



Informatica® PowerExchange
10.5.1

CDC Guide for Linux, UNIX, and Windows

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.

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

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

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

Publication Date: 2022-08-09

Table of Contents

Preface	11
Informatica Resources.	11
Informatica Network.	11
Informatica Knowledge Base.	11
Informatica Documentation.	12
Informatica Product Availability Matrices.	12
Informatica Velocity.	12
Informatica Marketplace.	12
Informatica Global Customer Support.	12
 Part I: PowerExchange CDC Introduction.....	13
 Chapter 1: Change Data Capture Introduction.	14
PowerExchange CDC Overview.	14
Change Data Capture.	14
Change Data Extraction and Apply.	15
PowerExchange CDC Data Sources.	16
Db2 Data Sources.	17
Microsoft SQL Server Data Sources.	17
MySQL Data Sources.	17
Oracle Data Sources.	18
PostgreSQL Data Sources.	19
SAP HANA Data Sources.	20
IBM i and z/OS Data Sources with Offload Processing.	20
PowerExchange CDC Components.	20
PowerExchange Listener.	20
PowerExchange Logger for Linux, UNIX, and Windows.	21
PowerExchange Navigator.	21
PowerExchange CDC Architecture.	22
PowerExchange Integration with PowerCenter.	24
Environment Variable Incompatibilities Between PowerExchange and PowerCenter.	24
Summary of CDC Implementation Tasks.	25
 Part II: PowerExchange CDC Components.....	26
 Chapter 2: PowerExchange Listener.	27
PowerExchange Listener Overview.	27
Customizing the dbmover Configuration File for CDC.	27
CAPI_CONNECTION Statements.	28
CAPI_SRC_DFLT Statement.	29

CAPT_PATH Statement.	31
CAPT_XTRA Statement.	31
Starting the PowerExchange Listener.	32
Starting the PowerExchange Listener on Linux or UNIX.	32
Starting the PowerExchange Listener on Windows.	33
Stopping the PowerExchange Listener.	34
Displaying Active PowerExchange Listener Tasks.	34
 Chapter 3: PowerExchange Logger for Linux, UNIX, and Windows.	36
PowerExchange Logger Overview.	36
PowerExchange Logger Tasks.	38
PowerExchange Logger Files.	38
CDCT File.	38
PowerExchange Logger Log Files.	39
Lock Files.	40
Message Log Files.	41
File Switches.	41
PowerExchange Logger Operational Modes.	42
Continuous Mode.	42
Batch Mode.	43
PowerExchange Logger Usage Considerations.	43
Logging Data for IBM i or z/OS Sources to Remote PowerExchange Logger Logs.	44
Configuring the PowerExchange Logger.	44
Enabling a Capture Registration for PowerExchange Logger Use.	44
Customizing the PowerExchange Logger Configuration File.	45
Customizing the dbmover Configuration File for the PowerExchange Logger.	59
Using PowerExchange Logger Group Definitions.	62
Starting the PowerExchange Logger.	66
PWXCL Syntax and Parameters.	66
How the PowerExchange Logger Determines the Start Point for a Cold Start.	69
Cold Starting the PowerExchange Logger	70
Managing the PowerExchange Logger.	71
Controlling and Stopping PowerExchange Logger Processing.	71
Monitoring the PowerExchange Logger.	72
Determining If the PowerExchange Logger Captured Changes.	75
PowerExchange Logger Verbose Messages.	75
PWXUCDCT Commands for Maintaining the PowerExchange Logger CDCT and Log Files.	76
Backing Up PowerExchange Logger Files.	78
Re-creating the CDCT File After a Failure.	79

Part III: PowerExchange CDC Data Sources.....	80
Chapter 4: Db2 CDC on Linux, UNIX, or Windows.....	81
Db2 CDC on Linux, UNIX, or Windows Overview.	81
Planning for Db2 CDC.	82
Prerequisites.	82
Required User Authority.	82
Db2 Datatypes Supported for CDC.	83
Db2 CDC Considerations.	84
Configuring Db2 for CDC.	85
Configuring PowerExchange for Db2 CDC.	85
Configuring PowerExchange CDC without the PowerExchange Logger.	86
Configuring PowerExchange CDC with the PowerExchange Logger.	86
Creating the Capture Catalog Table.	87
Initializing the Capture Catalog Table.	87
Customizing the dbmover Configuration File for Db2 CDC.	88
Using a Db2 Data Map.	94
Task Flow for Db2 Data Map Use.	94
Managing Db2 CDC.	94
Stopping Db2 CDC.	95
Changing a Db2 Source Table Definition.	95
Reconfiguring a Partitioned Database or Database Partition Group.	96
Db2 CDC Troubleshooting.	97
Workaround for SQL1224 Error on AIX.	97
Chapter 5: Microsoft SQL Server CDC.....	99
Microsoft SQL Server CDC Overview.	99
Planning for SQL Server CDC.	100
SQL Server CDC Prerequisites.	100
Required User Authority for SQL Server CDC.	100
SQL Server Datatypes Supported for CDC.	101
SQL Server CDC Operational Considerations.	102
Extracting Data for Multiple Publication Databases.	104
Configuring SQL Server for CDC.	105
Configuring PowerExchange for SQL Server CDC.	105
Configuring PowerExchange CDC without the PowerExchange Logger.	105
Configuring PowerExchange CDC with the PowerExchange Logger.	106
Customizing the dbmover Configuration File for SQL Server CDC.	106
Managing SQL Server CDC.	113
Disabling Publication of Change Data for a SQL Server Source.	113
Changing a SQL Server Source Table Definition.	113
Changing the MULTIPUB Parameter Setting After Running Extractions.	114

Chapter 6: MySQL CDC.	116
MySQL CDC Overview.	116
About the Binary Log File.	117
DDL-Updated Catalog of MySQL Source Table Definitions.	117
MySQL CDC Operational Considerations.	118
MySQL Datatypes Supported for CDC.	121
Implementation Task Flow.	122
Preparing MySQL Sources.	123
Configuring PowerExchange for MySQL CDC.	124
Configuring the dbmover Configuration File.	124
Creating the DDL-Updated Catalog Tables.	128
Creating a Snapshot of Source Table Definitions.	129
Managing MySQL CDC.	129
Stopping Change Data Capture for a MySQL Source Table.	129
Stopping MySQL CDC Processing Temporarily.	130
Changing the Structure of a MySQL Source Table.	130
Changing the Binary Log Location or Base Name.	131
 Chapter 7: Express CDC for Oracle.	 132
Express CDC for Oracle Overview.	132
PowerExchange Express CDC for Oracle Benefits.	133
PowerExchange Express CDC for Oracle Architecture.	133
Gathering Information About the CDC Environment.	137
Express CDC Considerations.	139
PowerExchange Express CDC for Oracle Restrictions.	139
Oracle Datatypes Supported for Express CDC.	140
Operational Considerations.	142
Handling Oracle LOB Data.	146
Performance Considerations.	148
RAC Considerations.	148
ASM Considerations.	149
Checkpointing In-flight Transactions.	151
Dynamic Dictionary.	153
Oracle Data Guard Physical Standby Databases as Sources.	155
Oracle Multitenant Pluggable Databases as Sources.	159
Amazon RDS for Oracle Database Instances as Sources.	160
Source Tablespaces That Use Transparent Data Encryption.	163
Limiting the Redo Logs From Which Express CDC Reads Changes.	163
Implementation Task Flow.	164
Configuring Oracle for Express CDC.	165
Specifying an Archive Log Destination.	165
Enabling ARCHIVELOG Mode.	166

Creating an Oracle User and Granting User Privileges.	166
Creating an ASM User.	168
Enabling Oracle Minimal Global Supplemental Logging.	169
Configuring PowerExchange for Express CDC.	169
Configuring the dbmover Configuration File	171
Customizing the PowerExchange Express CDC for Oracle Configuration File.	179
Managing PowerExchange Express CDC for Oracle.	201
Monitoring PowerExchange Express CDC for Oracle.	202
Adding Another Capture Registration.	202
Stopping CDC Processing for a Table.	203
Stopping CDC Processing Temporarily.	203
Changing the Structure of an Oracle Source Table	203
Reporting DDL Operations for Registered Oracle Source Tables.	204
Chapter 8: PostgreSQL CDC.	206
PostgreSQL CDC Overview.	206
PostgreSQL CDC Considerations.	207
PostgreSQL Datatypes Supported for CDC.	208
Implementation Task Flow.	210
Preparing PostgreSQL CDC Sources.	211
Manually Creating the Replication Store Table.	212
Configuring PowerExchange for PostgreSQL CDC.	212
Configuring the dbmover Configuration File.	212
Managing PostgreSQL CDC.	215
Stopping Change Data Capture for a PostgreSQL Table.	216
Stopping PostgreSQL CDC Processing Temporarily.	216
Changing the Structure of a PostgreSQL Source Table.	216
Chapter 9: SAP HANA CDC.	218
SAP HANA CDC Overview.	218
SAP HANA CDC Considerations.	218
SAP HANA Datatypes Supported for CDC.	219
SAP HANA CDC Log Tables.	220
Implementation Task Flow.	220
Preparing the SAP HANA Server for PowerExchange CDC.	221
Configuring ODBC on Linux.	221
Configuring ODBC on Windows.	222
Configuring JDBC on Windows or Linux.	223
Preparing the SAP HANA Database for PowerExchange CDC.	223
Creating a User for CDC.	224
Granting Additional Permissions.	224
Manually Creating the Log Tables.	224
Manually Creating Triggers.	224

Configuring PowerExchange for SAP HANA CDC.	225
Configuring the dbmover Configuration File.	225
CAPI_CONNECTION - HANA Statement.	225
Managing SAP HANA CDC.	227
Stopping Change Data Capture for a SAP HANA Source Table.	227
Stopping SAP HANA CDC Processing Temporarily.	227
Chapter 10: Remote Logging of Data.	228
Remote Logging Overview.	228
Remote Logging of Data from Sources on IBM i or z/OS Systems.	229
Remote Logging of Data from Linux, UNIX, or Windows Sources.	230
Requirements for Capture Registrations.	233
Security Settings for Data from z/OS Sources.	233
Configuration Tasks for Remote Logging.	234
Customizing the PowerExchange Logger Configuration File for Remote Logging.	234
Customizing the dbmover Configuration File on the System to Which Data Is Logged.	238
Customizing the dbmover Configuration File on the PowerCenter Integration Service System.	239
Configuring Capture Registrations for the PowerExchange Logger.	239
Configuring PowerCenter Connection Attributes for Extracting Data from the Log Files.	239
Example of Remote Logging from a z/OS Data Source.	240
Example of Remote Logging from a Db2 for i Data Source.	242
Example of Remote Logging from a PowerExchange Express CDC for Oracle Source.	244
Part IV: Change Data Extraction.	247
Chapter 11: Introduction to Change Data Extraction.	248
Change Data Extraction Overview.	248
Extraction Modes.	249
PowerExchange-Generated Columns in Extraction Maps.	250
Uses of BI and CI Fields in Extraction Maps.	256
Restart Tokens and the Restart Token File.	257
Multiple-Source Processing in CDC Sessions.	258
Commit Processing with PWXPC.	259
Tuning Options.	261
Chapter 12: Extracting Change Data.	262
Overview of Extracting Change Data.	262
Security Considerations for Extracting z/OS Data.	263
Task Flow for Extracting Change Data.	264
Testing an Extraction Map.	264
Configuring PowerCenter CDC Sessions.	266
Changing Default Values for Session and Connection Attributes.	266
Configuring Application Connection Attributes.	267

Examples of Controlling Commit Processing.	275
Recovery and Restart Processing for CDC Sessions.	277
PowerCenter Recovery Tables for Relational Targets.	278
PowerCenter Recovery Files for Nonrelational Targets.	279
Application Names.	280
Restart Processing for CDC Sessions by Start Type.	280
Creating Restart Tokens for Extractions.	283
Displaying Restart Tokens.	283
Configuring the Restart Token File.	284
Restart Token File Statements.	285
Example Restart Token File.	288
Chapter 13: Managing Change Data Extractions.	289
Starting PowerCenter CDC Sessions.	289
Cold Start Processing.	290
Warm Start Processing.	290
Recovery Processing.	291
Stopping PowerCenter CDC Sessions.	291
Stop Command Processing.	292
Terminating Conditions.	292
Changing PowerCenter CDC Sessions.	293
Examples of Adding Sources and Creating Restart Tokens.	293
Recovering PowerCenter CDC Sessions.	295
Example of Session Recovery.	296
Part V: Monitoring and Tuning.	298
Chapter 14: Monitoring CDC Sessions.	299
Monitoring Overview.	299
Monitoring CDC Sessions in PowerExchange.	299
Read Progress Messages.	300
Extraction Statistics Messages.	300
Multithreaded Processing Statistics.	301
PowerExchange Listener DISPLAY ACTIVE or LISTTASK Command.	302
PowerExchange Listener DISPLAYSTATS Command.	302
PowerExchange Logger for Linux, UNIX, and Windows Monitoring Statistics	304
Monitoring CDC Sessions in PowerCenter.	307
Session Log Messages.	307
Performance Details in Workflow Monitor.	308
Viewing Performance Details in Workflow Monitor.	310
Chapter 15: Tuning CDC Sessions.	311
Tuning Overview.	311

PowerExchange DBMOVER Statements for Tuning CDC Sessions.	312
PowerCenter Connection Attributes and Session Properties.	315
PowerCenter Connection Attributes for Tuning CDC Sessions	315
PowerCenter Session Properties for Tuning Buffer Memory.	318
CDC Offload Processing.	318
Rules and Guidelines for CDC Offload Processing.	319
Enabling Offload Processing for CDC Sessions.	319
Example of CDC Offload Processing with an Oracle Source.	320
Multithreaded Processing.	320
Rules and Guidelines for Multithreaded Processing.	321
Enabling Multithreaded Processing for CDC Sessions.	321
Appendix A: DTL__CAPXTIMESTAMP Time Stamps.....	322
Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source.	322
Index.	324

Preface

Use the *Informatica® PowerExchange® CDC Guide for Linux, UNIX, and Windows* to learn how to configure, implement, and manage PowerExchange change data capture (CDC) from relational data sources on Linux, UNIX, and Windows systems. This guide also describes how you can use the remote logging feature of the PowerExchange Logger for Linux, UNIX, and Windows to process change data from remote Db2 for i (i5/OS) and Db2 for z/OS data sources.

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. Setting incompatibilities include, but may not be restricted to, the inclusion of parameters such as “CDC_EXCLUDE_JOBNAME” parameters which prevent data from the job being captured and populated to the CA-Datcom maintained CDC 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 eSupport option.

Part I: PowerExchange CDC Introduction

This part contains the following chapter:

- [Change Data Capture Introduction, 14](#)

CHAPTER 1

Change Data Capture Introduction

This chapter includes the following topics:

- [PowerExchange CDC Overview, 14](#)
- [PowerExchange CDC Data Sources, 16](#)
- [PowerExchange CDC Components, 20](#)
- [PowerExchange CDC Architecture, 22](#)
- [PowerExchange Integration with PowerCenter, 24](#)
- [Environment Variable Incompatibilities Between PowerExchange and PowerCenter, 24](#)
- [Summary of CDC Implementation Tasks, 25](#)

PowerExchange CDC Overview

PowerExchange change data capture (CDC) works in conjunction with PowerCenter® to capture change data from source tables and replicate the changes to target tables and files. This guide describes PowerExchange CDC for supported relational database sources on Linux, UNIX, or Windows operating systems.

After materializing target tables or files with PowerExchange bulk data movement, you can use PowerExchange CDC to synchronize the targets with their corresponding source tables. Synchronization is faster when you replicate only the change data rather than all of the data.

The change data replication process consists of following high-level steps:

1. *Change data capture.* PowerExchange captures change data for the source tables. PowerExchange can read change data directly from the RDBMS log files or database. Optionally, you can use the PowerExchange Logger for Linux, UNIX, and Windows to capture change data to its log files.
2. *Change data extraction.* PowerExchange, in conjunction with PowerCenter, extracts captured change data for movement to the target.
3. *Change data apply.* PowerExchange, in conjunction with PowerCenter, transforms and applies the extracted change data to target tables or files.

Change Data Capture

PowerExchange can capture change data directly from Db2 database logs, Microsoft SQL Server distribution databases, MySQL binary logs, Oracle redo logs, or a PostgreSQL replication slot. Because SAP HANA does not provide direct access to its log files, the capture process uses triggers on source tables to determine when to capture change data.

If you use the offloading feature in combination with the PowerExchange Logger for Linux, UNIX, and Windows, the PowerExchange Logger process can capture change data from these relational data sources and also from data sources on an IBM i (i5/OS) or z/OS system.

If you do not retain the source database log files long enough for CDC processing to complete, use the PowerExchange Logger for Linux, UNIX, and Windows. The PowerExchange Logger writes change data to its log files. PowerExchange can then extract change data from the PowerExchange Logger log files rather than from the database log files.

For each source table, you must define a capture registration in the PowerExchange Navigator. The capture registration provides metadata for the columns that are selected for change capture.

PowerExchange captures changes that result from successful SQL INSERT, DELETE, and UPDATE operations. Depending on the statement type, PowerExchange captures the following data images:

- For INSERTS, PowerExchange captures after images only. An *after image* reflects a row just after an INSERT operation. PowerExchange passes these changes as INSERTs to PowerCenter.
- For DELETES, PowerExchange captures before images only. A *before image* reflects a row just prior to the last DELETE operation. PowerExchange passes these changes as DELETES to PowerCenter.
- For UPDATES, PowerExchange captures the following image types:
 - Both before and after images if you select an image type of "BA" in the CDC application connection attributes for PowerCenter. PowerExchange passes an UPDATE to PowerCenter as a DELETE of the before-image data followed by an INSERT of the after-image data.
 - After images if you select an image type of "AI" in the CDC application connection attributes. PowerExchange passes only the after-image data for an updated row, unless you also request before-image data. PowerExchange passes an UPDATE to PowerCenter as an UPDATE or INSERT.

Change Data Extraction and Apply

PowerExchange works with PowerCenter to extract change data and write 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.

When you create a capture registration for a source table, the PowerExchange Navigator generates a corresponding extraction map and application name for the extraction. The extraction map describes the columns for which to extract change data. You can edit the extraction map to remove columns from extraction processing. Also, you can create alternative extraction maps, each for a subset of the columns that are registered for capture. For Db2 data sources only, you can create a data map if you have user-defined or multi-field columns for which you want to manipulate data before loading it to the target.

From PowerCenter, you run a CDC workflow and session that extracts and applies change data. To define a data source in PowerCenter, you can import the extraction map or import the table definition from the source database through PowerExchange. For Db2 only, you can import a Db2 data map instead of the extraction map. In most situations, Informatica recommends that you import the extraction map.

Also, you must define a mapping, session, and workflow in PowerCenter. Optionally, you can include transformations in the mapping to manipulate the change data. When you define a CDC session, you must specify a connection type. The connection type determines the extraction mode and access method that PowerExchange uses to extract data.

To extract change data directly from Db2 transaction logs, the Microsoft SQL Server distribution database, the MySQL binary log, Oracle redo log files, or PostgreSQL replication slot, you must use the real-time extraction mode. To extract change data from PowerExchange Logger log files, you can use either the batch extraction mode or continuous extraction mode.

The following table describes these extraction modes:

Extraction Mode	Description
Real-time extraction mode	Reads change data directly from the database log files in near real time, on an ongoing basis. When the PowerExchange Listener receives an extraction request, it pulls the change data from the log files and transmits the data to PowerCenter for extraction and apply processing. This mode provides the lowest latency for change data extraction but potentially the highest impact on system resources.
Batch extraction mode	Reads change data from PowerExchange Logger log files that are in a closed state when an extraction request is made. After processing the log files, the extraction request ends. This mode provides the highest latency for change data extraction but minimizes the impact on system resources.
Continuous extraction mode	Reads change data continuously from open and closed PowerExchange Logger log files in near real time. This mode also minimizes database log accesses and the log retention period that is required for CDC.

To initiate change data extraction and apply processing, run a CDC workflow and session from PowerCenter.

During extraction processing, PowerExchange extracts changes from the change stream in chronological order based on the unit of work (UOW) end time. PowerExchange passes only the successfully committed changes to PowerCenter for processing. PowerExchange does not pass ABORTed or UNDO changes. If you are capturing changes from Db2 database logs or Oracle redo logs, changes that were contiguous in the change stream might not be contiguous in the reconstructed UOW that PowerExchange passes to PowerCenter.

To properly resume extraction processing, PowerExchange maintains restart tokens for each source table. Restart tokens are used for all extraction modes. To generate current restart tokens, you can use the PowerExchange Navigator, the special override statement in the restart token file, or the DTLUAPPL utility.

RELATED TOPICS:

- [“Introduction to Change Data Extraction” on page 248](#)

PowerExchange CDC Data Sources

PowerExchange can capture change data from many types of data sources on Linux, UNIX, or Windows systems.

The following types of data sources are supported:

- Db2 for Linux, UNIX, and Windows
- Microsoft SQL Server
- MySQL
- Oracle
- PostgreSQL
- SAP HANA

In the PowerExchange Navigator, you must create a capture registration for each source table. The PowerExchange Navigator generates a corresponding extraction map and application name. You import the extraction map into PowerCenter to define the source definition for extraction and apply processing.

If you use the PowerExchange Logger for Linux, UNIX, and Windows in combination with the offloading feature, you can also process change data from remote data sources on IBM i (i5/OS) and z/OS systems.

Db2 Data Sources

PowerExchange captures change data from Db2 recovery log files for Db2 source tables on a Linux, UNIX, or Windows system.

For CDC to work, archive logging must be active for the database. Also, you must create a PowerExchange capture catalog table in the source database. The capture catalog table stores information about the source tables and columns, including Db2 log positioning information.

If you have a source table with user-defined fields or multi-field columns, you can create a data map to manipulate these fields with expressions. For example, you might want to create data map to manipulate packed data in a CHAR column. If you create a data map, you must still create a capture registration and merge the data map with the extraction map that is generated for the capture registration.

RELATED TOPICS:

- [“Db2 CDC on Linux, UNIX, or Windows” on page 81](#)

Microsoft SQL Server Data Sources

PowerExchange CDC uses Microsoft SQL Server transactional replication technology to access data in SQL Server distribution databases. For CDC to work, you must enable SQL Server Replication on the system from which change data is captured. Also, verify that each source table in the distribution database has a primary key. If your database has a high volume of change activity, use a distributed server as the host of the distribution database. When the extraction process runs, the Microsoft SQL Server Agent must also be running.

RELATED TOPICS:

- [“Microsoft SQL Server CDC” on page 99](#)

MySQL Data Sources

PowerExchange can capture change events for MySQL source tables from MySQL binary logs by using the MySQL binary log reader, mysqlbinlog.

The binary log reader and PowerExchange capture process must run on the same Linux or Windows machine. This machine can be remote from the MySQL source database server. Use of the PowerExchange Logger for Linux, UNIX, and Windows is optional.

PowerExchange extracts the change records from the real-time change stream or from PowerExchange Logger for Linux, UNIX, and Windows log files and makes the changes available to PowerCenter CDC sessions.

PowerExchange uses the DataDirect ODBC driver for MySQL to retrieve change data and source metadata from the MySQL database server. This ODBC driver is included in the PowerExchange installation.

RELATED TOPICS:

- [“MySQL CDC” on page 116](#)

Oracle Data Sources

PowerExchange Express CDC for Oracle captures change data from Oracle sources. Informatica strongly recommends that you use it with the PowerExchange Logger for Linux, UNIX, and Windows.

The following table identifies how PowerExchange Express CDC supports selected Oracle features:

Feature	PowerExchange Express CDC for Oracle Support
PowerExchange Logger for Linux, UNIX, and Windows use	Use of the PowerExchange Logger is strongly recommended
Multithreaded processing	Yes
CDC processing speed	Fast
Oracle redo logs processing	Reads change data directly from active logs and from archived redo logs, including copies of the archived logs that were created outside of the Oracle archive process.
Oracle data dictionary use	Transparently stores the data dictionary in memory.
Checkpointing	Periodically writes state information for in-flight transactions to tables or file-system files for checkpointing.
Capture from Oracle RAC and ASM environments	Yes
Capture from Oracle Data Guard environments	Can capture data from Data Guard logical and physical standby databases.
Capture from Oracle multitenant environments	Can capture data from a single pluggable database (PDB)
Capture from Amazon Elastic Compute Cloud (EC2) and Amazon RDS for Oracle environments	Yes
Capture from Oracle Exadata machines	Yes
Capture from tables that use Oracle Exadata Hybrid Columnar Compression (EHCC)	Yes, except direct-path operations.
Capture from objects that use Oracle Advanced Security Transparent Data Encryption (TDE)	Can capture data from encrypted tablespaces but not from encrypted columns.
Capture from objects that use Oracle Advanced Compression	Can capture conventional and direct-path DML operations from tables and table partitions and subpartitions that use Advanced Compression.
Direct-path operations	Can capture direct-path operations except for tables that use EHCC.

Feature	PowerExchange Express CDC for Oracle Support
DDL operations	Does not capture CREATE TABLE...AS SELECT operations because the table cannot be registered for CDC. Tolerates ALTER TABLE ADD, ALTER TABLE ADD PARTITION, ALTER TABLE ADD CONSTRAINT, CREATE USER, ALTER USER, and DROP USER operations. The DDL change is not captured but CDC processing of other changes continues.
Oracle RESETLOGS events	Can capture data across a RESETLOGS boundary in the archive logs.
EXCHANGE PARTITION operations	Does not capture the exchange operation or any rows it generates. Can capture subsequent DML changes on the table or partition that was the target of the exchange, if registered for CDC.
Source data loaded with the SQL*Loader utility	Can capture the data if the utility load type is conventional path and the load method is Insert, with the exception of data that is compressed with EHCC.
Capture from index-organized tables (IOTs)	Yes
Capture from materialized views	Can capture data from the master tables that underlie the views.
Capture from tables that use system partitioning or reference partitioning	Yes
Capture from tables in a sorted hash cluster	No
Capture from virtual columns with derived data	No
Capture from LOB columns	Yes

RELATED TOPICS:

- [“Express CDC for Oracle” on page 132](#)

PostgreSQL Data Sources

PowerExchange reads DML and Truncate operations for registered PostgreSQL source tables from a PostgreSQL logical replication slot for PowerExchange use. PowerExchange then persists the changes in the PowerExchange *replication store table* in the PostgreSQL database.

PowerExchange automatically creates the replication slot and replication store table when capture processing starts. If you want to customize the table for your environment, you can manually create the table before starting change capture.

Use of the PowerExchange Logger for Linux, UNIX, and Windows is optional. When a CDC session starts, PowerExchange can retrieve change records from either the replication slot in real time or from PowerExchange Logger for Linux, UNIX, and Windows log files.

PowerExchange captures Begin, Insert, Update, Delete, Commit, and Truncate operations for source transactions. PowerExchange does not capture DDL changes, other than Truncate operations, that are made to the PostgreSQL source.

Each source table must have a primary key.

RELATED TOPICS:

- [“PostgreSQL CDC” on page 206](#)

SAP HANA Data Sources

PowerExchange CDC for SAP HANA captures inserts, updates, and deletes from SAP HANA tables on Windows or Linux. Because SAP HANA does not provide direct access to its log files, the capture process uses triggers on source tables to determine when to capture change data.

The PowerExchange Navigator can optionally create scripts for the PKLOG and PROCESSED log tables and trigger DDL when you create capture registrations for the source tables. If you do not use the Navigator to create the tables, PowerExchange creates the log tables and triggers when the extraction process runs.

The triggers write insert, update, and delete changes to the PKLOG transaction log table. These changes are read in table order and are then available for extraction to PowerCenter CDC sessions or PowerExchange CDC Publisher. PowerExchange provides only after-image records for SAP HANA CDC.

Each source table must have a primary key.

IBM i and z/OS Data Sources with Offload Processing

You can use CDC offload processing in combination with the PowerExchange Logger for Linux, UNIX, and Windows to log change data from data sources on systems other than the system where the PowerExchange Logger runs.

With offload processing, a PowerExchange Logger process on Linux, UNIX, and Windows can log change data from IBM i (i5/OS) and z/OS systems as well as from other Linux, UNIX, or Windows systems. For example, a PowerExchange Logger process can log change data from a Db2 instance on z/OS.

PowerExchange CDC Components

Several PowerExchange components are involved in change data capture (CDC).

These components are:

- **PowerExchange Listener.** Required, unless PowerExchange and the PowerCenter Integration Service are installed on the same physical machine.
- **PowerExchange Logger for Linux, UNIX, and Windows.** Optional.
- **PowerExchange Navigator.** Required.

PowerExchange Listener

The PowerExchange Listener manages capture registrations and extraction maps for all CDC data sources. It also manages data maps if you create them for Db2 tables. The PowerExchange Listener maintains this information in the following files:

- CCT file for capture registrations
- CAMAPS directory for extraction maps
- DATAMAPS directory for Db2 data maps

The PowerExchange Listener also handles PowerCenter extraction requests for both change data replication and bulk data movement.

When you create, edit, or delete capture registrations or extraction maps in the PowerExchange Navigator, the PowerExchange Navigator uses the location value in the registration group and extraction group to contact the PowerExchange Listener. This location corresponds to a NODE statement in the dbmover.cfg file. For example, when you open a registration group for a RDBMS instance, the PowerExchange Navigator communicates with the PowerExchange Listener to get all capture registrations defined for that instance.

A PowerExchange Listener is not required if PowerExchange and the PowerCenter Integration Service run on the same physical machine.

RELATED TOPICS:

- [“PowerExchange Listener” on page 27](#)

PowerExchange Logger for Linux, UNIX, and Windows

The PowerExchange Logger for Linux, UNIX, and Windows can capture change data from Db2 database logs, the Microsoft SQL Server distribution database, MySQL binary logs, Oracle redo logs, or the PostgreSQL table in which PowerExchange stores data from a PostgreSQL logical replication slot. The PowerExchange Logger then writes the captured data to PowerExchange Logger log files.

Use of the PowerExchange Logger is optional. To use the PowerExchange Logger, run one PowerExchange Logger process for each database type and instance. The PowerExchange Logger writes all successful UOWs in chronological order based on end time to its log files. This practice maintains transactional integrity. You can extract the change data from the PowerExchange Logger log files in either batch or continuous mode.

Benefits of the PowerExchange Logger include:

- Source database overhead is reduced because PowerExchange makes fewer accesses to the source log files or database to read change data. For Oracle, this overhead reduction can be significant.
- You do not need to retain the source RDBMS log files longer than normal for CDC.
- PowerExchange does not need to reposition its point in the logs from which to resume reading data. This feature can significantly reduce restart times.

Tip: Informatica strongly recommends that you use the PowerExchange Logger rather than real-time extraction mode for PowerExchange Express CDC for Oracle sources.

RELATED TOPICS:

- [“PowerExchange Logger for Linux, UNIX, and Windows” on page 36](#)

PowerExchange Navigator

The PowerExchange Navigator is the graphical user interface from which you define and manage capture registrations, extraction maps, and data maps.

You must define a capture registration for each source table. The PowerExchange Navigator generates the corresponding extraction map. For Db2 sources, you can also define data maps if you need to perform column-level processing, such as adding user-defined columns and building expressions to populate them. You can import the extraction maps into PowerCenter so that they can be used for moving change data to the target.

Note: PowerExchange uses SQL Server services when creating capture registrations. For Db2, Microsoft SQL Server, MySQL, Oracle, PostgreSQL, and SAP HANA data sources, you do not need the RDBMS client software. Instead, from the PowerExchange Navigator, you can point to the PowerExchange Listener on the

machine that contains the source Db2 database, Microsoft SQL Server server, MySQL server, Oracle instance, PostgreSQL server, or SAP HANA server.

For more information about the PowerExchange Navigator, see the *PowerExchange Navigator User Guide*.

PowerExchange CDC Architecture

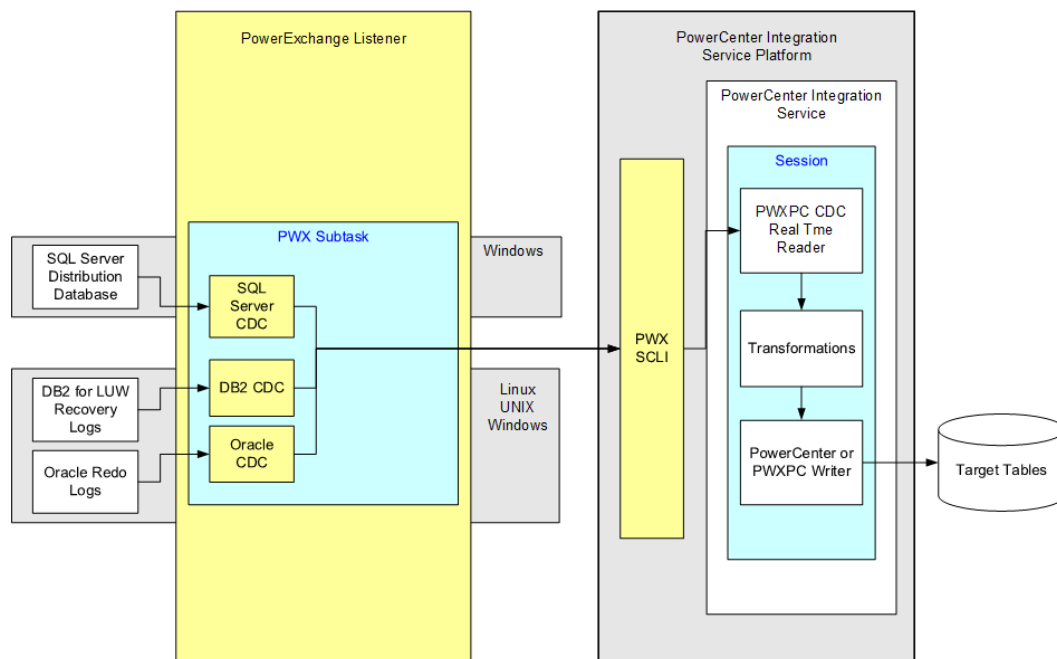
The PowerExchange CDC architecture is sufficiently flexible to handle many change data replication scenarios.

You can use PowerExchange in conjunction with PowerCenter to replicate change data from multiple sources of the same RDBMS type to multiple targets of different types in a single session.

