a real-time flush to commit the change data and restart tokens to the targets. PWXPC also writes message PWXPC_10081 in the session log.

PWXPC resets the UOW count after a real-time flush occurs because of the UOW count limit or the real-time flush latency interval.

Valid values for UOW count are:

- -1 or 0. PWXPC does not use the **UOW Count** attribute to control commit processing.
- 1 through 999999999. PWXPC flushes change data after reading the specified number of UOWs.

Default is 1.

The lower you set the UOW count value, the faster the PowerCenter Integration Service commits change data to the target. If you require the lowest possible latency, enter a UOW count of 1. However, a low latency might result in the session using more system resources on the PowerCenter Integration Service and the target systems.

Attention: In the session properties, verify that the **Commit Type** attribute specifies **Source** and that the **Commit at End of File** attribute is disabled. The **Commit at End of File** attribute is enabled by default. If you accept the default, the PowerCenter Integration Service writes additional data to the targets after the CDC reader has committed the restart tokens and shut down. When you restart the CDC session, the session might write duplicate data to the targets.

Examples of Controlling Commit Processing

Review the following examples to learn how to use the commitment control attributes to control commit processing with PWXPC.

Example 1. Subpacket Commit and UOW Count

This example uses the **Maximum Rows Per commit** and **UOW Count** attributes to control commit processing.

The change data is composed of UOWs of the same size. Each UOW contains 1,000 change records.

The following table describes the commitment control attribute values that this example uses:

Attribute	Value
Maximum Rows Per commit	300
Minimum Rows Per commit	0, which disables this attribute
Real-time Flush Latency in milli-seconds	0, which is equivalent to 2 seconds
UOW Count	1

Based on the maximum rows value, PWXPC flushes the data buffer after reading the first 300 records in a UOW. This action commits the change data to the targets. PWXPC continues to commit change data to the targets every 300 records.

PWXPC commits on UOW boundaries only for the UOW count and real-time flush latency interval. If the real-time flush latency interval expires before PWXPC reads 300 change records, PWXPC still commits based on the maximum rows value because that threshold is met before a UOW boundary occurs.

When the end of the UOW is read, PWXPC commits the change data because the **UOW Count** value is 1. PWXPC resets the UOW and maximum row counters and the real-time flush latency timer each time it commits. Because all of the UOWs have the same number of change records, PWXPC continues to read change data and to commit the data to the targets at the same points in each UOW.

In this example, PWXPC commits change data at the following points:

- 300 change records based on the maximum rows value
- 600 change records based on the maximum rows value
- 900 change records based on the maximum rows value
- 1,000 change records based on the UOW count value

Example 2. UOW Count and Time-Based Commits

This example uses the **UOW Count** and **Real-time Flush Latency in milli-seconds** attributes to control commit processing. The change data consists of UOWs of varying sizes.

The following table describes the commitment control attribute values that this example uses:

Attribute	Value
Maximum Rows Per commit	0, which disables this attribute
Minimum Rows Per commit	0, which disables this attribute
Real-time Flush Latency in milli-seconds	5000, which is equivalent to 5 seconds
UOW Count	1000

Initially, PWXPC reads 900 complete UOWs in 5 seconds. Because the real-time flush latency interval has expired, PWXPC flushes the data buffer to commit the change data to the targets. PWXPC then resets both the UOW counter and real-time flush latency timer. When PWXPC reaches UOW 1,000, PWXPC does not commit change data to the targets because the UOW counter was reset to 0 after the last commit.

PWXPC reads the next 1,000 UOWs in 4 seconds, which is less than the real-time flush latency timer. PWXPC commits this change data to the target because the UOW counter has been met. After this commit, PWXPC then resets the real-time flush latency timer and the UOW counter.

PWXPC continues to read change data and commit the data to the targets, based on the UOW count or the real-time flush latency flush time, whichever limit is met first.

In this example, PWXPC commits change data at the following points:

- After UOW 900 because the real-time latency flush latency timer matched first.
- After UOW 1,900 because the UOW count matched first during the second commit cycle.

Example 3. Minimum Rows and UOW Count

This example uses the **Minimum Rows Per commit** and **UOW Count** attributes to control commit processing.

The change data consists of UOWs of the same size. Each UOW contains ten change records.

The following table describes the commitment control attribute values that this example uses:

Attribute	Value
Maximum Rows Per commit	0, which disables this attribute
Minimum Rows Per commit	100
Real-time Flush Latency in milli-seconds	-1, which disables this attribute
UOW Count	10

PWXPC passes the minimum rows value to PowerExchange and requests change data from the change stream. Because the minimum rows value is 100, PowerExchange skips the commit records of the first nine UOWs. When PowerExchange reads the last change record in the tenth UOW, the minimum rows limit is met. So, PowerExchange passes the commit record for the tenth UOW to PWXPC and resets the minimum rows counter. PWXPC increases the UOW counter to one.

PowerExchange and PWXPC continue to read the change data until the UOW counter is 10. At this point, PWXPC flushes the data buffer to commit the change data to the targets and resets the UOW counter.

PWXPC commits change data after 1,000 change records, or after every 10 UOWs, because each UOW contains 10 change records and the **UOW Count** is 10.

Recovery and Restart Processing for CDC Sessions

If you select the **Resume from last checkpoint** option for the **Recovery Strategy** attribute in a CDC session that extracts change data, PWXPC and PowerCenter provide recovery and restart processing for that session.

If a session fails, the PowerCenter Integration Service recovers the session state of operation, and PWXPC recovers the restart information.

PWXPC saves restart information for all sources that are in a CDC session. The restart information for CDC sessions, including the restart tokens, originates from PowerExchange on the system from which the change data is extracted. You can include both relational and nonrelational targets in a single CDC session. PWXPC uses one of the following locations to store and retrieve restart information, based on the target type:

- For relational targets, PWXPC uses recovery state tables in the target databases. PWXPC, in conjunction with the PowerCenter Integration Service, commits both the change data and the restart tokens for that data in the same commit operation. This commit ensures that the applied data and the restart tokens are in-sync.
- For nonrelational targets, PWXPC uses the recovery state file that is in the shared location on the PowerCenter Integration Service machine. PWXPC, in conjunction with the PowerCenter Integration Service, writes the change data to the target files and then writes the restart tokens to the recovery state file. As a result, duplicate data might be applied to the targets when you restart the failed CDC sessions.

The PowerCenter Integration Service saves the session state of operation and maintains target recovery tables. The PowerCenter Integration Service stores the session state of operation in the shared location that is specified in \$PMStorageDir. The PowerCenter Integration Service saves relational target recovery information in the target database.

When you run a CDC session that uses a resume recovery strategy, PWXPC writes the following message to the session log to indicate that recovery is in effect:

```
PWXPC_12094 [INFO] [CDCRestart] Advanced GMD recovery in effect. Recovery is automatic.
```

When you recover or restart a CDC session, PWXPC uses the saved restart information to resume reading the change data from the point of interruption. The PowerCenter Integration Service restores the session state of operation, including the state of each source, target, and transformation. PWXPC, in conjunction with the PowerCenter Integration Service, determines how much of the source data it needs to reprocess.

PowerExchange and PWXPC use the restart information to determine the correct point in the change stream from which to restart extracting change data and then applying it to the targets.

If you run a session with resume recovery strategy and the session fails, do not change the mapping, the session, or the state information before you restart the session. PowerCenter and PWXPC cannot guarantee recovery if you make any of these changes.

Restriction: If any of the targets in the CDC session use the PowerCenter File Writer to write CDC data to flat files, do not use a resume recovery strategy. Restart tokens for all targets in the CDC session, including relational targets, will be compromised if a flat file target is in the same session. Data loss or duplication might occur.

RELATED TOPICS:

- [“PowerCenter Recovery Tables for Relational Targets” on page 112](#)
- [“PowerCenter Recovery Files for Nonrelational Targets” on page 113](#)
- [“Restart Processing for CDC Sessions by Start Type” on page 114](#)

PowerCenter Recovery Tables for Relational Targets

When the PowerCenter Integration Service runs a CDC session that has a resume recovery strategy, it writes information to recovery tables on the target database system.

When the PowerCenter Integration Service recovers the session, it uses the information in the recovery tables to determine where to begin loading data to target tables. PWXPC also uses information in the recovery tables to determine where to begin reading the change stream.

If you want the PowerCenter Integration Service to create the recovery tables, grant table creation privileges to the database user name that is configured in the target database connection. Otherwise, you must create the recovery tables manually.

For relational targets, the PowerCenter Integration Service creates the following recovery tables in the target database:

PM_RECOVERY

Contains target load information for the session run. The PowerCenter Integration Service removes the information from this table after each successful session and initializes the information at the beginning of subsequent sessions.

PM_TGT_RUN_ID

Contains information the PowerCenter Integration Service uses to identify each target on the database. The information remains in the table between session runs. If you manually create this table, you must create a row and enter a value other than zero for LAST_TGT_RUN_ID to ensure that the session recovers successfully.

PM_REC_STATE

Contains state and restart information for CDC sessions. PWXPC stores the application name and restart information for all sources in the CDC session. The PowerCenter Integration Service stores any state information for the session. Unlike the session state information, restart information persists in this table across successful sessions. The PowerCenter Integration Service updates it with each commit to the target tables.

If you edit or drop the recovery tables before you recover a session, the PowerCenter Integration Service cannot recover the session. Also, PWXPC cannot restart the CDC session from the point of interruption.

If you disable recovery, the PowerCenter Integration Service does not remove the recovery information from the target database. Also, PWXPC no longer updates the restart information in the target database.

Recovery State Table

The recovery state table, PM_REC_STATE, contains state and CDC restart information for a CDC session. This table resides in the same target database as the target tables.

The PowerCenter Integration Service creates an entry in the state table for each CDC session. These entries can comprise more than one row. CDC sessions with heterogeneous target tables have state table entries in each unique relational target database and an entry in a state file on the PowerCenter Integration Service machine for each nonrelational target. For example, a CDC session that targets Oracle and SQL Server tables

and a MQ Series queue has an entry in the state table in the target Oracle database, in the state table in the target SQL Server database, and in the state file on the PowerCenter Integration Service machine.

Each session entry in a state table contains a number of repository identifiers and execution state data such as the checkpoint number and CDC restart information. The following columns can contain PWXPC-specific restart information:

APPL_ID

Contains the value the PWXPC creates by appending the task instance ID of the CDC session to the value that you specify in the **Application Name** attribute in the source PWX CDC application connection. When this value matches an APPL_ID value for a row in the state table, the PowerCenter Integration Service, in conjunction with PWXPC, selects the row from the state table for the CDC session.

STATE_DATA

Contains the restart information for the session in a variable-length, 1,024-byte binary column. When the PowerCenter Integration Service commits change data to the targets tables, it also commits the restart information for that data in this column. PWXPC uses the restart information from this column to perform restart processing for the CDC session.

If the amount of restart information for a session exceeds 1,024 bytes, the PowerCenter Integration Service adds additional rows to accommodate the remainder of the restart information. For each row added, the PowerCenter Integration Service increases the value of the SEQ_NUM column by one, starting from zero.

PowerCenter Recovery Files for Nonrelational Targets

If you configure a resume recovery strategy for a CDC session, the PowerCenter Integration Service stores the session state of operation in the shared location, \$PMStorageDir, on the PowerCenter Integration Service machine. For nonrelational targets, the PowerCenter Integration Service also stores the target recovery status in a recovery state file in the shared location on the PowerCenter Integration Service machine. PWXPC stores the restart information for nonrelational target files in this state file.

Recovery State File

For all nonrelational targets in a CDC session, the PowerCenter Integration Service uses a recovery state file on the PowerCenter Integration Service machine.

Nonrelational target files include MQ Series message queues, PowerExchange nonrelational targets, and other PowerCenter nonrelational targets.

CDC sessions with heterogeneous target tables have state table entries in each unique relational target database and an entry in a state file on the PowerCenter Integration Service machine for each nonrelational target.

The PowerCenter Integration Service creates the recovery state file in the shared location, \$PMStorageDir. The file name has the following prefix:

```
pm_rec_state_appl_id
```

PWXPC creates the value for the *appl_id* variable in the file name by appending the task instance ID of the CDC session to the value that you specify in the **Application Name** attribute in the source PWX CDC application connection. The PowerCenter Integration Service uses various task and workflow repository attributes to complete the file name. The message CMN_65003, which the PowerCenter Integration Service writes to the session log, contains the complete file name.

Application Names

PWXPC, in conjunction with the PowerCenter Integration Service, uses the application name you specify as part of the key when it stores and retrieves the restart information for a CDC session.

When you configure the PWX CDC application connection for a CDC session, enter a unique value for the **Application Name** attribute. PWXPC appends the repository task instance ID for the CDC session to this value to create the APPL_ID value in the recovery state table and the *appl_id* portion of the recovery state file name.

Because the value of the APPL_ID column and the state recovery file contains the task instance ID for the CDC session, changes to the session can affect restart processing. If you add or remove sources or targets for a CDC session, you must use the restart token file to provide restart tokens and then cold start the session.

Restart Processing for CDC Sessions by Start Type

How you start a CDC session affects how PWXPC determines the restart points for sources in the session. Each source has its own restart point.

For each start type, PWXPC determines the restart point as follows:

- For a cold start, PWXPC uses the restart token file to acquire restart tokens for all data sources. PWXPC does not read the state tables or state file and does not attempt to recover the session. The CDC session continues to run until it is stopped or interrupted.
- For a warm start, PWXPC reconciles the restart tokens that are in the restart token file with the restart tokens in the state tables and state file. If necessary, PWXPC performs recovery processing. The session continues to run until it is stopped or interrupted.
- For a recover start, PWXPC reads the restart tokens from any applicable state tables and state file. If necessary, PWXPC performs recovery processing. PWXPC updates the restart token file with the restart tokens for each source in the CDC session, and then the session ends.

Before you run a CDC session for the first time, create and populate the restart token file with restart token pair for each source in the session. Each restart token pair should match a point in the change stream where the source and target are in a consistent state.

For example, materialize a target table and stop update activity on the source. To define a start or restart point, specify a special override statement that contains the CURRENT_RESTART option in the restart token file. Use the restart token file that has the file name that matches the restart token file name in the PWX CDC application connection. When you cold start the CDC session, PWXPC requests that PowerExchange use the current end-of-log as the extraction start point. You can then resume update activity on the sources.

If you cold start a CDC session and a restart token file does not exist, the PowerCenter Integration Service runs the session. PWXPC passes Null restart tokens for all sources to PowerExchange. PowerExchange issues message PWXPC_12060 to indicate that the restart tokens for each source are Null and then assigns the default restart point to each source.

Attention: If you use Null restart tokens, the CDC session might have incorrect results. Provide valid restart tokens when you cold start CDC sessions.

Default Restart Points for Null Restart Tokens

If PowerExchange receives null restart tokens for all sources in a CDC session, it uses the default restart points.