The targets can be tables or files on the same system as the source or on other systems. The PowerCenter Integration Service can write data to tables in some RDBMSs as well as to flat files and XML files. If you installed PowerExchange or PowerExchange (PowerCenter Connect) products that provide connectivity to additional nonrelational or relational targets, you can also load data to those targets, for example, Db2 for z/OS tables, VSAM data sets, IMS segments, or WebSphere MQ.

You can run multiple instances of PowerExchange CDC components on a single system. For example, you might want to run a separate PowerExchange Logger for each source RDBMS to create separate sets of log files for each RDBMS type.

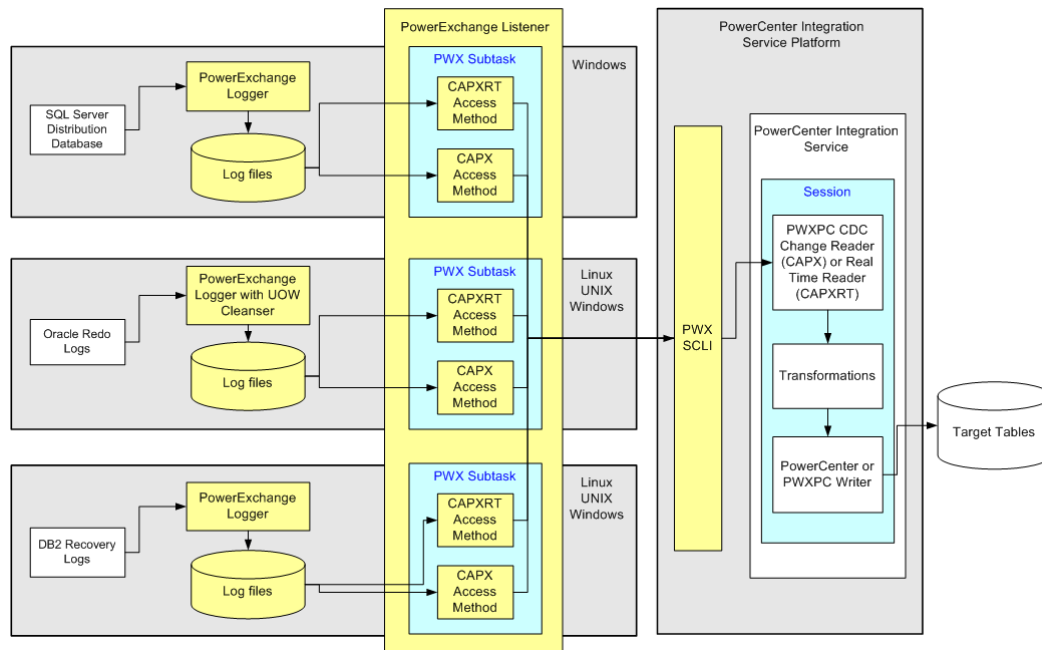
The following image shows a simple CDC configuration that uses real-time extraction mode to access change data directly from the change stream without the PowerExchange Logger:



In this real-time configuration, PowerExchange CDC uses the CAPXRT access method to capture change data from a SQL Server distribution database, Db2 database logs, and Oracle redo logs. 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 created 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 or instance to multiple target tables in a single extraction process.

The following image shows a CDC configuration that uses the PowerExchange Logger in both batch extraction mode and continuous extraction mode:



In this configuration, the PowerExchange Logger captures change data from the change stream for Microsoft SQL Server, Oracle, and Db2 tables and writes that data to its log files. After the data is in the PowerExchange log files, the source RDBMS log files can be deleted, if necessary. When an extraction session runs, PWXPC contacts the PowerExchange Listener. The PowerExchange Listener reads the PowerExchange Logger log files and calls the SCLI on the PowerCenter Integration Service machine to transmit the change data to PowerCenter.

For some source tables, PWXPC extracts change data from the PowerExchange Logger log files in batch extraction mode with the CAPX access method. In this mode, the extraction session stops after it completes processing the log files. For other source tables, PWXPC extracts change data in continuous mode with the CAPXRT access method. In this mode, the extraction session extracts change data on an ongoing basis. In PowerCenter, you can create one source definition and one mapping that covers both extraction modes. However, batch and continuous extractions must run as separate sessions. For a batch extraction session, use a PWX CDC Change application connection. For a continuous extraction session, use a PWX CDC Real Time application connection. For example, you can run batch extractions to replicate change data to targets that need to be synchronized periodically, and run continuous extractions to replicate change data to targets that need to be synchronized in near real time. Batch and continuous extraction sessions can run concurrently.

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 directly from the change stream and from PowerExchange Logger for Linux, UNIX, and Windows log 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*.

Environment Variable Incompatibilities Between PowerExchange and PowerCenter

When PowerCenter® and PowerExchange are installed on the same Linux, UNIX, or Windows machine, in certain cases, they have conflicting requirements for the PATH and LD_LIBRARY_PATH environment variables. To run correctly in these cases, PowerExchange and PowerCenter must run in separate environments.

This requirement applies when the PowerCenter Integration Service or PowerCenter Repository Service runs on the same machine as one of the following PowerExchange components:

- PowerExchange Listener
- PowerExchange Logger for Linux, UNIX, and Windows
- PowerExchange Navigator
- Any PowerExchange utility except the createdatamaps utility

The following table describes the restrictions that apply to the PATH and LD_LIBRARY_PATH variables in the PowerExchange and PowerCenter environments:

Environment	PATH	LD_LIBRARY_PATH
PowerExchange	\$INFA_HOME must not precede \$PWX_HOME. Otherwise, you cannot start the PowerExchange Listener or Logger from the command line.	LD_LIBRARY_PATH must not contain an entry for PowerCenter. This requirement ensures that PowerExchange utilities pick up their libraries from \$PWX_HOME only.
PowerCenter	The \$PWX_HOME entry must not precede the \$INFA_HOME entry.	The \$LD_LIBRARY_PATH variable definition must include both \$INFA_HOME and \$PWX_HOME, and \$INFA_HOME must precede \$PWX_HOME. For example: <code>\$INFA_HOME/server/ bin:\$PWX_HOME:\$LD_LIBRARY_PATH</code>

To set the correct environment for PowerExchange or PowerCenter instances on the same machine, use one of the following strategies:

- Always start PowerExchange and PowerCenter using separate user accounts, and set the environment variables appropriately for each account.
- Run the pwxsettask.sh or pwxsettask.bat script each time you start a PowerExchange component.

Summary of CDC Implementation Tasks

After you install PowerExchange, you can configure change data capture and extraction, materialize targets, and start extraction processing.

The following table identifies the tasks that you perform to implement change data capture and extraction processing for a Linux, UNIX, or Windows data source:

Step	Task	References
1	Configure parameters in the dbmover.cfg file for the PowerExchange Listener.	"Customizing the dbmover Configuration File for CDC" on page 27
2	Start the PowerExchange Listener on the machine with the source database.	"Starting the PowerExchange Listener" on page 32
3	Perform RDBMS-specific configuration tasks for CDC.	<ul style="list-style-type: none"> - Chapter 4, "Db2 CDC on Linux, UNIX, or Windows" on page 81 - Chapter 5, "Microsoft SQL Server CDC" on page 99 - Chapter 6, "MySQL CDC" on page 116 - Chapter 7, "Express CDC for Oracle" on page 132 - Chapter 8, "PostgreSQL CDC" on page 206
4	(Optional) Configure the PowerExchange Logger.	"Configuring the PowerExchange Logger" on page 44
5	(Optional) Start the PowerExchange Logger.	"Starting the PowerExchange Logger" on page 66
6	From the PowerExchange Navigator, define and activate capture registrations and extraction maps for the data sources.	<i>PowerExchange Navigator User Guide</i>
7	For Db2 sources that have user-defined or multi-field columns that you want to manipulate, create Db2 data maps.	<i>PowerExchange Navigator User Guide</i>
8	Materialize the target from the source.	<i>PowerExchange Bulk Data Movement Guide</i>
9	Establish a start point for the extraction.	"Restart Tokens and the Restart Token File" on page 257
10	From PowerCenter, configure mappings, workflows, connections, and sessions. Then run the workflow.	<ul style="list-style-type: none"> - <i>PowerExchange Interfaces for PowerCenter</i> - <i>PowerCenter Designer Guide</i> - <i>PowerCenter Workflow Basics Guide</i>

Part II: PowerExchange CDC Components

This part contains the following chapters:

- [PowerExchange Listener, 27](#)
- [PowerExchange Logger for Linux, UNIX, and Windows, 36](#)

CHAPTER 2

PowerExchange Listener

This chapter includes the following topics:

- [PowerExchange Listener Overview, 27](#)
- [Customizing the dbmover Configuration File for CDC, 27](#)
- [Starting the PowerExchange Listener, 32](#)
- [Stopping the PowerExchange Listener, 34](#)
- [Displaying Active PowerExchange Listener Tasks, 34](#)

PowerExchange Listener Overview

In a change data capture (CDC) environment, a PowerExchange Listener can provide some or all of the following services:

- Store and manage capture registrations, extraction maps, and data maps for CDC data sources.
- Provide captured change data to PowerCenter when you run a PowerCenter CDC session.
- Provide captured change data or source table data to the PowerExchange Navigator when you perform a database row test of an extraction map or a data map.
- Interact with other PowerExchange Listeners on other nodes to facilitate communication among the PowerExchange Navigator, PowerCenter Integration Service, data sources, and any system to which PowerExchange processing is offloaded.

Customizing the dbmover Configuration File for CDC

You must configure some statements in the dbmover configuration file for CDC processing.

The PowerExchange Listener uses the dbmover statements to perform the following functions:

- Connect to source relational databases and objects to capture change data.
- Determine the directory in which to store capture registrations, extraction maps, and PowerExchange Logger log files.
- Connect to the system where the PowerExchange Logger log files reside to extract change data.

The following key dbmover statements are required for CDC and pertain to all source RDBMSs that PowerExchange supports on Linux, UNIX, or Windows:

- CAPI_CONNECTION statements
 - Source-specific CAPI_CONNECTION statements, which are described for each source type
 - A CAPX CAPI_CONNECTION if you use continuous extraction mode
- CAPI_SRC_DFLT statement
- CAPT_PATH statement
- CAPT_XTRA statement

Review the descriptions of each of these parameters. For more information about other dbmover.cfg statements, see the *PowerExchange Reference Manual*.

CAPI_CONNECTION Statements

PowerExchange requires that you define CAPI_CONNECTION statements in the dbmover configuration file on any Linux, UNIX, or Windows system where PowerExchange captures or extracts change data.

PowerExchange uses the CAPI_CONNECTION statements to connect to the change stream for a source to extract change data.

For each data source, you must define one of the following source-specific types of CAPI_CONNECTION statements:

- For Db2 sources on Linux, UNIX, or Windows: UDB CAPI_CONNECTION
- For Microsoft SQL Server sources: MSQL CAPI_CONNECTION
- For MySQL sources: MYSQL CAPI_CONNECTION
- For Express CDC for Oracle sources: ORAD CAPI_CONNECTION
- For PostgreSQL sources: PG CAPI_CONNECTION
- For SAP HANA sources: HANA CAPI_CONNECTION

If you use continuous extraction mode to extract change data from PowerExchange Logger log files, you must also define a CAPX CAPI_CONNECTION statement.

You can specify up to eight source-type CAPI_CONNECTION statements in a dbmover configuration file, excluding CAPX CAPI_CONNECTION statements. You can identify one of these statements as the overall default CAPI_CONNECTION statement. If you define multiple CAPI_CONNECTION statements for the same source type, you can also specify a source-specific default. In addition to or in lieu of the defaults, you can define specific CAPI_CONNECTION overrides in multiple ways. The order of precedence that PowerExchange uses to determine which CAPI_CONNECTION statement to use is described in the *PowerExchange Reference Manual*.

Note: To perform database row tests for data sources that are defined by capture registrations local to the PowerExchange Navigator, you must specify the appropriate CAPI_CONNECTION statements on the PowerExchange Navigator machine. Otherwise, you do not need to specify CAPI_CONNECTION statements to perform database row tests.

RELATED TOPICS:

- [“CAPI_CONNECTION - CAPX Statement” on page 60](#)
- [“CAPI_CONNECTION - MSQL Statement” on page 107](#)
- [“CAPI_CONNECTION - MYSQL Statement” on page 125](#)
- [“CAPI_CONNECTION - UDB Statement” on page 89](#)

- [“CAPI_CONNECTION - ORAD Statement” on page 175](#)
- [“CAPI_CONNECTION - PG Statement” on page 213](#)

CAPI_SRC_DFLT Statement

The CAPI_SRC_DFLT statement specifies the CAPI_CONNECTION statement that PowerExchange uses by default for a specific data source type when a CAPI connection override is not supplied.

Informatica recommends that you specify this statement for each source type to ensure that the appropriate type of CAPI_CONNECTION statement is used for accessing a source. If you have multiple source types that run on different platforms and do not specify a CAPI_SRC_DFLT statement for each source type, source access problems might occur in certain situations during PowerCenter data previews or CDC sessions.

Operating Systems: All

Data Sources: All

Related Statements: CAPI_CONN_NAME and CAPI_CONNECTION

Required: No

Syntax:

```
CAPI_SRC_DFLT=(source_type
               , capi_connection_name)
```

Parameters:

source_type

Required. The CDC source type. The following table identifies the valid source-type options:

Option	Source Type
ADA	Adabas sources
AS4	Db2 for i (i5/OS) sources
CAPX	Sources for which you are extracting data in continuous extraction mode and using the PowerExchange Logger for Linux, UNIX, or Windows or PowerExchange Condense
DB2	Db2 for z/OS sources
DCM	CA Datacom sources
HAN	SAP HANA CDC sources
IDL	CA IDMS/DB log-based CDC sources
IML	IMS log-based CDC sources
IMS	IMS synchronous CDC sources
MSS	Microsoft SQL Server sources
MYS	MySQL sources
ORA	Oracle sources

Option	Source Type
PGS	PostgreSQL sources
UDB	Db2 sources on Linux, UNIX, or Windows
VSAM or VSM	VSAM sources

capi_connection_name

Required. Unique name of the CAPI_CONNECTION statement to use as the default for the specified source type. This name must match the NAME value in a CAPI_CONNECTION statement that has a TYPE value that is compatible with the CAPI_SRC_DFLT *source_type*.

The following table shows, for each CAPI_SRC_DFLT option, the compatible CAPI_CONNECTION type:

CAPI_SRC_DFLT Option	CAPI_CONNECTION Statement Type
AS4	UOWC
CAPX	CAPX
ADA, DB2, DCM, IDL, IDM, IML, IMS, VSAM, or VSM	UOWC
HAN	HANA
ORA	ORAD for PowerExchange Express CDC for Oracle
MSS	MSQL
MYS	MYSQL
PGS	PG
UDB	UDB

Usage Notes:

- If you define multiple CAPI_CONNECTION statements for a source type, you can define a CAPI_SRC_DFLT statement to identify the default CAPI_CONNECTION for that source type. The CAPI_SRC_DFLT statement must point to a CAPI_CONNECTION statement of a compatible type.
- You can optionally define a CAPI_CONN_NAME statement that specifies an overall default statement, out of all of the CAPI_CONNECTION statements in the DBMOVER file.
- Instead of or in addition to specifying defaults, you can use the following CAPI connection name overrides to point to a specific CAPI_CONNECTION statement for CDC sessions or database row tests:
 - For CDC sessions, use the **CAPI Connection Name Override** attribute on the PWX CDC application connection.
 - For PowerExchange Condense, use the CONN_OVR parameter in CAPTPARM configuration file.
 - For the PowerExchange Logger for Linux, UNIX, and Windows, use the CONN_OVR parameter in pwxcl.cfg configuration file.

- For DTLUAPPL utility operations that generate restart tokens, use the CONN_OVR parameter in the DTLUAPPL control statement.
- For CAPXRT database row tests in the PowerExchange Navigator, use the **CAPI Connection Name** value in the **CAPXRT Advanced Parameters** dialog box. If you add an SQL statement for generating restart tokens, you can include the CONNAME parameter to point to the override CAPI_CONNECTION.
- For PowerExchange ODBC connections, use the DTLCONN_OVR parameter in the odbc.ini file or the SQL escape sequence override DTLCONNOVR.

CAPT_PATH Statement

The CAPT_PATH statement specifies the path to a directory on a Linux, UNIX, or Windows system for storing CDC control files.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Db2, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL sources on Linux, UNIX, or Windows, and if you use remote logging, data sources on IBM i (i5/OS) or z/OS

Related Statements: CAPT_XTRA

Required: Yes for CDC sources on Linux, UNIX, and Windows

Syntax:

```
CAPT_PATH=path
```

Value: Enter the path to the directory that stores the following CDC control files:

- CCT file, which contains capture registrations
- CDEP file, which contains application names for any PowerCenter extractions that use ODBC connections
- CDCT file, which contains information about PowerExchange Logger for Linux, UNIX, and Windows log files

This directory can be one that you created specifically for these control files or another directory.

To specify a Windows network path, use three leading backslashes (\\) so that PowerExchange can parse the network path correctly. For example:

```
CAPT_PATH=\\host\Shared Folders\C\CDC_SHARED\capture
```

Default location is the PowerExchange installation directory.

Usage Notes:

- Informatica recommends that you use a unique directory name to separate these CDC objects from the PowerExchange code. This practice makes migrating to a another PowerExchange version easier.
- PowerExchange C-ISAM control files, such as the CCT, CDEP, and CDCT files, must be stored on local disk. Do not locate these files in SAN or NAS storage.
- To specify a path to the directory that contains extraction maps, use the CAPT_XTRA statement.

CAPT_XTRA Statement

The CAPT_XTRA statement specifies the path to a directory on a Linux, UNIX, or Windows system for storing extraction maps for CDC.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Db2, Microsoft SQL Server, MySQL, Oracle, and PostgreSQL sources on Linux, UNIX, or Windows

Related Statements: CAPT_PATH

Required: Yes for CDC sources on Linux, UNIX, and Windows

Syntax:

```
CAPT_XTRA=path
```

Value: Enter the path to the directory for storing extraction maps.

This directory can be a directory that you created specifically for extraction maps or another directory.

To specify a Windows network path, use three leading backslashes (\\) so that PowerExchange can parse the network path correctly. For example:

```
CAPT_XTRA=\\host\Shared Folders\C\CDC_SHARED\capture\xtramaps
```

Default location is the PowerExchange installation directory.

Usage Note: To specify a directory for storing CCT and CDEP control files for CDC, use the CAPT_PATH statement.

Starting the PowerExchange Listener

To start the PowerExchange Listener, you can run the dtllst program or use other system-specific methods.

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

Starting the PowerExchange Listener on Linux or UNIX

To start the PowerExchange Listener on a Linux or UNIX system, use one of the following methods:

- Enter dtllst at the command line to run the PowerExchange Listener in foreground mode. Syntax is:

```
dtllst node1 [config=directory/myconfig_file] [license=directory/mylicense_key_file]
```

Include the optional config and license parameters if you want to specify configuration and license key files that override the original dbmover.cfg and license.key files.

Add an ampersand (&) at the end to run the PowerExchange Listener in background mode. Also, you can add the prefix nohup at the beginning to run the PowerExchange Listener persistently. Syntax is:

```
nohup dtllst node1 [config=directory/myconfig_file] [license=directory/  
mylicense_key_file] &
```

- Run the startlst script, which is supplied by PowerExchange installation. This script deletes the detail.log file and then starts the PowerExchange Listener.

Caution: If you run PowerExchange and PowerCenter on the same machine, using the same user account, you must create separate environments for PowerExchange and PowerCenter. To create the appropriate PowerExchange environment and start the PowerExchange Listener, run the pwxsettask.sh script.

Use the following syntax:

```
pwxsettask.sh dtllst node_name ["config=directory/config_file"] ["license=directory/  
license_key_file"]
```

The quotation marks are optional.

For more information, see [“Environment Variable Incompatibilities Between PowerExchange and PowerCenter” on page 24.](#)

Starting the PowerExchange Listener on Windows

To start the PowerExchange Listener on a Windows system, use one of the following methods:

- Run the PowerExchange Listener as a Windows service by completing one of the following actions:
 - From the Windows Start menu, click **Start > Programs > Informatica PowerExchange > Start PowerExchange Listener**.
 - Use the dtllstsi program to enter the start command from a Windows command prompt:

```
dtllstsi start "service_name"
```

- Enter dtllst.

The syntax is the same as that for Linux and UNIX except that the & and nohup operands are not supported. You must have a product license that allows you to manually run dtllst.

If you use the PowerExchange Logger for Linux, UNIX, and Windows, the PowerExchange Listener must run under a user ID that has READ access to the PowerExchange Logger log files.

If you run the PowerExchange Listener as an application service in the Informatica domain, enable the PowerExchange Listener Service from the Informatica Administrator tool to start it. For more information, see the *Informatica Application Service Guide*.

Caution: If you run PowerExchange and PowerCenter on the same machine, using the same user account, you must create separate environments for PowerExchange and PowerCenter. To create the appropriate PowerExchange environment and start the PowerExchange Listener, run the pwxsettask.bat script.

Use the following syntax:

```
pwxsettask dtllst node_name ["config=directory/config_file"] ["license=directory/  
license_key_file"]
```

The quotation marks are required.

For more information, see ["Environment Variable Incompatibilities Between PowerExchange and PowerCenter" on page 24](#).

Stopping the PowerExchange Listener

To stop the PowerExchange Listener, use the CLOSE or CLOSE FORCE command. To stop active PowerExchange Listener tasks, use the STOPTASK command.

The following table describes these commands and the syntax for issuing each command from the command line against a PowerExchange Listener task that is running in foreground mode:

Command	Description	Command Line Syntax
CLOSE	Stops the PowerExchange Listener after all of the following subtasks complete: <ul style="list-style-type: none">- CDC subtasks, which stop at the next commit of a unit of work (UOW)- Bulk data movement subtasks- PowerExchange Listener subtasks	On Linux, UNIX, or Windows: C
CLOSE FORCE	Forces the cancellation of all user subtasks and stops the PowerExchange Listener. PowerExchange waits 30 seconds for current user subtasks on the PowerExchange Listener to complete. Then PowerExchange cancels any remaining user subtasks and stops the PowerExchange Listener. This command is useful if you have long-running subtasks on the PowerExchange Listener.	On Linux , UNIX, or Windows: C F
STOPTASK	Stops a PowerExchange Listener task for a specific extraction application process. PowerExchange waits to stop the PowerExchange Listener until either the end UOW or commit threshold is reached.	On Linux, UNIX, or Windows: STOPTASK <i>app_name</i> The <i>app_name</i> is the name of an active change data extraction process. You can get this name from a PWX-00712 message in the PowerExchange Listener DISPLAY ACTIVE command output.

Alternatively, you can use any of the following methods:

- On a Linux, UNIX, or Windows system, use the pwxcmd program to issue the close, closeforce, or stoptask command to a PowerExchange Listener running in foreground or background mode, on the local system or a remote system. You can issue these pwxcmd commands from the command line or include them in scripts or batch files.
- On a Linux or UNIX system, if the PowerExchange Listener is running in background mode, use the standard operating system commands to find the PowerExchange Listener process ID and then “kill” that process. A “kill” operation is similar to a CLOSE operation.
- On a Windows system, if the PowerExchange Listener does not respond to a CLOSE FORCE command, press Ctrl + C once to issue CLOSE or press Ctrl + C twice to issue CLOSE FORCE.

Displaying Active PowerExchange Listener Tasks

You can use the DISPLAY ACTIVE command to display information about each active PowerExchange Listener task that is running in foreground mode on a Linux, UNIX, or Windows system. This information includes the TCP/IP address, port number, application name, access type, and status.

On a Linux, UNIX, or Windows system, enter the following command at the command line on the screen where the PowerExchange Listener task is running in foreground mode:

D

Alternatively, on a Linux, UNIX, or Windows system, you can issue the `pwxcmd listtask` command from a command line, script, or batch file to a PowerExchange Listener running on the local system or a remote system. The `pwxcmd listtask` command produces the same output as the `DISPLAY ACTIVE` command.

CHAPTER 3

PowerExchange Logger for Linux, UNIX, and Windows

This chapter includes the following topics:

- [PowerExchange Logger Overview, 36](#)
- [PowerExchange Logger Tasks, 38](#)
- [PowerExchange Logger Files, 38](#)
- [File Switches, 41](#)
- [PowerExchange Logger Operational Modes, 42](#)
- [PowerExchange Logger Usage Considerations, 43](#)
- [Logging Data for IBM i or z/OS Sources to Remote PowerExchange Logger Logs, 44](#)
- [Configuring the PowerExchange Logger, 44](#)
- [Starting the PowerExchange Logger, 66](#)
- [Managing the PowerExchange Logger, 71](#)

PowerExchange Logger Overview

The PowerExchange Logger for Linux, UNIX, and Windows captures change data from PowerExchange data sources and writes that data to PowerExchange Logger log files. The PowerExchange Logger writes only the successful units of work (UOWs) to its log files, in chronological order based on end time.

When a PowerCenter CDC session runs, it extracts change data from the log files instead of from the change stream.

Note: The PowerExchange Logger for Linux, UNIX, and Windows is similar in function to PowerExchange Condense on IBM i (i5/OS) or z/OS systems.

The PowerExchange Logger can capture change data from Db2 database logs, a Microsoft SQL Server distribution database, MySQL binary logs, Oracle redo logs, or PostgreSQL replication store table that records changes retrieved from a logical replication slot. If you use the remote logging feature, a PowerExchange Logger process on Linux, UNIX, or Windows can also process data from data sources on IBM i or z/OS systems.

Use the PowerExchange Logger to reduce the overhead of CDC processing. With the PowerExchange Logger, PowerExchange accesses the source database fewer times to read change data, which reduces database I/O. Also, because change data is extracted from the PowerExchange Logger log files, you do not need to extend the retention period for source database log files to accommodate CDC processing.

You must run one PowerExchange Logger process for each source type and instance, as defined in a registration group. The PowerExchange Logger can run on the source database server, PowerCenter Integration Service machine, or another system in either continuous mode or batch mode.

Multiple PowerExchange Logger instances can run under the same PowerExchange Listener and dbmover.cfg configuration. However, because a single dbmover.cfg can contain a maximum of eight CAPI_CONNECTION statements, the number of source instances and PowerExchange Logger instances that can run under a single Listener and dbmover.cfg is limited. For more information, see [“CAPI_CONNECTION Statements” on page 28](#).

When you create capture registrations for data sources, including IBM i and z/OS data sources for which processing is offloaded, set the **Condense** option to **Part**. The PowerExchange Logger supports only partial condense processing. For IBM i or z/OS data sources, if you set the **Condense** option to **Full** in capture registrations, the PowerExchange Logger ignores the registrations and does not process change data from those sources.

For each PowerExchange Logger process, you must define a configuration file. PowerExchange provides a sample configuration file named pwxcl.cfg. The configuration file contains parameters for controlling the PowerExchange Logger and for identifying the source instance. Use the COLL_END_LOG parameter to control whether the PowerExchange Logger runs in continuous mode or batch mode.

When PowerCenter CDC sessions run, the PowerExchange Log Reader process extracts change data from the PowerExchange Logger log files in continuous extraction mode or batch extraction mode.

You can secure sensitive data that is stored in PowerExchange Logger log files, such as Social Security numbers, by enabling AES encryption of the log files. In the PowerExchange Logger configuration file, you can select the AES encryption algorithm that you want to use. To enable encryption, you must also specify an encryption password either in the PowerExchange Logger configuration file or in the pwxcl command that you use to cold start the PowerExchange Logger from the command line. If you specify the encryption password in the pwxcl command for a cold start and need to restore the CDCT file later, you must enter the same encryption password for the restore operation.

Tip: To reduce the risk of unauthorized access to the encryption password, Informatica recommends that you specify the password in the pwxcl command for cold starting the Logger rather than specify the password in the configuration file.

RELATED TOPICS:

- [“PowerExchange Logger Operational Modes” on page 42](#)
- [“PowerExchange Logger Tasks” on page 38](#)
- [“PowerExchange Logger Files” on page 38](#)
- [“File Switches” on page 41](#)
- [“Configuring the PowerExchange Logger” on page 44](#)
- [“Managing the PowerExchange Logger” on page 71](#)

PowerExchange Logger Tasks

The PowerExchange Logger for Linux, UNIX, and Windows includes the following task and subtasks:

Controller task

Loads parameter settings from the PowerExchange Logger pwxcl configuration file. Loads the capture registrations from the CCT file. After loading this information, the Controller starts the Command Handler subtask and then the Writer subtask.

Command Handler subtask

Processes PowerExchange Logger commands from various sources, including user stdin and the pwxcmd program. If the PROMPT parameter is set to Y in the pwxcl.cfg file, the Command Handler waits for the Writer subtask to initialize before accepting a user command.

Writer subtask

Performs most of the PowerExchange Logger work that uses CPU time. The Writer initializes the CAPI for the source database, determines the start or restart point in the change stream, reads change data from the change stream, and writes change data to PowerExchange Logger log files. The Writer also writes records to the CDCT file during a file switch, deletes expired CDCT records, and rolls back CDCT records when you warm start the PowerExchange Logger from an earlier point in time. If the PROMPT parameter is set to Y in the pwxcl configuration file, the Writer waits for you to respond to confirmation prompts before proceeding with a cold start or a rollback of CDCT records.

Note: The Log Reader, which extracts change data from PowerExchange Logger log files, runs as an independent process.

PowerExchange Logger Files

A PowerExchange Logger process writes information to the CDCT file, PowerExchange Logger log files, and PowerExchange message logs.

The PowerExchange Logger also uses lock files during processing.

RELATED TOPICS:

- [“CDCT File” on page 38](#)
- [“PowerExchange Logger Log Files” on page 39](#)
- [“Lock Files” on page 40](#)
- [“Message Log Files” on page 41](#)

CDCT File

The PowerExchange Logger stores log file and restart information in the CDCT file.

When a PowerCenter CDC session runs in continuous extraction mode or batch extraction mode, the PowerExchange Listener reads the CDCT file to determine the PowerExchange Logger log files from which to extract change data.

The PowerExchange Logger creates the CDCT file in the directory that is specified by the CAPT_PATH statement in the dbmover configuration file that is on the system where the PowerExchange Logger runs. The

CDCT file must reside on local disk. If the CAPT_PATH statement is not specified, the CDCT file is in the local directory from which the PowerExchange Logger is invoked.

The CDCT file name has the format CDCT_dbid, where dbid is the DBID value in the PowerExchange Logger configuration file.

The first time the PowerExchange Logger receives data of interest after startup or a file switch, the PowerExchange Logger opens a log file to which to write data and creates an entry for this log file in the CDCT file. After the PowerExchange Logger finishes writing data to the log file, it marks the file as closed in the CDCT file. The PowerExchange Logger also updates the restart information in the CDCT file during periods when no changes of CDC interest are received.

If a PowerExchange Logger failure occurs and leaves an open log file, the PowerExchange Logger marks the CDCT entry for that log file for deletion. The next time the PowerExchange Logger starts, it deletes that log file entry and creates a new log file. The log files contain records from which the CDCT entry for the log can be rebuilt.

PowerExchange automatically generates a backup of the CDCT file at PowerExchange Logger initialization and normal termination. These backups are located in the same directory as the CDCT file and have file names with the following formats: CDCT_dbid.INIT.bkp and CDCT_dbid.TERM.bkp.

Tip: You can use the PWXUCDCT utility to print information about CDCT records, back up the CDCT file, restore the CDCT file from a backup, re-create the CDCT file based on PowerExchange Logger log files, and delete expired CDCT records.

RELATED TOPICS:

- [“PWXUCDCT Commands for Maintaining the PowerExchange Logger CDCT and Log Files” on page 76](#)

PowerExchange Logger Log Files

The PowerExchange Logger creates log files for storing change data records when it first encounters changes for source tables and columns of interest. These source tables and columns must be defined in active capture registrations.

The PowerExchange Logger creates log files based on the EXT_CAPT_MASK parameter in the pwxccf configuration file. This parameter specifies a path to the directory where the log files are stored and a prefix for the log file names. Log file names have the following format:

path/prefix.CND.CPyymmdd.Thhmmnnn

Where:

- *path/prefix* is the EXT_CAPT_MASK value.
- *yymmdd* is the date when the file is created.
- *hhmm* is a 24-hour time when the file is created.
- *nnn* is a generated sequence number that makes each file name unique.

The log files remain open until a file switch occurs or the PowerExchange Logger shuts down.

When you run a PowerCenter CDC session in continuous extraction mode or batch extraction mode, PowerExchange extracts change data from the PowerExchange Logger log files.

Lock Files

During initialization, a PowerExchange Logger process creates lock files to prevent other PowerExchange Logger processes from accessing the same CDCT file and log files concurrently.

As long as the PowerExchange Logger process holds a lock on the lock files, locking is in effect for the resources for which the lock files were created.

PowerExchange Logger locking works on local disks on Linux, UNIX, or Windows systems. It also works on the following shared file systems on Linux or UNIX systems:

- Veritas Storage Foundation™ Cluster File System by Symantec
- IBM General Parallel File System
- EMC Celerra network-attached storage (NAS) with Network File System (NFS) protocol version 3
- NetApp NAS with NFS version 3

The PowerExchange Logger creates lock files in the following order:

1. A lock file for the CDCT file for a source instance. The PowerExchange Logger generates the lock file name and location based on the directory that is specified in the CAPT_PATH parameter of the dbmover configuration file.
2. One of the following lock files:
 - If you use a group definition file, a lock file for each set of the PowerExchange Logger log files that is defined by the GROUP statements in the group definition file. The PowerExchange Logger generates the lock file names and locations based on the external_capture_mask parameter in each GROUP statement. In this case, the PowerExchange Logger ignores the EXT_CAPT_MASK parameter in the pwxcl configuration file when creating lock files and processing log files.
 - If you do not use a group definition file, a lock file for PowerExchange Logger log files. The PowerExchange Logger generates the lock file name and location based the directory and file-name prefix that are specified in the EXT_CAPT_MASK parameter of the pwxcl configuration file.

Lock file names end with _lockfile.lck. For example, a lock file for the CDCT file could have the name CDCT_oracoll1_lockfile.lck.

When the PowerExchange Logger process ends, it unlocks the lock files to enable other PowerExchange Logger processes to access the previously locked resources.