For DB2 for i5/OS sources, the default restart points vary by extraction mode, as follows:

- For batch extraction mode and continuous extraction mode, the default restart point is the oldest condense file that is recorded in the CDCT file.
- For real-time extraction mode, the default restart point is the oldest journal receiver that is still attached on the journal receiver chain.

PowerExchange uses the default restart point only if all sources in a CDC session have null restart tokens. If some sources have non-null restart tokens, PWXPC assigns the oldest restart point from those tokens to any sources for which no restart tokens are specified.

For example, a new CDC session contains the sources A, B, and C. The restart token file contains restart tokens for sources A and B. The restart point for source A is older than that for source B. Source C does not have existing or supplied restart tokens. Because some sources in the CDC session have explicit restart points, PWXPC does not assign null restart tokens to source C. Instead, PWXPC assigns the restart point for source A to source C because this restart point is the oldest one supplied.

Determining the Restart Tokens for Cold Start Processing

When you cold start a CDC session, PWXPC uses the restart token file to determine the restart tokens for all sources. PWXPC ignores any entries in the state tables or state file for the sources in the CDC session.

More specifically, PWXPC uses one of the following methods to determine the restart tokens:

- If the restart token file is empty or does not exist, PWXPC assigns null restart tokens to all sources in the CDC session.
- If the restart token file contains only explicit override statements, PWXPC performs the following processing:
 - Assigns the restart tokens in the explicit override statements to the specified sources.
 - Assigns the oldest supplied restart point to any sources for which an explicit override statement was not specified.
- If the restart token file contains only the special override statement, PWXPC assigns the restart tokens in the special override statement to all sources.
- If the restart token file contains a special override statement and explicit override statements, PWXPC performs the following processing:
 - Assigns the restart tokens in the explicit override statements to the specified sources.
 - Assigns the restart tokens in the special override statement to all remaining sources.

Determining the Restart Tokens for Warm Start Processing

When you warm start a CDC session, PWXPC uses the state tables and state file, in conjunction with restart token file, to determine the restart tokens for all sources.

More specifically, PWXPC uses one of the following methods to determine the restart tokens:

- If the restart token file is empty or does not exist and there is no matching entry in a state table or state file, PWXPC assigns null restart tokens to all sources in the session.

- If the restart token file is empty or does not exist and if some but not all sources have a matching entry in a state table or a state file, PWXPC performs the following processing:
 - Assigns any restart tokens found in a state table and state file to the appropriate sources.
 - Assigns the oldest available restart point to all sources that do not have restart tokens.
- If the restart token file is empty or does not exist and if all sources have an entry in a state table or state file, PWXPC uses the restart tokens from the state tables or state file.
- If the restart token file contains explicit override statements and no sources have a matching entry in a state table or no state file, PWXPC performs the following processing:
 - Assigns the restart tokens in the explicit override statements to the specified sources.
 - Assigns the oldest supplied restart point to all sources that do not have restart tokens.
- If the restart token file contains explicit override statements and if some but not all sources have a matching entry in a state table or a state file, PWXPC performs the following processing:
 - Assigns the restart tokens in the explicit override statements to the specified sources.
 - Assigns restart tokens from a state table or state file to the appropriate sources, provided that the tokens have not been supplied in the restart token file.
 - Assigns the oldest available restart point to all sources that do not have restart tokens supplied in the restart token file or from a state table or state file.
- If the restart token file contains explicit override statements and if all sources have an entry in a state table or a state file, PWXPC performs the following processing:
 - Assigns the restart tokens in the explicit override statements to the specified sources.
 - Assigns the restart tokens from state tables or the state file to all remaining sources that do not have restart tokens supplied in the restart token file.
- If the restart token file contains only the special override statement, PWXPC assigns the restart tokens in the special override statement to all sources.
- If the restart token file contains a special override statement and explicit override statements, PWXPC performs the following processing:
 - Assigns the restart tokens in the explicit override statements to the specified sources.
 - Assigns the restart tokens in the special override statement to all remaining sources.

Creating Restart Tokens for Extractions

Before you begin extracting change data, you must create restart tokens to indicate the extraction start point.

The optimal start point matches the point in the change stream at which you last synchronized the source and target. This point marks the end of the change stream, or current end-of-log (EOL), if you stop update activity on the source, as recommended, until after target materialization and restart token generation are complete.

To create current restart tokens for the current EOL, use one the following methods:

PWXPC restart token file

To generate current restart tokens for a CDC session that uses real-time or continuous extraction mode, specify the `CURRENT_RESTART` option on the `RESTART1` and `RESTART2` special override statements in the PWXPC restart token file. When the CDC session runs, PWXPC requests that PowerExchange provide

restart tokens for the current EOL. PWXPC uses this restart information to locate the extraction start point.

Database row test

In the PowerExchange Navigator, perform a database row test with a `SELECT CURRENT_RESTART` SQL statement.

DTLUAPPL utility

Run the DTLUAPPL utility with the `GENERATE RSTKKN` option.

If you use the DTLUAPPL utility or PowerExchange Navigator to generate restart tokens, enter the token values in the PWXPC restart token file before you start the CDC session.

Displaying Restart Tokens

You can display restart token values in the output from a database row test, extraction session, or DTLUAPPL PRINT function.

If you run a database row test on an extraction map from the PowerExchange Navigator, the output includes a restart token pair for each row of change data. The following columns show the token values:

- `DTL__CAPXRESTART1` shows the sequence token value.
- `DTL__CAPXRESTART2` shows the restart token value.

If you include the `DTL__CAPXRESTART1` and `DTL__CAPXRESTART2` columns in the PowerCenter source definition, PowerExchange provides the restart tokens for each row when you extract change data in a CDC session.

When a CDC session runs, PowerExchange and PWXPC display restart token values in the following messages:

- In messages `PWX-04565` and `PWX-09959`, the sequence token is in the Sequence field and restart token is in the PowerExchange Logger field.
- In messages `PWXPC_12060` and `PWXPC_12068`, the sequence token is in the Restart Token 1 field and the restart token is in the Restart Token 2 field.
- In messages `PWXPC_10081`, `PWXPC_10082`, and `PWXPC_12128`, the sequence token is the first token value and the restart token is the second token value.

If you use the DTLUAPPL utility to generate restart tokens, you can use the `PRINT` statement to display the generated values. In the `PRINT` output, DTLUAPPL displays the sequence token, without the usual trailing eight zeros, in the Sequence field and displays the restart token in the Restart field.

Configuring the Restart Token File

When you configure the CDC session in PowerCenter, specify the name and location of the restart token file.

To specify the restart token file, enter the following attributes on the PWX CDC application connection for the source:

RestartToken File Folder

Enter the name of the directory that contains the restart token file. If you use the default value of `$PMRootDir/Restart` and the `Restart` directory does not exist, PWXPC creates the directory. PWXPC does not create any restart token directory under another name.

RestartToken File Name

Enter a unique name for the restart token file. If you do not specify this value, PWXPC uses the value in the **Application Name** attribute, if present. Otherwise, PWXPC uses the workflow name. Because this name must be unique, Informatica recommends that you always specify a value for the **RestartToken File Name** attribute.

When you run a CDC session, PWXPC verifies that the restart token file exists. If one does not exist, PWXPC uses the name specified in this attribute to create an empty restart token file.

Restriction: The value of **RestartToken File Name** attribute in must be unique for every CDC session. Non-unique file names can cause unpredictable results, such as change data loss and session failures.

To find the restart token file name for a CDC session, use the following methods:

- For CDC sessions that have run, look for message `PWXPC_12057` in the session log. This message indicates the restart token file directory and file name.
- In Workflow Manager, look for the restart token file folder and file name in the attributes on the PWX CDC application connection associated with the source in the CDC session. If the restart token file name is not present, PWXPC uses the application name, if specified. Otherwise, PWXPC uses the workflow name.

Before you run a CDC session the first time, configure the restart token file to indicate the point in the change stream from which to start extracting change data. Later, you might need to modify the restart token file to add sources to a CDC session or to indicate the point from which to restart change data extraction.

Restart Token File Statements

You can optionally specify explicit override statements, a special override statement, and comments in the restart token file.

These statements have the following uses:

- **Explicit override statements.** Specifies a restart token pair or the `CURRENT_RESTART` option for a specific source. If you use the PowerExchange Logger for Linux, UNIX, and Windows, you can use an explicit override statement to define a restart point based on a date and time in the Logger logged data. In all cases, you must provide the PowerExchange extraction map name for the source object. You can define multiple explicit override statements in the restart token file, each for a different source.
- **Special override statement.** Specifies a restart token pair or the `CURRENT_RESTART` option for all sources in a CDC session. You can provide a specific restart token pair or request that PowerExchange use the current restart point. You can define only one special override statement in the restart token file. You can also define explicit override statements in the same file to specify source-specific restart points.
- **Comments.** Specifies any comments that you want to add to the restart token file.

General Syntax Rules and Guidelines

When defining explicit override statements, special override statements, and comments in a restart token file, use the following rules and guidelines:

- Statements can begin in any column.
- All statements are optional.
- Do not include blank lines between statements.

- Comment lines must begin with <!--.
- In a restart token file, you can specify one or more explicit override statements and only one special override statement.
- Explicit override statements for a source take precedence over the special override statement, if defined.
- On warm start, explicit override statements and the special override statement take precedence over any restart token values stored for sources in the start table or file.

Explicit Override Statements

Use explicit override statements to specify an extraction restart point for a specific source in a CDC session. You can specify multiple explicit override statements, each for a different source.

When you warm start a CDC session, the explicit override statement for a source object overrides the restart tokens in the state table or state file for that source. You can use explicit override statements in conjunction with special override statements to provide override restart tokens for all sources in a CDC session.

An explicit override statement for a source can specify either a pair to restart tokens that define a specific point in the change stream or the CURRENT_RESTART option for the current end of the change stream. Alternatively, if you use the PowerExchange Logger for Linux, UNIX, and Windows with a CDC application connection, you can optionally define an explicit override statement that specifies a time-based restart point for extraction processing in the Logger log files.

To specify restart tokens for a source, enter a pair of statements, each containing the extraction map name and a sequence token (restart1) or restart token (restart2) value. Use the following syntax:

```
extractionMapName=restart1_token
extractionMapName=restart2_token
```

Note: Because a source can have multiple extraction maps with distinct names, you might need to define multiple pairs of explicit override statements for a source.

For z/OS sources, if you use the default value of Y for the FUZZYSTART parameter in the LRAP CAPI_CONNECTION statement, you can enter a restart2 position other than a begin-uow position from which to start extraction processing in the PowerExchange Logger for z/OS logs. In the following example, the *restart2_token* value points to a RBA position that does not coincide with a begin-uow record and the *restart1_token* value points to a position before the *restart2_token* position:

- For the *restart1_token* value, specify 00000000000100. This value consists of 48 zeroes with a "1" in the twelfth position. With this value, the *restart2_token* value will determine the records to be returned by the extraction.
- For the *restart2_token* value, specify a value that consists of a 6-byte Logger ID in EBCDIC format (padded with spaces) + a 6-byte RBA position + 8 zeroes. The RBA position can be any RBA in the logs near where you want to restart extraction processing. For example: E2C2F2D34040000000AEF4000000000

With these restart token settings, PowerExchange begins extraction processing at the first record that is at or past the *restart1_token* position and in the first UOW that is returned based on the *restart2_token* position.

To specify the current end of the change stream as the restart point, use the following syntax:

```
extractionMapName=CURRENT_RESTART
```

To specify a time-based restart point in the PowerExchange Logger logged data, enter a pair of statements, one containing the RESTART_TIME option and the other containing a date and time value. Use the following syntax:

```
extractionMapName=RESTART_TIME
extractionMapName=datetime
```

Parameter descriptions:

extractionMapName

The name of an extraction map for the data source. To determine the extraction map name, use one of the following methods:

- For CDC data map sources, see the **Schema Name Override** and **Map Name Override** attributes in the session properties. These attributes override the schema name and map name in the source extraction map. Or, in Designer, see the **Schema Name** and **Map Name** values in the source Metadata Extensions.
- For relational sources, see the **Extraction Map Name** attribute in the session properties.

Note: After the extraction map has been used to extract change data, the table name is appended to this value in the format *extractionMapName_tableName*. Use the full name when defining an explicit override statement.

restart1_token

The sequence token portion of a restart token pair. This value varies based on data source type.

restart2_token

The restart token portion of a restart token pair. This value based on data source type.

CURRENT_RESTART

The option that generates a pair of restart tokens that mark the current end of the change stream. The PWXPC CDC reader opens a separate connection to PowerExchange, requests the generation of current restart tokens, and then provides the token values to the applicable source.

Optionally, you can generate current restart tokens in the **Database Row Test** dialog box of the PowerExchange Navigator.

Restriction: Use the CURRENT_RESTART option only for CDC sessions that use real-time extraction mode or continuous extraction mode.

RESTART_TIME

If you use the PowerExchange Logger for Linux, UNIX, and Windows log files, this option allows you to specify a date and time value as the restart point for extraction processing in the Logger log files.

datetime

The date and time to use as the restart point in PowerExchange Logger for Linux, UNIX, and Windows log files. This value must be in the format YYYYMMDDhhmmssuuuuuu, where YYYY is a four-digit year, MM is the month, DD is the day of the month, hh is hours, mm is minutes, ss is seconds, and uuuuuu is microseconds.

Special Override Statement

Use a special override statement to specify a pair of restart tokens or the CURRENT_RESTART option as the restart point for a CDC session.

When you warm start the CDC session, the special override statement overrides the restart tokens in the state table or state file for the session. You can use the special override statement in conjunction with explicit override statements, which pertain to specific sources. In this case, the special override statement sets a restart point for all sources in the CDC session except those for which an explicit override statement is defined.

A special override statement is composed of a pair of RESTART1 and RESTART2 statements. Use the following syntax:

```
RESTART1={restart1_token|CURRENT_RESTART}  
RESTART2={restart2_token|CURRENT_RESTART}
```


You can specify only one pair of these statements in the restart token file.

For z/OS sources, if you use the default value of Y for the FUZZYRSTART parameter in the LRAP CAPI_CONNECTION statement, you can enter a *restart2_token* position other than a begin-uow position from which to start extraction processing in the PowerExchange Logger for z/OS logs. In the following example, the *restart2_token* value points to a RBA position that does not coincide with a begin-uow record and the *restart1_token* value points to a position before the *restart2_token* position:

- For the *restart1_token* value, specify 000000000000100. This value consists of 48 zeroes with a "1" in the twelfth position. With this value, the *restart2_token* value will determine the records to be returned by the extraction.
- For the *restart2_token* value, specify a value that consists of a 6-byte Logger ID in EBCDIC format (padded with spaces) + a 6-byte RBA position + 8 zeroes. The RBA position can be any RBA in the logs near where you want to restart extraction processing. For example: E2C2F2D340400000000AEF4000000000

With these restart token settings, PowerExchange begins extraction processing at the first record that is at or past the *restart1_token* position and in the first UOW that is returned based on the *restart2_token* position.