To identify a PowerExchange Logger process that holds a lock, look up the process ID (PID) in the Task Manager on a Windows system or issue the ps command on a UNIX or Linux system.

Also, the PowerExchange Logger writes messages to the PowerExchange message log that indicate the locking status. Look for the following key messages:

- To verify that lock files are created, look for PWX-25802 messages, such as:

```
PWX-25802 Process pwxcl.exe pid 5428 locked file C:\capture\captpath
\CDCT_instance_lockfile.lck
```
- To verify that lock files are unlocked, look for PWX-25803 messages, such as:

```
PWX-25803 Process pwxcl.exe pid 5428 unlocked file C:\capture\extcapt
\loggerfiles_lockfile.lck
```
- If the PowerExchange Logger process cannot find the lock file that it needs to access some resources, it writes message PWX-25800:

```
PWX-25800 Could not find lock file file_name
```


- If a lock file is locked by another process, the PowerExchange Logger process writes some or all of the following messages, depending on if it can acquire a lock before the maximum retry interval that is specified in PWX-25814 elapses:

```
PWX-25804 Error trying to lock PowerExchange Logger files
PWX-25811 File file_name is locked by process process_name pid process_id on host
host_name date date time time
PWX-25812 File file_name is locked by pid process_id start offset length bytes
PWX-25813 No information is available on process which locked file file_name
PWX-25814 Trying to lock file file_name until number seconds elapses
PWX-25815 File file_name is locked by another process and no more waiting is allowed.
```

If a PowerExchange Logger process ends abnormally with message PWX-25815 and return code 25815, try to determine the status of the other PowerExchange Logger process that is holding the lock. This other process is identified in message PWX-25811. For example, the other process might not have completely shut down, or both processes might be trying to use the same files because of an error in their pwxcl configuration files.

Message Log Files

The PowerExchange Logger writes messages to the PowerExchange message log file.

By default, on Linux, UNIX, and Windows, this file is named detail.log and is located in the working directory where the PowerExchange Logger process runs. However, you can optionally specify another directory for PowerExchange message log files. You can also enable the use of alternative log files.

To specify a unique directory for PowerExchange message log files, include the LOGPATH parameter in the dbmover configuration file. You can then find the PowerExchange message log files more easily.

Also, you can implement alternative logging by specifying the TRACING statement in the dbmover configuration file. When alternative logging is enabled, PowerExchange creates a set of alternative log files for each PowerExchange process, including each PowerExchange Logger process, in a separate directory. When an alternative log file becomes full, PowerExchange switches to another alternative log file. This automatic rotation of message log files prevents out-of-space conditions. Also, PowerExchange buffers messages before writing them to the alternative log files on disk at a specific flush interval. This mode of writing messages can reduce I/O activity on the alternative log files.

File Switches

When running in continuous mode, the PowerExchange Logger periodically closes its open log files if they contain data and then opens a new set of log files. This process is called a *file switch*.

The PowerExchange Logger automatically performs a file switch when the criteria in the following parameters of the pwxcl configuration file are met:

- FILE_SWITCH_CRIT
- FILE_SWITCH_VAL

If the open log files do not contain data when the file-switch criteria in these parameters are met, the file switch does not occur. The PowerExchange Logger waits until the next time the file-switch criteria are met. If the files still do not contain data, the PowerExchange Logger continues to check the log files at set intervals. Only when the log files contain data does the file switch occur.

Also, you can force a file switch by entering the fileswitch command from the command line. Alternatively, on Linux, UNIX, or Windows, you can send a pwxcmd fileswitch command to a PowerExchange Logger process running on the local system or a remote system.

If a file switch is initiated automatically or by the `fileswitch` command when the PowerExchange Logger is not on a commit boundary in the change stream, the PowerExchange Logger waits 10 seconds for the commit to occur and then forces the file switch.

PowerExchange Logger Operational Modes

A PowerExchange Logger process can operate in continuous mode or batch mode.

To set the operational mode, use the `COLL_END_LOG` parameter in the `pwxccl` configuration file.

Continuous Mode

In continuous mode, the PowerExchange Logger process runs continuously until you manually stop it.

Use continuous mode in the following situations:

- You have a database with a high level of change activity that occurs continuously.
- You have a database with intermittent activity that occurs at unpredictable intervals.
- You want to avoid the overhead of scheduling PowerExchange Logger runs.
- You cannot restart the PowerExchange Logger process often enough to keep up with the change volume.

To enable continuous mode, set the `COLL_END_LOG` parameter to 0.

In continuous mode, each time the Writer subtask completes a logging cycle, the PowerExchange Logger process is temporarily suspended. The next logging cycle is triggered by any of the following events:

- The wait interval that is defined in the `NO_DATA_WAIT` parameter of the `pwxccl` configuration file elapses.
- The `CONDENSE` command is manually entered at the command line or with the `pwxcmd` program.
- The `FILESWITCH` command is manually entered at the command line or with the `pwxcmd` program.

The PowerExchange Logger process continues to run until you enter the `SHUTDOWN` or `SHUTCOND` command. To prevent log files from becoming too large, the PowerExchange Logger process periodically performs a file switch. Log files that are very large can prolong restart times for CDC sessions that run in continuous extraction mode or batch extraction mode.

You can use the `NO_DATA_WAIT2` parameter in the `pwxccl` configuration file to prevent the PowerExchange Logger from consuming too much CPU time when PowerExchange is not receiving change data. For example, if you set the `NO_DATA_WAIT2` parameter to 30 seconds, the PowerExchange Logger sleeps for 30 seconds, provided that no updates are received, and then performs another processing cycle. However, a large `NO_DATA_WAIT2` value can delay processing of a `SHUTDOWN` command. If you need to reduce the amount of time that the PowerExchange Logger sleeps on a quiet system, you can adjust the `FILE_FLUSH_VAL` and `FILE_SWITCH_VAL` parameters.

Also, specify the `RSTRADV` time interval in the source-specific `MSQL` or `UDB CAPI_CONNECTION` statement or the `OPTIONS` statement of the PowerExchange Express CDC for Oracle configuration file to enable the PowerExchange Logger to advance its restart and sequence tokens even when UOWs do not contain any change data of interest for the data sources.

When you run the PowerExchange Logger in continuous mode, you can use either continuous or batch extraction mode for workflows that extract change data from the PowerExchange Logger log files.

Tip: Run the PowerExchange Logger in continuous mode unless you have a reason to use batch mode. On Linux or UNIX, you can run a continuous PowerExchange Logger process in background mode and use the `pwxcmd` program to send commands to that background PowerExchange Logger process.

Batch Mode

In batch mode, the PowerExchange Logger process shuts down after it reaches the end-of-log (EOL) and waits for the interval that is specified in the NO_DATA_WAIT2 parameter of the pwxcl configuration file without receiving additional change data.

Use batch mode in the following situations:

- You run the PowerExchange Logger on a scheduled basis, after batch applications that update the database complete.
- You run the PowerExchange Logger manually for testing or other purposes.

To enable batch mode, set the COLL_END_LOG parameter to 1 in the pwxcl configuration file.

When you run the PowerExchange Logger in batch mode, use batch extraction mode for any workflows that extract change data from the PowerExchange Logger log files.

PowerExchange Logger Usage Considerations

Before you run the PowerExchange Logger for Linux, UNIX, or Windows, review the following usage considerations:

- Informatica recommends that you run the PowerExchange Logger for Linux, UNIX, and Windows and the process that extracts data from the PowerExchange Logger log files on the same system. Configure a PowerExchange Listener on the PowerExchange Logger system and specify that node location in the **Location** attribute of the PowerCenter PWX CDC connection. If you run the PowerExchange Logger and extraction process on different systems and use NFS to access the log files, disable attribute and data caching for the NFS mount point. Otherwise, session failures might occur.
- On Linux and UNIX, the PowerExchange Logger requires sufficient amounts of main memory and virtual memory to process change data. If the memory is not sufficient, PowerExchange writes the error messages PWX-00271 and PWX-00904 to the PowerExchange message log file when you attempt to start the PowerExchange Logger on Linux or UNIX.
To prevent this problem, use the Linux or UNIX ulimit command to set the size limits for maximum memory and virtual memory to unlimited. The specific ulimit syntax varies by platform and shell. For more information about this command, see the documentation for your Linux or UNIX operating system.
- On Linux and UNIX, you can run a PowerExchange Logger process in background mode. For background PowerExchange Logger processes, Informatica recommends that you set the COLL_END_LOG parameter to 0 in the pwxcl configuration file to run the PowerExchange Logger continuously. Also, set the PROMPT parameter to N. If you use PROMPT=Y, the PowerExchange Logger ignores this setting and issues an error message. To send commands to a PowerExchange Logger process that is running in the background, use the pwxcmd program. To enable pwxcmd use, define the CONDENSENAME statement in the pwxcl configuration file and define the SVCNODE statement in the dbmover configuration file.

Logging Data for IBM i or z/OS Sources to Remote PowerExchange Logger Logs

You can log data for a data source that resides on an IBM i (i5/OS) or z/OS system to PowerExchange Logger log files on a Linux, UNIX, or Windows system.

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 system. This practice can reduce the amount of time for CDC processing on the IBM i or z/OS system.

To provide the highest level of security for z/OS data sources, set the first parameter in the SECURITY statement to 2 in the z/OS DBMOVER configuration file. With this setting, the PowerExchange Logger for Linux, UNIX, and Windows can log data from z/OS systems only if its user credentials pass z/OS security checking. The PowerExchange Logger must use a valid z/OS user ID and password combination that has READ access to CAPX.REG.* resource profiles in the FACILITY class, which are managed by your z/OS security product.

For more information, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“Remote Logging Overview” on page 228](#)

Configuring the PowerExchange Logger

To configure the PowerExchange Logger, you must define a PowerExchange Logger configuration file for each source type and instance, as defined in a registration group.

Also, verify that the **Condense** option is set to **Part** in the capture registrations for all source tables that the PowerExchange Logger will process.

If you want the PowerExchange Logger to create separate sets of log files for different groups of tables, create a PowerExchange group definition file. The group definition file defines the group name, the path and file-name prefix for the log files, and the registrations in the group.

RELATED TOPICS:

- [“Customizing the PowerExchange Logger Configuration File” on page 45](#)
- [“Customizing the dbmover Configuration File for the PowerExchange Logger” on page 59](#)
- [“Using PowerExchange Logger Group Definitions” on page 62](#)
- [“Enabling a Capture Registration for PowerExchange Logger Use” on page 44](#)

Enabling a Capture Registration for PowerExchange Logger Use

For the PowerExchange Logger to use a capture registration, the registration must have a status of active and a **Condense** setting of **Part**.

If the PowerExchange Logger does not find any active capture registration, the PowerExchange Logger issues error message PWX-06427 and ends.

To enable a capture registration for PowerExchange Logger use:

1. In the PowerExchange Navigator, open the capture registration.
2. In the Resource Inspector, select **Active** in the **Status** list.
3. In the **Condense** list, select **Part**.

Customizing the PowerExchange Logger Configuration File

Before you start the PowerExchange Logger, configure its parameters in the PowerExchange Logger configuration file.

PowerExchange provides an example configuration file, named `pwxccl`, in the PowerExchange installation directory that is specified in the `PWX_HOME` environment variable on Linux or UNIX or in the `PATH` environment variable on Windows. Use this example file as a starting point for creating a customized file. To preserve the example file in its original state, rename it and copy it to another directory. Then customize the copy. You must specify the `CS` parameter when you start the PowerExchange Logger to identify the custom configuration file path and file name.

When customizing the configuration file, if you enter a parameter value such as a Windows path that contains one or more spaces, enclose the value in double quotation marks ("). Make sure that you use straight quotation marks.

Note: If you used the similar PowerExchange Condense component in a PowerExchange release earlier than 8.6.1, you can copy its `dtlca.cfg` configuration file and then customize the copy. Rename the file to `pwxccl` or use the `CS` execution parameter. The PowerExchange Condense component is no longer supported on Linux, UNIX, and Windows.

Parameter Descriptions

Specify PowerExchange Logger parameters in the `pwxccl.cfg` configuration file.

This topic describes each parameter.

The parameter syntax uses the following notational conventions:

- *Italics* indicate a variable.
- Curly braces { } enclose alternative options. Enter only one option. Do not type the braces when you enter the option.
- A vertical bar | indicates a mutually exclusive choice. When used with braces, you must enter one of the items.
- Underlining indicates a default value.

Parameters:

CAPT_IMAGE={AI|BA}

Type of data images that the PowerExchange Logger stores in its log files. Use this parameter to control whether the PowerExchange Logger stores after images only or both before and after images of the data in its log files.

This parameter affects the amount of storage that you use for PowerExchange Logger log files and whether before image data is available for use in extraction processing.

Enter one of the following options:

- **AI**. Stores only after images in the PowerExchange Logger log files.
- **BA**. Stores both before and after images in the PowerExchange Logger log files.

Default is **AI**. With **AI**, less storage is required for PowerExchange Logger log files. However, the following CDC limitations apply:

- You cannot use before images of the data in extraction processing. If you add before image (BI) fields to extraction maps, PowerCenter CDC sessions that reference the BI fields fail.
- If you add change indicator (CI) fields to extraction maps, PowerCenter CDC sessions that reference the CI fields fail.

Informatica recommends that you enter **BA** if you have sufficient storage for larger log files. The sample `pwxccl` configuration file that PowerExchange supplies specifies **BA**.

Note: If you use **BA** and add CI columns to the extraction maps, any Insert and Delete operations on the source result in Null values in the CI columns. Any Update operations on the source result in the Y or N indicator in the CI columns.

CAPTURE_NODE={node_name|local}

Optional. The node name that the PowerExchange Logger uses to retrieve capture registrations and change data. Specify this parameter only if you use the PowerExchange Logger to capture change data from a source on a remote system.

Enter the node name of the remote source system, as specified in a `NODE` statement in the `dbmover` configuration file on the system where the PowerExchange Logger runs. The PowerExchange Logger uses the specified node name to connect to the PowerExchange Listener on the remote source node to read capture registrations and change data. The PowerExchange Logger then writes the change data to its local log files.

Default is `local`. Do not specify this parameter if the capture registrations and change data are on the local machine where the PowerExchange Logger runs.

You can also specify an optional user ID and password to control connection to the specified node. For more information, see the `CAPTURE_NODE_UID` parameter and the `CAPTURE_NODE_EPWD` or `CAPTURE_NODE_PWD` parameter.

CAPTURE_NODE_EPWD=encrypted_password

An encrypted password that is associated with the user ID specified in the `CAPTURE_NODE_UID` parameter. This password, in conjunction with the `CAPTURE_NODE_UID` value, is used to control PowerExchange access to capture registrations and change data.

Tip: You can create an encrypted password in the PowerExchange Navigator by selecting **File > Encrypt Password**.

If you use remote logging of data from a data source on IBM i (i5/OS) or z/OS to a PowerExchange Logger for Linux, UNIX, and Windows instance, you can enter an encrypted PowerExchange passphrase instead of an encrypted password. Do not encrypt a passphrase that contains invalid characters, such as double-quotation marks, single quotation marks, or currency symbols.

Note: If you specify `CAPTURE_NODE_EPWD`, do not also specify `CAPTURE_NODE_PWD`.

CAPTURE_NODE_PWD=password

A clear text password that is associated with the user ID specified in the `CAPTURE_NODE_UID` parameter. This password, in conjunction with the `CAPTURE_NODE_UID` value, is used to control PowerExchange access to capture registrations and change data.

If you use remote logging of data from a data source on IBM i (i5/OS) or z/OS to a PowerExchange Logger for Linux, UNIX, and Windows instance, you can use a valid PowerExchange passphrase instead

of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

- Uppercase and lowercase letters
- The numbers 0 to 9
- Spaces
- The following special characters:

' - ; # \ , . / ! % & * () _ + { } : @ | < > ?

Note: The first character is an apostrophe.

Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.

When entering a passphrase, you must enclose it with double-quotation marks ("), for example:

```
CAPTURE_NODE_PWD="This is a passphrase!"
```

Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.

To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the *PowerExchange Reference Manual*.

Note: If you specify CAPTURE_NODE_PWD, do not also specify CAPTURE_NODE_EPWD.

CAPTURE_NODE_UID=user_id

A user ID that is used to control access to capture registrations and change data on the local machine or on the remote node that is specified in the CAPTURE_NODE parameter.

Whether this parameter is required depends on the operating system of the local or remote node and the SECURITY setting in its DBMOVER configuration file on that node.

If CAPTURE_NODE specifies a remote IBM i (i5/OS) or z/OS node, the SECURITY setting has the following effects:

- If the SECURITY setting is 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 the SECURITY setting is 1, 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 the SECURITY setting is 2, 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.

If CAPTURE_NODE specifies a Linux, UNIX, or Windows local or remote node, enter a user ID that is valid for your data source type:

- For Db2 sources on Linux, UNIX, or Windows, enter a valid operating system user ID that has Db2 DBADM or SYSADM authority.
- For Microsoft SQL Server instances that use SQL Server Authentication, enter a database user ID that permits access to the SQL Server distribution database. For SQL Server instances that use Windows Authentication, PowerExchange uses the user ID under which the PowerExchange Listener was started. In this case, do not specify this parameter unless you want to specify another user.

- For MySQL sources on Linux or Windows, enter a database user ID that allows access to the MySQL binary logs. This user must have been granted the privileges that are required for MySQL CDC. For more information, see [“Preparing MySQL Sources” on page 123](#).
- For Oracle sources, enter the ORACAPTL user ID that you defined, which permits access to the Oracle online and archive redo logs.
- For PostgreSQL sources on Windows, enter a database user ID that allows access to the PostgreSQL source database. This user must have been granted the privileges that are required for PostgreSQL CDC. For more information, see [“Preparing PostgreSQL CDC Sources” on page 211](#).

Note: If you specify CAPTURE_NODE_UID, you must enter a password or encrypted password in either the CAPTURE_NODE_PWD or CAPTURE_NODE_EPWD parameter, but not both.

COLL_END_LOG={0|1}

Required. The PowerExchange Logger operational mode.

Enter one of the following options:

- **0.** Runs the PowerExchange Logger continuously until you manually stop it. After the Writer subtask completes a processing cycle, it waits for the number of minutes specified in the NO_DATA_WAIT parameter before starting another processing cycle.
- **1.** Runs the PowerExchange Logger in batch mode. The PowerExchange Logger shuts down after the seconds specified in the NO_DATA_WAIT2 parameter elapse and no data has been received.

Default is **0** for continuous mode.

COND_CDCT_RET_P={days|60}

Recommended. Retention period, in days, for CDCT records and PowerExchange Logger log files. Log files that are older than this period and their corresponding CDCT records are deleted automatically during PowerExchange Logger cleanup processing. Cleanup processing occurs during startup, file switch, or shutdown processing.

Enter a number greater than 0. Default is 60.

When setting this parameter, try to minimize the size of the CDCT file while preserving the log files that contain the earliest change data that you might need to access. Use the following guidelines:

- If you set the retention period to a low value, ensure that PowerExchange extracts change data from the PowerExchange Logger log files during this period. Otherwise, the log files for which the retention period has elapsed are deleted and you can lose change data. For example, if the retention period is 5 and you plan not to run extractions during a 10-day holiday, increase the retention period to 15. This approach ensures that the log files with the change data you need are not deleted until the extractions run again.
- If you set the retention period to a high value, the CDCT can become very large, depending on the number active capture registrations. Also, the number of PowerExchange Logger log files might increase. For continuous extraction mode, you can use the PowerExchange Logger FILE_SWITCH parameters to decrease the number of log files and increase their size instead.
- If you use continuous extraction mode, PowerExchange reads the CDCT file each time the interval specified in the FILEWAIT parameter of the CAPX CAPI_CONNECTION statement elapses. If a CDCT file becomes large, this read activity can increase I/O, system resource use, and latency of change data extraction. If you use batch extraction mode, this high read activity is not a consideration.

CONDENSENAME=service_name

Optional. A name for the command-handling service for a PowerExchange Logger for Linux, UNIX, and Windows process to which infacmd pwx or pwxcmd commands are issued.

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 infacmd pwx or pwxcmd commands.

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

Tip: If you run the PowerExchange Logger as a background process in continuous mode, specify this parameter so that you can use the pwxcmd program to issue commands to the PowerExchange Logger. Without the use of pwxcmd, you cannot shut down a PowerExchange Logger process that is running in the background or send status information to a computer that is remote from where the PowerExchange Logger runs.

CONDENSE_SHUTDOWN_TIMEOUT={seconds |600}

Maximum amount of time, in seconds, that the PowerExchange Logger waits after receiving the SHUTDOWN or pwxcmd shutdown command before stopping.

Enter a number from 0 through 2147483647. Default is 600.

During a shutdown, the PowerExchange Logger updates the CDCT file for each capture registration that is used to capture change data. If you have a large number of capture registrations, you might need to increase this timeout period.

CONN_OVR=capi_connection_name

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.

Enter a valid CAPI_CONNECTION name for the source type.

Informatica recommends that you specify CONN_OVR because it is the only type of override that the PowerExchange Logger can use.

DBID=instance_name

Required. A source identifier, sometimes called the *instance* 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 matches the instance or database name that is displayed in the Resource Inspector of the PowerExchange Navigator for the registration group that contains the capture registrations.

For sources on Linux, UNIX, and Windows, enter one of the following options:

- For Db2 sources on Linux, UNIX, or Windows, enter the **Database** name that is displayed for the registration group in the **Resource Inspector**.

- For Microsoft SQL Server, this value depends on whether you also specify the optional DISTSRV and DISTDB parameters in the PowerExchange Logger configuration file and whether you entered the **Instance** identifier when creating the registration group:
 - If you specify the optional DISTSRV and DISTDB parameters to have the single PowerExchange Logger instance log data for all of the registered articles that are published to the distribution database, enter a name that serves as the collection identifier for all of the registrations. This name must be one to eight characters in length and start with a letter. This name overrides the instance name that is associated with the individual registrations.

Note: When you perform a CAPXRT database row test, you must enter this name in the **MSS LUW DBID** field in the CAPXRT Advanced Parameters dialog box. When you define a PWXPC connection for PowerCenter CDC sessions that extract data from PowerExchange Logger log files, enter this value for the **Logger DBID** attribute on the PowerCenter PWX MSSQL CDC Real Time connection.
 - If you do not specify the DISTSRV and DISTDB parameters, enter the value from the **Instance** field that is displayed for the registration group in the PowerExchange Navigator Resource Inspector. The instance identifier is either the unique user-defined identifier that was optionally entered for the database name and database server combination during registration group creation in the PowerExchange Navigator or the instance identifier that PowerExchange generated if you did not specify an instance identifier.
- For MySQL, enter the **Instance** name that is displayed for the registration group in the **Resource Inspector**.
- For Oracle, enter the **Instance** name that is displayed for the registration group in the **Resource Inspector**. This value also should match the first positional parameter of the ORACLEID statement in the dbmover configuration file.
- For PostgreSQL, enter the **Instance** name that is displayed for the registration group in the **Resource Inspector**.

If you use the PowerExchange Logger to log data from remote data sources on IBM i (i5/OS) or z/OS, enter one of the following options:

- For Adabas, enter the **Instance** name that is displayed for the registration group.
- For Datacom, enter the **MUF Name** value that is displayed for the registration group.

Alternatively, if you use Datacom synchronous CDC, enter the value of the MUF parameter in the DTLINPUT data set specified in the MUF JCL. Or, if you use Datacom table-based CDC, enter the value of REG_MUF parameter in the ECCRD CMP member of the RUNLIB library.
- For Db2 for i, 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.
- For Db2 for z/OS, enter the **Instance** name that is displayed for the registration group. This name should match the RN parameter value in the DB2 statement in the RUNLIB(REPDB2OP) member.
- For IDMS Log-based CDC, enter the **Logsid** value that is displayed for the registration group. This value should match the LOGSID parameter value in the RUNLIB(ECCRIDLP) member.
- For IMS, enter the **IMSID** value that is displayed for the registration group. For IMS log-based CDC, this value should match the first parameter value in the IMSID statement in the RUNLIB(CAPTIMS) member.
- For VSAM, enter the **Instance** name that is displayed for the registration group.

DB_TYPE={ADA|AS4|DB2|DCM|IDL|IMS|MSS|MYS|ORA|PGS|UDB|VSM}

Required. Source database type.

For sources on Linux, UNIX, or Windows, enter one of the following options:

- **MSS** for Microsoft SQL Server sources
- **MYS** for MySQL sources
- **ORA** for Oracle sources
- **PGS** for PostgreSQL sources
- **UDB** for Db2 sources on Linux, UNIX, or Windows

If you use the PowerExchange Logger to log data from remote data sources on IBM i (i5/OS) or z/OS systems, enter one of the following options:

- **ADA** for Adabas sources
- **AS4** for Db2 for i sources
- **DB2** for Db2 for z/OS sources
- **DCM** for Datacom sources.
- **IDL** for IDMS log-based CDC sources
- **IMS** for IMS sources
- **VSM** for VSAM sources

DISTDB=distribution_database_name

For Microsoft SQL Server sources, the name of the distribution database. Enter this parameter if you want the single PowerExchange Logger instance to read data for all of the registered articles that are published to the distribution database, regardless of their source publication databases. Otherwise, you must run a separate PowerExchange Logger instance for each source database.

This parameter is required if MULTIPUB=Y is specified in the MSQL CAPI_CONNECTION statement. Otherwise, this parameter is optional.

If you specify this parameter, you must also specify the DISTSRV parameter.

Note: For the PowerExchange Logger to extract change data from the distribution database for articles in multiple publication databases in one pass, you must also set the MULTIPUB parameter in the MSQL CAPI_CONNECTION statement to Y. Otherwise, the extraction fails with message PWX-15757. Also, you must specify the DBID parameter value for the **Logger DBID** attribute on the PowerCenter PWX MSSQL CDC Real Time connection.

DISTSRV=distribution_database_server

For Microsoft SQL Server sources, the network name of the server that hosts the distribution database. This name might be different from the network name of the SQL Server instance if the distribution database resides on a different server. Enter this parameter if you want the single PowerExchange Logger instance to read data for all of the registered articles that are published to the distribution database, regardless of their source publication databases. Otherwise, you must run a separate PowerExchange Logger instance for each source database.

This parameter is required if MULTIPUB=Y is specified in the MSQL CAPI_CONNECTION statement. Otherwise, this parameter is optional.

If you specify this parameter, you must also specify the DISTDB parameter.

Note: For the PowerExchange Logger to extract change data from the distribution database for articles in multiple publication databases in one pass, you must also set the MULTIPUB parameter in the MSQL CAPI_CONNECTION statement to Y. Otherwise, the extraction fails with message PWX-15757. Also, you must specify the DBID parameter value for the **Logger DBID** attribute on the PowerCenter PWX MSSQL CDC Real Time connection.

ENCRYPTPWD=encrypted_encryption_password

A password in encrypted format for enabling the encryption of PowerExchange Logger log files. With this password, the PowerExchange Logger can generate a unique encryption key for each Logger log file. The password is stored in the CDCT file in encrypted format. For security purposes, the password is not stored in CDCT backup files and is not displayed in the CDCT reports that you can generate with the PWXUCDCT utility.

You can set the AES algorithm to use for log file encryption in the ENCRYPTOPT parameter. The default is AES128.

If you specify this parameter, do not also specify the ENCRYPTPWD parameter in the same pwxcl.cfg file.

If you specify this parameter and cold start the PowerExchange Logger with a pwxcl command that includes the encryptpwd parameter, the ENCRYPTPWD parameter in the configuration file takes precedence.

If you change this ENCRYPTPWD password after log files have been encrypted, you must cold start the PowerExchange Logger. Otherwise, the change is ignored.

Tip: For optimal security, Informatica recommends that you specify the encryption password in a pwxcl command for cold starting the PowerExchange Logger rather than in the pwxcl.cfg configuration file. This practice can reduce the risk of malicious access to the encryption password for the following reasons: 1) The encryption password is not stored in the pwxcl.cfg file, and 2) You can remove the password from the command line after a successful cold start. If you specify the encryption password in a pwxcl command for a cold start and then need to restore the CDCT file later, you must enter the same encryption password in the RESTORE_CDCT command of the PWXUCDCT utility.

To *not* encrypt PowerExchange Logger log files, do not enter an encryption password in the pwxcl.cfg configuration file or in the pwxcl command for a cold start.

ENCRYPTOPT={AES128|AES192|AES256}

The AES encryption algorithm that you want to use for encrypting PowerExchange log files. To enable encryption, you must also specify an encryption password in the ENCRYPTPWD or ENCRYPTPWD parameter in the pwxcl.cfg configuration file or specify the encryptpwd parameter in a pwxcl command that you use to cold start the PowerExchange Logger.

Default is AES128.

ENCRYPTPWD=clear_text_encryption_password

A password in clear text format for enabling the encryption of PowerExchange Logger log files. With this password, the PowerExchange Logger can generate a unique encryption key for each Logger log file. The password is stored in the CDCT file in encrypted format. For security purposes, the password is not included in CDCT backup files and is not displayed in the CDCT reports that you can generate with the PWXUCDCT utility.

You can set the AES algorithm to use for log file encryption in the ENCRYPTOPT parameter. The default is AES128.

If you specify this parameter, do not also specify the ENCRYPTPWD parameter in the same pwxcl.cfg file.

If you specify this parameter and cold start the PowerExchange Logger with a pwxcl command that includes the encryptpwd parameter, an error occurs. Do not specify the ENCRYPTPWD parameter in the configuration file and also specify the encryptpwd parameter in the PWXCCL command.

If you change this ENCRYPTPWD password after log files have been encrypted, you must cold start the PowerExchange Logger. Otherwise, the change is ignored.

Tip: For optimal security, Informatica recommends that you specify the encryption password in a `pwxccl` command for cold starting the PowerExchange Logger rather than in the `pwxccl.cfg` configuration file. This practice can reduce the risk of malicious access to the encryption password for the following reasons: 1) The encryption password is not stored in the `pwxccl.cfg` file, and 2) You can remove the password from the command line after a successful cold start. If you specify the encryption password in a `pwxccl` command for a cold start and then need to restore the CDCT file later, you must enter the same encryption password in the `RESTORE_CDCT` command of the `PWXUCDCT` utility.

To *not* encrypt PowerExchange Logger log files, do not enter an encryption password in the `pwxccl.cfg` configuration file or in the `pwxccl` command for a cold start.

EPWD

Deprecated. Use `CAPTURE_NODE_EPWD` instead. If both `CAPTURE_NODE_EPWD` and `EPWD` are specified, `CAPTURE_NODE_EPWD` takes precedence.

EXT_CAPT_MASK=path/prefix

Required. An existing directory path and a unique prefix to be used for generating the PowerExchange Logger log files.

Maximum length is 256 characters.

For example:

```
/capture/pwxlog
```

If you enter a value that include spaces, you must enclose the value in double quotation marks (").

If you need to enter a Windows network path, use three leading backslashes (\\\) for PowerExchange to parse the network path correctly. For example:

```
EXT_CAPT_MASK=\\host\Shared Folders\C\CDC_SHARED\logfiles
```

Verify that no existing files match the path and prefix. PowerExchange considers any file that matches this path and prefix to be a PowerExchange Logger log file, even if it is unrelated to PowerExchange Logger processing.

To create the log files, the PowerExchange Logger appends the following information:

```
.CND.CPyymmdd.Thhmmssnnn
```

Where:

- *yymmdd* is a date composed of a two-digit year, a month, and a day.
- *hhmmss* is 24-hour time value, including hours, minutes, seconds.
- *nnn* is a generated sequence number that ensures uniqueness of the file name.

For example:

```
/capture/pwxlog.CND.CP080718.T1545001
```

FILE_FLUSH_VAL={seconds|-1}

Recommended. File flush interval in seconds. When this interval elapses, the PowerExchange Logger writes any outstanding change data that it read from the source to log files on disk. After the change data is flushed to disk, CDC sessions that use continuous extraction mode can read the change data. This parameter affects the latency of continuous change data extractions.

Valid values are:

- **-1.** The PowerExchange Logger does *not* flush outstanding change data to the current log file on disk based on this parameter. Enter this value only if you use batch extraction mode. If you use continuous extraction mode, this value can increase the latency of your continuous extraction sessions.
- **0.** The PowerExchange Logger flushes outstanding change data after every record.
- *Any value from 1 through 86400.* The PowerExchange Logger flushes outstanding change data at the specified interval.

Default is -1.

Warning: A value of 0 can degrade the performance of the PowerExchange Logger and file system.

Set this value as appropriate for your CDC environment. Values that are too high can increase change extraction latency, and values that are too low can degrade PowerExchange Logger and system performance. Informatica recommends that you set this parameter to a value that is equal to or greater than the NO_DATA_WAIT2 value because file flushes cannot occur until the NO_DATA_WAIT2 period expires.

FILE_SWITCH_CRIT={M|R}

Type of units to use for the FILE_SWITCH_VAL parameter, which determines when to do an automatic file switch.

Enter one of the following options:

- **M** for minutes.
- **R** for records.

Default is **M**.

FILE_SWITCH_ON_CHECKPOINT={Y|N}

Controls whether automatic file switching occurs when the PowerExchange Logger is notified of a checkpoint event. When the Logger receives notification of a checkpoint, it initiates a file switch to harden the captured data to disk and advance the restart point. Options are:

- **Y.** Enables automatic file switches when checkpoint events occur.
- **N.** Disables automatic file switches when checkpoint events occur.

Default is Y.

FILE_SWITCH_VAL={minutes_or_records|30}

Number of minutes or change records, as determined by FILE_SWITCH_CRIT, that must elapse before PowerExchange performs a file switch.

Enter a number greater than 0. Default is 30. Entering a very large number might cause unpredictable results.

For example, if you use 30 and enter FILE_SWITCH_CRIT=R, the PowerExchange Logger performs a file switch every 30 records. If FILE_SWITCH_CRIT=M, the PowerExchange Logger performs a file switch every 30 minutes.

If the PowerExchange Logger log files contain no data when the FILE_SWITCH_VAL threshold is reached, the file switch does not occur.

This value affects the size of the PowerExchange Logger log files. Specify a value that results in log files of the appropriate size for your environment.

Tip: When using continuous extraction mode, set this parameter such that you have larger log files and a smaller CDCT file. When using batch extraction mode, set this parameter to a value that causes file switches to occur within the time frame that meets your change extraction latency requirements.