Parameter descriptions:

restart1_token

The sequence token portion of a restart token pair. This value varies based on data source type.

restart2_token

The restart token portion of a restart token pair. This value varies based on data source type.

CURRENT_RESTART

The option for generating a pair of restart tokens that mark the current end of the change stream. The PWXPC CDC reader opens a separate connection to PowerExchange, requests the generation of current restart tokens, and then provides the token values to all applicable sources.

Optionally, you can generate current restart tokens in the **Database Row Test** dialog box of the PowerExchange Navigator.

Restriction: Use the CURRENT_RESTART option only for CDC sessions that use real-time extraction mode or continuous extraction mode.

Comment Statement

You can use the comment statement anywhere in the restart token file. Comment statements must begin with <!--.

For example:

```
<!-- my comments
```

Example Restart Token File

This example restart token file is for a CDC session with seven source tables. The file includes explicit override statements that provide restart tokens for three source tables and a special override statement that provides restart tokens for the remaining tables.

The restart token file contains the following statements:

```
<!-- Restart Tokens for existing tables -->
Restart1=000000AD775600000000000000AD77560000000000000000
Restart2=C1E4E2D34040000000AD5F2C00000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_001 -->
dldsn9.rrtb0001_RRTB_SRC_001=0000060D1DB2000000000000060D1DB20000000000000000
dldsn9.rrtb0001_RRTB_SRC_001=C1E4E2D340400000013FF362000000000
```


CHAPTER 8

Monitoring CDC Sessions

This chapter includes the following topics:

- [Monitoring Overview, 123](#)
- [Monitoring CDC Sessions in PowerExchange, 123](#)
- [Monitoring CDC Sessions in PowerCenter, 128](#)

Monitoring Overview

PowerExchange, PWXPC, and PowerCenter issue messages that you can use to monitor the progress of CDC sessions.

PWXPC can also display progress and statistical information about CDC sessions in PowerCenter Workflow Monitor.

RELATED TOPICS:

- [“Monitoring CDC Sessions in PowerExchange” on page 123](#)
- [“Monitoring CDC Sessions in PowerCenter” on page 128](#)

Monitoring CDC Sessions in PowerExchange

You can use certain PowerExchange messages and commands to monitor the extraction of change data by CDC sessions.

Use the following types of PowerExchange messages and output for monitoring extractions:

- Read progress messages. You can request that PowerExchange write messages that indicate the number of change records read by a CDC session.
- Extraction statistics messages. When extraction sessions end, PowerExchange writes messages that include statistical information about the change records processed.
- Multithreaded processing statistics messages. You can request that PowerExchange write statistical information about CDC sessions that use multithreaded processing.
- DISPLAY ACTIVE or LISTTASK command. Use one of these PowerExchange Listener commands, based on your operating system and mode of command execution, to list active CDC sessions. For more information about these commands, see the *PowerExchange Command Reference*.

Read Progress Messages

You can request that PowerExchange write read progress messages to the PowerExchange message log file. These messages indicate the number of change records read for a CDC session.

If you select the **Retrieve PWX log entries** option on the PWX CDC application connection, PWXPC also writes these messages in the session log.

To have PowerExchange write read progress messages, include the following statements in the DBMOVER configuration file:

PRGIND=Y

Enter Y to have PowerExchange write PWX-04587 messages to the PowerExchange message log file. These messages indicate the number of records read for a CDC session. Default is N.

PRGINT=records

Enter the number of records that PowerExchange must read before writing PWX-04587 messages to the PowerExchange message log file. Default is 250 records.

For example, to have PowerExchange write read progress messages after reading 100 records, specify the following statements:

```
PRGIND=Y
PRGINT=100
```

PWX-04587 messages have the following format:

```
PWX-04587 int_server/workflow_name/session_name: Records read=records
```

Where:

- *int_server* is the name of the PowerCenter Integration Service.
- *workflow_name* is the name of the workflow that contains the CDC session.
- *session_name* is the name of the CDC session.
- *records* is the cumulative number of records read since the CDC session started.

For example, for a CDC session named *s_cdc_DB2_SQL_stats* runs, PowerExchange writes the following messages:

```
PWX-04587 intserv/wf_cdc_mon_stats/s_cdc_DB2_SQL_stats: Records read=100
PWX-04587 intserv/wf_cdc_mon_stats/s_cdc_DB2_SQL_stats: Records read=200
PWX-04587 intserv/wf_cdc_mon_stats/s_cdc_DB2_SQL_stats: Records read=300
```

PowerExchange continues to write PWX-04587 messages for this CDC session until the session ends. In the PowerExchange message log file, each of these messages has a date and time stamp. Use this information to determine the speed with which PowerExchange processes change data from the change stream.

Extraction Statistics Messages

When a CDC session ends, PowerExchange writes messages that contain statistical information about extraction processing for the session.

These messages are:

- PWX-04578. PowerExchange writes this message for each source in the CDC session. The message includes the number of Insert, Update, Delete, Commit, and total records read for the source.
- PWX-04588. PowerExchange writes this message for the entire CDC session. This message includes the total number of records read for the session.

Important: The statistical information in the PowerExchange messages represents the change data that PowerExchange read for a CDC session. This information might not reflect the data that was applied to the targets. For statistical information about the change data applied to a target, review the session log.

Multithreaded Processing Statistics

If you use multithreaded processing, you can configure PowerExchange to issue messages that contain statistics on multithreaded extraction processing.

To issue these messages, you must specify the `SHOW_THREAD_PERF` statement in the `DBMOVER` configuration file on the PowerCenter Integration Service machine:

```
SHOW_THREAD_PERF=number_of_records
```

This statement specifies the number of records PowerExchange must process before writing statistics messages about multithreaded extraction processing to the PowerExchange message log file. For more information about this statement, see the *PowerExchange Reference Manual*.

If you select the **Retrieve PWX log entries** attribute on the application connection for the CDC session, PWXPC writes these messages in the session log. Also, you must specify 1 or greater for the **Worker Threads** attribute on the application connection to implement multithreaded processing so that statistics can be generated.

PowerExchange writes the following messages during each statistics interval:

- PWX-31255. Cycle time, which is the total time that PowerExchange on the PowerCenter Integration Service machine spent processing the change data before passing it to PWXPC. This message includes the total percentage of time and average, minimum, and maximum times in microseconds.
- PWX-31256. I/O time, which is the time that PowerExchange on the PowerCenter Integration Service machine spent reading change data from the PowerExchange Listener on the source system. This message includes the I/O percentage of the total time and average, minimum, and maximum times in microseconds.
- PWX-31257. Parsing time, which is the time that PowerExchange on the PowerCenter Integration Service machine spent in column-level processing for the change records on all threads. This message includes the parsing percentage of the total time and average, minimum, and maximum times in microseconds.
- PWX-31258. External time, which is the time that PowerExchange on the PowerCenter Integration Service machine spent combining the change records from all threads back into a single UOW to pass to PWXPC and for PWXPC to flush the data to PowerCenter. This message includes the external percentage of the total time and average, minimum, and maximum times in microseconds.
- PWX-31259. Delay time, which is the time that the PowerExchange on the PowerCenter Integration Service machine waited to receive new change records to process from the PowerExchange Listener on the source system. This message includes the delay percentage of the total time and average, minimum, and maximum times in microseconds.

For example, if you specify `SHOW_THREAD_PERF=10000`, PowerExchange writes the following messages after reading 10,000 change records and reaching the next UOW boundary:

```
PWX-31254 PowerExchange threading stats for last 10000 rows. Cycle (array) size is 25
rows. 0 out of array occurred.
PWX-31255 Cycle time: 100% (avg:      5709 min:      4741 max:      7996 usecs)
PWX-31256 IO time: 4% (avg:         235 min:         51 max:      1021 usecs)
PWX-31257 Parse time: 79% (avg:     4551 min:     4102 max:     5495 usecs)
PWX-31258 Extern time: 20% (avg:     1145 min:      618 max:     3287 usecs)
PWX-31259 Delay time: 0% (avg:         7 min:         4 max:         165 usecs)
PWX-31254 PowerExchange threading stats for last 100000 rows. Cycle (array) size is 25
rows. 0 out of array occurred.
PWX-31255 Cycle time: 99% (avg:      5706 min:      4735 max:      7790 usecs)
PWX-31256 IO time: 4% (avg:         234 min:         51 max:         950 usecs)
```

```

PWX-31257 Parse time: 79% (avg: 4549 min: 4108 max: 5425 usecs)
PWX-31258 Extern time: 20% (avg: 1144 min: 616 max: 3242 usecs)
PWX-31259 Delay time: 0% (avg: 7 min: 4 max: 115 usecs)

```

If the parsing and external processing times are greater than the I/O time, you can increase the number of threads for the CDC session to try to improve throughput.

PowerExchange Listener DISPLAY ACTIVE or LISTTASK Command

Run the PowerExchange Listener DISPLAY ACTIVE or LISTTASK command to display CDC sessions that are active in the PowerExchange Listener.

The specific command name and syntax depends on how you issue the command, as follows:

- Issue the DISPLAY ACTIVE command from the command line on the system where the PowerExchange Listener runs. For more information, see the *PowerExchange Command Reference*.
- Use the pwxcmd program to issue the listtask command to a PowerExchange Listener that runs on the local system or a remote system. For more information, see the *PowerExchange Command Reference*.
- In the PowerExchange Navigator, issue the LISTTASK command from the **Database Row Test** dialog box. For more information, see the *PowerExchange Navigator User Guide*.
- If you run the PowerExchange Listener as an application service in the Informatica domain, run the infacmd pwx program to issue the ListTaskListener command. For more information, see the *Informatica Command Reference*.

In the command output, the **PwrCtrSess** field displays the PowerCenter session name in the following format:

```
integration_server_name/workflow_name/session_name
```

For example, when two CDC sessions are active, the DISPLAY ACTIVE or LISTTASK command produces the following output:

```

PWX-00711 Active tasks:
PWX-00712 TaskId=1, Partner=10.10.10.01, Port=2480, PwrCtrSess=intserv1/workflow1/cdc_sess1,
Application=appl_name1, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=
PWX-00729 Userid=userid, Client=clientid, File=capture_registration_file_name, Table=table_name,
DB=database_name
PWX-00712 TaskId=2, Partner=10.10.10.02, Port=2480, PwrCtrSess=intserv2/workflow2/cdc_sess2,
Application=appl_name2, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=
PWX-00729 Userid=userid, Client=clientid, File=capture_registration_file_name, Table=table_name,
DB=database_name
PWX-00713 2 active tasks
PWX-00709 0 Dormant TCBS

```

PowerExchange Listener DISPLAYSTATS Command

You can use the PowerExchange Listener DISPLAYSTATS or pwxcmd displaystats command to publish monitoring statistics for a PowerExchange Listener that runs on i5/OS, Linux, zLinux, UNIX, Windows, or z/OS.

Before you run the command, configure the following statements in the DBMOVER configuration file:

- Specify the MONITOR parameter in the STATS statement in the DBMOVER configuration file to enable PowerExchange to collect these statistics. You can include the *interval* subparameter to publish the statistics at a regular interval as well as on demand.
- For the proper display of monitoring output on z/OS, set the LOG_LINE_LIMIT statement to 132. Otherwise, the lines might wrap awkwardly, making the output hard to read.

You can issue the command in any of the followings ways:

- From the command line on the Linux, UNIX, Windows, or zLinux system where the PowerExchange Listener runs.

- With the MVS MODIFY (F) command on the z/OS system where the PowerExchange Listener runs.
- With the pwxcmd program from a remote Linux, UNIX, and Windows system to a Listener on any supported operating system.

Note: You must use this method to publish monitoring statistics for a PowerExchange Listener on i5/OS on demand.

The command syntax depends on the operating system type and whether you use pwxcmd. For more information, see the *PowerExchange Command Reference*.

Depending on which command parameter you use, you can publish one of the following types of reports:

- **Listener.** Reports PowerExchange Listener summary statistics on memory usage, CPU processing time, and activity on behalf of client requests. These statistics include counts of client tasks, connections, number of messages sent and received, bytes of data sent and received, and netport jobs (z/OS only). These statistics include both bulk data movement and CDC tasks.

Note: If you run a PowerExchange Listener Service in the Informatica Domain, you can use the infacmd pwx displayStatsListener command to publish these statistics. For more information, see the *Informatica Command Reference*.

- **Accessmethods.** Reports statistics on PowerExchange Listener message and data transfer activity by client task and access method. For each active task and access method combination, these statistics include the number of rows read and written, bytes of data read and written, the source or target file name or data map file name, and the CPU processing time. For CDC requests that use the CAPX or CAPXRT access method, the report also includes counts of the SQL inserts, updates, and deletes that the task processed.
- **Clients.** Reports information about the active client tasks that are running under the PowerExchange Listener. For each task, the statistics show some or all of the following information: the status, access method, read or write mode, process name and session ID if available, CPU processing time, and start date and time. The statistics also include the client port number and IP address. If the client is PowerCenter, the statistics include the PowerCenter session ID and the application name for CDC.

By default, the Listener report is published.

The reports for a PowerExchange Listener on z/OS are similar to those for a PowerExchange Listener on i5/OS, Linux, zLinux, UNIX, or Windows.

The following example Listener report is for a PowerExchange Listener on z/OS:

```

PWX-00723 Command <displaystats Listener> succeeded
PWX-37101 Listener <PWXLST > ASID=375 (x'177') UserID=AUSRID
PWX-37102 Memory
PWX-37103 Region below 16-MB line: In Use      108 KB Limit Value      9192 KB Free      9084 KB
PWX-37104 Region above 16-MB line: In Use    53912 KB Limit Value    1675264 KB Free    1621352 KB
PWX-37117 CPU Time
PWX-37118 TCB Time = 0 SRB Time = 0 zIIP-NTime = 0
PWX-37119 Listener = 0 hrs, 0 mins, 1 secs, 275762 mcrcs
PWX-37106 Cumulative Requests
PWX-37107 Total Tasks=      11 Active Tasks =      3 HWM Tasks =      3 Maxtasks =      50
PWX-37108 Connections=      11 Accepted =      11 Active =      0
PWX-37109 Msgs Sent =      0 Msgs Received=      22
PWX-37110 Data Sent =      0 Data Received=      7304
PWX-37111 NetportJobs=      0

```

The **Memory**, **TCB Time**, **SRB Time**, and **NetportJobs** values are specific to the PowerExchange Listener on z/OS. For a PowerExchange Listener on i5/OS, Linux, UNIX, or Windows, the report displays the total memory usage.

You can use this report determine if the number of client tasks is reaching the limit that is set in the MAXTASKS statement of the DBMOVER configuration file. Compare the **HWM Tasks** value to the **Maxtasks** value. If the HWM Tasks value reaches the MAXTASKS limit, PowerExchange Listener processing might be delayed, which can cause reduced throughput and connection timeouts.