GROUPDEFS=*path/file_name*

Path and file name of the optional PowerExchange Logger group definition file. This file defines groups of capture registrations that the PowerExchange Logger uses to capture change data to separate sets of log files. It also defines the path that the PowerExchange Logger uses to create the log files that contain the change data for each group.

This parameter is optional and has no default.

If you need to enter a Windows network path, use three leading backslashes (\\\) for PowerExchange to parse the network path correctly.

Maximum length is 255 characters.

LOGGER_DELETES_EXPIRED_CDCT_RECORDS={Y|N}

Controls whether the PowerExchange Logger deletes CDCT records for log files for which the retention period has expired.

Enter one of the following options:

- **Y.** The PowerExchange Logger deletes expired CDCT records during file switches. You cannot use the PWXUCDCT utility DELETE_EXPIRED_FILES command to manually delete expired log files and their related CDCT records.
- **N.** The PowerExchange Logger does not delete expired CDCT records. However, you can use the PWXUCDCT utility DELETE_EXPIRED_FILES command to manually delete expired log files and their related CDCT records.

Note: This parameter does not affect PowerExchange Logger deletions of CDCT records that are rolled back because of a cold start or a warm start to a prior point in time.

Default is **Y**.

NO_DATA_WAIT={minutes|1}

If you run the PowerExchange Logger in continuous mode, specify the number of minutes that the PowerExchange Logger must wait before starting the next logging cycle.

Enter 0 or a number greater than 0. Default is 1.

A value of 0 causes no waiting to occur between PowerExchange Logger processing cycles. If source data is not available, the CAPI sleeps.

For continuous extraction mode, enter 0 for no waiting, or enter a low value so that the next logging cycle starts shortly after the current one completes.

If the value of FILE_SWITCH_CRIT is M and the value of FILE_SWITCH_VAL is less than the value of NO_DATA_WAIT, the PowerExchange Logger uses the FILE_SWITCH_VAL value instead.

NO_DATA_WAIT2={seconds|600}

The number of seconds that PowerExchange waits at the end-of-log for more change data before returning control to the PowerExchange Logger. If this wait period elapses and no new change data has been received, PowerExchange returns control to the PowerExchange Logger, and the PowerExchange Logger then stops the current logging cycle.

Enter a number greater than 0. Default is 600.

Informatica recommends a value of 10. If you enter a greater value, execution of commands for the PowerExchange Logger might be delayed.

Use the same value for the FILE_FLUSH_VAL parameter.

PROMPT={Y|N}

When you run the PowerExchange Logger in foreground mode, controls whether PowerExchange displays a user confirmation prompt and waits for a response when you perform one of the following actions:

- Cold start the PowerExchange Logger.
- Warm start the PowerExchange Logger from a previous position in the change stream. This situation occurs only if the CDCT file still contains records related to the deleted files.

Enter one of the following options:

- **Y.** Displays the confirmation message PWX-33236 for a cold start or PWX-33242 for a warm start. You must respond to the message for startup processing to continue.
- **N.** Does not display the confirmation messages. PowerExchange attempts to start without first prompting for user confirmation.

If you run the PowerExchange Logger in foreground mode, the default is Y.

If you run the PowerExchange Logger in background mode or as a PowerExchange Logger Service in the Informatica domain, the default is N. In this case, if you enter PROMPT=Y in the pwxcl.cfg file, the PowerExchange Logger ignores this setting, issues error message PWX-33253, and continues processing.

PWD

A deprecated parameter. Use CAPTURE_NODE_PWD instead. If both CAPTURE_NODE_PWD and PWD are specified, CAPTURE_NODE_PWD takes precedence.

RESTART_TOKEN and SEQUENCE_TOKEN

A pair of token values that define a restart point for starting change data processing when a PowerExchange Logger is cold started.

Depending on how you set these parameters, PowerExchange Logger processing starts from one of the following restart points during a cold start:

- If you do not specify these parameters, processing starts from the current end-of-log position.
- If you enter 0 for both parameters, processing starts from the default start location for the source type, as follows:
 - For Db2, the default location is the current log position at the time the PowerExchange capture catalog was created.
 - For Microsoft SQL Server, the default location is the oldest data available in the publication database.
 - For MySQL, the default location is the oldest data available in the binary log. This position is the beginning of the binary log that has the log name that includes the lowest numeric suffix value. You can use the SHOW BINARY LOGS statement to identify this log.
 - For Oracle, the default location is the beginning of the most recent archive log.
 - For PostgreSQL, the default location is the oldest data stored in the replication store table. If the replication store table is empty, the default location is the oldest record waiting to be read from the logical replication slot.
- If you enter restart token and sequence token values other than 0, processing resumes from the specific restart point defined by these token values. To perform a special start of the PowerExchange Logger, you must specify the RESTART_TOKEN and SEQUENCE_TOKEN parameters with specific values, and the SEQUENCE_TOKEN value must be greater than the sequence token in the CDCT file.

If you use remote logging of change data from IBM i (i5/OS) z/OS data sources, see the PowerExchange Condense chapter in the *PowerExchange CDC Guide for i5/OS* or *PowerExchange CDC Guide for z/OS* for information about what to enter for these parameters.

SIGNALLING={N|Y}

Indicates whether the PowerExchange Logger attempts to take automatic action in the event of certain errors.

Enter one of the following options:

- **N.** The PowerExchange Logger does not automatically trap and handle system errors. Instead, the operating system uses default error handling. Usually, the default handling is to report the program line in error and dump memory.
- **Y.** The PowerExchange Logger automatically handles certain errors such as memory corruption. After the PowerExchange Logger handles the error, it attempts to shut down in a controlled manner.

Default is **N**.

STATS=(MONITOR[,interval|0])

Enables PowerExchange Logger collection of the following monitoring statistics:

- PowerExchange Logger process ID (PID)
- Status of the PowerExchange Logger Writer task
- CPU time used by the PowerExchange Logger - total and for Writer task functions.
- Memory use (current/total/maximum) in kilobytes, total and for the Controller, Command Handler, and Writer tasks
- Record counts, including the number of inserts, updates, deletes, and commits that the PowerExchange Logger processed - total since the Logger started and for the current active log file and the active logging cycle
- Total number of UOWs and bytes that the Writer task processed

Also enables collection of the following statistics for PowerExchange Logger group definitions, if defined:

- The number of DML operations and commits processed for each group
- Then number of change records that have not yet been flushed to a Logger log file on disk
- The name of the open Logger log file for each group and the file open timestamp

Logger statistics are printed to the PowerExchange message log and on screen when any of the following events occur:

- You enter the DL or DG command at the command line, or enter the `pxwcmd displaystats -tp {logger|groups}` command from a remote Linux, UNIX, or Windows system.
- You issue the SHUTCOND or SHUTDOWN command to a PowerExchange Logger that runs in continuous mode.
- A PowerExchange Logger that runs in batch mode finishes its run and shuts down.

For more information about the monitoring commands and related reports, see the *PowerExchange Command Reference*.

Optionally, include the *interval* subparameter in the STATS statement to publish the Logger statistics at a regular interval.

{interval|0}

Optional. The interval, in minutes, after which PowerExchange publishes monitoring statistics for the PowerExchange Logger. The interval-based statistics that are written to the PowerExchange message log file are the same as those published by the DL (or DS) command and `pwxcmd displaystats -tp logger` command. However, a subset of the message output is displayed on screen to prevent flooding the screen with messages over time.

Note: The Logger still issues monitoring messages at shutdown, regardless of whether you specify the *interval* parameter.

Valid values are 0 through 120. Default is 0, which disables interval-based reporting of PowerExchange Logger monitoring statistics. With the default value, PowerExchange writes these statistics only when one of the following Logger commands is issued: DL, `pwxcmd displaystats -tp logger`, SHUTCOND, or SHUTDOWN.

UID

A deprecated parameter. Use CAPTURE_NODE_UID instead. If both CAPTURE_NODE_UID and UID are specified, CAPTURE_NODE_UID takes precedence.

VERBOSE={Y|N}

Indicates whether the PowerExchange Logger writes verbose or terse messages to the PowerExchange message log file for activities that it performs frequently, such as cleanup, condense, and file-switch processing.

Enter one of the following options:

- **Y.** Verbose messaging. The PowerExchange Logger logs multiple messages at various processing points, such as when starting or ending a cycle of reading source data or doing a file switch. Verbose messaging often includes processing statistics such as records processed and elapsed time.
- **N.** Terse messaging. The PowerExchange Logger logs a single terse message for each file switch.

Default is **Y**.

Example pwxccl Configuration File

PowerExchange provides an example PowerExchange Logger configuration file, `pwxccl`, in the PowerExchange installation directory. You can use this file to create a custom configuration file.

The following example shows basic configuration statements:

```
/* Name for PWXCMD control
/*CONDENSENAME=PWXCCL1

DBID=ORACOLL1
DB_TYPE=ORA
CAPTURE_NODE_UID=user_id
CAPTURE_NODE_EPWD=encrypted_password
/* CAPTURE_NODE_PWD=plain_text_password

PROMPT=Y

EXT_CAPT_MASK=/capture/condense0
COND_CDCT_RET_P=50
LOGGER_DELETES_EXPIRED_CDCT_RECORDS=Y

/* 0 = continuous, 1 = Stop at end-of-log (batch)
COLL_END_LOG=0

/* Number of minutes to wait between CAPI read cycles
NO_DATA_WAIT=0
/* Number of seconds to wait at the end-of-log for more change data
NO_DATA_WAIT2=60
```

```

/* Number of seconds before flushing, or writing, data to the current log file on disk
/* -1 = No flush, 0 = flush every record, 1 to N flush every N seconds
/*FILE_FLUSH_VAL=60
/* Minimum number of FILE_SWITCH_CRIT units after new CDCT source entry
(normal,coldstart)
FILE_SWITCH_CRIT=M
FILE_SWITCH_VAL=20

CAPT_IMAGE=BA
SEQUENCE_TOKEN=00
RESTART_TOKEN=00

```

Customizing the dbmover Configuration File for the PowerExchange Logger

To use the PowerExchange Logger, you must define the CAPT_PATH statement and certain source-specific statements in the dbmover configuration file.

Also, you can include some optional parameters to help make finding messages for the PowerExchange Logger easier or to send commands to a PowerExchange Logger process that is running in background mode.

Use the following key parameters:

CAPT_PATH

Required. Path to the local directory on a Linux, UNIX, and Windows system that contains the control files for CDC, including the CCT and CDCT files. The CCT file contains information about capture registrations. The CDCT file contains information about the PowerExchange Logger log files, such as file names and number of records.

CAPX CAPI_CONNECTION

Required for continuous extraction mode. If you want the CAPI to use continuous extraction mode for the extraction of change data from PowerExchange Logger log files, you must define a CAPX CAPI_CONNECTION statement.

LOGPATH

Optional. A unique path and directory for PowerExchange message log files on a Linux, UNIX, or Windows system. Use this parameter to create message log files in a directory that is separate from your current working directory so that you can find the message log files more easily.

SVCNODE

Optional. The TCP/IP port on which a command-handling service for a PowerExchange Logger process listens for commands that you issue with the pwxcmd program. You must define this parameter if you run the PowerExchange Logger process in background mode on a Linux or UNIX system. For more information about pwxcmd commands, see the *PowerExchange Command Reference*.

TRACING

Optional. Enables alternative logging. PowerExchange creates a set of alternative log files for each PowerExchange process in a separate directory. You can specify the directory location, the number of log files, and the log file size in MB. When a log file reaches the specified size, PowerExchange switches to the next log file and begins overwriting any data in that file. Alternative logging is faster and enables you to customize the amount of data logged for long-running jobs, such as a PowerExchange Logger process that runs in continuous mode. If you specify this statement, also specify the LOGPATH statement.

In addition to these parameters, the PowerExchange Logger requires source-specific statements.

For more information about all DBMOVER configuration parameters, see the *PowerExchange Reference Manual*.

CAPX_CONNECTION - CAPX Statement

The CAPX CAPX_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files.

Operating Systems: Linux, UNIX, and Windows

Required: Yes for continuous extraction mode

Syntax:

```
CAPI_CONNECTION=( [DLLTRACE=trace_id]
                  ,NAME=capi_connection_name
                  [,TRACE=trace_name]
                  ,TYPE=(CAPX
                        ,DFLTINST=instance_name
                        [,FILEWAIT={seconds|1}]
                        [,NOSEQVAL={N|Y}]
                        [,RSTRADV=seconds]
                        [,RSTRANMODE={N|Y}]
                        [,VALIDATEREGS={N|Y}]
                        )
                  )
```

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=(CAPX, ...)

Required. Type of CAPI_CONNECTION statement. For continuous extraction mode, this value must be CAPX.

DFLTINST=instance_name

Required. A source instance identifier that is specified for the registration group. This value must match the DBID value that is specified in the PowerExchange Logger configuration file.

To determine this value, view the registration group properties in the PowerExchange Navigator. Depending on the source type, enter one of the following values:

- For Adabas, Db2 for i (i5/OS), Db2 for z/OS, MySQL, Oracle, PostgreSQL, and VSAM, enter the name that is displayed in the **Instance** field for the registration group in the **Resource Inspector**.
- For Datacom, enter the name of the Multi-User Facility (MUF) in the **MUF Name** field.
- For a Db2 source on Linux, UNIX, or Windows, enter the name of the database in the **Database** field.

- For an IDMS log-based source, enter the name of the database in the **DB Name** field.
- For an IMS source, enter the recon identifier for the database in the **RECON ID** field.
- For Microsoft SQL Server, this value depends on whether you specify the optional DISTSRV and DISTDB parameters in the PowerExchange Logger configuration file:
 - If you specify the DISTSRV and DISTDB parameters, enter the DBID name that you use as the collection identifier for all of the registrations. This name overrides the instance name that is associated with the individual registrations.
 - If you do not specify the DISTSRV and DISTDB parameters, enter the value that the PowerExchange Navigator generates and displays in the **Instance** field of the **Resource Inspector** for the registration group. The generated value is composed of the first four characters of the database name followed by a generated number, which starts at 000.

Maximum length is eight alphanumeric characters.

FILEWAIT={seconds|1}

Optional. The number of seconds that PowerExchange waits before checking for new PowerExchange Logger log files.

For the *seconds* variable, enter a number from 1 through 86400. Default is 1.

NOSEQVAL={N|Y}

Optional. If you receive error message PWX-36944 after starting a CDC session, the sequence token that PWXPC passed to PowerExchange is earlier than the sequence token that is recorded in the PowerExchange Logger CDCT file. If you want the session to continue and you can tolerate some data loss, you can set this parameter to Y. The Log Reader then begins extracting the earliest available data in the log files. With the default value of N, the session fails.

RSTRADV=seconds

Optional. The time interval, in seconds, that PowerExchange waits before advancing the restart and sequence tokens for a registered data source during periods when UOWs contain no changes of CDC interest for a data source. When the wait interval expires, PowerExchange returns the next committed "empty UOW," which includes only updated restart information.

For the *seconds* variable, enter a number from 0 through 86400. No default value is provided. A value of 0 causes PowerExchange to return an empty UOW after each UOW processed. Consequently, a value of 0 can degrade performance.

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

- PowerExchange completes processing a UOW that includes changes of CDC interest.
- PowerExchange returns an "empty UOW" because the RSTRADV wait interval expired without any new changes of CDC interest having been received.

For example, if you specify 5, PowerExchange waits 5 seconds after it completes processing the last UOW or after the previous RSTRADV 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 specify the RSTRADV parameter with any valid value, PowerExchange always advances the restart and sequence tokens when the Log Reader reaches the end of a Logger log file, even if the RSTRADV interval has not expired. This behavior ensures that restart and sequence tokens are advanced even when the CDC session run time is shorter than the RSTRADV interval. This situation is most likely to occur if you have source tables that have a low level of update activity.

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, even those not of CDC interest, 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.

RSTRANMODE={N|Y}

Optional. If you are migrating from real-time extraction mode to continuous extraction mode for CDC sessions that have not previously used the PowerExchange Logger for Linux, UNIX, and Windows, set this parameter to Y. This setting enables PowerExchange to convert restart token information to the format that PowerExchange Logger requires, when CDC sessions run. Retain the Y setting until PowerExchange completes converting the restart tokens for all registered source tables and all CDC sessions that use this CAPX CAPI_CONNECTION. Then, set this parameter to N or delete it. Default is N.

VALIDATEREGS={N|Y}

Optional. Controls whether the PowerExchange extraction of data from PowerExchange Logger for Linux, UNIX, and Windows ends or continues if it detects a source entry in the CDC interest list for which the PowerExchange Logger is *not* capturing change data. In this situation, the extraction process issues message PWX-36947, either as error message or informational message.

Options are:

- **Y.** The PowerExchange Logger extraction process issues message PWX-36947 as an error message and then ends.
- **N.** The PowerExchange Logger extraction process issues message PWX-36947 as an informational message and continues processing.

Default is N.

Using PowerExchange Logger Group Definitions

To create separate sets of PowerExchange Logger log files for groups of tables, create a PowerExchange Logger group definition file. Then, specify its path and file name in the GROUPDEFS parameter of the pwxccf configuration file.

When the PowerExchange Logger process starts, it reads the group definition file and creates a separate set of log files for each defined group.

Group definitions can help improve the efficiency of extraction sessions because the extractions target a more specific set of PowerExchange Logger log files.

By default, the PowerExchange Logger processes change data for all tables that reside on the instance specified by the DBID parameter and that have active capture registrations with the **Condense** option set to **Part**. Changes for all of these tables are written to a single set of log files. For a table with a low level of change activity, the extraction process might need to read many change records in the PowerExchange Logger log files before finding the changes of interest.

With group definitions, you can define a group that includes a subset of capture registrations. The PowerExchange Logger then writes change data to a separate set of log files for the tables that are associated with these registrations. When an extraction process runs, it is more likely to find the change data for a table in the group faster because it reads only the log files for that group.

For example, if you have five source tables with a low level of change activity and one table with a high level of change activity, you can define a group that includes the low-activity tables and another group that includes only the high-activity table only. Then, in PowerCenter, define a CDC session that extracts change data from the PowerExchange Logger log files for the low-activity group, and define another CDC session that extracts change from the log files for the high-activity group. This configuration enables the CDC session for the low-activity tables to find and extract the few change records for these tables much more quickly.

If you have multiple tables with the same table name but different schemas, you can define a single capture registration for the table and specify it once, under a single group, in the group definition file. For any other group that includes the same table with a different schema, you can override the schema name in the group definition by using a SCHEMA statement. By using the SCHEMA statement, you can avoid creating multiple capture registrations and specifying each one in the group definition file. For example, if you have an EMPLOYEE table with different schemas for the north, south, east, and west regions, you can register the north EMPLOYEE table only and specify the capture registration name in the NORTH group. Then specify only the override schemas in the EAST, WEST, and SOUTH groups.

Note: SCHEMA statements are optional for Db2 for i (i5/OS) sources and for Db2 and Oracle sources on Linux, UNIX, and Windows. SCHEMA statements are not supported for SQL Server or SAP HANA sources on Windows or any data source on z/OS.

On Linux, UNIX, and Windows, PowerExchange requirements for unregistered versions of tables, those for which you specify a SCHEMA statement instead of a REG statement in the group definition file, vary by source type:

- For Db2 on Linux, UNIX, or Windows, you must define any unregistered version of a table with the Db2 DATA CAPTURE CHANGES clause.
- For Microsoft SQL Server, you must register all versions of a table in PowerExchange and specify a REG statement in the group definition file.
- For MySQL, you must register all versions of a table in PowerExchange and specify a REG statement in the group definition file.
- For Oracle, you must create an Oracle supplemental log group for the unregistered table, which is similar to the supplemental log group that was created for the registered copy of the table at registration completion.
- For PostgreSQL, you must register all versions of a table in PowerExchange and specify a REG statement in the group definition file.
- For SAP HANA on Linux, you must register all versions of a table in PowerExchange and specify a REG statement in the group definition file.

Tip: When using group definitions, you can optimize extraction efficiency by defining a CDC session in PowerCenter for each group of tables defined in the group definition file.

RELATED TOPICS:

- [“PowerExchange Logger Group Definition File” on page 63](#)
- [“Example Group Definition File” on page 65](#)

PowerExchange Logger Group Definition File

A PowerExchange Logger group definition file contains one or more GROUP statements. Each GROUP statement contains REG or SCHEMA parameters that directly or indirectly identify a group of capture registrations and tables for which you want to create separate sets of PowerExchange Logger log files.

For the PowerExchange Logger to use the group definition file, you must specify the path and file name of the file in the GROUPDEFS parameter of the pwxcl.cfg file.

Note: If you specify the GROUPDEFS parameter, the PowerExchange Logger ignores the EXT_CAPT_MASK parameter in the pwxcl.cfg file when creating log files.

The following table describes the statements and parameters in the group definition file:

Statement	Positional Parameter	Description	Data Type and Length
GROUP	<i>group_name</i>	A unique user-defined name for the group. This parameter is required.	VARCHAR(255)
	<i>external_capture_mask</i>	A unique path and file-name prefix for the PowerExchange Logger log files that are created for tables in the group. This parameter is required. Note: This path and prefix is used for the group instead of the path and prefix that are specified in the EXT_CAPT_MASK parameter of the pwxcl.cfg file.	VARCHAR(255)
REG	<i>registration_name</i>	Optional. Registration name that is specified in the Name field of a capture registration. This lowercase name can be the full registration name or the first part of the name followed by an asterisk (*) wildcard. This parameter is optional. If omitted, the PowerExchange Logger assumes REG=.*.	VARCHAR(8)
SCHEMA	<i>schema_name</i>	Optional. Name of the override schema. You can optionally use this parameter for Db2 for i (i5/OS) sources and for Db2 LUW, MySQL, and Oracle sources on Linux, UNIX, or Windows. Note: This parameter is not supported for SQL Server sources on Windows. If you use the offloading feature to have the PowerExchange Logger process data from z/OS sources, this parameter is also not supported for the z/OS sources.	VARCHAR(255)

Use the following rules and guidelines when you create a PowerExchange Logger group definition file:

- Each *group_name* must be unique within the group definition file.
- Each *external_capture_mask* must be unique on the system.
- SCHEMA statements are optional for Db2 for i (i5/OS) sources and for Db2 LUW, MySQL, and Oracle sources on Linux, UNIX, or Windows. SCHEMA statements are not supported for SQL Server sources on Windows or any data source on z/OS.
- If you use a SCHEMA statement, you must define a capture registration in the group. You can specify multiple SCHEMA statements under a GROUP if you want the tables with those schemas to be included in the group.
- REG statements apply to the preceding SCHEMA statement. If a SCHEMA statement is not present, the REG statements apply to the preceding GROUP statement.
- If the file contains a SCHEMA or REG statement without a preceding GROUP statement, the PowerExchange Logger issues a syntax error.
- Do not include the same *schema.table* value in more than one group. If a table is included in multiple groups, only the first group that includes the table logs changes for it.
- If you do not define at least one REG statement for a GROUP, the PowerExchange Logger includes all of the active capture registrations that are defined for the specified DBID instance and for which the **Condense** option is set to **Part**.

- If a registration belongs to multiple groups, the PowerExchange Logger logs changes for that registration only under the first group in the group definition file that includes the registration.

Example Group Definition File

PowerExchange provides an example group definition file, `pwxcclgrp.cfg`, in the PowerExchange installation directory. Use this example as a starting point when creating your group definition file.

The example file contains the following statements:

```
GROUP=(Company1People,"/user/logger_files/people/company1/condense")
REG=Emp*
REG=Manager
GROUP=(UK_People,"/user/logger_files/people/UK/condense")
SCHEMA=Company2
REG=Manager
REG=Emp*
REG=Em*
SCHEMA=Company3
REG=Manager
REG=Emp*
GROUP=(All_Managers,"/user/logger_files/people/managers/condense")
SCHEMA=Company1
REG=Manager
SCHEMA=Company2
REG=Manager
SCHEMA=Company3
REG=Manager
GROUP=(AllCompany3_Locations,"/user/logger_files/locations/company3/condense")
REG=loc*
GROUP=(Company2Jobs,"/user/logger_files/jobs/company2/condense")
REG=Job*
```

Note: Because this example is for a group definition file on a Linux or UNIX system, the paths include forward slashes. A group definition file on Windows system would be similar but have back slashes.

This example file defines the following groups:

- **Company1People group.** Groups all tables associated with capture registrations that have names beginning with "Emp" or the name "Manager." Changes for these tables are logged to log files that have file names beginning with "condense" and that are located at "/user/logger_files/people/company1/."
- **UK_People group.** Groups tables that have the schema Company2 and that are associated with capture registrations that have names beginning with "Emp" or "Em" or the name "Manager." Changes for these tables are logged to log files that have names beginning with "condense" and that are located at "/user/logger_files/people/UK/."
- **All_Managers group.** Groups tables that have the schema Company1, Company2, or Company3 and that are associated with the capture registration with the name "Manager." Changes for these tables are logged to log files that have names beginning with "condense" and that are located at "/user/logger_files/people/managers/."
- **AllCompany3_Locations group.** Groups all tables that are associated with capture registrations that have names beginning with "loc." Changes for these tables are logged to log files that have names beginning with "condense" and that are located at "/user/logger_files/locations/company3/."
- **Company2Jobs group.** Groups all tables that are associated with capture registrations that have names beginning with "Job." Changes for these tables are logged to log files that have names beginning with "condense" and that are located at "/user/logger_files/jobs/company2/."

Some tables might be included in more than one group. For example, the table `COMPANY2.MANAGERS` is in the `Company1People`, `UK_People`, and `All_Managers` groups. However, changes for this table are logged only under the `Company1People` group because it is the first group in the file that includes this table.

Starting the PowerExchange Logger

You can cold start, warm start, or special start a PowerExchange Logger process.

You must start the PowerExchange Logger under a user ID that has READ and WRITE access to PowerExchange Logger log files. Also, the PowerExchange Listener must be running under a user ID that has READ access to the PowerExchange Logger log files.

Choose the start method that is appropriate for your current situation:

- A *cold start* uses the restart and sequence tokens, if present, in the pwxcl configuration file to determine the point in the change stream from which the PowerExchange Logger starts reading changes. If you are starting the PowerExchange Logger for the first time, you must perform a cold start. When you enter the pwxcl command to start the PowerExchange Logger, set the coldstart parameter to Y.
- A *warm start* uses the restart and sequence tokens in the CDCT file to resume CDC processing. You can perform a warm start only if you have run the PowerExchange Logger previously and have a recent CDCT file for the database instance. When you enter the pwxcl command for starting the PowerExchange Logger, set the coldstart parameter to N or omit the parameter.
- A *special start* uses the restart and sequence tokens in the pwxcl.cfg file to override the token values from the CDCT file for the PowerExchange Logger run. None of the data that was captured prior to the special start is lost. You must enter the SEQUENCE_TOKEN and RESTART_TOKEN parameters in the pwxcl.cfg. The SEQUENCE_TOKEN value must be greater than or equal to the sequence token in the CDCT file.

Use a special start to avoid capturing changes from problematic portions of the logs. For example, perform a special start in the following situations:

- You do not want the PowerExchange Logger to capture an upgrade of an Oracle catalog. In this case, stop the PowerExchange Logger before the upgrade. After the upgrade is complete, generate new sequence and restart tokens for the PowerExchange Logger based on the post-upgrade SCN. To perform a special start, you must specify the SEQUENCE_TOKEN and RESTART_TOKEN parameters in the pwxcl.cfg file. Then special start the PowerExchange Logger.
- You do not want the PowerExchange Logger to reprocess old, unavailable logs that were caused by outstanding UOWs that are not of CDC interest. In this case, stop the PowerExchange Logger. Edit the RESTART_TOKEN value to reflect the SCN of the earliest available log, and then perform a special start. If any of the outstanding UOWs are of CDC interest, data might be lost.

Note: You cannot use the pwxcmd or infacmd program to start the PowerExchange Logger.

RELATED TOPICS:

- [“How the PowerExchange Logger Determines the Start Point for a Cold Start” on page 69](#)
- [“Cold Starting the PowerExchange Logger ” on page 70](#)
- [“PWXCL Command Parameters” on page 68](#)

PWXCL Syntax and Parameters

To start the PowerExchange Logger process, run the pwxcl program, which is located in the PowerExchange installation directory by default.

PWXCCL Command Syntax

The `pwxccl` command that is used to start the PowerExchange Logger has the following syntax:

```
pwxccl
[coldstart={Y|N}]
[specialstart={Y|N}]
[config=path/pwx_config_file]
[cs=path/pwxlogger_config_file]
[license=path/license_file]
[encryptepwd=encrypted_password]
```

Use the following rules and guidelines when you enter the `pwxccl` command:

- To cold start the PowerExchange Logger, set the `coldstart` parameter to Y. The default is N.
- To special start the PowerExchange Logger from a specific point in the change stream, set the `specialstart` parameter to Y. The default is N. You must also specify the `SEQUENCE_TOKEN` and `RESTART_TOKEN` parameters in the `pwxccl.cfg` file.
- All parameters on the `pwxccl` command are optional. However, if you specify the `config` or `license` parameter, the `cs` parameter is required.
- In the `config`, `cs`, and `license` parameters, the full path is required only if the file is not in the default location.
- On Linux and UNIX, append an ampersand (&) at the end of the statement to run the PowerExchange Logger in background mode. For example:

```
pwxccl [coldstart=Y|N] [specialstart={Y|N}] [config=directory/myconfig_file]
[cs=directory/mycondense_config_file]
[license=directory/mylicense_key_file] &
```

For more information about `pwxccl` syntax, see the *PowerExchange Command Reference*.

Caution: If you run PowerExchange and PowerCenter on the same machine and with the same user account, you must create separate environments for PowerExchange and PowerCenter. To create the appropriate PowerExchange environment and start the PowerExchange Logger, run the `pwxcsettask.bat` script on Windows or the `pwxcsettask.sh` script on Linux or UNIX.

On Windows, use the following syntax:

```
pwxcsettask pwxccl
["coldstart={Y|N}"]
["config=path/pwx_config_file"]
["cs=path/pwxlogger_config_file"]
["license=path/license_file"]
["encryptepwd=encrypted_password"]
```

The quotation marks are required on Windows.

On Linux and UNIX, use the following syntax:

```
pwxcsettask.sh pwxccl
[coldstart={Y|N}]
[config=path/pwx_config_file]
[cs=path/pwxlogger_config_file]
[license=path/license_file]
[encryptepwd=encrypted_password]
```

The quotation marks are optional on Linux and UNIX.

For more information, see [“Environment Variable Incompatibilities Between PowerExchange and PowerCenter” on page 24](#).

PWXCL Command Parameters

You can specify several optional parameters in the `pwxccl` command that starts the PowerExchange Logger.

The following table describes each parameter:

Parameter	Description
<code>coldstart</code>	<p>Indicates whether to cold start or warm start the PowerExchange Logger.</p> <p>Enter one of the following values:</p> <ul style="list-style-type: none">- Y. Cold start the PowerExchange Logger. You must specify <code>coldstart=Y</code> to perform a cold start. If the CDCT file contains log entries, the PowerExchange Logger deletes these entries.- N. Warm start the PowerExchange Logger from the restart point that is indicated in the CDCT file. If no restart information exists in the CDCT file, the PowerExchange Logger ends with error message PWX-33239. <p>Default is N.</p>
<code>config</code>	<p>The full path and file name for a dbmover configuration file that overrides the default dbmover configuration file in the installation directory. The override file must have a path or file name that is different from that of the default file.</p> <p>This override file takes precedence over any other override configuration file that you optionally specify with the <code>PWX_CONFIG</code> environment variable.</p>
<code>cs</code>	<p>The full path and file name for the PowerExchange Logger configuration file. Use this parameter to specify a PowerExchange Logger configuration file that overrides the default <code>pwxccl</code> configuration file in the installation directory. The override file must have a path or file name that is different from that of the default file.</p>
<code>encryptepwd</code>	<p>A password in encrypted format for enabling the encryption of PowerExchange Logger log files. With this password, the PowerExchange Logger can generate a unique encryption key for each Logger log file. The password is stored in the CDCT file in encrypted format. For security purposes, the password is not stored in CDCT backup files and is not displayed in the CDCT reports that you can generate with the <code>PWXUCDCT</code> utility. You can generate an encrypted password from the PowerExchange Navigator.</p> <p>If you specify this parameter, you must also specify <code>coldstart=Y</code> in the same <code>pwxccl</code> command.</p> <p>If you specify this command-line parameter and the <code>ENCRYPTEPWD</code> parameter in the PowerExchange Logger configuration file, the parameter in the configuration file takes precedence. If you specify this command-line parameter and the <code>ENCRYPTPWD</code> parameter in the PowerExchange Logger configuration file, an error occurs.</p> <p>You can set the AES algorithm to use for log file encryption in the <code>ENCRYPTOPT</code> parameter of the <code>pwxccl.cfg</code> file. The default is <code>AES128</code>.</p> <p>Tip: For optimal security, Informatica recommends that you specify the encryption password in a <code>pwxccl</code> command for cold starting the PowerExchange Logger rather than in the <code>pwxccl.cfg</code> configuration file. This practice can reduce the risk of malicious access to the encryption password for the following reasons: 1) The encryption password is not stored in the <code>pwxccl.cfg</code> file, and 2) You can remove the password from the command line after a successful cold start. If you specify the encryption password in a <code>pwxccl</code> command for a cold start and then need to restore the CDCT file later, you must enter the same encryption password in the <code>RESTORE_CDCT</code> command of the <code>PWXUCDCT</code> utility.</p> <p>To <i>not</i> encrypt PowerExchange Logger log files, do not enter an encryption password in the <code>pwxccl</code> command for a cold start or in the <code>pwxccl.cfg</code> configuration file.</p>

Parameter	Description
license	<p>The full path and file name for a license key file that overrides the default license.key file in the installation directory. The override file must have a file name or path that is different from that of the default file.</p> <p>This override file takes precedence over any other override license key file that you optionally specify with the PWX_LICENSE environment variable.</p>
specialstart	<p>Indicates whether to perform a special start of the PowerExchange Logger. A special start begins PowerExchange capture processing from a point in the change stream that you specify. This start point overrides the restart point based on information in the CDCT file. A special start does not delete any content from the CDCT file.</p> <p>Use this parameter to progress the restart point beyond problematic parts in the source logs. None of the data that was captured prior to the special start is lost.</p> <p>Enter one of the following values:</p> <ul style="list-style-type: none"> - Y. Perform a special start of the PowerExchange Logger from the point in the change stream that is defined by the SEQUENCE_TOKEN and RESTART_TOKEN parameter values in the pwxcl.cfg configuration file. You must specify valid token values in the pwxcl.cfg file to perform a special start. These token values override the restart point based on information in the CDCT file. To generate valid token values, contact Informatica Global Customer Support. Ensure that the SEQUENCE_TOKEN value in the pwxcl.cfg file is greater than or equal to the current sequence token from the CDCT file. <p>Note: Because assistance from Customer Support is required to generate valid sequence and restart tokens, Informatica recommends that you use the specialstart parameter only at the direction of Customer Support.</p> <ul style="list-style-type: none"> - N. Do not perform a special start. Perform a cold start or warm start as indicated by the coldstart parameter. <p>Default is N.</p> <p>Do not specify both coldstart=Y and specialstart=Y. If you do, the coldstart=Y parameter takes precedence.</p>

Note: In these parameters, the full path is required only if the file is not in the default location.