The following example accessmethods report is for a PowerExchange Listener on z/OS, but the same fields are displayed for a PowerExchange Listener on i5/OS, Linux, UNIX, Windows, or zLinux:

```

PWX-00723 Command <displaystats AccessMethods> succeeded
PWX-37201 Active Access Methods
PWX-37202 Task ID = 42412 AM = CAPXRT
PWX-37203 Rows read = 1029 Rows written = 0
PWX-37204 Bytes read = 116277 Bytes written = 0
PWX-37205 File = d2ivd0.d002root_ROOT
PWX-37206 Table = <Capture Extract Realtime>
PWX-37208 Inserts = 564 Updates = 0 Deletes = 465
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 299809 mcrcs
PWX-37202 Task ID = 42414 AM = NRDB
PWX-37203 Rows read = 10 Rows written = 0
PWX-37204 Bytes read = 570 Bytes written = 0
PWX-37205 File = ABC.VSAM.MASTER_REC
PWX-37206 Table = <Non-relational source>
PWX-37202 Task ID = 42414 AM = KSDS
PWX-37203 Rows read = 10 Rows written = 0
PWX-37204 Bytes read = 800 Bytes written = 0
PWX-37205 File = XYQ.TEST.V1.KSDS
PWX-37206 Table = XYQ.TEST.V1.KSDS
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 76151 mcrcs

```

For the CAPXRT and CAPX access methods, the report includes the number of SQL inserts, updates, and deletes that the task processed for a CDC request.

A client task can have multiple access methods, for example, one for reading source data and one for mapping nonrelational source data to a relational format. In the example output, task 42414 uses the NRDB access method with the data map file specified in the **File** field to map nonrelational data to a relational format. The same task uses the KSDS access method to retrieve data from the KSDS data set specified in the **File** field.

The following example clients report is for a PowerExchange Listener on Windows, but the same fields are displayed for a PowerExchange Listener on i5/OS, Linux, zLinux, UNIX, or z/OS:

```

PWX-00723 Command <displaystats Clients> succeeded
PWX-37112 Active Tasks
PWX-37113 Task ID = 41942 Status = Active
PWX-37114 Port = 2480 Partner = 127.0.0.1
PWX-37115 PwrCtrSess = N/A
PWX-37207 Application = N/A
PWX-37116 AM = NRDB Mode = Read Process = DTLST3 SessionId =
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 62400 mcrcs
PWX-37122 Start time = 2014-05-01 14:21:37
PWX-37113 Task ID = 41943 Status = Active
PWX-37114 Port = 2480 Partner = 127.0.0.1
PWX-37115 PwrCtrSess = N/A
PWX-37207 Application = N/A
PWX-37116 AM = NRDB Mode = Read Process = DTLST3 SessionId =
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 124800 mcrcs
PWX-37122 Start time = 2014-05-01 14:22:01

```

The **Partner** field displays the IP address of the client that issued the request that caused the PowerExchange Listener to create the task. This value begins with `::ffff` for an IPv6 address.

For more information about the fields in each of these reports, see the *PowerExchange Command Reference*.

Monitoring CDC Sessions in PowerCenter

In PowerCenter, you can monitor the progress of CDC sessions.

Use the following information to monitor CDC sessions:

- Messages in the session log. PWXPC writes messages to the session log.
- Performance details in Workflow Monitor. If you configure a CDC session to report performance details, you can monitor the progress of the session in Workflow Monitor.

For more information about PowerCenter monitoring options, see the *PowerCenter Performance Tuning Guide*.

Session Log Messages

You can use messages that PWXPC and PowerCenter write to the session log to monitor the progress of CDC sessions.

When PWXPC flushes change data, PWXPC writes one of the following messages in the session log to indicate the reason for the flush:

```
PWXPC_10081 [INFO] [CDCDispatcher] raising real-time flush with restart tokens
[restart1], [restart2] because the UOW Count [count] is reached

PWXPC_10082 [INFO] [CDCDispatcher] raising real-time flush with restart tokens
[restart1], [restart2] because Real-time Flush Latency [latency] is reached

PWXPC_12128 [INFO] [CDCDispatcher] raising real-time flush with restart tokens
[restart1], [restart2] because the Maximum Rows Per commit [count] is reached
```

You can use the restart tokens in these PWXPC flush messages to monitor the processing of the change data.

For each PWXPC flush message, PowerCenter writes a WRT_8160 message after committing change data to the targets. This message displays the source-based commit statistics.

Performance Details in Workflow Monitor

In Workflow Monitor, you can view performance details in the run properties for a CDC session to assess the efficiency of the CDC session and extraction processing.

If session performance is degraded, you can use data in the Performance Counter column to determine the bottleneck.

PWXPC does not store performance details in the repository so you cannot view performance details for previous executions of a CDC session.

To enable the collection of performance details, select **Collect performance data** on the **Properties** tab of the CDC session properties.

When the CDC session runs, PWXPC refreshes performance statistics every 10 seconds.

If you enable a resume recovery strategy for the CDC session, PWXPC displays data for all Performance Counter fields.

The following table describes the Performance Counter fields:

Performance Counter Field	Description
1 PowerExchange CDC Reader Status:	Current status of the PWXPC reader, as indicated by one of the following values: <ul style="list-style-type: none">- No Data To Process. In the last read, PowerExchange did not pass data to PWXPC.- Restart Advance. PowerExchange passed restart tokens to PWXPC but did not pass change data.- Processing Data. PowerExchange passed change data and restart tokens to PWXPC for processing.
1.1 Time Last Data Row Read	Time, in milliseconds, that PWXPC took to read the data last received from PowerExchange.
1.2 Data Rows In Current Interval	Number of change records received from PowerExchange during the current statistics interval.

Performance Counter Field	Description
1.3 End Packets In Current Interval	Number of UOWs received from PowerExchange during the current statistics interval.
1.4 Data Read Rate In Current Interval (rows/sec)	<p>Number of change records read per second by PowerExchange during the current statistics interval.</p> <p>The value varies, based on the quantity of change data:</p> <ul style="list-style-type: none"> - If PowerExchange is reading large amounts of change data from the change stream, this value is usually large and reflects the maximum PowerExchange throughput. - If PowerExchange is waiting for change data at the end of the change stream, this value is small. <p>The following factors can increase this value:</p> <ul style="list-style-type: none"> - Large network bandwidth - CDC offload processing - Multithreaded processing
1.5 Mean Data Read Rate (rows/sec)	Mean number of change records that PowerExchange read per second, from the start of the CDC session.
1.6 Max Data Read Rate (rows/sec)	Maximum number of change records that PowerExchange read per second during a statistics interval, from the start of the CDC session.
2 PowerCenter Processing Status:	<p>Overall status of the CDC session, as indicated by one of the following values:</p> <ul style="list-style-type: none"> - Idle. Waiting for change data. - Processing Data. Data is being processed. - Recovery Disabled. If a resume recovery strategy is not enabled, the PWXPC CDC reader cannot get PowerCenter status information.
2.1 Time Of Last Commit	Time stamp of the last commit to a target.
2.2 Rows Processed To Commit In Current Interval	Number of change records that the PWXPC reader flushed during the current statistics interval. This count includes the change records in all committed UOWs. Some of these UOWs might have started before the current statistics interval began.
2.3 Commit Rate In Current Interval (rows/sec)	<p>Processing rate, in number of change records per second, for the change records for the UOW that was last committed during the current statistics interval. This processing includes reading the UOW from PowerExchange and committing the change data to the targets.</p> <p>The following factors can affect this rate:</p> <ul style="list-style-type: none"> - Number of available DTM buffers - Responsiveness of the target - Number of transformations in the pipeline
2.4 Mean Commit Rate (rows/sec)	<p>Mean number of change records per second for the rate displayed in 2.3 Commit Rate In The Current Interval.</p> <p>This value differs from the 2.6 Mean Throughput Rate value in that it takes into account only the time when the session is actively processing data. This value does not reflect processing overlap in PowerCenter.</p>
2.5 Max Commit Rate (rows/sec)	Maximum number of change records per second for the commit rate displayed in 2.3 Commit Rate In The Current Interval, from the start of the CDC session.

Performance Counter Field	Description
2.6 Mean Throughput (rows/sec)	Mean rate of processing for the CDC session.
2.7 Max Throughput (rows/sec)	Maximum throughput for the CDC session.
2.8 Commits In Current Interval	Number of commits processed to completion by the target during the current statistics interval.
2.9 Commits Pending	Number of commits that the PWXPC reader issued but that have not yet reached the targets. A large value might indicate problems with target responsiveness.
3 Capture Timestamps	-
3.1 Timestamp On Last End Packet Read	The capture timestamp, DTL__CAPXTIMESTAMP, from the last UOW read for a source in the CDC session.
3.2 Timestamp On Last Target Commit	The capture timestamp, DTL__CAPXTIMESTAMP, from the last UOW committed to the target.
4 Totals	-
4.1 Elapsed Time	Total elapsed time for the CDC session.
4.2 Rows Read	Total number of change records read from PowerExchange.
4.3 End Packets Read	Total number of UOWs read.
4.4 Time in PowerExchange Processing	Total time of PowerExchange processing for the CDC session.
4.5 Rows Processed	Total number of change records processed through PowerCenter and committed to the targets.
4.6 Commits to Target	Total number of flushes that the PWXPC reader issued and that were committed to the targets.
4.7 TS on Last Commit minus TS at Commit (2.1-3.2)	Result from subtracting 3.2 Timestamp On Last Target Commit value from the 2.1 Time Of Last Commit value. If this result is negative, the value is enclosed in parentheses.

Viewing Performance Details in Workflow Monitor

In Workflow Monitor, view performance details for a CDC session to assess the efficiency of the CDC sessions.

1. In Workflow Monitor, right-click a session and select **Get Run Properties**.
2. In the **Properties** window, click the **Performance** area.

The **Performance Counter** column displays a source qualifier from the CDC session. The **Counter Value** column displays the PowerCenter node name.

3. To view performance details, select the source qualifier.

Note: For PWXPC to display performance details for a CDC session that ended, you must select performance details while the session is still running.

PowerCenter displays data in the **Performance Counter** fields in the **Performance** area.

CHAPTER 9

Managing Change Data Extractions

This chapter includes the following topics:

- [Starting PowerCenter CDC Sessions, 133](#)
- [Stopping PowerCenter CDC Sessions, 136](#)
- [Changing PowerCenter CDC Sessions, 137](#)
- [Recovering PowerCenter CDC Sessions, 140](#)

Starting PowerCenter CDC Sessions

Use Workflow Manager, Workflow Monitor, or pmcmd to start a workflow or task for a CDC session. You can do a cold start, warm start, or recovery start. The method you use determines how PWXPC gets the restart information.

Also, you can start the entire workflow, part of a workflow, or a task in the workflow.

Use one of the following methods to start a CDC session:

Cold start

To cold start a CDC session, use the Cold Start command in Workflow Manager or Workflow Monitor. You can also use the pmcmd starttask or startworkflow commands with the norecovery option. A CDC session that uses real-time or continuous extraction mode runs continuously until it is stopped or interrupted. A CDC session that uses batch extraction mode runs until it reaches the end of log (EOL) or it is stopped or interrupted.

When you cold start a CDC session, PWXPC uses the restart token file to acquire restart tokens for all sources. PWXPC does not read the state tables or file or makes any attempt to recover the session.

Warm start

To warm start a CDC session, use the Start or Restart commands in Workflow Manager or Workflow Monitor. You can also use the pmcmd starttask or startworkflow commands. A CDC session that uses real-time or extraction mode runs continuously until it is stopped or interrupted. A CDC session that uses batch extraction mode runs until it reaches EOL or it is stopped or interrupted.

When you warm start a CDC session, PWXPC reconciles any restart tokens provided in the restart token file with any restart tokens that exist in the state tables or file. If necessary, PWXPC performs recovery processing.

Recovery start

To start recovery for a CDC session, use the Recover command from Workflow Manager or Workflow Monitor. You can also use the pmcmd recoverworkflow command or the starttask or startworkflow commands with the recovery option. When recovery completes, the CDC session ends.

When you recover a CDC session, PWXPC reads the restart tokens from any applicable state tables or state file. If necessary, PWXPC performs recovery processing. PWXPC updates the restart token file with the restart tokens for each source in the CDC session. Then the session ends. To begin extracting change data again, either cold start or warm start the session.

RELATED TOPICS:

- [“Cold Start Processing” on page 134](#)
- [“Warm Start Processing” on page 134](#)
- [“Recovery Processing” on page 135](#)

Cold Start Processing

To cold start workflows and tasks, use the Cold Start command in Workflow Manager or Workflow Monitor. Alternatively, you can use the pmcmd starttask or startworkflow commands with the norecovery option.

After you request a cold start for a CDC session, the following processing occurs:

1. PWXPC writes the following message in the session log:

```
PWXPC_12091 [INFO] [CDCRestart] Cold start requested
```
2. PWXPC reads the restart tokens from only the restart token file and associates a restart token with each source in the session.
3. PWXPC creates the initialization restart token file with the initial restart tokens.
4. PWXPC commits the restart tokens for each source to the appropriate state tables or state file and then writes message PWXPC_12104 to the session log.
5. PWXPC passes the restart tokens to PowerExchange. PowerExchange begins extracting change data and passing the data to PWXPC for processing.
6. PWXPC continues processing change data from PowerExchange and committing the data and restart tokens to the targets. This processing continues until the session ends or you stop it.

Warm Start Processing

To warm start workflows and tasks, use the Start or Restart command in Workflow Manager or Workflow Monitor. Alternatively, you can use the pmcmd starttask or startworkflow commands.

When you warm start a workflow or task, PWXPC automatically performs recovery. You do not need to recover failed workflows and tasks before you restart them.

After you request a warm start for a CDC session, the following processing occurs:

1. PWXPC writes the following message in the session log:

```
PWXPC_12092 [INFO] [CDCRestart] Warm start requested. Targets will be resynchronized automatically if required
```
2. PWXPC queries the PowerCenter Integration Service about the commit levels of all targets. If all targets in the session have the same commit level, PWXPC skips recovery processing.
3. PWXPC reconciles the restart tokens from the restart token file and from the state tables or file.

Restriction: If a CDC session requires recovery processing, PWXPC does not use the restart token file. Consequently, you cannot override restart tokens for sources.

4. PWXPC creates the initialization restart token file with the reconciled restart tokens.
5. If recovery is required, PWXPC re-reads the change data for the last unit-of-work (UOW) that was committed to the targets with the highest commit level and then flushes the data to the targets with lower commit levels. The PowerCenter Integration Service commits flushed change data and restart tokens to any relational targets and updates any nonrelational files.
6. If recovery is not required and the reconciled restart tokens differ from those in the state tables or state file, PWXPC commits the reconciled restart tokens and then writes message PWXPC_12104 to the session log.
7. PWXPC passes the restart tokens to PowerExchange. PowerExchange begins extracting change data and passing the data to PWXPC for processing.
8. PWXPC continues processing change data from PowerExchange and commits the data and restart tokens to the targets. This processing continues until the session ends or you stop it.

Recovery Processing

To recover workflows and tasks, use the Recover command in Workflow Manager or Workflow Monitor. Alternatively, you can use the pmcmd recoverworkflow command, or the starttask or startworkflow command with the recovery option.

Use the recovery start method to populate the restart token file with the restart tokens for all sources in a CDC session. You can then cold start the CDC session or verify that the targets and restart tokens are in a consistent state. However, you do not need to recover failed workflows and tasks before you restart them because PWXPC automatically performs recovery processing when you warm start a workflow or task.

After you request recovery for a CDC session, the following processing occurs:

1. PWXPC writes the following message in the session log:

```
PWXPC_12093 [INFO] [CDCRestart] Recovery run requested. Targets will be
resynchronized if required and processing will terminate
```
2. PWXPC queries the PowerCenter Integration Service about the commit levels of all targets. If all targets in the session have the same commit level, PWXPC skips recovery processing.
3. PWXPC reads the restart tokens from the recovery state tables or state file.
Restriction: If a CDC session requires recovery processing, PWXPC does not use the restart token file. Consequently, you cannot override restart tokens for sources.
4. PWXPC creates the initialization restart token file with the reconciled restart tokens.
5. If recovery is required, PWXPC re-reads the change data for the last UOW that was committed to the targets with the highest commit level and then flushes the data to the targets with lower commit levels. The PowerCenter Integration Service commits any flushed change data and restart tokens to any relational targets, and updates any nonrelational files.
6. PWXPC updates the restart token file with the final restart tokens, creates the termination restart token file, and ends.

To process change data from the point of recovery, warm start or cold start the workflow or task.

Stopping PowerCenter CDC Sessions

You can stop CDC sessions from PowerCenter or PowerExchange.

In PowerCenter, issue the Stop or Abort command in Workflow Monitor. Alternatively, use the `pmcmd stoptask`, `stopworkflow`, `aborttask`, or `abortworkflow` commands.

- If you issue the Stop command in Workflow Monitor or use the `pmcmd stoptask` or `stopworkflow` command, the PWXPC CDC reader and PowerCenter Integration Service complete processing all of the data in the pipeline and shut down. Then, the CDC session ends.
- If you issue the Abort command in Workflow Monitor or use the `pmcmd aborttask` or `abortworkflow` command, the PowerCenter Integration Service waits 60 seconds to allow the readers and writers to complete processing all of the data in the pipeline and shut down. If the PowerCenter Integration Service cannot finish processing and committing data within this period, it kills the DTM process and ends the CDC session.

For more information about these PowerCenter commands, see the *Informatica Command Reference* or *PowerCenter Workflow Basics Guide*.

In PowerExchange, issue the PowerExchange Listener `STOPTASK` command in one of the following ways:

- From the command line on the system where extraction processing occurs
- From the PowerExchange Navigator
- With the `DTLUTSK` utility
- With the `pwxcmd` program

When you issue the `STOPTASK` command, PowerExchange stops the extraction task in the PowerExchange Listener and passes an EOF to the PowerCenter Integration Service. Then the CDC session ends. For more information about the `STOPTASK` command, see the *PowerExchange Command Reference*.

Stop Command Processing

After you issue a stop command in PowerCenter or PowerExchange, the following processing occurs:

Note: To stop CDC sessions and workflows, you can use the Stop command in Workflow Monitor or the `pmcmd stoptask` or `stopworkflow` command. Alternatively, you can use the PowerExchange `STOPTASK` command.

1. If you use a PowerCenter stop command, the PowerCenter Integration Service requests PWXPC to stop. If you use the PowerExchange `STOPTASK` command, PowerExchange sends an EOF to PWXPC.
2. When PWXPC receives an EOF, it flushes any complete and uncommitted UOWs and the associated restart tokens to the targets. PWXPC then writes messages `PWXPC_12101` and `PWXPC_12068` to the session log.
3. The PowerCenter Integration Service processes all of data in the pipeline and writes it to the targets.
4. The PowerCenter Integration Service sends an acknowledgment to PWXPC indicating that the targets have been updated.
5. PWXPC writes the termination restart token file, and then writes the message `PWXPC_12075` to the session log.
6. The PWXPC CDC reader shuts down.
7. The PowerCenter Integration Service performs any post-session tasks and ends the session.

Terminating Conditions

You can have CDC sessions stop based on user-defined events or at EOL if you configure certain terminating conditions.

When PWXPC encounters a terminating condition, it stops reading change data from sources, flushes change data to the targets, and passes an EOF to the PowerCenter Integration Service. The PowerCenter Integration Service commits the data to the targets and ends the CDC session.

Use the following connection attributes and features as terminating conditions:

Event table processing

Create an event table and a capture registration for the table. Then specify the extraction map for the table in the **Event Table** attribute of the PWX CDC Real Time application connection for the CDC session. After PowerExchange reads a change record for the event table, it passes an EOF to PWXPC to end the CDC session.

Idle time

Enter 0 for the **Idle Time** attribute on a PWX CDC Real Time application connection. Then, whenever PowerExchange reaches EOL, it passes an EOF to PWXPC to end the CDC session.

Batch extraction mode

If you use batch extraction mode, PowerExchange reads all closed PowerExchange Condense condense files or PowerExchange Logger for Linux, UNIX, and Windows log files. Then PowerExchange passes an EOF to PWXPC to end the CDC session.

Changing PowerCenter CDC Sessions

Use this procedure to change CDC sessions. You might need to add or remove sources and targets.

After you change a CDC session, you must cold start it. Because a cold start is required, you must also get the latest restart tokens for the original sources before restarting the session. To do so, you can perform a recovery.

To change a CDC session:

1. Stop the workflow.
2. After the workflow ends, recover the CDC session.
When you recover tasks, PWXPC writes the ending restart tokens for all sources in a CDC session to the restart token file that you specified on the PWX CDC application connection.
3. Make changes to the session or workflow, if necessary.
4. Verify that the restart token file in the source CDC connection points to the same restart token file updated in the recovery.
5. If you add sources to the CDC session, add statements to the restart token file that provide restart tokens for the additional sources.
6. If you remove sources from the CDC session, update the restart token file to remove their restart tokens.
7. Cold start the CDC session.

Examples of Adding Sources and Creating Restart Tokens

The following examples show how to add sources to CDC sessions and create restart tokens for those sources.

The first example uses the `CURRENT_RESTART` option of the special override statement in the restart token file to generate current restart tokens. The second example uses `DTLUAPPL` to generate current restart tokens.

Example 1. Creating Current Restart Tokens with Special Override Statements

This example adds a source table, `RRTB_SRC_004`, to a CDC session that has three other sources. You edit the restart token file to generate restart tokens that represent the current end of the change stream for the additional source.

In the restart token file, you define special override statements with `CURRENT_RESTART` option for the `RRTB_SRC_004` source.

For the other three sources, you retain the existing restart points.

To add a source with `CURRENT_RESTART` restart tokens:

1. In Workflow Monitor, use the Stop command to stop the workflow.
2. After the workflow stops, select the Recover Task command to run a recovery session.

PWXPC writes the following messages in the session log:

```
PWXPC_12060 [INFO] [CDCRestart]
=====
Session restart information:
=====
Extraction Map Name      Restart Token 1      Restart Token 2      Source
dldsn9.rrtb0002_RRTB_SRC_002  000000AD220F00000000000000AD220F0000000000000000  C1E4E2D34040000000AD0D9C000000000  GMD storage
dldsn9.rrtb0001_RRTB_SRC_001  000000AD220F00000000000000AD220F0000000000000000  C1E4E2D34040000000AD0D9C000000000  GMD storage
dldsn9.rrtb0003_RRTB_SRC_003  000000AD220F00000000000000AD220F0000000000000000  C1E4E2D34040000000AD0D9C000000000  GMD storage
```

PWXPC also writes the restart tokens in the restart token file that is identified the CDC application connection attributes.

3. Edit the mapping, session, and workflow to add the source `RRTB_SRC_004`.
4. Edit the restart token file to add `RESTART1` and `RESTART2` special override statements that specify the `CURRENT_RESTART` option for the `RRTB_SRC_004` source.

The updated file appears as follows:

```
<!-- existing sources
dldsn9.rrtb0001_RRTB_SRC_001=000000AD220F00000000000000AD220F0000000000000000
dldsn9.rrtb0001_RRTB_SRC_001=C1E4E2D34040000000AD0D9C000000000
dldsn9.rrtb0002_RRTB_SRC_002=000000AD220F00000000000000AD220F0000000000000000
dldsn9.rrtb0002_RRTB_SRC_002=C1E4E2D34040000000AD0D9C000000000
dldsn9.rrtb0003_RRTB_SRC_003=000000AD220F00000000000000AD220F0000000000000000
dldsn9.rrtb0003_RRTB_SRC_003=C1E4E2D34040000000AD0D9C000000000
<!-- new source
RESTART1=CURRENT_RESTART
RESTART2=CURRENT_RESTART
```

5. Cold start the session.

PWXPC connects to PowerExchange and generates restart tokens that match the current end of the change stream for the `RRTB_SRC_004` source. PWXPC passes the generated restart tokens to PowerExchange to begin change data extraction. Because the restart points for the other sources are earlier than the restart point for `RRTB_SRC_004`, PWXPC does not pass any change data for `RRTB_SRC_004` until it reads the first change after the generated restart point.

Example 2. Creating Current Restart Tokens with the DTLUAPPL Utility

This example adds the source table, RRTB_SRC_004, to a CDC session that has three other sources. You use the DTLUAPPL utility to generate restart tokens that represent the current end of the change stream.

For the other three sources, you retain the existing restart points.

1. In Workflow Monitor, use Stop command to stop the workflow.
2. After the workflow stops, select the Recover Task command to run a recovery session.

PWXPC writes the following messages in the session log:

```
PWXPC_12060 [INFO] [CDCRestart]
=====
Session restart information:
=====
Extraction Map Name      Restart Token 1      Restart Token 2      Source
dldsn9.rrtb0002_RRTB_SRC_002 000000AD220F00000000000000AD220F0000000000000000 C1E4E2D3404000000AD0D9C00000000 GMD storage
dldsn9.rrtb0001_RRTB_SRC_001 000000AD220F00000000000000AD220F0000000000000000 C1E4E2D3404000000AD0D9C00000000 GMD storage
dldsn9.rrtb0003_RRTB_SRC_003 000000AD220F00000000000000AD220F0000000000000000 C1E4E2D3404000000AD0D9C00000000 GMD storage
```

PWXPC also writes the restart tokens in the restart token file that is identified in the CDC application connection attributes.

3. Edit the mapping, session, and workflow to add the source RRTB_SRC_004.
4. Run the DTLUAPPL utility with RSTTKN GENERATE parameter to generate restart tokens that represent the current end of the change stream for the additional source.

Use the following DTLUAPPL control cards:

```
mod APPL dummy DSN7 rsttkn generate
  mod rsttkn rrtb004
end appl dummy
print appl dummy
```

The PRINT command produces the following output:

```
Registration name=<rrtb004.1> tag=<DB2DSN7rrtb0041>
Sequence=<00000DBF240A00000000000000DBF240A00000000>
Restart =<C1E4E2D340400000DBF238200000000>
```

You can add eight zeros to the end of the Sequence value to create the sequence value for the restart token file.

5. Edit the restart token file to add the source and its restart tokens.

The updated file contains the following lines:

```
<!-- existing sources
dldsn9.rrtb0001_RRTB_SRC_001=000000AD220F00000000000000AD220F0000000000000000
dldsn9.rrtb0001_RRTB_SRC_001=C1E4E2D3404000000AD0D9C00000000
dldsn9.rrtb0002_RRTB_SRC_002=000000AD220F00000000000000AD220F0000000000000000
dldsn9.rrtb0002_RRTB_SRC_002=C1E4E2D3404000000AD0D9C00000000
dldsn9.rrtb0003_RRTB_SRC_003=000000AD220F00000000000000AD220F0000000000000000
dldsn9.rrtb0003_RRTB_SRC_003=C1E4E2D3404000000AD0D9C00000000
<!-- new source
dldsn9.rrtb0004_RRTB_SRC_004=00000DBF240A0000000000000DBF240A000000000000000
dldsn9.rrtb0004_RRTB_SRC_004=C1E4E2D3404000000DBF2382000000000
```

6. Cold start the session.

PWXPC passes the restart tokens to PowerExchange to begin change data extraction. Because the restart points for the other sources are earlier than the restart point for RRTB_SRC_004, PWXPC does not pass any change data for RRTB_SRC_004 until it reads the first change after the generated restart point.

Recovering PowerCenter CDC Sessions

You can use Workflow Manager, Workflow Monitor, or pmcmd to recover an entire workflow or a task in a workflow for a CDC session that fails.

A CDC session can fail for the following reasons:

- Permanent errors, such as source or target data errors
- Transitory or environmental errors, such as infrastructure problems, server failures, and network availability issues

If you run a session with a resume recovery strategy and the session fails, do not edit the state information or the mapping for the session before you restart the session.

If a session fails because of transitory or environmental errors, restart the session after you have corrected the errors. When you warm start a CDC session, PWXPC automatically performs recovery, if required. Alternatively, you can recover a CDC session, and then restart the session.

If a CDC session fails because of permanent errors, such as SQL or other database errors, you must correct the errors before restarting the CDC session. With some failures, you can correct the error and then restart the CDC session. In other cases, you might need to rematerialize the target table from the source table before you start extracting and applying change data again. If you rematerialize the target table, provide restart tokens that match the materialization point in the change stream, and then cold start the CDC session.

Restriction: If a CDC session requires recovery processing, you cannot override the restart tokens because PWXPC does not read the restart token file.

Example of Session Recovery

This example describes recovery processing for a CDC session with relational targets.

Assume that you aborted the CDC session from the Workflow Monitor and then issued the Restart Task command to restart the session.

PWXPC automatically performs a recovery processing when the session warm starts and writes the following message in the session log:

```
PWXPC_12092 [INFO] [CDCRestart] Warm start requested. Targets will be resynchronized automatically if required
```

PWXPC then reads the restart tokens from the state tables and writes message PWXPC_12060 in the session log. This message records the restart tokens for the session and its sources, for example:

```
PWXPC_12060 [INFO] [CDCRestart]
=====
Session restart information:
=====
Extraction Map Name      Restart Token 1      Restart Token 2      Source
dldsn8.rrtb0004_RRTB_SRC_004 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0009_RRTB_SRC_009 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0005_RRTB_SRC_005 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0006_RRTB_SRC_006 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0008_RRTB_SRC_008 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0003_RRTB_SRC_003 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0002_RRTB_SRC_002 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0001_RRTB_SRC_001 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
dldsn8.rrtb0007_RRTB_SRC_007 00000FCA65840000000000000D2E004A00000000FFFFFFFFF C1E4E2D3404000000D21B1A500000000 GMD storage
```

If PWXPC detects that recovery is required, PWXPC writes message PWXPC_12069 in the session log. This message usually includes the restart tokens for both the begin-UOW and end-UOW for the oldest uncommitted UOW that PWXPC re-reads during recovery. PWXPC usually stores end-UOW restart tokens in the state table or state file. However, if you specify a **Maximum Rows Per commit** threshold on the connection, PWXPC can commit change data and restart tokens between UOW boundaries. As a result, the restart tokens might not represent an end-UOW.