How the PowerExchange Logger Determines the Start Point for a Cold Start

When you cold start a PowerExchange Logger for Linux, UNIX, and Windows process, it uses the RESTART_TOKEN and SEQUENCE_TOKEN parameters, if present, in the pwxcl configuration file to determine the point in the change stream at which to start reading changes.

Based on how you set these parameters, the PowerExchange Logger starts from one of the following points in the change stream:

- If you enter valid token values in the RESTART_TOKEN and SEQUENCE_TOKEN parameters, the PowerExchange Logger starts from the point in the change stream that the token values specify. Use this method to start the PowerExchange Logger from a specific point.
- If you do not define the RESTART_TOKEN and SEQUENCE_TOKEN parameters, the PowerExchange Loggers starts from the current end-of-log (EOL), or current point in time in the change stream.

Tip: You can generate restart and sequence tokens for the current EOL by running the DTLUAPPL utility with the RSTTKN GENERATE parameter or by performing a database row test with the SELECT CURRENT_RESTART SQL statement in PowerExchange Navigator.

- If you enter only zeroes (a single 0, or an even number of 0s) in the `RESTART_TOKEN` and `SEQUENCE_TOKEN` parameters, the PowerExchange Logger processing starts from one of the following start positions, depending on the data source type:
 - For Db2 sources on Linux, UNIX, or Windows, processing starts from the position at which the DTLUCUDB utility created the Db2 catalog snapshot to initialize the PowerExchange capture catalog table. However, you can change this default restart position with the DTLUCUDB `UPDTDRP` command.
 - For Microsoft SQL Server sources, processing starts from the position of the oldest available data in the distribution database.
 - For PowerExchange Express CDC for Oracle sources, processing starts from the beginning of the most recent Oracle archive log.
 - For MySQL sources, processing starts from the oldest data available in the binary log. This position is the beginning of the binary log that has the log name that includes the lowest numeric suffix value. You can use the `SHOW BINARY LOGS` statement to identify this log.
 - For PostgreSQL sources, processing starts from the oldest data available in the replication store table. If the replication store table is empty, processing starts from the oldest record waiting to be read from the logical replication slot.
 - For remote Db2 for i (i5/OS) sources, processing starts from the beginning of the oldest receiver in the current chain of receivers.
 - For remote z/OS data sources, processing starts from the beginning of the PowerExchange Logger for z/OS active log files.

Cold Starting the PowerExchange Logger

Use this procedure to cold start the PowerExchange Logger. In the start statement, you must include the parameter `COLDSTART=Y`.

During a cold start, the PowerExchange Logger deletes the records in the CDCT file.

1. If you previously ran the PowerExchange Logger and have existing CDCT and log files, retain these files for historical purposes.

You can move or rename the files, as long as another PowerExchange Logger process is not using them. Do not delete them if you want to retain change processing history.

Warning: If you delete, move, or rename the CCT file, the capture registrations are no longer available.

2. In the `pwxcl` configuration file, set the `RESTART_TOKEN` and `SEQUENCE_TOKEN` parameters in a manner that causes the PowerExchange Logger to start from the appropriate point in the change stream.
3. To cold start the PowerExchange Logger, enter the following command at the command line:

```
pwxcl coldstart=y
```

The `coldstart` parameter must be set to `y`.

Include the optional `config`, `cs`, and `license` parameters if you want to override the default `dbmover.cfg`, `pwxcl.cfg`, and `license.key` files. On Linux and UNIX systems, you can add an ampersand (&) at the end of the statement to run the PowerExchange Logger in background mode. For more information about PowerExchange Logger start syntax, see the *PowerExchange Command Reference*.

Managing the PowerExchange Logger

To assess the status of the PowerExchange Logger for Linux, UNIX, and Windows, you can display messages about PowerExchange Logger processing, memory use, and CPU use.

Occasionally, you might need to stop the PowerExchange Logger.

RELATED TOPICS:

- [“Monitoring the PowerExchange Logger” on page 72](#)
- [“PowerExchange Logger Verbose Messages” on page 75](#)
- [“Controlling and Stopping PowerExchange Logger Processing” on page 71](#)
- [“PWXUCDCT Commands for Maintaining the PowerExchange Logger CDCT and Log Files” on page 76](#)
- [“Backing Up PowerExchange Logger Files” on page 78](#)
- [“Re-creating the CDCT File After a Failure” on page 79](#)

Controlling and Stopping PowerExchange Logger Processing

PowerExchange provides commands for stopping the PowerExchange Logger, manually initiating a file switch, or starting another logging cycle.

You can enter these commands from the command line or by using the `pwxcmd` program on a remote Linux, UNIX, or Windows system. The output is displayed on screen and written to the PowerExchange message log.

Note: To use the `pwxcmd` program, you must specify the `CONDENSENAME` parameter in the `pwxccl.cfg` file and the `SVCNODE` statement in the `dbmover.cfg` file.

The following table describes each of these commands:

Command-line Command	pwxcmd Command	Description
CONDENSE	condense	When the PowerExchange Logger is running in continuous mode, manually starts a new PowerExchange Logger logging cycle before the wait period for starting another cycle has elapsed. The wait period is defined by the <code>NO_DATA_WAIT</code> parameter in <code>pwxccl.cfg</code> .
FILESWITCH	fileswitch	Closes open PowerExchange Logger log files if they contain data and then switches to a new set of log files. If the log files do not contain data, the file switch does not occur. If you use batch extraction mode, you can use this command to make change data in the current log files available for extraction processing before the next file switch is due to occur. To issue the <code>fileswitch</code> command from a script or batch file, you must use the <code>pwxcmd</code> program. Usually, you do not need to perform manual file switches if you use continuous extraction mode.

Command-line Command	pwxcmd Command	Description
SHUTCOND	shutcond	Stops the PowerExchange Logger in a controlled manner after initiating and completing a final logging cycle. The final logging cycle enables the PowerExchange Logger to capture all of the changes up to point when the command is issued. After the logging cycle completes, the PowerExchange Logger closes open log files, updates the CDCT file, closes the CAPI, stops the Writer and Command Handler subtasks, and then ends the pwxcl program. Use this command if a logging cycle has not run recently.
SHUTDOWN	shutdown	Stops the PowerExchange Logger in a controlled manner after closing any open PowerExchange Logger log files and writing the latest restart position to the CDCT file. During shutdown processing, the PowerExchange Logger closes open log files, updates the CDCT file, closes the CAPI, stops the Writer and Command Handler subtasks, and then ends the pwxcl program. Use this command to stop a PowerExchange Logger process that is running in continuous mode.

For more information about command syntax, example output, and pwxcmd use, see the *PowerExchange Command Reference*.

Monitoring the PowerExchange Logger

PowerExchange provides several commands that you can use for monitoring PowerExchange Logger processing and performance.

You can enter these commands from the command line or by using the pwxcmd program on a remote Linux, UNIX, or Windows system. The output is displayed on screen and written to the PowerExchange message log.

Note: To use the pwxcmd program, you must specify the CONDENSENAME parameter in the pwxcl.cfg file and the SVCNODE statement in the dbmover.cfg file.

The following table summarizes these commands:

Command-line Command	pwxcmd Command	Description
DG	pwxcmd displaystats -tp groups	<p>Displays monitoring statistics for each PowerExchange Logger group that is defined, if any. The statistics include:</p> <ul style="list-style-type: none"> - The group name and the number of capture registrations in the group - The total number of insert, update, and delete records that the PowerExchange Logger processed for the group - The number of commits that the PowerExchange Logger processed for the group - The number of change records that the PowerExchange Logger has not yet flushed from memory to its log files on disk - The file name of the open Logger log file and the timestamp for when the file was opened <p>To use this command, you must specify the STATS=(MONITOR) parameter in the pwxccl.cfg configuration file.</p>
DISPLAY CPU	pwxcmd displaycpu	<p>Displays the CPU time spent, in microseconds, for PowerExchange Logger processing during the current logging cycle, by processing phase. Processing phases include:</p> <ul style="list-style-type: none"> - Reading source data - Writing data to PowerExchange Logger log files - Performing file switches - Performing "other processing," such as initialization and Command Handler processing of commands <p>Also includes the total CPU time for all PowerExchange Logger processing.</p>
DISPLAY EVENTS	pwxcmd displayevents	Displays events that the PowerExchange Logger Controller, Command Handler, and Writer tasks are waiting on. Also indicates if the Writer is processing data or is in a sleep state while waiting for an event or timeout to occur.
DISPLAY MEMORY	pwxcmd displaymemory	Displays PowerExchange Logger memory use, in bytes, for each PowerExchange Logger task and subtask, with totals for the entire PowerExchange Logger process.
DISPLAY RECORDS	pwxcmd displayrecords	Displays counts of inserts, updates, deletes, and commits that the PowerExchange Logger processed during the current processing cycle. If the PowerExchange Logger did not receive changes during the current cycle, displays counts of change records for the current set of PowerExchange Logger log files.

Command-line Command	pwxcmd Command	Description
DISPLAY STATUS	pwxcmd displaystatus	Displays the status of the PowerExchange Logger Writer subtask.
DL	pwxcmd displaystats -tp logger	<p>Displays monitoring statistics for a PowerExchange Logger for Linux, UNIX, and Windows process and its tasks. The statistics include:</p> <ul style="list-style-type: none"> - The PowerExchange Logger process ID - The status of the PowerExchange Logger Writer subtask at the time the command is issued - The CPU time used by the PowerExchange Logger since it started - PowerExchange Logger memory use by the Controller, Command Handler, and Writer tasks. For tasks, memory use is reported in the following categories: Current, Total, and Maximum. - Counts of inserts, updates, deletes, and commits that the PowerExchange Logger has processed, total and for the open Logger log file and the active logging cycle <p>To use this command, you must specify the STATS=(MONITOR) parameter in the pwxccl.cfg configuration file.</p>

For more information about these commands and sample output, see the *PowerExchange Command Reference*.

If you specify the optional *interval* subparameter in the STATS=(MONITOR) parameter in the pwxccl.cfg configuration file, you can publish the same monitoring statistics that are reported by the DL command at a specific interval:

```
STATS=(MONITOR,interval)
```

Fewer interval-based statistics messages are displayed on screen than are written to the message log to prevent flooding the screen with messages over time. For example, the following messages are displayed on the screen:

```
PWX-37132 Controller: (981/983/1849) KB Command Handler: (0/0/34) KB Writer: (5127/5147/5181) KB
PWX-37135 Status 7144 Totals I=0000000024344 U=0000000000000 D=0000000024336 C=000000004004
Total=0000000052684
PWX-37136 CurrFileOpened : 2015-08-11 13:20:39 I=0000000024344 U=0000000000000 D=0000000024336 C=000000004004
Total=0000000052684
PWX-37137 Active Cycle : 2015-08-11 13:21:01 I=0000000024344 U=0000000000000 D=0000000024336 C=000000004004
Total=0000000052684
```

The following interval-based statistics are written to the PowerExchange message log:

```
PWX-37130 PWXCCL pid = 7144 Writer status = Reading or waiting for source data
PWX-37134 CPU Time = 0:00:02.589616
PWX-37131 Memory (Current/Total/Maximum)
PWX-37132 Controller: (981/983/1849) KB Command Handler: (0/0/34) KB Writer: (5127/5147/5181) KB
PWX-37135 Status 7144 Totals I=0000000024344 U=0000000000000 D=0000000024336 C=000000004004
Total=0000000052684
PWX-37136 CurrFileOpened : 2015-08-11 13:20:39 I=0000000024344 U=0000000000000 D=0000000024336 C=000000004004
Total=0000000052684
PWX-37137 Active Cycle : 2015-08-11 13:21:01 I=0000000024344 U=0000000000000 D=0000000024336 C=000000004004
Total=0000000052684
```

Also, the PowerExchange Logger publishes summary monitoring statistics when it shuts down, provided that you specified STATS=(MONITOR) in the pwxccl.cfg file, either with or without the *interval* subparameter. A Logger that runs in continuous mode shuts down when you issue a SHUTCOND or SHUTDOWN command. A Logger that runs in batch mode shuts down when it reaches the end of its batch run. These summary monitoring statistics are included the shutdown output. For example output, see the Logger SHUTCOND and SHUTDOWN commands in the *PowerExchange Command Reference*.

Determining If the PowerExchange Logger Captured Changes

To determine if the PowerExchange Logger for Linux, UNIX, and Windows 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 the PowerExchange Logger read all of the changes that were available at the start the logging cycle. Look for this message if a PowerExchange Logger log file does not receive change data within the time period you expect. Delays can occur for various reasons. For example, if you cold start the PowerExchange Logger from the current restart point but a recent copy of the Oracle online catalog is not available in the archived redo logs, PowerExchange might need to read many archived logs before determining the point from which to begin capturing changes.

This message also indicates the point at which the PowerExchange Logger CATINT, CATBEGIN, and CATEND parameters take effect. These parameters control how often Oracle copies the catalog and the time period within which the copy operation occurs.

PowerExchange Logger Verbose Messages

If you enter VERBOSE=Y in the pwxcl.cfg configuration file, the PowerExchange Logger for Linux, UNIX, and Windows produces more detailed messages during initialization, condense, fileswitch, record expiration, and shutdown processing. You can use these messages to assess PowerExchange performance and processing status.

For example, the following verbose messages indicate CPU use by the Writer subtask:

- Message PWX-33274 is issued before the Writer subtask starts reading source data after initialization and before the PowerExchange Logger shuts down:

```
PWX-33274 CPU Total number. CAPI Read number. Writing number. File switching number.  
Other number
```

- Message PWX-33279 issued after each file switch:

```
PWX-33279 CPU total number. This file total number. CAPI Reads number. Writing file  
number. Other number
```

If you do not use verbose messaging, you can use the DL, DG, DISPLAY CPU, and DISPLAY RECORDS commands to gather statistics that are useful for assessing PowerExchange Logger performance and status.

PWXUCDCT Commands for Maintaining the PowerExchange Logger CDCT and Log Files

You can use the PWXUCDCT utility to maintain the PowerExchange Logger CDCT file and log files.

The following table describes the PWXUCDCT commands that you can use to perform maintenance tasks:

Command	Description
CONVERT_CDCT	<p>If you upgrade to 9.5.1 HotFix 1 or later from an earlier release, you can issue this command to manually perform a one-time conversion of the CDCT file to the new format. Alternatively, the first time the PowerExchange Logger is warm started, it automatically converts the CDCT file to the new format.</p> <p>The conversion creates a CDCT_<i>dbid</i> file instance from the original CDCT file. Ensure that the <i>dbid</i> value in the CDCT file name matches the DBID parameter value in the PowerExchange Logger pwxocl configuration file under which you run the command.</p> <p>Note: If the old CDCT file contains information for multiple database instances, you must run this command multiple times, once for each instance. Each time you run the command, ensure that the CS parameter points to the correct pwxocl configuration file for the instance.</p>
CREATE_CDCT_BACKUP	<p>Manually creates a backup of all records in a CDCT file instance for a source database based on the latest configuration incarnation.</p> <p>Note: The PowerExchange Logger automatically generates a backup at initialization and at termination.</p>
DELETE_EXPIRED_CDCT	<p>This command is deprecated but is still supported for backward compatibility. Use DELETE_EXPIRED_FILES instead.</p>
DELETE_EXPIRED_FILES	<p>Delete the log files for which the retention period has expired and the CDCT records that reference those expired logs. For this command to work, you must set the LOGGER_DELETES_EXPIRED_CDCT_RECORDS parameter to N in the pwxocl configuration file.</p>
DELETE_ORPHAN_FILES	<p>Delete PowerExchange Logger log files that are not referenced by any record in the CDCT file.</p>
DERIVE_CDCT_BACKUP	<p>If the CDCT file is corrupted or deleted and if a CDCT backup is not available or the latest available backup would result in significant reprocessing of data, use this command to derive a backup text file for recovery purposes.</p> <p>The command uses the EXTERNAL_CAPTURE_MASK parameter value from the PowerExchange Logger configuration file or the <i>external_capture_mask</i> positional parameter from the group definition file to generate a list of PowerExchange Logger log files. The command then uses the content of these log files to generate a text file that can be used as input to the RESTORE CDCT command.</p> <p>Do not use this command if the PowerExchange Logger log files were also corrupted or deleted.</p> <p>Tip: Use the PREVBKUPFILE parameter to supply the name of the last available backup file. By using a previous backup file, you preserve more historic information in the CDCT file. Also, the utility will add any log files that were created since the backup was taken to the derived backup file.</p>

Command	Description
REPORT_CDCT	<p>Print the contents of the CDCT file. This information is primarily for debugging purposes.</p> <p>For the current Logger configuration incarnation, the report shows:</p> <ul style="list-style-type: none"> - Incarnation identifier, status, and reason (Rsn) for creation. The reason can be a cold start or a change in the configuration. - Source instance (or DBID) name and image type. - Number of groups defined in the group definition file. If no groups are defined, the default of 1 is used. - Begin and end timestamps. - Begin and end restart and sequence tokens. <p>For each Logger group, the report shows:</p> <ul style="list-style-type: none"> - Group number and name. - Incarnation to which the group belongs. - Path to the group log files. - Registration count. - Log file count and the first and current log sequence numbers. - Oldest log file timestamp. <p>For each registration, the report shows:</p> <ul style="list-style-type: none"> - Registration tag name and status. - Incarnation and group to which the registration belongs. - Default schema name.
REPORT_CDCT_FILES	<p>Report the following information for each log file that is recorded in the CDCT:</p> <ul style="list-style-type: none"> - Log file name and sequence number. - Configuration incarnation and group to which the log file belongs. - Record count, commit count, and whether any uncommitted data exists. - Begin and end timestamps. - Begin and end restart and sequence tokens. - File open timestamp. - File close timestamp.
REPORT_CONFIG	<p>List the parameter settings that are defined in the associated PowerExchange Logger pwxcl configuration file.</p> <p>If you created a group definition file and specified it in the GROUPDEFS parameter in the pwxcl file, the command also reports the group statements in the group definition file.</p>
REPORT_EXPIRED_CDCT	<p>This command is deprecated but is still supported for backward compatibility. Use REPORT_EXPIRED_FILES instead.</p>
REPORT_EXPIRED_FILES	<p>List the PowerExchange Logger log files for which the retention period has elapsed.</p>
REPORT_FILES_BY_NAME	<p>List PowerExchange Logger log files by file name. This information is based on directory information for the log files and not on the CDCT file.</p> <p>For each file, the command reports the following information:</p> <ul style="list-style-type: none"> - Date and time when the file was written. - Sequence number - Path and file name. <p>Also, the command reports the number of log files that match the default mask that is specified in the EXT_CAPT_MASK parameter of the pwxcl configuration file. If you specified a group definition file in the GROUPDEFS parameter of the pwxcl file, the command reports the number of log files that match any masks in the group definition file.</p>

Command	Description
REPORT_FILES_BY_TIME	<p>List PowerExchange Logger log files in the order in which they were created, from earliest to latest. This information is based on directory information for the log files and not on the CDCT file.</p> <p>For each file, the command reports the following information:</p> <ul style="list-style-type: none"> - Date and time when the file was written. - Sequence number. - Path and file name. <p>Also, the command reports the number of log files that match the default mask that is specified in the EXT_CAPT_MASK parameter of the pwxcl configuration file. If you specified a group definition file in the GROUPDEFS parameter of the pwxcl file, the command reports the number of log files that match any masks in the group definition file.</p>
REPORT_ORPHAN_FILES	List PowerExchange Logger log files that are not referenced by any record in the CDCT file.
RESTORE_CDCT	<p>Restore the CDCT file from a backup, up to a specific point in time. The PowerExchange Logger will reprocess any data that is later than this point in time.</p> <p>After the restore operation completes, run the DELETE_ORPHAN_FILES command.</p>

In the command syntax, include the CS parameter to specify the path to a specific PowerExchange Logger pwxcl configuration file. For more information about the PWXUCDCT utility commands, see the *PowerExchange Utilities Guide*.

Backing Up PowerExchange Logger Files

The PowerExchange Logger automatically creates a backup of the CDCT file at initialization and normal termination. You must manually back up the log files.

If a recent generated CDCT backup is not available, you can use the PWXUCDCT utility to manually create a CDCT backup. Use the PWXUCDCT utility CREATE_CDCT_BACKUP command to back up the CDCT file based on the latest Logger configuration incarnation.

Alternatively, if a failure causes the CDCT file and its recent backups to become damaged or deleted, you can use the DERIVE_CDCT_BACKUP command to derive a backup based on the available PowerExchange Logger log files, optionally in conjunction with the last available backup file. This previous backup can be an automatically generated backup or one that you created. By using the previous backup file, you preserve more historic information in the CDCT file. The utility will add any log files that were created since that backup was taken to the derived backup file.

Tip: If you manually back up the CDCT file and the log files, try to perform the backup during a period of low database activity, when no or little data is being written to the log files.

Re-creating the CDCT File After a Failure

If the CDCT file and its recent backups are damaged or deleted, you can re-create the CDCT file based on the available PowerExchange Logger log files. After you derive the CDCT backup, you can use it to restore the CDCT file.

This procedure assumes that PowerExchange Logger log files are available. Do not use this procedure if the log files were also damaged or deleted.

1. Issue the PWXUCDCT utility DERIVE_CDCT_BACKUP command to derive a backup from available PowerExchange Logger log files.

Tip: Include the PREVBACKUPFILE parameter to supply the name of the last available CDCT backup file. By using this previous backup file, you preserve more historic information in the CDCT file. The utility will add any log files that were created after this backup was taken to the derived backup file.

2. To restore the CDCT file from the derived backup, issue the PWXUCDCT utility RESTORE_CDCT command.
3. Verify that the restore operation was successful as follows:
 - Verify that the return code from the PWXUCDCT utility is zero.
 - Verify that messages PWX-25140 through PWX-25145 provide reasonable record counts for the records read from the backup file and for the records that were changed in the CDCT file.
4. Run the DELETE_ORPHAN_FILES command to delete log files that are no longer referenced by the restored CDCT file.

After you warm start the PowerExchange Logger, it re-creates the CDCT content for those files.

For more information about the PWXUCDCT utility commands, see the *PowerExchange Utilities Guide*.

Part III: PowerExchange CDC Data Sources

This part contains the following chapters:

- [Db2 CDC on Linux, UNIX, or Windows, 81](#)
- [Microsoft SQL Server CDC, 99](#)
- [MySQL CDC, 116](#)
- [Express CDC for Oracle, 132](#)
- [PostgreSQL CDC, 206](#)
- [SAP HANA CDC, 218](#)
- [Remote Logging of Data, 228](#)

CHAPTER 4

Db2 CDC on Linux, UNIX, or Windows

This chapter includes the following topics:

- [Db2 CDC on Linux, UNIX, or Windows Overview, 81](#)
- [Planning for Db2 CDC, 82](#)
- [Configuring Db2 for CDC, 85](#)
- [Configuring PowerExchange for Db2 CDC, 85](#)
- [Using a Db2 Data Map, 94](#)
- [Managing Db2 CDC, 94](#)
- [Db2 CDC Troubleshooting, 97](#)

Db2 CDC on Linux, UNIX, or Windows Overview

PowerExchange captures change data from Db2 database logs for source tables on Linux, UNIX, or Windows. PowerExchange uses the PowerExchange Client for PowerCenter (PWXPCL) to coordinate with PowerCenter to move the captured change data to one or more targets.

For PowerExchange to capture Db2 change data, you must perform the following configuration tasks in Db2:

- Ensure that archive logging is active for the database.
- Create a PowerExchange capture catalog table in the database. The capture catalog table stores information about all tables in the source database, including column definitions and Db2 log positions.

Also, perform the following configuration tasks in PowerExchange:

- 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. Optionally, you can define an additional extraction map.
- If a source table contains columns in which you store data in a format that is inconsistent with the column datatype, you can optionally 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 and prepare that data for loading to a target. You must merge the data map with the extraction map for the source table during capture registration creation.

- If you want to use the PowerExchange Logger for Linux, UNIX, and Windows to capture change data and write it to PowerExchange Logger log files, configure the PowerExchange Logger. The change data is then extracted from the PowerExchange Logger log files. Benefits of the PowerExchange Logger include fewer database accesses, faster CDC restart, and no need to prolong retention of Db2 log files for change capture. PowerExchange works in conjunction with PowerCenter to extract change data from Db2 database logs or PowerExchange Logger log files and load that data to one or more targets.

RELATED TOPICS:

- [“Planning for Db2 CDC” on page 82](#)
- [“Configuring PowerExchange for Db2 CDC” on page 85](#)
- [“Using a Db2 Data Map” on page 94](#)
- [“Managing Db2 CDC” on page 94](#)
- [“Introduction to Change Data Extraction” on page 248](#)
- [“PowerExchange Logger Overview” on page 36](#)

Planning for Db2 CDC

Before you configure Db2 CDC for sources on Linux, UNIX, or Windows, verify that the following prerequisites and user authority requirements are met. Also, review the restrictions so that you can properly configure CDC.

Prerequisites

PowerExchange CDC has the following prerequisites:

- Archive logging must be active for the database that contains the source tables from which change data is to be captured.
- Db2 source tables must be defined with the `DATA CAPTURE CHANGES` clause for capture processing to occur.
- A valid Db2 environment must exist for the PowerExchange user. On Linux and UNIX, the path to the Db2 client must be specified in the `PATH` and library path environment variables.

Required User Authority

For PowerExchange to read change data from Db2 logs, the user ID that you specify for database access must have SYSADM or DBADM authority. Usually, you specify this user ID in the `UDB CAPL_CONNECTION` statement in the `dbmover.cfg` file.

If the user has SYSADM authority, ensure that the user also has `SELECT`, `INSERT`, `UPDATE`, and `DELETE` permissions to the PowerExchange capture catalog in the source database. You can use the `GRANT DATAACCESS` statement to grant these permissions on the database to the user.

Db2 Datatypes Supported for CDC

PowerExchange supports most Db2 datatypes for CDC.

The following table identifies the Db2 source datatypes that PowerExchange supports and does not support for CDC:

DB2 Datatype	Supported for CDC?	Comments
BIGINT	Yes	-
BLOB	No	If you register a table with large object (LOB) columns, PowerExchange does not capture changes for the LOB columns but can capture changes for other columns in the table.
CHAR	Yes	-
CLOB	No	If you register a table with LOB columns, PowerExchange does not capture changes for the LOB columns but can capture changes for other columns in the table.
DATE	Yes	-
DBCLOB	No	If you register a table with LOB columns, PowerExchange does not capture changes for the LOB columns but can capture changes for other columns in the table.
DECFLOAT	No	If you register a table with DECFLOAT columns, PowerExchange does not capture changes for the DECFLOAT columns but can capture changes for other columns in the table.
DECIMAL	Yes	-
DOUBLE	Yes	-
GRAPHIC	Yes	-
INTEGER	Yes	-
LONG VARCHAR	Yes	-
LONG VARGRAPHIC	Yes	-
REAL	Yes	-
REF	No	DB2 does not allow change data capture for tables with REF columns.
SMALLINT	Yes	-
TIME	Yes	-
TIMESTAMP	Yes	-
UDTs ¹	No	PowerExchange does not capture change data for tables with UDT columns.
VARCHAR	Yes	-

DB2 Datatype	Supported for CDC?	Comments
VARGRAPHIC	Yes	-
XML	No	If you register a table with XML columns, PowerExchange does not capture changes for the XML columns but does capture changes for other columns in the table.
1. User-defined datatypes, such as DISTINCT and STRUCT.		

Db2 CDC Considerations

Consider the following CDC capabilities and restrictions when planning Db2 CDC processing.

- To extract change data on a Db2 client machine that is remote from the Db2 server where the change data is captured, both machines must have the same architecture. Otherwise, change data capture processing might fail with the error message PWX-20628.
- For Db2 9.7 and later sources, PowerExchange can capture change data from tables that use Db2 row compression. These tables were created or altered with the COMPRESS YES option.
- If the source tables are compressed, ensure that you have a compression dictionary that is compatible with the compressed Db2 log records from which PowerExchange reads change data for the tables. Otherwise, Db2 cannot decompress the log records for PowerExchange read requests. Usually, the compatible compression dictionary is available because Db2 maintains the current compression dictionary and a backup of the previous compression dictionary on disk.

If you run the Db2 REORG TABLE utility or the Db2 LOAD utility with the REPLACE or RESUME NO option against compressed source tables, Informatica recommends that you specify the KEEPDICTIONARY option for the utility. The KEEPDICTIONARY option forces Db2 to retain the current compression dictionary if it exists. If you use the RESETDICTIONARY option, Db2 rebuilds compression dictionary. In this case, the previous compression dictionary that matches the Db2 log records might not be available any longer.

- PowerExchange cannot capture change data for the following Db2 datatypes:
 - DECFLOAT, LOB, and XML datatypes. You can create a capture registration for a table that includes columns with DECFLOAT, LOB, and XML datatypes. However, the registration does not include these columns, and PowerExchange does not capture change data for them. PowerExchange does capture change data for the other columns in the registered table that have supported datatypes.
 - User-defined datatypes. Tables that include columns with user-defined datatypes cannot be registered for change data capture. PowerExchange cannot capture change data for these tables.
- To add or drop partitions in a partitioned database and then redistribute table data across the updated partition group, or to reconfigure a database partition group, you must use a special procedure. Otherwise, PowerExchange might not be able to resume change data capture properly.
- If you alter a column datatype to or from FOR BIT DATA, PowerExchange does not detect the datatype change. PowerExchange continues to use the datatype that is specified in the existing capture registration.
- If you alter a source table to change the DEFAULT value of a Db2 column of CDC interest, PowerExchange does not detect this DDL change during capture processing. As a result, the correct DEFAULT value is not available when PowerExchange performs the following operations:
 - Delivers the pre-existing short rows for a table to which columns were added.

- Delivers rows for source tables that use the VALUE COMPRESSION option and that include a column with the COMPRESS SYSTEM DEFAULT option and a default value.
- In a partitioned database, if an UPDATE to a table row changes the partition key and that change causes the row to move to another partition, PowerExchange processes the UPDATE as two operations: a DELETE and an INSERT. However, based on the Db2 log information, PowerExchange cannot predictably determine the order in which to perform the DELETE and INSERT operations. If the INSERT is processed first, both the original row and the updated row appear on the target until the DELETE is processed.
- The maximum length of a row from which PowerExchange can capture change data is 128,000 bytes.
- PowerExchange uses multithreaded processing for change data capture. By default, PowerExchange uses up to nine threads. To configure the number of threads, specify the THREADING parameter in the UDB CAPI CONNECTION statement. If you have a Db2 partitioned database, you can use a maximum of one thread for each database partition node plus two additional threads for the CAPI and merge processing.
- PowerExchange does not support Db2 10.5 sources on zLinux.

Configuring Db2 for CDC

To configure Db2 for CDC on Linux, UNIX, or Windows, perform the following tasks:

1. In the Db2 Control Center Configure Database Logging wizard, enable archive logging for the Db2 database. For more information, see the IBM Db2 documentation.
If archive logging is not enabled, PowerExchange issues the error messages PWX-20204 and PWX-20229 during CDC.
2. Set the following user environment variables in any process that runs PowerExchange CDC or the DTLUCUDB program:
 - Set DB2NOEXITLIST to ON.
 - Set DB2CODEPAGE to 1208.
3. Verify that the DB2 source tables are defined with the DATA CAPTURE CHANGES clause.
4. To enable PowerExchange to report the authorization ID and application that is associated with a DB2 transaction in monitoring message PWX-20177, set the DB2_LOGGING_DETAIL registry variable to APPLINFO in Db2.
To set this variable for the current DB2 instance, enter the following command:


```
db2set DB2_LOGGING_DETAIL=APPLINFO
```

 To set this variable for all Db2 instances on the system, enter the following command:


```
db2set -g DB2_LOGGING_DETAIL=APPLINFO
```
5. If a table that is selected for change data capture includes columns with a LONG datatype, use the INCLUDE LONGVAR COLUMNS clause to alter the table so that PowerExchange can capture data for the LONG columns. Otherwise, PowerExchange might issue the error message PWX-20094 during CDC processing.

Configuring PowerExchange for Db2 CDC

The tasks that you perform to configure PowerExchange for Db2 CDC depend on whether you want to use the PowerExchange Logger for Linux, UNIX, and Windows and the extraction mode you plan to use.

RELATED TOPICS:

- [“Configuring PowerExchange CDC without the PowerExchange Logger” on page 86](#)
- [“Configuring PowerExchange CDC with the PowerExchange Logger” on page 86](#)
- [“Creating the Capture Catalog Table” on page 87](#)
- [“Initializing the Capture Catalog Table” on page 87](#)
- [“Customizing the dbmover Configuration File for Db2 CDC” on page 88](#)

Configuring PowerExchange CDC without the PowerExchange Logger

If you plan to run extractions in real-time extraction mode and *not* use the PowerExchange Logger for Linux, UNIX, and Windows, complete the following tasks to configure PowerExchange CDC:

1. Create the PowerExchange capture catalog table.
2. Run the DTLUCUDB SNAPSHOT command to initialize the capture catalog table.
3. When you configure the dbmover.cfg file, include the following statements:
 - CAPT_PATH
 - CAPT_XTRA
 - UDB CAPI_CONNECTION
4. In the PowerExchange Navigator, create a capture registration for each source table. The PowerExchange Navigator generates a corresponding extraction map. Optionally, create a data map if you want to perform field-level processing.