The following example PWXPC_12069 message includes “from” restart tokens that are the same as those in the example PWXPC_12060 message:

```
PWXPC_12069 [INFO] [CDCRestart] Running in recovery mode. Reader will resend the oldest uncommitted UOW to resync targets:  
from: Restart 1 [00000FCA65840000000000000D2E004A00000000FFFFFFFF] : Restart 2 [C1E4E2D3404000000D21B1A500000000]  
to: Restart 1 [00000FCA65840000000000000D300D8000000000FFFFFFFF] : Restart 2 [C1E4E2D3404000000D21B1A500000000].
```

Because this session specifies a maximum rows threshold, the restart token values in the Restart 2 fields in both the “from” and “to” restart tokens is the begin-UOW value. The sequence token values in the Restart 1 fields represent the start and end change records in the UOW that is displayed in the Restart 2 field.

During recovery processing, PWXPC reads the change data records between the restart points that are defined by the two restart token values in the PWXPC_12069 message. Then PWXPC issues a commit for the change data and restart tokens. The PowerCenter Integration Service writes the change data to the target tables and writes the restart tokens to the state table. Then the session ends.

CHAPTER 10

Tuning CDC Sessions

This chapter includes the following topics:

- [Tuning Overview, 142](#)
- [PowerExchange DBMOVER Statements for Tuning CDC Sessions, 143](#)
- [PowerCenter Connection Attributes and Session Properties, 146](#)
- [CDC Offload Processing, 149](#)
- [Multithreaded Processing, 151](#)

Tuning Overview

PowerExchange and PowerCenter provide options that you can use to tune CDC sessions. These tuning options can help you increase throughput, reduce overhead on the source system, and improve CDC efficiency.

Use any of the following options to tune CDC sessions:

- PowerExchange DBMOVER statements. Customize certain statements in the DBMOVER configuration file to make tuning adjustments such as changing buffer sizes or disabling compression or traces.
- PowerCenter connection attributes. Customize PWX CDC application connection attributes to make tuning adjustments such as disabling encryption or compression, reducing commit processing, or enabling offload processing and multithreaded processing.
- Buffer memory. Set the PowerCenter **DTM Buffer Size** and **Default Buffer Block Size** session properties to generate a large number of small blocks. For CDC, this strategy improves session performance and prevents wasted buffer space.
- Offload processing. Use offload processing to transfer column-level extraction processing from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine. Also, if the data source type requires use of the UOW Cleanser (UOWC), offloading transfers UOWC processing to the PowerCenter Integration Service machine. Offloading helps increase throughput when resources available for the PowerExchange Listener are constrained on the source system.
- Multithreaded processing. Enable the use of multiple worker threads for resource-intensive, column-level extraction processing. You can use multithreading on the source system to process data from Linux, UNIX, or Windows data sources if the PWX connection for the CDC session has a location of local. You can also use multithreading for extracting change data from the systems other than the source system when offload processing is in effect. Enable multithreading only when extractions appear to be CPU bound.

- Workload Manager (WLM) service classes. Assign each of the following types of PowerExchange CDC started tasks or jobs to an appropriate WLM service class based on your business requirements: PowerExchange Listener, PowerExchange Agent, PowerExchange Logger, Post-Log Merge jobs, PowerExchange ECCRs, and PowerExchange Condense. A service class includes a goal and importance level, which WLM uses to prioritize work requests for z/OS shared resources.

Note: You can also log data to a PowerExchange Logger for Linux, UNIX, and Windows instance on a system that is remote from the source system. In certain situations, this configuration can reduce resource consumption on the source system, move column-level and UOW Cleanser processing to the remote system, and reduce the network overhead of data transfer. For more information, see [Chapter 5, “Remote Logging of Data” on page 73](#).

RELATED TOPICS:

- [“PowerCenter Connection Attributes for Tuning CDC Sessions ” on page 146](#)
- [“PowerExchange DBMOVER Statements for Tuning CDC Sessions” on page 143](#)
- [“Tuning Commit Processing” on page 148](#)

PowerExchange DBMOVER Statements for Tuning CDC Sessions

You can customize certain statements in the dbmover.cfg configuration file to tune CDC sessions.

Customize any of the following parameters to try to increase throughput or reduce CPU use:

APPBUFSIZE=bytes

The maximum application data buffer size, in bytes, that PowerExchange uses to read or write data. This buffer type can exist on a source or target system.

If you use a remote target system, PowerExchange usually writes change data to its application data buffer on the source system until the buffer is full. PowerExchange then sends the change data to a sending TCP/IP buffer on the source system. TCP/IP transports the change data to a receiving TCP/IP buffer on the target system. PowerExchange on the target system reads the change data from the TCP/IP buffer into its application data buffer. PWXPC then reads the change data and passes it to PowerCenter. PowerCenter processes the data and applies it to the targets.

Enter an APPBUFSIZE value that is greater than the maximum size of any single data row to be sent.

Valid values are from 34816 through 8388608. Default is 256000.

If the target is remote, enter the same APPBUFSIZE value in the DBMOVER configuration files on the source and target systems.

When the APPBUFSIZE value is not optimal, PowerExchange writes message PWX-01295 in the PowerExchange message log file on the source system. This message recommends a minimum application buffer size.

If dynamic application buffer sizing is enabled, the APPBUFSIZE statement defines the initial size of the application data buffer for all connections made during a PowerExchange Listener run. PowerExchange resizes the application data buffer dynamically for individual connections as needed. Dynamic application buffer sizing is enabled by default. You can explicitly enable it by specifying Y for the APPBUFSIZEDYN statement in the DBMOVER configuration file.

APPBUFSIZEDYN={N|Y}

Specifies whether to enable dynamic application buffer sizing.

The DBMOVER APPBUFSIZE statement defines the initial size of the application buffer for all connections made during a PowerExchange Listener run. If APPBUFSIZEDYN=Y, PowerExchange resizes the application buffers for individual connection as needed.

The APPBUFSIZEDYN statement applies to PowerExchange connections to data sources with either fixed-length or variable-length records. A variable-length record is a record with at least one variable-length field. A variable-length field has a datatype of VARCHAR or VARBIN.

For each connection to a data source with variable-length records, PowerExchange resizes the application buffer when it encounters a record that is too large to fit into the buffer. PowerExchange increases the size of the application buffer to a value of ten times the size of the record that has overflowed, up to the maximum application buffer size of 8 MB. The new size remains in effect for the duration of the Listener run or until the application buffer is resized again. PowerExchange never decreases the application buffer size for a connection after the Listener run has started.

For each connection to a data source with fixed-length records, PowerExchange determines the record length when the connection is opened and resizes the application buffer once, up to the maximum application buffer size of 8 MB, as needed.

CAPI_CONNECTION=(..., (TYPE={UDB|UOWC}, MEMCACHE=*cache_size*, ...))

The maximum memory cache size, in kilobytes, that PowerExchange can allocate to reconstruct complete UOWs. This MEMCACHE parameter is specified only in the UDB or UOWC CAPI_CONNECTION statements.

Enter a number from 0 through 2147483647. Default is 1024. If you enter 0, the memory cache size is unlimited.

PowerExchange keeps all changes in each UOW in cache until processing the end-UOW record. PowerExchange incrementally allocates memory cache up to the limit that this parameter specifies. If the MEMCACHE value is too small to hold all of the changes in a UOW in cache, the changes spill to a disk file.

Each UOW spill file contains one UOW. A UOW might require multiple UOW spill files to hold all of the changes for that UOW. If the change stream contains multiple large UOWs and the memory cache is insufficient, PowerExchange might create numerous UOW spill files.

PowerExchange processes the change stream more efficiently if it does not need to use UOW spill files. In addition to degrading extraction performance, large numbers of UOW spill files can cause a disk space shortage.

The default value of 1024 is appropriate if the change stream contains many small UOWs. If you have UOWs larger than 1024 KB, increase this value or enter 0. PowerExchange processes a UOW more efficiently if all of the changes are cached in memory. For most environments, a good starting value is 10240.

Attention: PowerExchange allocates memory cache for each connection for change data extraction processing. To prevent excessive memory use, use a MEMCACHE value that is reasonable for the extraction processing load and number of CDC sessions that run concurrently. If the value is too large and you run many concurrent sessions, memory constraints might occur.

CAPI_CONNECTION=(..., (TYPE={MSQL|UDB|UOWC}, RSTRADV=*rstr_seconds*, ...))

Time interval, in seconds, that PowerExchange waits before advancing restart and sequence tokens for a registered data source during periods when UOWs do not include any changes of interest for the data source. When the wait interval expires, PowerExchange returns the next committed "empty UOW," which includes only updated restart information.

This RSTRADV parameter is specified only in CAPI_CONNECTION statements of the following types:

- MSQL
- UDB
- UOWC

Enter a number from 0 through 86400.

If you do not specify RSTRADV, PowerExchange does not advance restart and sequence tokens for a registered source during periods when PowerExchange receives no changes of interest. In this case, when PowerExchange warm starts, it reads all changes, including those not of interest for CDC, from the restart point.

PowerExchange resets the wait interval to 0 when one of the following events occur:

- PowerExchange completes processing a UOW that includes changes of interest.
- PowerExchange returns an empty UOW because the wait interval expired without PowerExchange receiving any changes of interest.

For sources with low change activity, you can use the RSTRADV parameter to periodically advance to the restart tokens for those sources. Advancing the restart tokens speeds up restart processing for CDC sessions by minimizing the amount of change data that must be reprocessed.

For example, if you specify 5, PowerExchange waits 5 seconds after it completes processing the last UOW or after the previous wait interval expires. Then PowerExchange returns the next committed empty UOW that includes the updated restart information and resets the wait interval to 0.

A low value can cause the **UOW Count** option on the PWX CDC connection to match more quickly than expected. When the UOW counter matches, PWXPC flushes the data buffer and commits restart tokens to the targets. Excessive flush activity can adversely affect performance on the PowerCenter Integration Service machine and on the target databases.

Attention: A value of 0 can degrade performance. In addition to the UOWs that contain changes for registered sources of interest, PowerExchange returns an empty UOW for every UOW that does not contain changes for the registered sources of interest.

LISTENER=(node_name,TCPIP,port,send_bufsize,receive_bufsize,send_size,receive_size, ...)

A TCP/IP port on which a named PowerExchange Listener process listens for work requests.

The *send_bufsize* and *receive_bufsize* positional parameters define the data portion of the TCP/IP send and receive buffer sizes that PowerExchange uses. If you do not specify these values, PowerExchange uses the operating system defaults.

To increase throughput, try increasing the *send_bufsize* and *receive_bufsize* values in the LISTENER statement in the DBMOVER configuration file on the source system. For help in determining the best values to use, contact your network administrator.

NODE=(node_name,TCPIP,host_name,port,send_bufsize,receive_bufsize,send_size,receive_size, ...)

A TCPIP host name and port that PowerExchange uses to contact a PowerExchange Listener process.

The *send_bufsize* and *receive_bufsize* positional parameters define the data portion of the send and receive buffer sizes that PowerExchange uses. If you do not specify these values, PowerExchange uses the operating system defaults.

To increase throughput, try increasing the *send_bufsize* and *receive_bufsize* values in the NODE statement in the DBMOVER configuration file on the target system. For help in determining the best values to use, contact your network administrator.

TRACE=(trace_id,trace_level,99)

Activates PowerExchange diagnostic traces that Informatica Global Customer Support uses to solve problems with PowerExchange code.

TRACE statements can severely impact PowerExchange performance. Use these statements only at the direction of Informatica Global Customer Support.

After Informatica Global Customer Support diagnoses the problem, remove or comment out all TRACE statements in the DBMOVER configuration files on all systems.

For more information about these DBMOVER statements, see the *PowerExchange Reference Manual*.

PowerCenter Connection Attributes and Session Properties

You can use certain PowerCenter connection attributes and session properties for tuning CDC sessions.

PowerCenter Connection Attributes for Tuning CDC Sessions

In PowerCenter, you can customize some attributes on the PWX CDC connections to tune CDC sessions.

The following table describes the connection attributes that you can optionally use for tuning:

Connection Option	Description	Tuning Suggestion
Compression	Controls whether to compress source data during the PowerCenter session. Default disables compression.	Do not use compression.
Encryption Type	Type of data encryption that PowerExchange uses. Default is None for no encryption.	Do not use encryption.

Connection Option	Description	Tuning Suggestion
Image Type	<p>Indicates how PWXPC passes captured Updates to CDC sessions that extract and apply the updates to the target.</p> <p>Options are:</p> <ul style="list-style-type: none"> - AI. Process Updates as Update operations. PWXPC passes each Update as a single Update record. An Update record includes after images of the data only, unless you add before image (BI) and change indicator (CI) fields to the extraction map that you import for the source definition for the CDC session. - BA. Process Updates as Deletes followed by Inserts. PWXPC passes each Update as a Delete record followed by an Insert record. The Delete record contains the before image of the data, and the Insert record contains the after image. <p>Default is BA.</p> <p>If you specify AI, you can still use before images of the data, if available, in extraction processing. PWXPC can embed before-image data and after-image data in the same Update row. To embed before-image data, you must complete the following configuration tasks:</p> <ul style="list-style-type: none"> - In the PowerExchange Navigator, add BI and CI fields to the extraction map that you plan to import for the source definition in PowerCenter. - If you use batch or continuous extraction mode, enter BA for the CAPT_IMAGE parameter in the PowerExchange Condense or PowerExchange Logger for Linux, UNIX, and Windows configuration file. This setting stores both before and after images in the PowerExchange Logger log files or PowerExchange Condense condense files. When CDC sessions run, they extract data from these files. 	Set to AI.
UOW Count	<p>The number of UOWs that PWXPC reads from the source before it flushes the data buffer to commit the change data to the targets.</p> <p>Default is 1.</p>	To improve efficiency on the PowerCenter Integration Service machine and the target databases, increase this value to reduce commit processing.
Real-time Flush Latency in milliseconds	<p>The frequency, in milliseconds, with which PWXPC flushes the data buffer to commit the change data to the targets.</p> <p>Default is 0, which is equivalent to 2 seconds.</p>	To improve efficiency on the PowerCenter Integration Service machine and the target databases, increase this value to reduce commit processing.
PWX Latency in seconds	<p>Maximum time, in seconds, that the PowerExchange instance on the source waits for more change data before flushing data to PWXPC on the PowerCenter Integration Service machine.</p> <p>Default is 2.</p>	Use the default value.

Connection Option	Description	Tuning Suggestion
Minimum Rows Per commit	Minimum number of change records that PowerExchange reads from the change stream before it passes any commit records to PWXPC. Default is 0, which means that PWXPC ignores this option.	If UOWs usually contain few changes, increase this value to increase the size of the UOWs. This practice can improve efficiency on the PowerCenter Integration Service machine and on the target databases by reducing commit processing.
Offload Processing	Controls whether PowerExchange uses CDC offload processing. Offload processing transfers resource-intensive column-level and UOW Cleanser processing from the source system to another system. Default is No.	If resource constraints exist on the source system and you need to increase CDC throughput, consider enabling offload processing.
Worker Threads	Controls whether PowerExchange uses multiple threads for resource-intensive, column-level extraction processing. You can use multithreading on the source system to process data from Linux, UNIX, or Windows data sources, or on another system for extraction processing when offload processing is in effect. Enable multithreading only when extractions appear to be CPU bound. Enter the number of threads that you want PowerExchange to use. Valid values are 1 through 64. Default is 0, which causes PowerExchange to not use multithreaded processing.	Enter a number greater than 0.
Array Size	If the Worker Threads value is greater than zero, indicates the size of the storage array, in number of records, for the threads. Valid values are 25 through 5000. Default is 25.	Informatica recommends that you use the default value of 25 unless you are able to test and determine that the extra memory that is allocated to a larger array size has been beneficial and has not degraded server performance. If you make these determinations, Informatica recommends that you use an array size of 500 to 1000 with offload and multithreaded processing enabled. Attention: If you enter a large array size value, have large records, or run many sessions that use multithreading concurrently, memory shortages might occur on the PowerCenter Integration Service machine.

For more information about PWX CDC connection attributes, see *PowerExchange Interfaces for PowerCenter*.

Tuning Commit Processing

To tune commit processing and performance of CDC sessions, you can adjust commitment control attributes on the PWX CDC application connection.

If the session log for a CDC session contains PWXPC flush messages followed by PowerCenter source-based commit messages, the session might be reading change data faster than the data is applied to the targets.

To try to resolve this issue, adjust the following commitment control attributes on the PWX CDC connection, based on the most prevalent type of flush message in the session log:

- If PWXPC_10081 flush messages are the most prevalent messages, try increasing the **UOW Count**.
- If PWXPC_10082 flush messages are the most prevalent messages, try increasing the **Real-time Flush Latency in milli-seconds**.

If PWXPC flushes change data too frequently, too many commitment control attributes might be specified on the PWX CDC connection. In this case, specify a single commitment control attribute and disable the other ones.

If the change stream contains many small UOWs, you can use the **Minimum Rows Per commit** option to create larger UOWs of more uniform size. PowerExchange and PWXPC can process a few large UOWs more efficiently than many small UOWs. By using the **Minimum Rows Per commit** option to increase the size of UOWs, you can improve CDC processing efficiency.

Also, performance of the target database can impact the performance of the CDC session. Contact your database administrator to verify that database access is optimal.

PowerCenter Session Properties for Tuning Buffer Memory

When you run a CDC session, the PowerCenter Data Transformation Manager (DTM) allocates buffer memory to the session based on the **DTM Buffer Size** value on the **Properties** tab of the session properties. The DTM divides the memory into buffer blocks based on the **Default Buffer Block Size** setting on the **Config Object** tab of the session properties.

If you suspect that buffer memory is insufficient, enable the collection of performance details in the CDC session. Then review the difference between the performance counters

4.1 Time in PowerExchange Processing and **4.4 Elapsed Time**. If the elapsed time is much larger than the PowerExchange processing time, buffer memory constraints might exist. To improve performance of the CDC session, try adjusting the **DTM Buffer Size** and **Default Buffer Block Size** properties.

For optimal CDC performance, set these session properties to create a large number of small blocks. Informatica recommends the following settings:

- For the **DTM Buffer Size**, specify 128 MB, 256 MB, 512 MB, 1 GB, or 2 GB.
- For the **Default Buffer Block Size**, specify 32 KB.

Do *not* set these session properties to **auto**. The **auto** option creates a small number of large blocks, which can degrade CDC session performance. The **auto** option is intended for bulk data load processing.

CDC Offload Processing

CDC offload processing transfers column-level processing of change data from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine.

For data sources for which PowerExchange uses the UOW Cleanser (UOWC), offload processing also transfers UOWC processing to the PowerCenter Integration Service machine. These data sources include z/OS data sources and Db2 for i (i5/OS) sources.

Use offload processing when resources on the source system are constrained. In this situation, offload processing can help increase throughput for CDC sessions.

RELATED TOPICS:

- [“Rules and Guidelines for CDC Offload Processing” on page 150](#)
- [“Example of CDC Offload Processing with a DB2 for i5/OS Source” on page 151](#)
- [“Enabling Offload Processing for CDC Sessions” on page 150](#)

Rules and Guidelines for CDC Offload Processing

Before implementing CDC offload processing, review the following rules and guidelines:

- You must copy the appropriate source-specific CAPI_CONNECTION statements from the DBMOVER configuration file on the source system to the PowerCenter Integration Service machine.
- PowerExchange does not support CDC offload processing for capture registrations that you create from data maps that use any of the following options:
 - User access methods
 - User-defined fields that invoke programs by using the CALLPROG function
 - Record-level exits

Enabling Offload Processing for CDC Sessions

To use CDC offload processing, you must configure some PWX CDC connection attributes. Also, you must add the source-specific CAPI_CONNECTION statements to the DBMOVER configuration file on the PowerCenter Integration Service machine.

1. Configure attributes for offload processing on the PWX CDC Real Time application connection for the CDC session.

The following table describes the attributes that are required for offload processing:

Connection Attribute	Description
Location	Specifies the node name of the system on which the change data resides. This node name must match the name of a NODE statement in the dbmover.cfg configuration file on the PowerCenter Integration Service machine.
Offload Processing	Controls whether PowerExchange uses CDC offload processing. When offload processing is enabled, PowerExchange transfers column-level processing of the change data and any UOW Cleanser (UOWC) processing from the source system to the PowerCenter Integration Service machine. Options are: <ul style="list-style-type: none">- No. Disables offload processing.- Yes. Enables offload processing.- Auto. PowerExchange determines whether to enable or disable offload processing. Default is No.
CAPI Connection Name	Specifies the name of the source CAPI_CONNECTION statement in the dbmover.cfg on the PowerCenter Integration Service machine.

2. Copy the source-specific CAPI_CONNECTION statements from the DBMOVER configuration file on the source system to the dbmover.cfg configuration file on the PowerCenter Integration Service machine.
For DB2 for i5/OS sources, copy CAPI_CONNECTION statements of the types AS4J and UOWC.

Example of CDC Offload Processing with a DB2 for i5/OS Source

In this example, you enable a CDC session with a PWX CDC Real Time connection to use offload processing for change data extraction from a DB2 for i5/OS source.

The source change data remains on i5/OS system but all column-level and UOW Cleanser (UOWC) processing is offloaded to the PowerCenter Integration Service machine.

1. Copy the UOWC and AS4J CAPI_CONNECTION statements from the DBMOVER member in the *datilib*/CFG library on the i5/OS source system to the dbmover.cfg configuration file on the PowerCenter Integration Service machine.

This example uses the following CAPI_CONNECTION statements:

```
CAPI_CONNECTION=(NAME=i5UOWC,TYPE=(UOWC,CAPINAME=i5_AS4J,RSTRADV=600,MEMCACHE=20480))
CAPI_CONNECTION=(NAME=i5_AS4J,TYPE=(AS4J,JOURNAL=(JRN=PRODDATA/
PRODJRN),INST=PROD,EOF=N,STOPI=(CONT=5),LIBASUSER=Y))
```

2. Stop the CDC session.
3. Update the following attributes on the PWX CDC Real Time application connection for the CDC session:
 - For the **Offload Processing** attribute, select **Yes**.
 - For the **CAPI Connection Name** attribute, enter the name of the UOWC CAPI_CONNECTION statement. In this example, the name is i5UOWC.
4. Restart the CDC session.

Multithreaded Processing

Multithreaded processing uses multiple worker threads to distribute resource-intensive, column-level processing across multiple CPUs. Use multithreading if a single CPU cannot optimally handle extraction processing.

By default, PWXPC uses a single thread to process change data on the PowerCenter Integration Service machine. When you enable multithreading, PWXPC uses multiple threads to process change records.

Rules and Guidelines for Multithreaded Processing

Multithreaded processing can help improve performance for CDC sessions in specific situations.

Use the following rule and guidelines to determine when multithreaded processing is useful and how to set the **Worker Threads** attribute:

- Use multithreaded processing when the PWX reader thread of a CDC session uses 100% of a single CPU on a multiple-CPU server on the PowerCenter Integration Service machine. In this situation, multithreading improves throughput by spreading PowerExchange processing across multiple threads. Otherwise, multithreading does not improve throughput.
- For optimal performance, verify that the value of the **Worker Threads** attribute does not exceed the number of installed or available processors on the PowerCenter Integration Service machine.
- When defining the PWX CDC application connection, you must either set the **Location** attribute to "local" to enable the extraction to access the source locally, or set the **Offload Processing** attribute to **Yes** to offload extraction processing.
- If processing slows or hangs for CDC sessions that use multiple worker threads, increase the MAXTASKS value in the DBMOVER configuration file to help improve performance.

Enabling Multithreaded Processing for CDC Sessions

To use multithreaded processing, you must configure some PWX CDC connection attributes.

The following table describes the PWX CDC Real Time application connection attributes that are required to enable multithreaded processing for a CDC session:

Connection Attribute	Description
Worker Threads	Specifies the number of threads that PowerExchange uses on the PowerCenter Integration Service machine to process change data. Default is 0.
Array Size	If the Worker Threads value is greater than zero, specifies the size of the storage array, in number of records, for each thread. Default is 25.

APPENDIX A

DTL__CAPXTIMESTAMP Time Stamps

This appendix includes the following topic:

- [Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source, 153](#)

Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source

The time stamp that PowerExchange reports in the generated DTL__CAPXTIMESTAMP field in change records depends on the data source type and on certain parameter settings.

For PowerExchange data sources on z/OS, the `TIMESTAMP` parameter in the `UOWC CAPI_CONNECTION` controls the type of time stamp that PowerExchange reports in the `DTL__CAPXTIMESTAMP` field. If you set the `TIMESTAMP` parameter to `COMMIT`, PowerExchange reports the time stamp of the transaction commit on the source for all changes in the transaction. If you use the default parameter value of `LOG`, PowerExchange retrieves the time stamp from the source database logs. In this case, the time stamp type depends on the source type.

The following table describes the time stamps that PowerExchange reports when you use the default value of `LOG` for the `TIMESTAMP` parameter:

Data Source Type	Time Stamp Type
Adabas	The <code>HDDATE</code> time stamp from the <code>PLOG</code> block header, which indicates when the block was written. Note: In Adabas environments with a low level of update activity, the same time stamp might be reported for multiple updates that occurred at different times.
Datacom table-based CDC	The Coordinated Universal Time (UTC) time or local time when the change record was written to the Datacom <code>LXX</code> log. The <code>LOCAL_TIME</code> parameter in the <code>ECCR</code> configuration member, <code>ECCRDCMP</code> , controls whether the UTC or local time is used.
Db2 for i (i5/OS)	An i5/OS journal time stamp that reflects when the change was recorded in the journal.
Db2 for z/OS	The time at which the DB2 <code>ECCR</code> captured the change data record. Each record in a <code>UOW</code> has a different time stamp. Usually, this time stamp is a UTC value that reflects the time zone of the DB2 for z/OS system.

Data Source Type	Time Stamp Type
IDMS	The time at which the change data record was written to the IDMS log file. This time stamp is equivalent to the storeclock (STCK) time stamp. It does not reflect the local time zone.
IMS log-based CDC	The time at which the change was recorded in the IMS logs.
IMS synchronous CDC	The time at which the change occurred.
Batch VSAM and CICS/VSAM	The time at which the change record was captured. Each record in a UOW has a different time stamp. Usually, this time stamp is a UTC value.

For other data sources that do not use the UOWC CAPI_CONNECTION statement, PowerExchange determines the appropriate time stamp to report in the DTL__CAPXTIMESTAMP field. For PowerExchange Express CDC for Oracle sources, the TIME_STAMP_MODE parameter in the OPTIONS statement of the Express CDC configuration file controls the time stamp type.

The following table describes the time stamp types that PowerExchange reports for these data sources:

Data Source Type	Time Stamp Type
Db2 on Linux, UNIX, or Windows	The time stamp of the transaction commit. This time stamp is an ascending virtual time stamp (VTS) of the Db2 system, which usually corresponds to the UTC value.
Microsoft SQL Server	The time at which the change was written to the distribution database.
MySQL	The time stamp of the change event that MySQL recorded in the binary log.
PowerExchange Express CDC for Oracle	The time stamp type is controlled by the TIME_STAMP_MODE parameter setting in the OPTIONS statement of the Express CDC configuration file. <ul style="list-style-type: none"> - If you use the default value of LOGTIME, PowerExchange reports the time stamp of the change on source database, as recorded in the redo logs. This time stamp reflects the local time zone. - If you specify COMMITTIME, PowerExchange reports the time stamp of the transaction commit on the source database. - If you specify BEGINTIME, PowerExchange reports the time stamp of the begin UOW log record.
PostgreSQL	The time of the transaction commit.