Tip: Set the **Condense** option to **Part** even though you do not plan to use the PowerExchange Logger, unless you have a specific reason not to do so. This practice prevents having to edit the capture registrations later if you decide to use the PowerExchange Logger. You might want to set the **Condense** option to **None** if you plan to run both real-time and continuous extractions against tables defined by the same capture registrations and you do not want the PowerExchange Logger to capture change data for some registered tables.

If capture registrations already exist for the source tables, delete the existing registrations and extraction maps and create new ones.

5. Activate the capture registrations. Usually, you do this task after materializing the targets.

Next Step: Configure and start extractions. You must use real-time extraction mode.

Configuring PowerExchange CDC with the PowerExchange Logger

If you plan to use the PowerExchange Logger for Linux, UNIX, and Windows and run extractions in batch or continuous extraction mode, complete the following tasks to configure PowerExchange CDC:

1. Create the PowerExchange capture catalog table.
2. Run the DTLUCUDB SNAPSHOT command to initialize the capture catalog table.
3. When you configure the dbmover.cfg file, include the following statements:
 - CAPT_PATH
 - CAPT_XTRA
 - UDB CAPI_CONNECTION
 - CAPX CAPI_CONNECTION (for continuous extraction mode only)
4. Configure the pwxcl.cfg file for the PowerExchange Logger.

5. In the PowerExchange Navigator, create a capture registration for each DB2 source table. You must select **Part** in the **Condense** drop-down list. The PowerExchange Navigator generates a corresponding extraction map.

If capture registrations already exist for these tables, delete the existing registrations and extraction maps and create new ones.

6. Activate the capture registrations. Usually, you do this task after materializing the targets.
7. Start the PowerExchange Logger.

Next Step: Configure and start extractions. You can use either batch extraction mode or continuous extraction mode.

Creating the Capture Catalog Table

The PowerExchange capture catalog table stores information about the CDC source tables, column definitions, and valid Db2 log positions. You must create this table in the same database that contains the source tables from which change data is captured.

If the database has multiple partitions, the capture catalog table stores positioning information for each partition. If the database has only a single partition, the capture catalog table still contains positioning information for the partition.

Use the following DDL to create the capture catalog table:

```
CREATE TABLE DTLCCATALOG (
    VTSTIME    TIMESTAMP          NOT NULL,
    VTSACC     INTEGER            NOT NULL,
    NODENUM    SMALLINT           NOT NULL,
    SEQ        INTEGER            NOT NULL,
    TBSHEMA    VARCHAR(128),
    TBNAME     VARCHAR(128),
    OP         VARCHAR(1024)      NOT NULL,
    PRIMARY KEY (VTSTIME, VTSACC, NODENUM, SEQ) )
;
```

In this DDL, the table name is DTLCCATALOG. If necessary, you can specify another table name.

Tip: Informatica recommends that you place the PowerExchange capture catalog table in the Db2 catalog partition.

Initializing the Capture Catalog Table

To initialize the PowerExchange capture catalog table, run the DTLUCUDB utility with the SNAPSHOT command. You should need to do this task only once.

To specify the command, use the following syntax:

```
DTLUCUDB SNAPSHOT [DB=database_name] [CCATALOG=capture_catalog_name] [UID=user_id]
[EPWD=encrypted_password] [REPLACE=Y|N]
```

If the capture catalog table contains existing rows of data, you must set the REPLACE parameter to Y to enable PowerExchange to overwrite the data. For a new capture catalog table, accept the default of N.

After the snapshot successfully completes, back up the capture catalog table to create a point of consistency for recovery.

Note: If you run the DTLUCUDB SNAPSHOT command while the Db2 catalog is being updated, the snapshot fails. If this failure occurs, run the SNAPSHOT command again after the Db2 catalog updates are complete.

Customizing the dbmover Configuration File for Db2 CDC

In the dbmover configuration file, add statements for Db2 CDC on Linux, UNIX, or Windows.

You must include the following statements for Db2 CDC:

CAPT_PATH statement

Path to the local directory on a Linux, UNIX, or Windows system that contains the control files for CDC.

These files are: the CCT file for capture registrations, the CDEP file for application names that are used for ODBC extractions, and the CDCT file for the PowerExchange Logger for Linux, UNIX, and Windows.

CAPT_XTRA statement

Path to the local directory on a Linux, UNIX, or Windows system that stores extraction maps for CDC.

UDB CAPI_CONNECTION statement

A named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and to control CDC processing for Db2 sources.

Add this statement to the dbmover.cfg file on the system where Db2 capture registrations reside. This location corresponds to the **Location** node that you specify when defining a registration group. Usually, this location is where the source database resides.

If you plan to use continuous extraction mode, you must also include the following statement:

CAPX CAPI_CONNECTION statement

A named set of parameters that the CAPI uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files.

Also, Informatica recommends including the LOGPATH and TRACING statements to make finding messages easier. The LOGPATH statement defines a directory specifically for PowerExchange message log files, and the TRACING statement enables PowerExchange to create an alternative set of message log files for each PowerExchange process.

For more information about all DBMOVER statements, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“CAPI_CONNECTION - CAPX Statement” on page 60](#)
- [“CAPI_CONNECTION - UDB Statement” on page 89](#)
- [“CAPT_PATH Statement” on page 31](#)
- [“CAPT_XTRA Statement” on page 31](#)

Example dbmover Statements for DB2

This example shows dbmover statements that are typically defined for DB2 for Linux, UNIX, and Windows CDC.

```
CAPT_PATH=c:/pwxcapt/Vnnn
CAPT_XTRA=c:/pwxcapt/Vnnn/extrmaps
CAPI_CONN_NAME=UDBCC
CAPI_CONNECTION=(NAME=UDBCC
                  ,DLLTRACE=bbbb
                  ,TYPE=(UDB
                        ,CCATALOG=mylib.captcat_tbl
                        ,USERID=db2admin
                        ,PASSWORD=db2admin))
```


CAPI_CONNECTION - UDB Statement

The UDB CAPI_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and to control CDC processing for Db2 for Linux, UNIX, and Windows sources.

Add this statement to the dbmover.cfg file on the system where Db2 capture registrations reside. This location corresponds to the **Location** node that you specify when defining a registration group. Usually, this location is where the source database resides.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Db2 for Linux, UNIX, and Windows

Required: Yes for CDC

Syntax:

```
CAPI_CONNECTION=( [DLLTRACE=trace_id]
                  ,NAME=capi_connection_name
                  [,TRACE=trace_name]
                  ,TYPE=(UDB
                        [,AGEOUTPERIOD=minutes]
                        [,CCATALOG={capture_catalog|creator.DTLCCATALOG}]
                        [,DBCONN=database_name]
                        [,EPWD=encrypted_password]
                        [,LARGEOPS=number_of_operations]
                        [,LIMITRESCAN={Y|N}]
                        [,LOGBUFSIZE={kilobytes}[kilobytes]]
                        [,MEMCACHE={cache_size|1024}]
                        [,MONITORINT={minutes|5}]
                        [,PASSWORD=password]
                        [,RSTRADV=seconds]
                        [,SPACEPRI={AUTO|MAX|NONE|nn}]
                        [,THREADING={AUTO|MAX|NONE|nn}]
                        [,UDBSchema=schema]
                        [,UPDINT={seconds|600}]
                        [,UPDREC={records|1000}]
                        [,USERID=user_id]
                        )
                  )
```

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=(UDB, ...)

Required. The type of CAPI_CONNECTION statement. For Db2 for Linux, UNIX, and Windows sources, this value must be UDB.

AGEOUTPERIOD=minutes

The number of minutes that must elapse before an outstanding UOW that has no change records of CDC interest will be removed from the calculation of the CDC restart point. The age is calculated as the difference between the start time of the outstanding UOW and the current time.

Use this parameter to prevent CDC failures that can occur if you shut down and then restart capture processing while the transaction is outstanding. After the restart, the Db2 transaction log in which the outstanding UOW started might not be available, causing the PowerExchange Db2 read process to fail.

Valid values are 60 to 43200. No default value is provided.

CCATALOG={capture_catalog|creator.DTLCCATALOG}

Optional. The name of the PowerExchange capture catalog table in the format *creator.table_name*.

Default is *creator.DTLCCATALOG*, where *creator* is the user ID that is used to connect to the database.

DBCONN=database_name

Optional. The name of the override database that you want to connect to for data extraction instead of the database that is specified for the registration group. The override database must contain tables and columns that are identical to those in the original database. The registration tag names and extraction map names include the original database name.

EPWD=encrypted_password

Optional. An encrypted password that is used with the user ID in the USERID parameter for database access.

If you specify this parameter, you must also specify either the USERID parameter. However, do not also specify the PASSWORD parameter.

Tip: You can create encrypted passwords in the PowerExchange Navigator.

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).

Note: If a committed transaction spans multiple partitions in a Db2 database, PowerExchange reports the number of SQL operations and transaction size across all of the partitions.

LIMITRESCAN={Y|N}

Optional. When PowerExchange change capture processing encounters a compressed record for a source table in the Db2 log, controls whether PowerExchange scans the log for records from propagatable tables up to an internally set limit or continues scanning until the buffer is full or until

the end-of-log (EOL). Use this parameter to avoid excessive scanning of the log, which can result in PowerExchange capture timeouts and degraded performance.

- **Y.** Limits PowerExchange scanning of log records for change capture. This option is recommended to avoid potential PowerExchange capture timeouts and failures.
- **N.** Allows PowerExchange scanning of log records to continue until the buffer is full or until the end-of-log (EOL). In this case, PowerExchange capture processing might end abnormally with a timeout error. This problem is more likely to occur when source tables have a low volume change activity.

Default is N.

LOGBUFSIZE={kilobytes_normal[,kilobytes_filtered]}

Optional. The buffer sizes, in kilobytes, that the PowerExchange capture process uses for reading Db2 log records in normal-read mode and filtered-read mode. In *filtered-read* mode, PowerExchange scans for compressed records from propagatable tables. You can specify a buffer size for one or both log read modes.

For the first normal-read buffer size, if you enter 0 or do not specify a value, 128 KB is used by default. For the second filtered-read buffer size, if you enter 0 or do not specify a value, the first normal-read buffer size is used by default for filtered read operations.

Note: If you set the PowerExchange capdl_bufsize environment variable at the direction of Informatica Global Customer Support, the environment variable value overrides the buffer sizes in the LOGBUFSIZE parameter.

MEMCACHE={cache_size|1024}

Optional. The maximum memory cache size, in kilobytes, that PowerExchange can allocate 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. Informatica recommends that you enter 0.

For each extraction session, PowerExchange keeps all changes for each pending 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 the pending UOWs, PowerExchange spills the changes in a UOW to sequential files, called UOW spill files, on disk.

Each UOW spill file contains change data from 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, the default value is often too small to eliminate UOW spill files.

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

- For Linux and UNIX, PowerExchange uses the current directory by default. To use a different directory, you must specify the TMPDIR environment variable.

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

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.

PASSWORD=password

Optional. A clear text password that is used with the user ID in the USERID parameter for database access.

If you specify this parameter, you must also specify either the USERID parameter. However, do not also specify the EPWD parameter.

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 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.

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.

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*|2147483647}

Optional. The amount of disk space, in bytes, that PowerExchange uses to allocate UOW spill files as temporary files.

Enter a number from 1 through 2147483647. Default is 2147483647 bytes.

THREADING={*AUTO*|*MAX*|*NONE*|*nn*}

Optional. Controls the number of threads that the UDB CAPI uses to capture change data from a Db2 database. Use this parameter to improve the performance of capture processing. If you have a partitioned database, you can use a maximum of one thread for each database partition node plus two additional threads for CAPI and merge processing.

Valid values are:

- **AUTO.** Use up to nine threads.
- **MAX.** Use one thread for each database partition plus two additional threads for CAPI and merge processing. The maximum number of threads is 99.
- **NONE.** Do not use multiple threads for capture processing.
- **nn.** A user-specified number of threads. Valid values are 1 to 99. For a partitioned database, if you enter a value that exceeds the sum of (*number_of_database_partitions* + 2), the CAPI does not use the excess threads.

Default is AUTO.

UDBSchema=*schema*

Optional. A schema name that overrides the schema name in capture registrations.

UPDINT={*seconds*|600}

Optional. The minimum number of seconds that PowerExchange must wait after encountering a virtual time stamp (VTS) in the Db2 log records for a partition before writing a positioning entry to the PowerExchange capture catalog table. The positioning entry, which contains a log sequence number (LSN) and VTS, indicates the location in the Db2 logs.

Enter a number from 1 through 2147483647. Default is 600 seconds.

Note: The minimum number of records that is specified in the UPDREC parameter must also be met before PowerExchange can write positioning entries to the capture catalog table.

UPDREC={*records*|1000}

Optional. The minimum number of Db2 log records that PowerExchange must read for a partition before writing a positioning entry to the PowerExchange capture catalog table. The positioning entry contains an LSN and VTS and indicates a location in the Db2 logs.

Enter a number from 1 through 2147483647. Default is 1000 records.

Note: The minimum wait period that is specified in the UPDINT parameter must also be met before PowerExchange can write positioning entries to the capture catalog table.

USERID=*user_id*

Optional. A database user ID that has SYSADM or DBADM authority.

If you specify this parameter, you must also specify the PASSWORD or EPWD parameter.

Using a Db2 Data Map

If you want PowerExchange to perform field-level processing on some records in a Db2 source table, you must use a data map.

For example, in some Db2 environments, a table can contain a single column that stores an array of fields in a format that is not consistent with the column datatype, such as a CHAR or VARCHAR column that stores multiple packed data fields. You can use an expression to modify this data before PowerCenter replicates it to a target. Also, if you add a user-defined field to a table in record view, you can build an expression to populate it. In the PowerExchange Navigator, you can define expressions only for data maps.

You might have data maps available for your source tables if you used PowerExchange bulk data movement to materialize your data targets. Bulk data movement requires data maps. You can use the bulk data maps for CDC if you merge them with the extraction maps for your data sources. The PowerExchange Navigator automatically generates an extraction map when you create a capture registration. Alternatively, you can manually add an extraction map.

Note: The field names in the data map must match the actual column names, as indicated in the Db2 capture registration.

Task Flow for Db2 Data Map Use

Perform the following tasks to use a Db2 data map for change data capture:

1. In the PowerExchange Navigator, create a capture registration for the Db2 source table.
2. Create a Db2 data map for the same Db2 source table if one is not available from a previous bulk data movement operation.
3. Merge the Db2 data map with the extraction map for the table.
4. Perform a row test on the merged extraction map.

For more information, see the *PowerExchange Navigator User Guide*.

Managing Db2 CDC

You might need to stop CDC for Db2 source tables occasionally, for example, to change the table definitions.

RELATED TOPICS:

- [“Stopping Db2 CDC” on page 95](#)
- [“Reconfiguring a Partitioned Database or Database Partition Group” on page 96](#)
- [“Db2 CDC Troubleshooting” on page 97](#)
- [“Changing a Db2 Source Table Definition” on page 95](#)

Stopping Db2 CDC

You might need to stop change data capture for a Db2 source table to perform troubleshooting or routine maintenance tasks, such as maintenance on the capture catalog table or redistribution of table data across reconfigured database partitions.

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

- Open the capture registration for a source 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.
- To temporarily stop change data capture, alter the Db2 table to specify the DATA CAPTURE NONE clause:

```
ALTER owner.table_name DATA CAPTURE NONE
```

When DATA CAPTURE NONE is specified, Db2 no longer writes changes to the Db2 log files in expanded format. Because CDC requires expanded format, PowerExchange can no longer capture change data for the table from the log files. If you set it back to DATA CAPTURE CHANGES, you might need to rematerialize the targets.

RELATED TOPICS:

- [“Stopping PowerCenter CDC Sessions” on page 291](#)

Changing a Db2 Source Table Definition

Occasionally, you might need to change the structural definition of a Db2 source table that is registered for change data capture.

If you make table definition changes that affect the columns from which change data is captured, perform this procedure to enable PowerExchange to switch to the updated table definition, while preserving access to previously captured data. These table definition changes include adding, altering, or dropping columns. Do not 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 table definition 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.

1. Stop all transactions, applications, and other activity that update the source 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. If you use the PowerExchange logger for Linux, UNIX, and Windows, shut down the Logger.
4. In the PowerExchange Navigator, delete the original capture registration and extraction map.
5. Use DDL to make the table changes.
6. In the PowerExchange Navigator, create a new capture registration that reflects the metadata changes and set its status to **Active**. A corresponding extraction map is generated.

Tip: When you create the capture registration, use the original registration name so that you will not need to edit the map name that is used by the PowerCenter CDC workflows that contain the changed source table.

7. If you shut down the PowerExchange Logger, warm start it.
The PowerExchange Logger begins capturing changes based on the new capture registration.
8. If necessary, change the target table definition to reflect the source table metadata changes.

9. In PowerCenter Designer, import the extraction map for the altered source table to create a new source definition. Edit the mapping if necessary.
If you also changed the target table, edit or re-create the target definition. Then edit the mapping, if necessary.
10. If necessary, rematerialize the target tables. After materialization completes, create new restart tokens.
11. Enable update activity on the source table again.
12. Restart the PowerCenter workflows.
If the table definition changes affected columns of CDC interest or you needed to edit the mapping, cold start the session. Otherwise, warm start the session.

Reconfiguring a Partitioned Database or Database Partition Group

In a Db2 partitioned database environment on Linux, UNIX, or Windows, you might need to perform some reconfiguration tasks.

Common tasks are:

- Add a new partition to a partitioned database, or drop an existing partition. Then reconfigure the database partition group or groups to reflect the change.
- Reconfigure a database partition group by adding or removing existing partitions.

Typically, after making these types of changes, you run the Db2 REDISTRIBUTE DATABASE PARTITION GROUP command to redistribute table data among the partitions in the updated database partition group.

If PowerExchange change data capture is active in the partitioned database environment, you must use the following procedure to properly resume change data capture after making the reconfiguration changes.

Adding or Dropping Database Partitions

Use this procedure to create a new partition in a partitioned database or to drop an existing partition, and then update the appropriate database partition group for the change.

1. Stop updates to the source tables.
2. Verify that any change data that was captured has completed extraction processing.
3. In PowerCenter, stop all CDC sessions that extract change data for the tables in the partitioned database instance.
4. For each table for which the DATA CAPTURE CHANGES clause is specified, specify DATA CAPTURE NONE.

Note: This step temporarily disables DB2 capture of changes to its log files. If you do not perform this step, DB2 records the data redistribution changes that result from the REDISTRIBUTE command as regular change data activity.

5. Execute the SQL for adding the new database partition or for dropping an existing partition.
6. Execute the ALTER DATABASE PARTITION GROUP SQL to add the new partition to or remove the dropped partition from the appropriate database partition group.
7. Run the DB2 REDISTRIBUTE DATABASE PARTITION GROUP command to redistribute table data among the partitions in the altered database partition group.
8. Back up the PowerExchange capture catalog table.
9. Run the PowerExchange DTLUCUDB SNAPUPDT command. Set the REPLACE option set to Y. This step updates the PowerExchange capture catalog table to reflect the reconfigured partitioned database.

Tip: Informatica recommends that you first perform a test run with the REPLACE option set to N.

10. For each table for which you specified DATA CAPTURE NONE in step 2, reinstate the DATA CAPTURE CHANGES clause.
11. Restart the PowerCenter CDC sessions to resume extraction processing.

Reconfiguring a Database Partition Group

Use this procedure to add a partition to or remove a partition from a database partition group without changing the partitioning of the partitioned database instance.

1. Stop updates to the source tables.
2. Verify that any change data that was captured has completed extraction processing.
3. In PowerCenter, stop all CDC sessions that extract change data for the tables in the partitioned database instance.
4. For each table for which the DATA CAPTURE CHANGES clause is specified, specify DATA CAPTURE NONE.

Note: This step temporarily disables DB2 capture of changes to its log files. If you do not perform this step, DB2 records the data redistribution changes that result from the REDISTRIBUTE command as regular change data activity.

5. Execute the ALTER DATABASE PARTITION GROUP SQL to add the new partition to or remove the dropped partition from the appropriate database partition group.
6. Run the DB2 REDISTRIBUTE DATABASE PARTITION GROUP command to redistribute table data among the partitions in the altered database partition group.
7. For each table for which you specified DATA CAPTURE NONE in step 2, reinstate the DATA CAPTURE CHANGES clause.
8. Restart the PowerCenter CDC sessions to resume extraction processing.

Db2 CDC Troubleshooting

If you encounter the following issue when running Db2 CDC on Linux, UNIX, or Windows, attempt the solution that is described. If you cannot resolve the problem, contact Informatica Global Customer Support.

Workaround for SQL1224 Error on AIX

On AIX systems only, you might receive the following PowerExchange message for a Db2 SQL1224 error when you connect locally to a Db2 database that has multiple other local connections:

```
PWX-20604 State=08001, Code=-1224, Msg=[IBM][CLI Driver] SQL1224N A database agent
could not be started to service a request, or was terminated as a result of a database
system shutdown or a force command. SQLSTATE=55032.
```

To circumvent this problem, implement a loopback TCP/IP connection for the local Db2 database. The database can then function as a remote client that uses TCP/IP instead of interprocess communications (IPC) over shared memory.

To implement a loopback connection without changing the database alias that users enter for database connection, issue the following Db2 commands:

```
db2 catalog tcpip node node_name1 remote server_name1 server port_number1
db2 uncatalog database database_name1
db2 catalog database database_name1 at node node_name1
db2 catalog database database_name1 as database_alias1
db2 catalog database database_alias1 as database_name1 at node node_name1
```

For more information about these commands, see your IBM Db2 documentation.

CHAPTER 5

Microsoft SQL Server CDC

This chapter includes the following topics:

- [Microsoft SQL Server CDC Overview, 99](#)
- [Planning for SQL Server CDC, 100](#)
- [Configuring SQL Server for CDC, 105](#)
- [Configuring PowerExchange for SQL Server CDC, 105](#)
- [Managing SQL Server CDC, 113](#)

Microsoft SQL Server CDC Overview

PowerExchange uses SQL Server transactional replication to capture change data from tables in SQL Server databases. PowerExchange uses the PowerExchange Client for PowerCenter (PWXPC) to coordinate with PowerCenter to move the captured change data to one or more targets.

For CDC to work, you must enable SQL Server Replication on the SQL Server instance from which change data is to be captured. If your database has a high volume of change activity, you should use a distributed server as the host of the distribution database.

To configure CDC in PowerExchange, 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 you want to use the PowerExchange Logger for Linux, UNIX, and Windows to capture change data and write it to PowerExchange Logger log files, configure the PowerExchange Logger. The change data is then extracted from the PowerExchange Logger log files. Benefits of using the PowerExchange Logger include fewer database accesses and a reduced volume of data held in the distribution database.

PowerExchange works with PowerCenter to extract change data from the SQL Server distribution database or PowerExchange Logger log files and load that data to one or more targets.

RELATED TOPICS:

- [“Planning for SQL Server CDC” on page 100](#)
- [“Configuring SQL Server for CDC” on page 105](#)
- [“Configuring PowerExchange for SQL Server CDC” on page 105](#)
- [“Managing SQL Server CDC” on page 113](#)
- [“PowerExchange Logger for Linux, UNIX, and Windows” on page 36](#)
- [“Introduction to Change Data Extraction” on page 248](#)

Planning for SQL Server CDC

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

SQL Server CDC Prerequisites

PowerExchange has the following prerequisites for SQL Server CDC:

- PowerExchange requires an edition of Microsoft SQL Server 2008 or later that supports transactional replication. You must configure and enable transactional replication on the source system to do CDC.
- The Microsoft SQL Server Agent and Log Reader Agent must be running on the Windows machine from which change data is extracted. Usually, the SQL Server Agent remains running after it is initially started. For more information, see your SQL Server documentation.
- Each source table registered for change capture in the publication database must have a primary key.
- The PowerExchange Listener can run on a Windows system or Linux system. However, the PowerExchange Listener cannot run on a UNIX system.
- If you use the PowerExchange Logger for Linux, UNIX, and Windows, you can run it on the system where the PowerExchange Listener runs or on another Linux, UNIX, or Windows system for which you configure remote logging.
- On the PowerExchange Listener and PowerExchange Navigator machines, ensure that you have a license.key file that contains a key that authorizes use of Microsoft SQL Server CDC.

Required User Authority for SQL Server CDC

Verify that you have the proper authority level to complete registration and SQL Server configuration tasks.

The following user authority levels are required:

- To enable transactional replication on the publication database, you must be assigned the System Administrator role.
- To create a PowerExchange publication that is associated with a distribution database, you must be assigned the System Administrator role.
Note: In the PowerExchange Navigator, the first time you add a registration group for a distinct publication database, PowerExchange creates the publication named "PowerExchange Change Capture" and enables transactional replication on the publication database if replication is not yet enabled. The creation of the PowerExchange publication requires you to have the System Administrator role. Thereafter, when you create additional registrations for the same publication database, you need the DB_OWNER role.
- To create capture registrations from the PowerExchange Navigator and allow PowerExchange to generate the corresponding SQL Server articles in the publication, you must be assigned the DB_OWNER role.
- To run change data extractions against a SQL Server distribution database, you must have read access to the database. Also, you need SELECT authority on the MSrepl_commands and MSrepl_transactions system tables in the distribution database.
- If you use Microsoft SQL Server NTLM and Active Directory authentication to control access to a SQL Server database, you can enter a user ID and password that has the proper authority when creating a registration group or performing a database row test. Enter the user ID in the format *domain\user_name*.

If you do not specify a user ID and password when creating a registration group, the PowerExchange Navigator and the extraction processes try to use your Windows user ID and password to connect to the SQL Server distribution database.

SQL Server Datatypes Supported for CDC

PowerExchange supports most SQL Server datatypes for CDC, with some exceptions.

The following table indicates the SQL Server datatypes that PowerExchange supports and does not support for CDC:

Datatype	Supported for CDC?	Comments
bigint	Yes	-
binary	Yes	-
bit	Yes	-
char	Yes	-
date	Yes	In PowerCenter, when you import source metadata from PowerExchange to create a source definition, PowerExchange converts date columns to timestamp columns. This conversion is for consistency with PowerCenter datatype handling.
datetime	Yes	-
datetime2	Yes	-
datetimeoffset	Yes	PowerCenter treats this datatype as varchar.
decimal	Yes	-
float	Yes	-
geography	No	-
geometry	No	-
hierarchyid	No	-
image ¹	No	Use varbinary(MAX) instead.
int	Yes	-
money	Yes	-
nchar	Yes	-
ntext ¹	No	Use nvarchar(MAX) instead.
numeric	Yes	-
nvarchar	Yes	-
real	Yes	-
smalldatetime	Yes	-

Datatype	Supported for CDC?	Comments
smallint	Yes	-
smallmoney	Yes	-
sql_variant	No	PowerExchange does not capture change data for sql_variant columns but does capture change data for other columns in the same table.
text ¹	No	Use varchar(MAX) instead.
time	Yes	-
timestamp	Yes	-
tinyint	Yes	-
uniqueidentifier	Yes	PowerCenter imports the uniqueidentifier datatype as a varchar datatype of 38 characters.
user-defined datatypes (UDTs)	Yes	PowerExchange treats a UDT in the same way as the datatype on which the UDT is based.
varbinary	Yes	-
varchar	Yes	-
xml	Yes	PowerExchange treats this datatype as varchar(MAX).
<p>1. PowerExchange might not be able to capture change data for columns that have the image, ntext, or text datatype because of SQL Server transactional replication restrictions on these column types. Instead, use the alternative datatypes that Microsoft recommends, as indicated in the Comments column.</p>		

SQL Server CDC Operational Considerations

PowerExchange for SQL Server CDC has the following operational considerations:

- PowerExchange can capture change data from SQL Server distribution databases for which Transparent Data Encryption (TDE) is enabled. No special configuration tasks are required.
- PowerExchange uses the DataDirect ODBC driver for Microsoft SQL Server to create capture registrations and capture change data from the distribution database. The PowerExchange installation provides the ODBC driver in the `powx_base_installation\ODBCversion\Drivers` directory. No configuration of the driver is required.
- PowerExchange does not capture change data for SQL Server system tables.
- The maximum length of a row for which PowerExchange can capture and process change data is 128,000 bytes.
- PowerExchange does not capture the user ID that is associated with the original transaction that updated the database.
- The timestamp that PowerExchange records for each captured change indicates when the change was written to the distribution database by the SQL Server Replication Log Reader, not when the original transaction occurred.

- PowerExchange does not capture change data for derived columns that are not persisted. SQL Server computes values for these columns at run-time based on an expression but does not store the values in a table.
- SQL Server publishes *deferred updates* to SQL Server tables as DELETEs followed by INSERTs rather than as UPDATEs. Consequently, PowerExchange propagates deferred updates as DELETEs followed by INSERTs, even if you select **AI** for the **Image Type** attribute in the CDC connection. PowerExchange does not include before image (BI) and change indicator (CI) information in DELETE and INSERT records. If you must capture a deferred update as an UPDATE for business reasons, set the SQL Server 8207 trace flag. This flag causes the SQL Server Replication Log Reader to combine the DELETE and INSERT pair into a single UPDATE. For more information about SQL Server processing of deferred updates and the SQL Server 8207 trace flag, see the SQL Server documentation
- PowerExchange does not support the use of local aliases when connecting to SQL Server and creating publications at registration creation.
- If you need to switch the status of multiple SQL Server capture registrations from active to inactive or from inactive to active, use the DTLUCBRG utility with the MSSOPTS UPDATESTATUS parameter. This optional parameter enables you to switch the status of many registrations in one operation and regenerate the associated SQL Server publications.
- If you run the PowerExchange Listener on a Linux system or on a Windows system that is remote from the system where the SQL Server distribution database runs, you must define a MSQL CAPI CONNECTION statement that provides connection information for the SQL Server distribution database in the DBMOVER configuration file on the Listener system. Also, in the DBMOVER configuration files on the PowerCenter Integration Service machine and the PowerExchange Navigator system, define a NODE statement that points to the PowerExchange Listener system. To connect to the SQL Server system, the PowerExchange Listener uses the DataDirect ODBC driver that PowerExchange supplies.
- If columns are added with the NOT NULL and DEFAULT options to a Microsoft SQL Server 2012 or later source table, PowerExchange adds an appropriate default value, which is based on the column datatype, to the captured before image so that extraction processing can continue. PowerExchange does not process the added columns until a subsequent Update or Delete occurs on the source. You cannot change the default values that PowerExchange uses. If the use of the default values is not acceptable in your environment, you must rebuild the source table after the columns are added.
- PowerExchange CDC use of SQL Server transactional replication does not support column-level collation settings on CHAR, VARCHAR, and TEXT columns with a code page that is different from that of the database collation. To prevent change data loss or corruption, either do not use column-level collation or change the CHAR, VARCHAR, and TEXT column datatypes to NCHAR, NVARCHAR, or NTEXT.
- If the license.key file on the PowerExchange Navigator machine does not authorize use of SQL Server CDC, when you try to create a capture registration for a SQL Server data source, the PowerExchange Navigator fails with a license key error. In this case, go the Informatica Network at <https://network.informatica.com> and access eSupport. Then create a case of type=shipping to request a key that includes SQL Server CDC.
- PowerExchange can capture changes that are written to an availability database in a SQL Server Always On Availability Group. An Availability Group consists of primary and secondary replica databases on multiple nodes in a Windows Server Failover Clustering (WSFC) cluster. Only the following configuration has been tested and certified for PowerExchange CDC:
 - The distribution database is installed on a node outside of the Always On Availability Group cluster, which is consistent with SQL Server requirements.
 - PowerExchange is installed on a node outside of the Always On Availability Group cluster.
 - When you create a registration group for the SQL Server Always On Availability Group source, you specify the Availability Group listener name in the **Database Server** field.

Note: If you need to use another configuration, contact Informatica Global Customer Support. Informatica will try to accommodate your request.

After CDC processing is running, if the primary database fails over to a secondary replica database on another node, PowerExchange can continue to capture change data from the distribution database without data loss.

- PowerExchange uses the DataDirect ODBC driver for SQL Server to connect to Microsoft SQL Server source databases. You can define the ODBC_CONN_PARAMS statement in the dbmover.cfg file. If you want to add ODBC parameters to the connection strings that are used on the servers you specify for the following processing:
 - Change data capture
 - Creating, deleting, or modifying capture registrations from the PowerExchange Navigator or DBLUCBRG utility
- For example, you might need to add ODBC parameters if your site policies require database connections to use SSL encryption, specific cryptographic protocols, or self-signed or third-party signed SSL certificates. For more information, see the *PowerExchange Reference Manual*.
- PowerExchange does not capture changes from SQL Server views.
 - Set the CAPI_CONNECTION parameter GUIDBRACES=N to remove braces {} around GUID values in data captured from registered SQL Server columns that have the uniqueidentifier datatype.

Extracting Data for Multiple Publication Databases

If you plan to extract data from a Microsoft SQL Server distribution database that contains information for articles in multiple publication databases in a single pass, you must set some parameters for extraction processing to succeed.

This requirement applies to extraction of data directly from the change stream in real time extraction mode or from PowerExchange Logger for Linux, UNIX, and Windows log files in continuous extraction mode.

Specify the following parameters:

- In the MSQL CAPI_CONNECTION statement in the dbmover configuration, verify that the MULTIPUB parameter is set to Y, the default setting. If you do not use this setting, the extraction will fail with message PWX-15757.

Tip: If you plan to extract data for single publication database, set this parameter to N for more efficient extraction processing.
- If you use the PowerExchange Logger for Linux, UNIX, and Windows, set the following parameters:
 - In the PowerExchange Logger configuration file, pwxccf, ensure that you define the DBID, DISTDB, and DISTSRV parameters. These parameters are usually required for PowerExchange Logger processing of SQL Server sources.
 - In the PowerCenter PWX MSSQL CDC Change or Real Time connection definition, specify the PowerExchange Logger DBID parameter value for the **Logger DBID** attribute.
 - To perform a database row test, specify the PowerExchange Logger DBID parameter value in the **MSS LUW DBID** field in the CAPXRT Advanced Parameters dialog box.

Configuring SQL Server for CDC

You must perform a few configuration tasks to prepare SQL Server for PowerExchange change data capture (CDC).

If your SQL Server tables have a high level of update activity, use a distributed server as the host of the distribution database from which to capture change data. This practice prevents competition between PowerExchange CDC and your production database for CPU use and disk storage.

1. Start the SQL Server Agent and Log Reader Agent if they are not running. For more information, see your Microsoft SQL Server documentation.
2. Verify that each source table that you want to register for change capture has a primary key.

Also, transactional replication must be enabled on the publication database. Normally, transactional replication is enabled the first time any table in the database is registered for change data capture. Each time you register an additional table, PowerExchange checks that transactional replication is still enabled and, if necessary, re-enables it. You can configure transactional replication settings in SQL Server as needed.

Tip: The default transactional retention period at the Distributor is 72 hours. If you are using the PowerExchange Logger, accept this default retention period. If you are not using the PowerExchange Logger, Informatica recommends that you increase the retention period to 14 days. However, you might need to decrease the value if you have a high volume of transactions or space constraints. For more information, see your Microsoft SQL Server documentation.

Configuring PowerExchange for SQL Server CDC

The tasks that you perform to configure PowerExchange for change data capture (CDC) depend on whether you want to use the PowerExchange Logger for Linux, UNIX, and Windows and the extraction mode you plan to use.

RELATED TOPICS:

- [“Customizing the dbmover Configuration File for SQL Server CDC” on page 106](#)
- [“Configuring PowerExchange CDC with the PowerExchange Logger” on page 106](#)
- [“Configuring PowerExchange CDC without the PowerExchange Logger” on page 105](#)

Configuring PowerExchange CDC without the PowerExchange Logger

If you plan to run extractions in real-time extraction mode and *not* use the PowerExchange Logger for Linux, UNIX, and Windows, complete the following tasks to configure PowerExchange CDC:

1. When you configure the dbmover.cfg file, define the following statements:
 - CAPT_PATH
 - CAPT_XTRA
 - MSQL CAPL_CONNECTION
2. In the PowerExchange Navigator, create a capture registration for each SQL Server source table. The PowerExchange Navigator generates a corresponding extraction map.

Tip: Set the **Condense** option to **Part** even though you do not plan to use the PowerExchange Logger, unless you have a particular reason not to do so. This practice prevents having to change the capture registrations later if you decide to use the PowerExchange Logger. You might want to set the **Condense** option to **None** if you run both real-time and continuous extractions against tables defined by the same capture registrations and do not want the PowerExchange Logger to capture change data for certain registered tables.

If capture registrations already exist for these tables, delete the existing registrations and extraction maps and create new ones.

The PowerExchange Navigator generates a corresponding extraction map for each capture registration.

3. Activate the capture registrations. Usually, you do this task after materializing the targets.

Next Step: Configure and start extractions. You must use real-time extraction mode.

Configuring PowerExchange CDC with the PowerExchange Logger

If you plan to run extractions in batch or continuous extraction mode and use the PowerExchange Logger for Linux, UNIX, and Windows, complete this procedure to configure PowerExchange CDC.

1. When you configure the dbmover configuration file, define the following statements:
 - CAPT_PATH
 - CAPT_XTRA
 - MSQl CAPI_CONNECTION
 - CAPX CAPI_CONNECTION (for continuous extraction mode only)
2. Configure the pwxcl configuration file for the PowerExchange Logger.
3. In the PowerExchange Navigator, create a capture registration for each SQL Server source table.

You must set the **Condense** option to **Part**. If capture registrations already exist for these tables, you can edit the **Condense** option without affecting the registration version.

The PowerExchange Navigator generates a corresponding extraction map.
4. Start the PowerExchange Logger.
5. Activate the capture registrations.

Usually, you do this task after materializing the targets.

Next Step: Configure and start extractions. You can use either batch extraction mode or continuous extraction mode.

Customizing the dbmover Configuration File for SQL Server CDC

In the dbmover configuration file, add statements for Microsoft SQL Server CDC.

You must include the following statements for SQL Server CDC:

CAPT_PATH statement

Path to the local directory on a Linux, UNIX, or Windows system that contains the control files for CDC.

These files are: the CCT file for capture registrations, the CDEP file for application names that are used for ODBC extractions, and the CDCT file for the PowerExchange Logger for Linux, UNIX, and Windows.

CAPT_XTRA statement

Path to the local directory on a Linux, UNIX, or Windows system that stores extraction maps.

MSQL CAPI_CONNECTION statement

A named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and control CDC processing for SQL Server sources.

Add this statement to the dbmover.cfg file on the system where SQL Server capture registrations are stored. This location corresponds to the **Location** node that you specify when defining a registration group. Usually, this location is where the source database resides.

If you plan to use continuous extraction mode, you must also include the following statement:

CAPX CAPI_CONNECTION statement

A named set of parameters that the CAPI uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files.

Also, Informatica recommends including the LOGPATH and TRACING statements to make finding messages easier. The LOGPATH statement defines a directory specifically for PowerExchange message log files, and the TRACING statement enables PowerExchange to create an alternative set of message log files for each PowerExchange process.

For more information about all DBMOVER statements, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“CAPI_CONNECTION - CAPX Statement” on page 60](#)
- [“CAPI_CONNECTION - MSQL Statement” on page 107](#)
- [“CAPT_PATH Statement” on page 31](#)
- [“CAPT_XTRA Statement” on page 31](#)

Example dbmover Statements for SQL Server

The following statements are typical of those included in a dmover.cfg for SQL Server CDC:

```
LOGPATH="C:\Informatica\PowerExchangeVnnn\Logs"
CAPT_XTRA="C:\Informatica\PowerExchangeVnnn\Capture\camaps"
CAPT_PATH="C:\Informatica\PowerExchangeVnnn\Capture"
CAPI_CONN_NAME=CAPIMSSC
CAPI_CONNECTION=(NAME=CAPIMSSC
                  ,TYPE=(MSQL,DISTSRV=AUX159908\PWXPC
                        ,DISTDB=distribution
                        ,RSTRADV=30))
```

Note: You must use non-curly double quotation marks around values that include a space.

CAPI_CONNECTION - MSQL Statement

The MSQL 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 Microsoft SQL Server sources.

Operating Systems: Windows

Data Sources: Microsoft SQL Server

Required: Yes for Microsoft SQL Server CDC

Syntax:

```
CAPI_CONNECTION=( [DLLTRACE=trace_id]
                  ,NAME=capi_connection_name
                  [,TRACE=trace_name]
                  ,TYPE=(MSQL
                        ,DISTDB=distribution_database
```

```

, DISTRV=distribution_server
[, BATCHSIZE=number]
[, DWFLAGS={flag1flag2flag3flag4|NNNN}]
[, ENABLELWM={N|Y}]
[, EOF={N|Y}]
[, GUIDBRACES={Y|N}]
[, MEMCACHE={cache_size|256}]
[, MULTIPUB={N|Y}]
[, POLWAIT={seconds|1}]
[, RECONNTRIES={number|12}]
[, RECONNWAIT={seconds|5}]
[, RSTRADV=seconds]
[, SQLNOLOCK={N|Y}]
[, UIDFMT={DBNAME|NONE}]
)
)

```

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=(MSQL, ...)

Required. Type of CAPI_CONNECTION statement. For Microsoft SQL Server sources, this value must be MSQL.

DISTDB=distribution_database_name

Required. The name of the distribution database.

DISTRV=distribution_database_server

Required. The network name of the server that hosts the distribution database. This name is different from the network name of the SQL Server publication instance if the distribution database resides on a different instance.

Note: If the database server uses a port number other than the default port number of 1433, append the non-default port number to the server name by using the following format:

`\server_name,port_number\`. Otherwise, capture processing fails.

BATCHSIZE=number

Optional. The number of rows from which PowerExchange captures change data before closing the cursor and then reopening it. This parameter allows resources to be released periodically to reduce the capture processing load on system memory and to reduce temporary tables in the tempdb database. Valid values are 0 through 2147483647. No default is provided.

Specify this parameter only at the direction of Informatica Global Customer Support. It can degrade CDC performance because PowerExchange issues the data read query more often.

DWFLAGS={*flag1flag2flag3flag4*|NNNN}

Optional. Series of four positional parameters that control whether processing stops or continues when data loss, truncation, schema changes, or unrecognized transaction log records occur.

Specify this statement only at the direction of Informatica Global Customer Support.

Enter the following positional parameters:

- *flag1*. Controls whether PowerExchange stops a change data extraction when PowerExchange retrieves data of an unexpected length from the distribution database. Enter Y to continue processing or enter N to stop processing.
- *flag2*. Controls whether PowerExchange stops a change data extraction when it detects a schema change. Enter Y to continue processing or enter N to stop processing.
- *flag3*. Controls whether PowerExchange stops a change data extraction when PowerExchange does not find the requested start sequence in the transaction log. Enter Y to continue processing or enter N to stop processing.
- *flag4*. Controls whether PowerExchange stops a change data extraction when PowerExchange finds an unrecognized record in the transaction log. Enter Y to continue processing after error message PWX-15742 or enter N to stop processing.

Default is NNNN, which indicates none of the parameters are set.

ENABLELWM={*N*|*Y*}

Optional. When you use the PowerExchange Logger for Linux, UNIX, and Windows, controls whether the PowerExchange consumer API (CAPI) connection process deletes data read from the SQL Server distribution database after the data has been hardened to PowerExchange Logger log files or after the PowerExchange publication expiry time has elapsed. You can use this parameter to improve distribution database performance and to prevent the distribution database from growing too large in size when the PowerExchange Logger is in use.

Enter one of the following options:

- **N**. The distribution database cleanup job deletes data from the distribution database after the expiry time for the PowerExchange publications elapses. This option might degrade the performance of the distribution-database cleanup job and cause excessive growth of the distribution database.

- **Y.** The CAPI connection process deletes processed data from the distribution database after the data has been hardened to the PowerExchange Logger log files. After a log file switch, the PowerExchange Logger sends a low water marker to the CAPI connection process to identify the last end UOW prior to the file switch. At the end of the next capture cycle, after the CAPI connection process has read to the end of the available data in the distribution database, the CAPI deletes all of the processed data for the PowerExchange publications up to and including the low water mark data from the distribution.dbo.MSrepl_commands table in the distribution database.

Note: The user ID under which the PowerExchange Logger runs must have delete authority on the MSrepl_commands table.

This option can help improve distribution-database performance and control distribution-database size. However, if the SQL Server Log Reader Agent is writing very large UOWs to the distribution database when the CAPI connection processes the low water mark data, the performance of the distribution database might be temporarily degraded because the CAPI connection process must wait for a lock on the MSrepl_commands table.

Note: If you run multiple extractions against a single distribution database for different publication databases and use ENABLELWM=Y for one CAPI connection and ENABLELWM=N with a RSTRADV value for another CAPI connection, PowerExchange might issue error message PWX-15756 for the connection with ENABLELWM=N. The message incorrectly reports that change data has been lost. To suppress this error, add the DWFLAGS=NNYN parameter to MSQL CAPI_CONNECTION statement.

Default is N.

EOF={N|Y}

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

Enter one of the following options:

- **N.** PowerExchange does not stop change data extractions when the EOL is reached.
- **Y.** PowerExchange stops change data extractions when the EOL is reached.

Default is N.

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

- For CDC sessions that use real-time extraction mode, enter 0 for the **Idle Time** attribute of the PWX MSSQL CDC Real Time application connection.
- For the PowerExchange Logger for Linux, UNIX, and Windows, enter 1 for the COLL_END_LOG statement in the pwxcl.cfg configuration file.
- For CDC sessions that use ODBC connections, enter 0 for the WAITTIME parameter in the ODBC data source.

GUIDBRACES={Y|N}

Optional. Controls whether PowerExchange retains or removes braces {} around GUID values in data captured from registered SQL Server columns that have the uniqueidentifier datatype. If the PowerCenter session will write the data to SQL Server target uniqueidentifier columns, set this parameter to N to remove the braces. Otherwise, the session will encounter writer errors.

- **N.** Remove braces.

- **Y.** Retain braces.

Default is Y.

MEMCACHE={cache_size|256}

The maximum size, in kilobytes, of the memory cache that stores change data for a single SQL operation that is captured from the SQL Server distribution database. The memory cache stores the full row image, which can include both the before image and after image and any LOB data.

Valid values are 0 through 2147483647. Default is 256. If you enter 0, the default value is used.

MULTIPUB={N|Y}

Optional. Indicates whether you capture change data from the distribution database for articles in a single publication database or in multiple publication databases. This option can affect the performance of CDC processing in real time extraction mode and in continuous extraction mode with the PowerExchange Logger for Linux, UNIX, and Windows.

Enter one of the following options:

- **N.** Specify this option if you capture change data for articles in a single publication database. Informatica recommends this option in this scenario because it causes PowerExchange to extract changes much more efficiently. It can also help reduce resource usage.
- **Y.** Use this option to extract change data for articles in multiple publication databases in a single CDC session or in a single PowerExchange Logger for Linux, UNIX, and Windows pass. If you do not use this option in this scenario, extraction processing fails with message PWX-15757.

This option might cause change records to be written to the distribution database more slowly.

To improve performance, add the following index to the distribution database:

```
USE [distribution]
GO
/***** Object: Index [IX_MSrepl_transactions] Script Date: 03/31/2012
11:56:07 *****/
CREATE NONCLUSTERED INDEX [IX_MSrepl_transactions] ON [dbo].
[MSrepl_transactions]
(
    [entry_time] ASC,
    [publisher_database_id] ASC,
    [xact_seqno] ASC,
    [xact_id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = ON, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS =
ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO
```

Default is Y.

You can change the MULTIPUB setting after capturing changes. To maintain data integrity, follow the procedure for changing the MULTIPUB parameter setting in the *PowerExchange CDC Guide for Linux, UNIX, and Windows*. If you use the PowerExchange Logger for Linux, UNIX, and Windows and change the setting from Y to N, you must cold start the PowerExchange Logger.

POLWAIT={seconds|1}

Optional. The maximum number of seconds that PowerExchange waits after reaching the end of log before polling the source database for more change data.

For Microsoft SQL Server sources, the polling frequency also depends on the PowerExchange Logger NO_DATA_WAIT2 parameter, or if you do not use the PowerExchange Logger, the polling frequency depends on the PWX Latency attribute on the PWX CDC application connection. If the NO_DATA_WAIT2 or PWX Latency value is less than the POLWAIT value, the lesser value takes

precedence. In this case, PowerExchange polls the source more frequently than expected based on the POLWAIT parameter only.

Valid values are 1 through 2147483647. Default is 1.

RECONNTRIES={number|12}

The maximum number of times that PowerExchange tries to reconnect to the Microsoft SQL Server database after the connection is dropped. Use this parameter in conjunction with the RECONNWAIT parameter if you get the following ODBC connection error and want to improve connection resiliency:

```
PWX-15790 ODBC driver for Microsoft SQL Server returned error [08S01]
[Informatica][ODBC SQL Server Wire Protocol driver]Unexpected Network Error.
ErrNum = 10054.
```

Valid values are 0 or any positive number. A value of 0 results in no connection retries. Default is 12.

RECONNWAIT={seconds|5}

The number of seconds that PowerExchange waits before any attempt to reconnect to a Microsoft SQL Server database after the connection has been dropped. Use this parameter in conjunction with the RECONNTRIES parameter if you get the PWX-15790 message for an ODBC driver error and want to improve connection resiliency.

Valid values are 0 through 3600. A value of 0 results in no waiting between connection retries. Default is 5.

RSTRADV=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.

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.

Valid values 0 through 86400. No default is provided.

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.

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.

Warning: 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.

SQLNOLOCK={N|Y}

Optional. Controls whether PowerExchange SELECT statements use the NOLOCK hint when querying the SQL Server distribution database for change data capture. The NOLOCK hint can avoid

lock contention with the SQL Server utilities but might cause PowerExchange to miss some change records.

Enter one of the following options:

- **N.** PowerExchange SELECT queries that retrieve data from the distribution database do *not* use the NOLOCK hint. If locks are held on some change records, PowerExchange queries cannot retrieve the data until the locks are released. With this setting, PowerExchange queries might take longer to complete. However, no changes are skipped and data integrity is preserved. Use this option only when the MULTIPUB parameter is set to Y.
- **Y.** PowerExchange SELECT queries that retrieve data from the distribution database use the NOLOCK hint. Use this option only when the MULTIPUB parameter is set to N. If the MULTIPUB parameter is set to Y, SQL Server might use allocation order scans to retrieve data for PowerExchange queries, which can result in missed change data and data corruption.

Tip: Instead of using SQLNOLOCK=Y, Informatica recommends that you set the isolation level for the distribution database to READ_COMMITTED_SNAPSHOT ON to avoid data integrity problems.

Default is **N** if MULTIPUB is set to Y, or **Y** if MULTIPUB is set to N.

UIDFMT={DBNAME |NONE}

Optional. Controls the type of value that PowerExchange uses to populate the generated DTL__CAPXUSER column in each change record. Options are:

- **DBNAME.** Returns the Microsoft SQL Server publication database name.
- **NONE.** Returns a null because a user ID is not available.

Default is NONE.

Managing SQL Server CDC

You might need to stop CDC for source tables occasionally, for example, to change the table definitions.

Disabling Publication of Change Data for a SQL Server Source

You can disable publication of change data for a SQL Server source. For example, you might disable publication to perform some database maintenance, change the table definition, or avoid capturing unwanted changes.

- Open the capture registration for the table, and change the **Status** setting from **Active** to **History**.

This action disables publication of the SQL Server article for the table to the distribution database, which causes change capture to stop.

Warning: After the registration status is set to **History**, you cannot activate the registration for CDC use again.

Changing a SQL Server Source Table Definition

If you change the definition of a SQL Server source table that is registered for change data capture, use this procedure to enable PowerExchange to use the updated table definition and preserve access to previously captured data. Table definition changes include adding, altering, or dropping columns.

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 SQL Server source table definition:

1. Stop DELETE, INSERT, and UPDATE activity against 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. Delete the capture registration and extraction map.
4. Use DDL to change the table definition in SQL Server.
5. In the PowerExchange Navigator, create a new capture registration that reflects the metadata changes and set its status to **Active**. PowerExchange creates a corresponding extraction map.
The newly activated capture registration becomes eligible for change data capture.
6. If necessary, change the target table definition to reflect the source table metadata changes.
7. In the PowerCenter Designer, import the altered source and target definitions. Edit the mapping if necessary.
8. If necessary, rematerialize the target tables. After materialization completes, create new restart tokens.
9. Create new restart tokens for the altered table.
10. Re-enable DELETE, INSERT, and UPDATE activity against the table.
11. Cold start the extraction workflows.

Changing the MULTIPUB Parameter Setting After Running Extractions

After you run change data extraction processing, you can change the MULTIPUB parameter setting in the MSQL CAPI_CONNECTION statement. You might need to do this task if you add or remove publication databases that include sources of CDC interest. To preserve data integrity, you must use the proper procedure.

The MULTIPUB parameter indicates whether you extract data for articles in a single publication database or in multiple publications. For a single publication database, Informatica recommends that you set MULTIPUB to N so that PowerExchange can use more efficient extraction processing. For multiple publications, you must set MULTIPUB to Y, the default setting. This parameter applies to real time extractions directly from the change stream, and PowerExchange Logger for Linux, UNIX, and Windows extractions in continuous extraction mode.

To switch the MULTIPUB setting from Y to N:

Use this procedure to switch the MULTIPUB from the default of Y to N. If you use the PowerExchange Logger for Linux, UNIX, and Windows, you must cold start it after making this change.

1. Stop extraction workflows that process the SQL Server distribution database and that are running in real-time extraction mode or continuous extraction mode.
2. If you use the PowerExchange Logger for Linux, UNIX, and Windows, stop the PowerExchange Logger.
3. In the dbmover configuration file, edit the MSQL CAPI_CONNECTION statement to switch the MULTIPUB parameter setting from Y to N.
4. Cold start the PowerExchange Logger.
5. Restart the extraction workflows.

Note: The sequence tokens no longer include a timestamp.

To switch the MULTIPUB setting from N to Y:

Use this procedure to switch the MULTIPUB from N back to Y. If you use the PowerExchange Logger for Linux, UNIX, and Windows, you do not need to cold start it after making this change.

1. Stop DELETE, INSERT, and UPDATE activity on the SQL Server source tables.
2. Wait for the extraction workflows to reach the end of log and then stop them.
3. In the dbmover configuration file, edit the MSQL CAPI_CONNECTION statement to switch the MULTIPUB parameter setting from N to Y.
4. If you use the PowerExchange Logger for Linux, UNIX, and Windows, ensure that the DISTSRV and DISTDB parameters are specified in the PowerExchange Logger pwxcl.cfg configuration file. These parameters are required when MULTIPUB=Y.

5. To help avoid performance degradation, define the following index on the distribution database:

```
USE [distribution]
GO
/***** Object: Index [IX_MSrepl_transactions] Script Date: 03/31/2012 11:56:07
*****/
CREATE NONCLUSTERED INDEX [IX_MSrepl_transactions] ON [dbo].[MSrepl_transactions]
(
    [entry_time] ASC,
    [publisher_database_id] ASC,
    [xact_seqno] ASC,
    [xact_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = ON, SORT_IN_TEMPDB = OFF,
IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF, ONLINE = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO
```

6. To get the current restart tokens for the end of log, use one of the following methods:
 - Run the DTLUAPPL utility with the GENERATE RSTKKN option.
 - In the PowerExchange Navigator, perform a database row test with a SELECT CURRENT_RESTART SQL statement.
 - 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.
7. Add the current restart tokens for the extractions to the restart token file.
8. Allow DELETE, INSERT, and UPDATE activity to resume on the SQL Server tables.
9. Cold start the extraction workflows.

Note: PowerExchange adds a timestamp in the sequence token to combine the data from multiple publication databases during extraction processing.

CHAPTER 6

MySQL CDC

This chapter includes the following topics:

- [MySQL CDC Overview, 116](#)
- [About the Binary Log File, 117](#)
- [DDL-Updated Catalog of MySQL Source Table Definitions, 117](#)
- [MySQL CDC Operational Considerations, 118](#)
- [MySQL Datatypes Supported for CDC, 121](#)
- [Implementation Task Flow, 122](#)
- [Preparing MySQL Sources, 123](#)
- [Configuring PowerExchange for MySQL CDC, 124](#)
- [Managing MySQL CDC, 129](#)

MySQL CDC Overview

PowerExchange uses the MySQL binary log reader, `mysqlbinlog`, to read change events for source tables from the MySQL binary log. PowerExchange then extracts the change records from the real-time change stream or from PowerExchange Logger for Linux, UNIX, and Windows log files and makes the changes available to PowerCenter CDC sessions.

The binary log reader and PowerExchange capture process must run on the same Linux or Windows machine. This machine can be remote from the MySQL source database server.

Use of the PowerExchange Logger is optional. MySQL CDC is certified with Logger instances that run on Linux or Windows.

If you have a MySQL Enterprise Edition source, PowerExchange uses the DataDirect ODBC driver for MySQL to retrieve source metadata from the MySQL database server. This ODBC driver is included in the PowerExchange installation for Linux. If you have a MySQL Community Edition source, you must use the MySQL native ODBC driver. PowerExchange does not supply the native driver.

PowerExchange works with the PowerExchange Client for PowerCenter (PWXPC) to extract change records from the real-time change stream or from the PowerExchange Logger logs and transmit the data to CDC sessions. The CDC sessions can load the data to one or more targets. Use of PowerCenter transformations is optional.

RELATED TOPICS:

- [“Preparing MySQL Sources” on page 123](#)
- [“Configuring PowerExchange for MySQL CDC” on page 124](#)
- [“Managing MySQL CDC” on page 129](#)

About the Binary Log File

For MySQL sources, PowerExchange uses the MySQL `mysqlbinlog` utility to read change events that are in binary format from binary log files.

If the `mysqlbinlog` utility is remote from the MySQL server where the binary logs reside, specify the location of the `mysqlbinlog` utility in the `MYSQLBINLOG` parameter of the `MYSQL CAPLCONNECTION` statement or in the `PATH` environment variable.

MySQL binary log files contain events that describe database changes, including data changes and DDL changes. The term *binary log file* refers to an individual log file that contains database events and that has a unique name. The term *binary log* refers to a series of binary log files plus the index file that contains the names of the binary log files that have been used.

A binary log file name contains a base name and generated numeric suffix in the following format: *base_name_binlog.numeric_suffix*, for example, `mysql_5_7_binlog.000001`. The numeric suffix is incremented with each new log file.

When MySQL creates a new physical binary log file, it increments the numeric suffix in the log file name. MySQL creates a new log file under the following conditions:

- The MySQL server is restarted.
- The binary log file reaches its maximum size, as defined in the MySQL `max_binlog_size` variable.

If a MySQL server is restarted with a new binlog base name, a new series of binary logs starts. The first log file in the new series has a numeric suffix of 000001. Queries based on the online binary log file name will not return information about the previous binary logs. However, the previous binary logs and index file remain available on the server disk.

Important: This scenario might cause data loss. To ensure data integrity, verify that the current log base name is correct when starting or restarting data capture.

DDL-Updated Catalog of MySQL Source Table Definitions

PowerExchange uses a catalog in a MySQL database to store MySQL source table definitions. When PowerExchange detects a DDL change of CDC interest to a source table, PowerExchange updates the source table definition in the catalog. The catalog of updated source table definitions helps PowerExchange avoid errors when reading change data for a DDL-updated table.

PowerExchange updates the source table definitions in the catalog only for DDL change events that affect CDC. Typically, these DDL changes are those that require you to re-create or modify the capture registrations and extraction maps for the source tables, for example, column add, drop, or rename operations and table drop or rename operations.

To prepare for catalog use, perform the following tasks:

1. Create the catalog tables, PWXCatTables and PWXCatUpdates, on a Linux or Windows system. The tables can be in the MySQL source database or in another local or remote MySQL database. Use the PWXCATMY utility.
Note: A single catalog can record MySQL source table definitions for multiple PowerExchange registration-group instances.
2. Verify that the catalog tables have the correct format. Use the PWXCATMY utility.
3. Check that an active capture registration exists for each source table.
4. Create a snapshot of the source table definitions and record the snapshot information in the catalog. The snapshot provides the baseline table definitions that will be updated by subsequent DDL changes. Use the PWXCATMY utility.
5. In the DBMOVER configuration file, edit the MySQL CAPI_CONNECTION statement to specify the schema of the catalog tables in the CATSCHEMA parameter. Optionally, you can also specify catalog connection parameters. These parameters will be used during change capture processing.

With the PWXCATMY utility, you can perform all of the following catalog-related tasks:

- Create the catalog tables.
- Show the DDL statements for creating the catalog tables.
- Verify that the catalog tables have the correct format.
- Create a snapshot of the source table definitions and write that information to the catalog.
- Remove, or *unregister*, source table definitions from the catalog.
- List the source table names for which table definitions exist in the catalog.
- Dump the source table definitions that are recorded in the catalog.
- Drop the catalog tables.

For more information about the PWXCATMY utility, see the *PowerExchange Utilities Guide*.

After the catalog is configured, keep in mind the following usage considerations:

- If you rename a registered table from which change data was captured and then switch back to the original table name, the second rename event is not recorded in the catalog.
- If you change the NOT NULL option of a registered source column to NULL, the source table becomes invalid and CDC processing ends. An attempt to warm start the CDC session will fail. However, if you change the NULL option to NOT NULL, the change is successfully recorded in the catalog and processing continues.
- If you replace or rename the binlogs after configuring the catalog, such that the original binlogs are no longer in use, you must remove (unregister) the source table definitions from the catalog, re-create snapshots of the source table definitions in the catalog, and then cold start the CDC sessions. For more information, see [“Changing the Binary Log Location or Base Name” on page 131](#).

MySQL CDC Operational Considerations

Review the following operational considerations for MySQL CDC:

- PowerExchange is certified with MySQL source databases that use the InnoDB storage engine. PowerExchange is not certified with the other types of storage engines that MySQL can use.

- PowerExchange does not support MySQL database deployed in a cloud environment.
- PowerExchange can capture change data from table spaces that use InnoDB tablespace encryption and from InnoDB tables that use Transparent Data Encryption (TDE).
- PowerExchange uses the MySQL mysqlbinlog utility to read change events from MySQL binary log files. Do not delete the binary log files until the PowerExchange has completed log file processing. If you delete the log files, PowerExchange might miss some change events.
- After you run CDC sessions that extract change data for MySQL sources, avoid changing the base name of the binary log files or the name or location of the directory that contains the binary log files to minimize the potential for change data loss. If you must change the binary log location or base name, see [“Changing the Binary Log Location or Base Name” on page 131](#).
- When reading the binary log file, if the binary log reader encounters a DDL event that changes a registered MySQL source table, capture processing usually ends with an error. PowerExchange recognizes and reacts to the following DDL events:

DDL	PowerExchange Reaction
TRUNCATE TABLE	If the ONTABLETRUNC parameter in the MYSQL CAPI_CONNECTION statement is set to FAIL, capture processing ends. If the ONTABLETRUNC parameter is set to WARN, PowerExchange issues a warning message and continues CDC processing with the next change record.
CREATE TABLE	If you are not using the PowerExchange Logger, capture processing continues and ignores the table. If you are using the PowerExchange Logger, the Logger continues to run if the table was not previously registered. If the table was previously registered and then re-created, the Logger ends.
DROP TABLE	Capture processing ends with an error.
RENAME TABLE	Capture processing ends with an error.
ALTER TABLE RENAME COLUMN	Capture processing ends with an error.
ALTER TABLE ADD COLUMN	Capture processing continues.
ALTER TABLE DROP COLUMN	Capture processing ends with an error.
ALTER TABLE CHANGE COLUMN	Capture processing ends with an error.

- You cannot create capture registrations for MySQL source tables that include one or more columns with a spatial datatype, such as GEOMETRY. Consequently, PowerExchange cannot capture change data from these tables.
- If MySQL source tables contain native JSON columns with binary data, PowerExchange converts the JSON binary data to a textual representation for CDC processing.

- In PowerExchange, you cannot register MySQL source tables that contain one or more columns that are defined with the any of the following character sets:

Character Set	Description
amrscii8	ARMSCII-8 Armenian
binary	Binary pseudo charset
dec8	DEC West European
eucjpm	UJIS for Windows Japanese
geostd8	GEOSTD8 Georgian
hp8	HP West European
keybcs2	DOS Kamenicky Czech-Slovak
koi8u	IOI8-U Ukrainian
swe7	7-bit Swedish

- When the mysqlbinlog utility tries to connect to a remote binary log, MySQL passes the MySQL password that is required for connection as clear text. The password is then visible to any tools that list executing processes and their startup parameters.
- When the mysqlbinlog utility reads a remote binary log, it opens a connection to the MySQL server using a specific server ID, as specified in the server-id option. PowerExchange uses the server-id of 369 to connect to MySQL for change data extraction. Ensure that no other tool or application uses the server-id of 369 to access MySQL. Also, ensure that only one CDC session is active at a time.
- PowerExchange uses the DataDirect ODBC driver for MySQL to connect to MySQL source databases. You can define the ODBC_CONN_PARAMS statement in the dbmover.cfg file If you want to add ODBC parameters to the connection strings that are used on the servers you specify for the following processing:
 - Change data capture
 - Creating, deleting, or modifying capture registrations from the PowerExchange Navigator or DBLUCBRG utility

For example, you might need to add ODBC parameters if your site policies require database connections to use SSL encryption, specific cryptographic protocols, or self-signed or third-party signed SSL certificates. For more information, see the *PowerExchange Reference Manual*.

MySQL Datatypes Supported for CDC

Verify that the MySQL columns for which you plan to capture change data have datatypes that PowerExchange supports. PowerExchange CDC supports all native MySQL datatypes except spatial types.

The following table shows the MySQL datatypes that PowerExchange supports and does not support for CDC:

Datatype	Supported for CDC?	Comments
bigint	Yes	-
binary	Yes	-
bit	Yes	-
blob	Yes	-
char	Yes	-
date	Yes	-
datetime	Yes	-
decimal	Yes	-
double	Yes	-
enum	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
float	Yes	-
int	Yes	-
json	Yes	PowerExchange truncates column data that is longer than 98,304 bytes. PowerExchange converts data in binary json columns to the internal varchar type and then to textual json data. PowerExchange uses the UTF-8 character set for json columns.
longblob	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
longtext	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
mediumblob	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
mediumint	Yes	-
mediumtext	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
set	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
smallint	Yes	-

Datatype	Supported for CDC?	Comments
spatial types	No	Spatial types include geometry, point, linestring, polygon, multipoint, multilinestring, multipolygon, and geometrycollection. Source tables that include spatial columns cannot be registered for CDC in the PowerExchange Navigator or with the DTLUCBRG utility.
text	Yes	-
time	Yes	-
timestamp	Yes	-
tinyblob	Yes	-
tinyint	Yes	-
tinytext	Yes	-
varbinary	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
varchar	Yes	PowerExchange truncates column data that is longer than 98,304 bytes.
year	Yes	-

Implementation Task Flow

To implement MySQL CDC, you need to complete some tasks in MySQL, PowerExchange, and PowerCenter. Use the following high-level task flow:

1. Prepare the MySQL sources. For more information, see [“Preparing MySQL Sources” on page 123](#).
2. Create capture registrations and extraction maps for the MySQL source tables with either the PowerExchange Navigator or the DTLUCBRG utility. For more information, see the *PowerExchange Navigator User Guide* and *PowerExchange Utilities Guide*.
3. Add a MySQL CAPI_CONNECTION statement and other required statements to the DBMOVER configuration file on the system where the PowerExchange capture registrations and extraction maps are stored. See [“Configuring the dbmover Configuration File” on page 124](#).
4. Create the PowerExchange catalog tables for recording DDL change events for source tables. For more information, see [“Creating the DDL-Updated Catalog Tables” on page 128](#).
5. Create snapshots of the source table definitions to populate the catalog. For more information, see [“Creating a Snapshot of Source Table Definitions” on page 129](#).
6. If you use the PowerExchange Logger for Linux, UNIX, and Windows, configure the PowerExchange Logger. See [“Configuring the PowerExchange Logger” on page 44](#).
7. Configure a CDC restart point. See [“Creating Restart Tokens for Extractions” on page 283](#).
8. Materialize the target tables. Use any tool of your choice. PowerExchange bulk data movement does not currently support MySQL sources.
9. Start the PowerExchange Logger, if configured. See [“Cold Starting the PowerExchange Logger” on page 70](#).