INDEX

A

- adding libraries of CDC interest
 - with instance of same table [66](#)
- Application Name attribute [103](#)
- application names [114](#)
- architectural diagram
 - real-time extraction processing [12](#)
- architecture, PowerExchange CDC [12](#)
- AS400EVENTMSGQ statement
 - DBMOVER configuration file [18](#)
- AS400USRJRNCODE statement
 - DBMOVER configuration file [19](#)
- AS400USRMSGQ statement
 - DBMOVER configuration file [20](#)
- AS4J CAPI_CONNECTION parameters
 - parameters and syntax [21](#)

B

- backing up PowerExchange Condense files [51](#)
- batch extraction mode
 - use for terminating CDC sessions [137](#)
- batch mode
 - PowerExchange Condense [41](#)
- before indicator (BI) fields
 - use cases [90](#)

C

- CAPI Connection Name Override attribute [101](#)
- CAPI connection statements
 - AS4J parameters [21](#)
 - MEMCACHE parameter [143](#)
 - RSTRADV parameter [143](#)
 - UOWC parameters [26](#)
- CAPI_CONN_NAME statement
 - DBMOVER configuration file [31](#)
- CAPI_CONNECTION - AS4J statement
 - DBMOVER configuration file [21](#)
- CAPI_CONNECTION - UOWC statement
 - DBMOVER configuration file [26](#)
- CAPI_CONNECTION statements
 - sample [31](#)
- CDC sessions
 - adding sources with DTLUAPPL-generated CURRENT_RESTART tokens [139](#)
 - adding sources with special override CURRENT_RESTART tokens [138](#)
 - changing and restarting [137](#)
 - cold starting [134](#)
 - commit processing [93](#)
 - default restart points [115](#)
 - defining terminating conditions [137](#)

- CDC sessions (*continued*)
 - methods of starting [114](#)
 - monitoring in PowerCenter [128](#)
 - monitoring in PowerExchange [123](#)
 - multithreaded processing [151](#)
 - offload processing [149](#)
 - performance details in Workflow Monitor [129](#)
 - PowerExchange Logger for LUW logging of data from remote source [73](#)
 - processing of multiple source definitions [92](#)
 - recovering [140](#)
 - recovery example [140](#)
 - recovery start [135](#)
 - restart points for warm starts [115](#)
 - restart token file [118](#)
 - session and connection attributes for CDC [99](#)
 - start methods [133](#)
 - stop command processing [136](#)
 - stopping [136](#)
 - tuning [142](#)
 - tuning buffer memory [148](#)
 - tuning overview [95](#)
 - warm starting [134](#)
- change data capture (CDC)
 - architecture [12, 36](#)
 - overview [10](#)
 - stopping [64](#)
- change data extraction
 - connection attributes for Logger for LUW log files remote from the source [80](#)
 - creating restart tokens [116](#)
 - extraction modes [84](#)
 - monitoring in PowerCenter [128](#)
 - monitoring in PowerExchange [123](#)
 - multithreaded processing [151](#)
 - offload processing [149](#)
 - overview [83](#)
 - overview of extracting change data [96](#)
 - task flow [97](#)
 - testing extraction maps [98](#)
 - tuning CDC sessions [142](#)
- change indicator (CI) fields [90](#)
- close (pwxcmd) [33](#)
- closeforce (pwxcmd) [33](#)
- cold starts
 - CDC sessions [134](#)
 - determining restart tokens [115](#)
- command handler task
 - PowerExchange Condense [38](#)
- commit processing
 - commitment control attributes [106](#)
 - examples [109](#)
 - in CDC sessions [93](#)
 - target latency [105](#)
 - tuning [148](#)

- configuration of CDC
 - CDC with PowerExchange Condense [63](#)
 - CDC without PowerExchange Condense [62](#)
- connection attributes
 - Application Name [103](#)
 - attributes to set for CDC [99](#)
 - CAPI Connection Name Override [101](#)
 - commitment control attributes [106](#)
 - Event Table [104](#)
 - Idle Time [102](#)
 - Image Type attribute [101](#)
 - PWX Latency in seconds [105](#)
 - restart control attributes [103](#)
 - RestartToken File Folder [103](#)
 - RestartToken File Name [103](#)
- continuous extraction mode [84](#)
- continuous mode
 - PowerExchange Condense [41](#)
- controller task
 - PowerExchange Condense [37](#)
- CPX_DIR statement
 - DBMOVER configuration file [32](#)

D

- database row tests
 - testing data access with an extraction map [98](#)
- datatypes
 - supported for DB2 for i5/OS CDC [53](#)
- DBMOVER configuration file
 - APPBUFSIZE statement [143](#)
 - NODE and LISTENER buffer size parameters [143](#)
 - PowerExchange Logger for LUW logging of DB2 for i5/OS data [78](#)
 - PowerExchange Logger for LUW logging of remote source data [79](#)
 - TRACE statement [143](#)
- DBMOVER statements
 - AS400EVENTMSGQ [18](#)
 - AS400USRJRNCODE [19](#)
 - AS400USRMSGQ [20](#)
 - CAPL_CONN_NAME [31](#)
 - CAPL_CONNECTION - AS4J [21](#)
 - CAPL_CONNECTION - UOWC [26](#)
 - CPX_DIR [32](#)
 - RMTRDBDIRE [61](#)
 - RMTSYSNAME [61](#)
- deleting libraries of CDC interest
 - with instance of same table [66](#)
- diagrams
 - real-time extraction processing [12](#)
- DISPLAY ACTIVE command [126](#)
- DISPLAY STATUS command [49](#)
- displaystatus (pwxcmd) [49](#)
- DTL__BI__columnname column
 - described [84](#)
- DTL__CAPXACTION
 - described [84](#)
- DTL__CAPXCASDELIND
 - described [84](#)
- DTL__CAPXRESTART1 column
 - described [84](#)
 - displaying sequence token [117](#)
- DTL__CAPXRESTART2 column
 - described [84](#)
 - displaying restart token [117](#)
- DTL__CAPXROWID column
 - described [84](#)

- DTL__CAPXRRN column
 - described [84](#)
- DTL__CAPXTIMESTAMP column
 - described [84](#)
- DTL__CAPXTIMESTAMP field
 - types of reported time stamps by data source [153](#)
- DTL__CAPXUOW column
 - described [84](#)
- DTL__CAPXUSER column
 - described [84](#)
- DTL__CI__columnname column
 - described [84](#)
- DTL__ST column
 - described [84](#)
- DTLUAPPL utility
 - displaying restart tokens in generated columns [117](#)
- DTLUTSK utility
 - stopping CDC sessions [136](#)

E

- event message queue [61](#)
- Event Table attribute [104](#)
- event table processing
 - guidelines for using [104](#)
 - implementing [104](#)
 - use for terminating CDC sessions [137](#)
- extraction map columns, PowerExchange-generated
 - DTL__BI__columnname [84](#)
 - DTL__CAPXACTION [84](#)
 - DTL__CAPXCASDELIND [84](#)
 - DTL__CAPXRESTART1 [84](#)
 - DTL__CAPXRESTART2 [84](#)
 - DTL__CAPXROWID [84](#)
 - DTL__CAPXRRN [84](#)
 - DTL__CAPXTIMESTAMP [84](#)
 - DTL__CAPXUOW [84](#)
 - DTL__CAPXUSER [84](#)
 - DTL__CI__columnname [84](#)
 - DTL__ST [84](#)
- extraction maps
 - BI and CI fields [90](#)
 - PowerExchange-generated columns [84](#)
- extraction modes [84](#)
- extraction of change data
 - connection attributes for Logger for LUW log files remote from the source [80](#)
 - creating restart tokens [116](#)
 - extraction modes [84](#)
 - monitoring in PowerCenter [128](#)
 - monitoring in PowerExchange [123](#)
 - multithreaded processing [151](#)
 - offload processing [149](#)
 - overview [83](#)
 - overview of extracting change data [96](#)
 - task flow [97](#)
 - testing extraction maps [98](#)
 - tuning CDC sessions [142](#)
- extraction process [14](#)

F

- fileswitch (pwxcmd) [50](#)
- flush latency [105](#), [106](#)

I

- Idle Time attribute
 - use for terminating CDC sessions [137](#)
- Image Type attribute [101](#)
- integration with PowerCenter [14](#)

J

- journal receivers
 - CDC requirements [57](#)
- journals
 - using remote journals [59](#)

L

- listtask (pwxcmd) [33](#), [126](#)
- LISTTASK command [126](#)

M

- Maximum Rows Per commit attribute [106](#)
- Minimum Row Per commit attribute [106](#)
- monitoring CDC sessions
 - methods [123](#)
 - performance details in Workflow Monitor [129](#)
 - PowerCenter [128](#)
 - PowerCenter session log messages [129](#)
 - PowerExchange extraction statistics messages [124](#)
 - PowerExchange multithreaded processing statistics [125](#)
 - PowerExchange read progress messages [124](#)
 - viewing performance details in PowerCenter [131](#)
- multiple journal condensing
 - PowerExchange Condense [38](#)
- multiple-source processing
 - in CDC sessions [92](#)
- multithreaded processing
 - guidelines for usage [151](#)
 - overview [95](#), [151](#)
 - statistics messages [125](#)

O

- offload processing
 - enabling for CDC sessions [150](#)
 - example of offload processing [151](#)
 - overview [95](#), [149](#)
 - rules and guidelines [150](#)

P

- performance
 - CDC session performance details [131](#)
 - multithreaded processing [151](#)
- planning considerations [11](#)
- PowerCenter Client for PowerCenter (PWXPC) [14](#)
- PowerCenter integration with PowerExchange [14](#)
- PowerExchange CDC components [11](#)
- PowerExchange Condense
 - backing up output files [51](#)
 - batch mode [41](#)
 - CDC configuration steps with Condense [63](#)
 - checkpoint file to CDCT file synchronization [40](#)

PowerExchange Condense (continued)

- command handler task [38](#)
- condense cycles [37](#)
- continuous mode [41](#)
- controller task [37](#)
- displaying status of [49](#)
- files used [38](#)
- message log file [40](#)
- overview [13](#), [35](#)
- progress messages and tracing [49](#)
- shutting down [50](#)
- PowerExchange Condense configuration file
 - CAPT_IMAGE parameter [42](#)
 - CHKPT_BASENAME parameter [42](#)
 - CHKPT_NUM parameter [42](#)
 - COLL_END_LOG parameter [42](#)
 - COND_CDCT_RET_P parameter [42](#)
 - CONDENSE_SHUTDOWN_TIMEOUT parameter [42](#)
 - CONDENSENAME parameter [42](#)
 - DB_TYPE parameter [42](#)
 - DBID parameter [42](#)
 - FILE_SWITCH_CRIT parameter [42](#)
 - FILE_SWITCH_VAL parameter [42](#)
 - JRNL parameter [42](#)
 - KEY_CHANGE_ALW parameter [42](#)
 - NO_DATA_WAIT parameter [42](#)
 - NO_DATA_WAIT2 parameter [42](#)
 - OBJLOC parameter [42](#)
 - OPER_WTOR_ENABLED parameter [42](#)
 - REG parameter [42](#)
 - RESTART_TOKEN parameter [42](#)
 - SEQUENCE_TOKEN parameter [42](#)
 - SIGNALLING parameter [42](#)
 - VERBOSE parameter [42](#)
- PowerExchange Listener
 - DBMOVER statements [18](#)
 - DISPLAY ACTIVE command [126](#)
 - LISTTASK command [126](#)
 - message log file [16](#)
 - starting the PowerExchange Listener [32](#)
- PowerExchange Logger for Linux, UNIX, and Windows
 - capture registrations for logging data from remote sources [79](#)
 - configuration tasks for remote logging [76](#)
 - connection attributes for log files remote from the source [80](#)
 - DBMOVER statements for logging data from remote sources [79](#)
 - DBMOVER statements for logging DB2 data to remote log files [78](#)
 - example of logging data from a remote source [80](#)
 - logging data from remote sources [73](#)
 - logging DB2 for i5/OS data to a remote Logger [77](#)
 - rules and guidelines for logging data from a remote source [76](#)
- PowerExchange-generated extraction map columns
 - DTL__BI_columnname [84](#)
 - DTL__CAPXACTION [84](#)
 - DTL__CAPXCASDELIND [84](#)
 - DTL__CAPXRESTART1 [84](#), [117](#)
 - DTL__CAPXRESTART2 [84](#), [117](#)
 - DTL__CAPXROWID [84](#)
 - DTL__CAPXRRN [84](#)
 - DTL__CAPXTIMESTAMP [84](#)
 - DTL__CAPXUOW [84](#)
 - DTL__CAPXUSER [84](#)
 - DTL__CI_columnname [84](#)
 - DTL__ST [84](#)
- progress messages and tracing
 - PowerExchange Condense [49](#)
- PWX Latency in seconds attribute [105](#)
- pwxcmd
 - close command [33](#)

pwxcmd (continued)
closeforce command [33](#)
displaystatus command [49](#)
fileswitch command [50](#)
listtask command [33](#), [126](#)
shutcond command [50](#)
shutdown command [50](#)
stoptask command [33](#)
PWXPC [14](#)

R

real-time extraction mode [84](#)
Real-Time Flush Latency in milli-seconds attribute [106](#)
recovery
CDC sessions [140](#)
example of session recovery processing [140](#)
PM_REC_STATE table [112](#)
PM_RECOVERY table [112](#)
PM_TGT_RUN_ID table [112](#)
recovery information for nonrelational targets [113](#)
recovery state file for nonrelational targets [113](#)
recovery tables for relational targets [112](#)
recovery and restart processing [111](#)
recovery start
CDC sessions [135](#)
refreshing the CDC interest list
after adding or deleting libraries [66](#)
DBMOVER configuration parameters [68](#)
refreshing the CDC library interest list
considerations for library refresh requests [67](#)
issuing a refresh request [68](#)
troubleshooting [70](#)
refreshing the library interest list
monitoring the status of refresh processing [69](#)
remote journals [59](#)
restart
default restart points [115](#)
determining the journal receiver for CDC restart [65](#)
earliest restart points [115](#)
methods of starting CDC sessions [133](#)
restart points with null restart tokens [115](#)
warm starting CDC sessions [134](#)
warm starts
CDC sessions [134](#)
restart control options
Application Name connection attribute [103](#)
RestartToken File Folder attribute [103](#)
RestartToken File Name attribute [103](#)
restart of CDC sessions
processing by start type [114](#)
restart token file
\$PMRootDir/Restart [117](#)
example file [121](#)
explicit override statements [119](#)
overview [91](#)
special override statement [120](#)
statement types [118](#)
syntax rules and guidelines [118](#)
restart tokens
creating for extraction sessions [116](#)
determining for cold starts [115](#)
displaying in DTL_CAPXRESTART2 column [117](#)
overview [91](#)
recovery state file [113](#)
recovery state table [112](#)
RestartToken File Folder attribute [103](#)

RestartToken File Name attribute [103](#)
RMTRDBDIRE statement [61](#)
RMTSYSNAME statement [61](#)
row tests
testing data access with an extraction map [98](#)
RTVSQLSTMT command
generating SQL statement to re-create objects for troubleshooting [71](#)

S

sample CAPI_CONNECTION statements [31](#)
security requirements [55](#)
security, journal [56](#)
sequence tokens
displaying in DTL_CAPXRESTART1 column [117](#)
session attributes
attributes to set for CDC [99](#)
SHOW_THREAD_PERF parameter [125](#)
shutcond (pwxcmd) [50](#)
shutdown (pwxcmd) [50](#)
shutting down
PowerExchange Condense [50](#)
SNDPWXCMD
FILESWITCH [50](#)
SHUTCOND [50](#)
SHUTDOWN [50](#)
SNDPWXCMD command [13](#)
source table definitions
DDL changes [64](#)
stoptask (pwxcmd) [33](#)
STOPTASK command
stopping CDC sessions [136](#)

T

table definitions
DDL changes [64](#)
target latency [105](#)
task flow
extracting change data [97](#)
terminating conditions
Idle Time attribute for CDC sessions [102](#)
troubleshooting
using the RTVSQLSTMT command to re-create source or target environments [71](#)
tuning
overview of tuning options [95](#)
tuning CDC sessions
APPBUFSIZE statement [143](#)
buffer memory [148](#)
CAPI_CONNECTION MEMCACHE parameter [143](#)
CAPI_CONNECTION RSTRADV parameter [143](#)
commit processing attributes [148](#)
DBMOVER tuning parameters [143](#)
methods [142](#)
NODE and LISTENER buffer size parameters [143](#)
PowerCenter CDC connection attributes [146](#)
TRACE statement [143](#)

U

UOW Count attribute [106](#)
UOWC CAPI_CONNECTION parameters
parameters and syntax [26](#)

usage scenarios [10](#)

use cases [90](#)

W

warm starts

restart points used [115](#)