10. Create PowerCenter CDC workflows that include the MySQL sources. Use a MySQL CDC application connection. For more information, see the *PowerExchange Interfaces for PowerCenter* publication.
11. Cold start the CDC workflows.

Preparing MySQL Sources

To prepare a MySQL source system for PowerExchange CDC, you must perform a few configuration tasks.

1. Verify that your MySQL version is supported and installed on a supported Red Hat Linux or Windows operating system.
2. Verify that the MySQL mysqlbinlog utility is installed on the system where change capture processing will occur. This location can be local to or remote from the source database. The following configurations meet this requirement:
 - Run PowerExchange and the mysqlbinlog utility on the MySQL source database server.
 - Run PowerExchange and the mysqlbinlog utility on a machine that is remote from the MySQL source database server. Either specify the mysqlbinlog path in the Path environment variable or specify the mysqlbinlog full path and file name in the MYSQLBINLOG parameter in the MYSQL CAPI_CONNECTION statement in the dbmover configuration file.
3. If PowerExchange capture will run on a Linux system, configure ODBC on that system so that PowerExchange will be able to use the ODBC driver to connect to the MySQL server. Perform the following steps:
 - Set the ODBCINI and ODBCINST environment variables. Use the following export statements:

```
export ODBCINI=$PWX_HOME/ODBC7.1/odbc.ini
export ODBCINST=$PWX_HOME/ODBC7.1/odbcinst.ini
```
 - Set the shared LD_LIBRARY_PATH environment variable. Use the following statements:

```
LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:$PWX_HOME/ODBC7.1/lib;
export LD_LIBRARY_PATH
```
 - Update the ODBC driver information in the odbc.ini and odbcinst.ini files. For example, add the following statement:

```
Driver=pxw_home/ODBC7.1/lib/DWmysql27.so
```

The `pxw_home` variable represents the local path to the DataDirect installation that PowerExchange uses.
4. Enable binary logging with the options that PowerExchange requires on the MySQL source database in one of the following ways:
 - If you start the MySQL database server from the command line, enter the following command:

```
mysqld --server-id[=server_id] --log-bin[=base_name] --binlog-format[=row]
--binlog-row-image[=full]
```
 - If you start the MySQL database server as a service on Windows or as a daemon on Linux, you can specify database configuration settings in an .ini or .cnf configuration file. The default file, my.ini, is located in the MySQL installation directory. To enable binary logging, add the following lines to your MySQL configuration file:

```
[mysqld]
server-id=server_id
log-bin=base_name
binlog-format=row
binlog-row-image=full
```

Notes:

- For MySQL 5.7.x, you must use a number greater than 0 for the server-id value.
 - Informatica recommends that you include the optional log-bin parameter to specify the base name for the sequence of binary log files. To create the binary log file names, MySQL adds a numeric suffix to the base name, which is incremented each time a new binary log is created. If you do not specify a base name, MySQL uses the default base name of *host_name-bin*.
 - PowerExchange requires row-based binary logging with the row image type of full. Verify that the binlog-format parameter is set to row and that the binlog-row-image parameter is set to full. These values are the default values.
5. Create a MySQL user that PowerExchange can use to connect to the MySQL database. Use the following SQL statement:

```
CREATE USER 'pwx_user'@'%' IDENTIFIED BY 'password';
```

6. Grant the following privileges that are required for CDC to the PowerExchange user:

```
GRANT SELECT ON database_name.* TO 'pwx_user'@'%';  
GRANT REPLICATION CLIENT ON *.* TO 'pwx_user'@'%';
```

If the user needs to access binary logs on a remote MySQL server, grant the following additional privilege:

```
GRANT REPLICATION SLAVE ON database_name.* TO 'pwx_user'@'%';
```

Configuring PowerExchange for MySQL CDC

You must complete several PowerExchange configuration tasks to prepare for MySQL CDC. The tasks depend on whether you use the PowerExchange Logger for Linux, UNIX, and Windows and the type of extraction mode you plan to use.

RELATED TOPICS:

- [“MySQL CDC Overview” on page 116](#)
- [“Preparing MySQL Sources” on page 123](#)
- [“Managing MySQL CDC” on page 129](#)

Configuring the dbmover Configuration File

In the dbmover configuration file on the system where the capture registrations and CDC control files are stored, add the statements that are required for MySQL CDC. This system is specified in the **Location** field of the registration group definition.

The following statements are also required:

MYSQL CAPI CONNECTION

A named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and control CDC processing for source tables on a MySQL source database server.

CAPT_PATH

Path to the local directory on a Linux or Windows system that stores the control files for CDC.

These files include the CCT file for capture registrations, the CDEP file for application names that are used for ODBC extractions, and the CDCT file for the PowerExchange Logger for Linux, UNIX, and Windows.

CAPT_XTRA

Path to the local directory on a Linux or Windows system that stores the extraction maps for CDC.

If you plan to use continuous extraction mode, you must also include the following statement:

CAPX CAPI_CONNECTION

A named set of parameters that the CAPI uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files.

Also, Informatica recommends including the LOGPATH and TRACING statements to make finding messages easier. The LOGPATH statement defines a directory specifically for PowerExchange message log files, and the TRACING statement enables PowerExchange to create an alternative set of message log files for each PowerExchange process.

For more information about all DBMOVER statements, see the *PowerExchange Reference Manual*.

CAPI_CONNECTION - MYSQL Statement

The MYSQL 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 MySQL sources.

Operating Systems: Linux and Windows

Data Sources: MySQL

Required: Yes for MySQL CDC

Syntax:

```
CAPI_CONNECTION=(NAME=capi_connection_name
                  [,DLLTRACE=trace_id]
                  ,TYPE=(MYSQL
                        ,SERVER={database_server|localhost}
                        ,CATSCHEMA=catalog_schema_name
                        [,CATSERVER=catalog_server]
                        [,CATUSERNAME=catalog_user_name]
                        [,CATPASSWORD=encrypted_catalog_user_password]
                        [,CATPASSWORD=catalog_user_password]
                        [,MYSQLBINLOG=path/binlog_file [option1 option2 ...]]
                        [,ONDATA_TRUNC={WARN|FAIL}]
                        [,ONTABLEDDL={WARN|WARN-UNTIL-EOL|FAIL}]
                        [,ONTABLE_TRUNC={WARN|FAIL}]
                        [,RECONNTRIES={reconnection_attempts|12}]
                        [,RECONNWAIT={seconds|5}]
                        [,ROWMEMMAX=bytes]
                        [,RSTRADV=seconds]
                        [,UOWREADAHEAD={minimum_transactions|5},{maximum_transactions|10}]
                        )
                  )
```

Parameters:

NAME=capi_connection_name

Required. A unique user-defined name for this CAPI_CONNECTION statement.

Maximum length is eight alphanumeric characters.

DLLTRACE=trace_ID

Optional. The 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.

TYPE=(MYSQL, ...)

Required. The type of CAPI_CONNECTION statement. For MySQL sources, this value must be MYSQL.

SERVER={server_name|localhost}[port_number]

Required. The host name or IP address of the MySQL server where the MySQL source database runs. You can enter "localhost" if the MySQL server and PowerExchange Listener run locally on the same machine.

If you want the source server to listen on a port number other than the default port, append the port number to the server name.

CATSERVER

Optional. The name of the server that contains the DDL-updated catalog tables that store MySQL source table definitions.

Default is the MySQL source server name. You can use the default if you created the catalog tables and installed the MySQL ODBC drivers on the source server.

CATSCHEMA

Required. The name of the schema for the DDL catalog tables.

CATUSERNAME

Optional. A user name that is used to connect to the catalog server.

Default is the MySQL source user name if the catalog tables exist on the source server.

CATEPASSWORD

Optional. The encrypted password of the specified catalog user. Specify either CATEPASSWORD or CATPASSWORD but do not specify both parameters.

Default is the password of the MySQL source user if the catalog tables exist on the source server.

CATPASSWORD

Optional. The clear-text password of the specified catalog user. Specify either CATPASSWORD or CATEPASSWORD but do not specify both parameters.

Default is the password of the MySQL source user if the catalog tables exist on the source server.

MYSQLBINLOG=path/binlog_file_name [option1 option2...]

Optional. If the MySQL server is not installed on the local host, specify the path to the mysqlbinlog utility in this parameter or in the PATH environment variable. PowerExchange uses the mysqlbinlog utility to read change events from the MySQL binary log. You can optionally include any of the options that the mysqlbinlog utility supports, such as the --ssl- options. PowerExchange passes the options to the utility when the CDC session runs. For information about the utility options, see the *MySQL Reference Manual*.

If any part of the MYSQLBINLOG value includes spaces, you must enclose the entire MYSQLBINLOG value in double-quotation marks ("). Furthermore, if any part of the path\binlog_file_name value or an option includes a space, escape the value with the backslash (\) character and also enclose the value in double-quotation marks, for example:

```
MYSQLBINLOG="\"c:\bin\test dir\mysqlbinlog.exe\" --ssl-mode required "
```

If the path\binlog_file_name value begins with the hyphen (-) character, the entire value is treated as a utility option, as if no path/file_name value was specified.

Note: The minimum version of the mysqlbinlog utility that PowerExchange supports is version 3.4. Do not use an earlier version.

ONDATATRUNC={WARN|FAIL}

Optional. Indicates whether PowerExchange issues a warning message and continues processing or ends abnormally when it needs to truncate data from MySQL columns that have a mediumblob, longblob, mediumtext, longtext, enum, json, set, varbinary, or varchar datatype and are longer than 98,304 bytes.

Default is FAIL.

ONTABLEDDL={WARN|WARN-UNTIL-EOL|FAIL}

Optional. Indicates whether PowerExchange issues a warning message and continues processing or ends abnormally when it encounters a DDL record for a source that is not consistent with the source registration. Options are:

- WARN. Issue a warning message and continue capture processing.
- WARN-UNTIL-EOL. Issue a warning message and continue capture processing until the end-of-log (EOL). Then issue an error message and terminate capture processing.
- FAIL. Issue an error message and terminate capture processing.

Default is WARN-UNTIL-EOL.

ONTABLETRUNC={WARN|FAIL}

Optional. Indicates whether PowerExchange issues a warning message and continues processing or ends abnormally when it encounters a TRUNCATE TABLE record in the change stream.

Default is FAIL.

RECONNTRIES={*reconnection_attempts*|12}

Optional. The maximum number of times that PowerExchange tries to reconnect to the MySQL Server database server after detecting that the database server has shut down or the network connection to the server has been dropped. When PowerExchange and the database server run on the same machine, PowerExchange processes change records to the end of log and then checks whether the server is still running. If PowerExchange connects to a remote database server and the server shuts down or the network connection to the server is severed, the binary log reader connection might be dropped. In this case, you can use this parameter in conjunction with the RECONNWAIT parameter to improve connection resiliency.

Valid values are 0 or any positive number. A value of 0 results in no connection retries. Default is 12.

RECONNWAIT={*seconds*|5}

Optional. The number of seconds that PowerExchange waits before any attempt to reconnect to a MySQL Server database server after detecting that the server has shut down or the network connection to the server has been dropped. Use this parameter in conjunction with the RECONNTRIES parameter to improve connection resiliency.

Valid values are 0 through 3600. A value of 0 results in no waiting between connection retries. Default is 5.

ROWMEMMAX=*maximum_bytes*

Optional. The maximum amount of memory, in bytes, that PowerExchange can use to store a row change from a MySQL source table. No default value is available. If no maximum value is provided, PowerExchange does not limit the memory for a captured row change.

RSTRADV=seconds

Optional. 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.

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.

Valid values 0 through 86400. No default is provided. A value of 0 disables restart advance processing.

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.

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.

UOWREADAHEAD=minimum_transactions_in_queue,maximum_transactions_in_queue

Optional. The minimum number of transactions and the maximum number of transactions that control I/O on the UOW read-ahead queue that PowerExchange uses to prefetch change data. The UOW read-ahead queue is a dynamic queue of buffered transactions that are waiting to be read by the PowerExchange capture process. When the queue reaches the maximum number of transactions, the queue reader thread stops and sleeps until enough transactions have been read from the queue to return it to the maximum value. The number of transactions in the queue might become less than the minimum number in the following cases:

- PowerExchange has read to the end of the binary log and no additional changes are available.
- The transactions in the queue are consumed faster than the changes are read from the binary log.

For the *minimum_transactions* field, the default value is 5 and the valid range of values is 0 through 100. For the *maximum_transactions* field, the default value is 10 and the maximum is 100.

Note: The maximum number of transaction that the queue can hold is 100.

Creating the DDL-Updated Catalog Tables

Before you begin capturing changes, you must create the catalog tables, PWXCatTables and PWXCatUpdates, in the MySQL source database or in another local or remote MySQL database. You can create the catalog tables either by running DDL statements manually or by using the PWXCATMY utility.

To create the catalog tables from the PWXCATMY utility, issue the following command from the PowerExchange root directory, where the utility is installed:

```
PWXCATMY OPERATION=CREATE CATHOSTNAME=catalog_host_name CATUSERNAME=catalog_user_name
CATPASSWORD=user_password CATSCHEMA=catalog_schema
```

The utility creates the tables on the specified catalog host.

To create the catalog tables manually, first get the DDL statements for creating the tables in the proper format by performing a SHOWDDL operation in the PWXCATMY utility. Issue the following utility command:

```
PWXCATMY OPERATION=SHOWDDL
```

Then execute the DDL statements that the command returns.

Tip: To verify that the catalog tables were created in the correct format, you can use the PWXCATMY OPERATION=VERIFY command.

For more information about using the utility, see the *PowerExchange Utilities Guide*.

Creating a Snapshot of Source Table Definitions

After you create the catalog tables, create a snapshot of the source table definitions to initially populate the catalog. After you start CDC processing, PowerExchange updates the baseline table definitions in the catalog when it detects DDL change events of CDC interest from the MySQL binary log.

To create a snapshot of the source table definitions, use the PWXCATMY utility. Issue the following command for a REGISTER operation:

```
PWXCATMY OPERATION=REGISTER source_connection_parameters catalog_connection_parameters  
table_parameters
```

In the *table_parameters*, you can enter any of the following criteria to select the source tables for which to register table definitions in the catalog:

- Specific table names
- Table-name masks that include MySQL wildcards
- The source instance name, as defined in the PowerExchange registration group for the source

For more information about this command, see the *PowerExchange Utilities Guide*.

Managing MySQL CDC

After CDC is running, you might need to perform some occasional tasks to maintain and manage your MySQL CDC environment.

These tasks include:

- Stopping MySQL CDC processing.
- Changing the structure of a source table.
- Adding a capture registration.

Stopping Change Data Capture for a MySQL Source Table

You might want to stop change data capture for a MySQL source table if the table has been dropped, change activity no longer occurs on the table, or the data in the table is no longer of interest for CDC processing.

1. In PowerExchange Navigator, open the capture registration and set the **Status** option to **History**.
A capture registration that has a status of **History** cannot be activated again.
2. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger and then warm start it.

This step refreshes the registration information that the PowerExchange Logger uses.

3. In PowerCenter, delete or update CDC workflows to make sure that no workflow processes the removed table.

Stopping MySQL CDC Processing Temporarily

You might need to temporarily stop MySQL CDC processing to troubleshoot issues or to perform a maintenance task on the target database.

If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger to stop CDC processing for all source tables. Later, you can warm start the PowerExchange Logger to resume change capture processing without any change data loss. This method is preferable.

Changing the Structure of a MySQL Source Table

Occasionally, you might need to make DDL changes to a registered MySQL source table that add, alter, or drop columns from which PowerExchange captures changes. You can switch to the new table definition in a manner that preserves access to previously captured data.

This topic describes the steps for properly switching to a new table definition.

Note: You do not need to perform these steps in the following situations:

- You selectively capture change data for a subset of columns, and the DDL changes do not affect any of these columns or their ordinal values.
- You need to stop change data extraction processing for a column. In this case, remove the column from the extraction map and do not edit the capture registration. PowerExchange still captures change data for the column but does not extract it when CDC sessions run.

1. Stop data change activity (inserts, updates, and deletes) on the table.
2. Verify that any change data that was captured under the current table definition has completed extraction processing. Then stop all PowerCenter workflows that extract change data for the table.
3. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the Logger.
4. In the PowerExchange Navigator, open the original capture registration and set its status to **History**. PowerExchange does not capture change data based on capture registrations that have a status of **History** or **Inactive**.

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

5. Make the DDL changes to the table.
6. In the PowerExchange Navigator, create a new capture registration for the table that reflects the DDL changes.

Make sure that you include these settings:

- In the **Condense** list, select **Part**.
 - In the **Status** list, select **Active**.
7. If you shut down the PowerExchange Logger, warm start it.
The PowerExchange Logger begins capturing changes based on the new capture registration.
 8. Change the target table definition to reflect the source table changes, if necessary.
 9. In PowerCenter Designer, import the new extraction map for the altered source table to create a new source definition. Also, if you changed the target table, edit or re-create the target definition. Then, edit the mapping, if necessary.

10. If necessary, rematerialize the target tables and then create new restart tokens.
11. Allow change activity on the table to resume.
12. Start the PowerCenter workflows again.
Extraction processing resumes.

Changing the Binary Log Location or Base Name

After you run CDC sessions that extract change data for MySQL sources, avoid changing the base name of the binary log files or the name or location of the directory that contains the binary log files to minimize the potential for change data loss.

If you must change the binary log location, perform the following tasks:

1. Stop all CDC sessions that process data from the MySQL source.
2. Shut down the MySQL server.
3. In the log-bin option for binary logging, add the new binary log location, including the path and directory, before the base name.
4. Copy the existing binary log files and index to the new location.
5. Restart the MySQL server.
6. Warm start the CDC sessions.

If you must change the binary log base name, or if you want to change the binary log location without copying the existing binary log files to the new location, perform the following tasks:

1. Stop all CDC sessions that process data from the MySQL source.
2. Shut down the MySQL server.
3. In the log-bin option for binary logging, add the new base name or add the new location with the new or existing base name.
4. Use the PWXCATMY utility to remove, or *unregister*, the source table definitions from the catalog. Then use the utility to re-create, or *re-register*, snapshots of the source definitions in the catalog.
5. Restart the MySQL server.
6. Cold start the CDC sessions.

CHAPTER 7

Express CDC for Oracle

This chapter includes the following topics:

- [Express CDC for Oracle Overview, 132](#)
- [PowerExchange Express CDC for Oracle Benefits, 133](#)
- [PowerExchange Express CDC for Oracle Architecture, 133](#)
- [Gathering Information About the CDC Environment, 137](#)
- [Express CDC Considerations, 139](#)
- [Implementation Task Flow, 164](#)
- [Configuring Oracle for Express CDC, 165](#)
- [Configuring PowerExchange for Express CDC, 169](#)
- [Managing PowerExchange Express CDC for Oracle, 201](#)

Express CDC for Oracle Overview

PowerExchange Express CDC for Oracle captures change data directly from Oracle active and archived redo logs or from copies of the archived redo logs and makes that data available to PowerCenter CDC sessions for propagation to targets.

PowerExchange Express CDC has its own log reader to get change data directly from the redo logs for real-time extraction processing. This architecture helps reduce the impact of log reading on the performance of the Oracle source system.

PowerExchange Express CDC capture components produce a change stream in which inserts, updates, and deletes are organized by transaction, and the transactions are arranged in their original commit order.

PowerExchange Express CDC can capture change data from most types of data source environments, including Oracle RACs that use Automatic Storage Management (ASM), Oracle Data Guard logical or physical standby databases, Oracle multitenant pluggable databases, and Oracle Exadata database machines.

Also, PowerExchange Express CDC can capture change data for cloud-based database instances deployed in an Amazon Elastic Compute Cloud (EC2) or Amazon Relational Database Service (RDS) for Oracle environment.

With PowerExchange Express CDC, use of the PowerExchange Logger for Linux, UNIX, and Windows is optional but strongly recommended. The PowerExchange Logger writes successful units of work (UOWs) in chronological order by end time to PowerExchange Logger log files. CDC sessions can then extract the change data from the PowerExchange Logger log files in either continuous extraction mode or batch

extraction mode. Benefits of using the PowerExchange Logger include fewer database accesses, faster CDC restart, and no need for prolonged retention of the Oracle redo files for CDC.

To configure PowerExchange Express CDC, you must add some Oracle-specific statements to the DBMOVER configuration file and define the separate PowerExchange Express CDC configuration file, which has the default file name of `pxorad.cfg`. Also, create capture registrations, configure PowerCenter CDC sessions, and configure restart processing, as normal. The restart token format for PowerExchange Express CDC for Oracle sources is different from that for any other data source type.

You must also perform some configuration tasks in Oracle. PowerExchange Express CDC requires the Oracle database to run in ARCHIVELOG mode with minimal global supplemental logging enabled.

PowerExchange Express CDC for Oracle Benefits

Consider the benefits of PowerExchange Express CDC for Oracle when comparing it to other Oracle change capture solutions.

- PowerExchange Express CDC for Oracle processing is multithreaded to improve throughput.
- PowerExchange Express CDC for Oracle can efficiently process changes in environments that have a high volume of changes and large UOWs.
- PowerExchange Express CDC for Oracle supports RAC and ASM environments.
- PowerExchange Express CDC supports Oracle databases deployed in the cloud by using Amazon RDS for Oracle or Amazon EC2.
- PowerExchange Express CDC supports Oracle LOBs.
- You do not need to configure periodic dumps of the Oracle data dictionary. PowerExchange Express CDC for Oracle copies the Oracle data dictionary into memory at initialization.

PowerExchange Express CDC for Oracle Architecture

Example configurations demonstrate the general PowerExchange Express CDC for Oracle architecture.

The configurations include the following components:

- Oracle source tables and redo logs
- PowerExchange Express CDC capture components, including the log reader
- PowerExchange Listener
- PowerExchange Logger for Linux, UNIX, and Windows, which is optional but strongly recommended
- PowerExchange Navigator
- PowerCenter and the PowerExchange Client for PowerCenter (PWXPC)

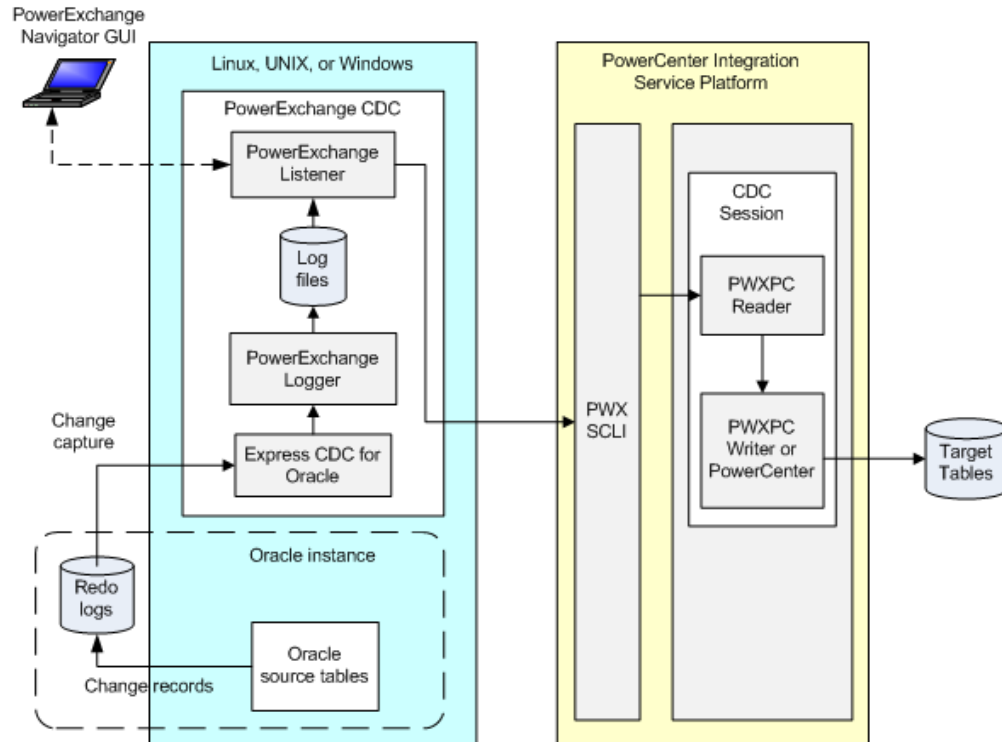
Note: PowerExchange Express CDC for Oracle does not use the PowerExchange UOW Cleanser.

Configuration 1: All Capture Components on the Oracle System

If you have a Linux, UNIX, or Windows system that has sufficient CPUs and disk space, Informatica recommends that you run the Oracle instance, Oracle Express change capture components, PowerExchange Listener, and PowerExchange Logger all on that system. The PowerExchange Logger log files reside on the

same system. This configuration avoids network-related degradation of performance and is the easiest one to configure and maintain.

The following figure shows Configuration 1:



The Express CDC log reader reads change records directly from the Oracle redo logs and forwards the committed changes to the PowerExchange Logger. The PowerExchange Logger logs the changes to its local log files. When a PowerCenter CDC session runs, change data is pulled from the PowerExchange Logger log files and sent to target tables, which are usually on a different system. Besides handling requests for change data, the PowerExchange Listener handles PowerExchange Navigator and PWXPC requests for Oracle metadata or data, registrations, and extraction maps for other functions such as database row tests.

With this configuration, the PowerExchange Express CDC system contains the PowerExchange Express CDC for Oracle configuration file, the PowerExchange Logger configuration file, and the dbmover.cfg configuration file. The dbmover.cfg file includes the ORAD CAPI_CONNECTION, CAPX CAPI_CONNECTION, ORACLEID, and ORACLE_CAPTURE_TYPE statements. The dbmover.cfg file also includes the CAPX CAPI_CONNECTION if you use continuous extraction mode.

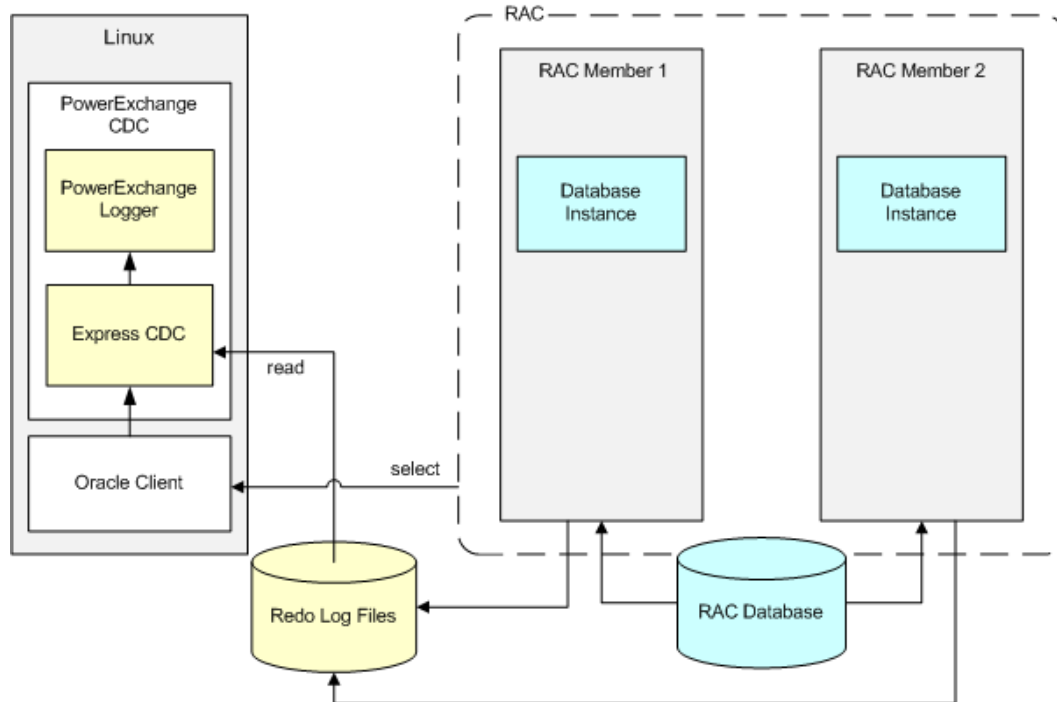
Configuration 2: PowerExchange Express CDC in a RAC Environment Without ASM

If you want to capture change data in an Oracle RAC environment, Informatica recommends that you run the PowerExchange Express CDC capture process and PowerExchange Logger on a server outside of the RAC. With this configuration, if a RAC member node fails, Express CDC continues running, and Oracle establishes a connection to another active RAC member node.

In this configuration, the Express CDC log reader still reads change records directly from the Oracle redo logs and forwards the committed changes to the PowerExchange Logger. The archived and online redo logs must exist on a shared disk that can be accessed from the server where the PowerExchange Logger runs. The PowerExchange Logger logs the changes to its local log files. When PowerCenter CDC sessions run, PWXPC extracts changes from these log files.

To read change data, the PowerExchange Express CDC log reader must run under a user ID and password that has read access to the online and archived redo logs. Also, the Oracle Client must run under a user ID and password that has been granted SELECT authority on the appropriate database objects, as described in the PowerExchange ora_orad.sql file.

The following figure shows a RAC with two member nodes and a separate Linux system with the PowerExchange Express CDC capture process and PowerExchange Logger:



Note: The PowerExchange Listener also runs on the Linux system.

In this scenario, a tnsnames.ora file resides on the Linux system. It specifies the FAILOVER option and the following connection descriptor that allows connection to either RAC member node:

```

ORATEST2=
(DESCRIPTION=
  (FAILOVER=ON)
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP) (HOST=rcln rac21.informatica.com) (PORT=1521))
    (ADDRESS=(PROTOCOL=TCP) (HOST=rcln rac22.informatica.com) (PORT=1521))
  )
  (CONNECT_DATA=
    (SERVICE_NAME=ORATEST2.informatica.com)
  )
)

```

The following PowerExchange files also reside on the Linux system:

- CCT, CDEP, and CDCT files
- PowerExchange Logger log files
- PowerExchange Express CDC for Oracle configuration file
- A dbmover.cfg configuration file that includes the ORAD CAPI_CONNECTION, CAPX CAPI_CONNECTION, ORACLEID, and ORACLE_CAPTURE_TYPE statements

In the PowerExchange Express CDC for Oracle configuration file, you must specify the following statement for CDC in a RAC:

```
RAC MEMBERS=2;
```

Note: The MEMBERS parameter specifies the maximum number of redo log threads that PowerExchange Express CDC for Oracle can track for member instances in the RAC, including open and closed threads.

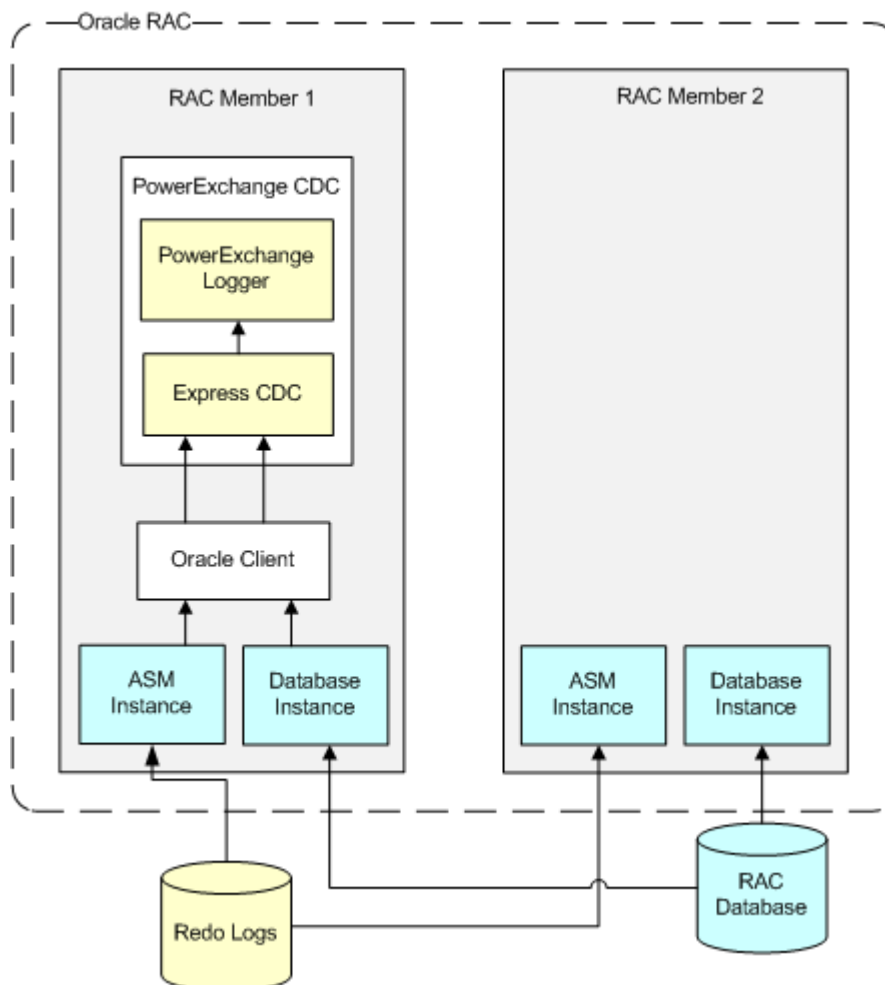
Configuration 3: PowerExchange Express CDC in a RAC Environment with ASM

If you want to capture change data in an Oracle RAC environment, Informatica recommends that you run the PowerExchange Express CDC capture process and PowerExchange Logger on a RAC member node with an ASM instance. This configuration provides the best performance.

The PowerExchange Express CDC log reader must be able to connect to the ASM instance as well as to the database to read the redo logs. A log reader that connects to an ASM instance must use an ASM login user ID that has SYSDBA or SYSASM authority. After the log reader reads the data, Express CDC forwards the committed changes to the PowerExchange Logger. The PowerExchange Logger logs the changes to its local log files. When PowerCenter CDC sessions run, PWXPC extracts changes from these log files.

In this sample configuration, the Express CDC log reader connects to an ASM instance and runs under a user ID and password that have SYSDBA authority. Because the Express log reader also connects to the Oracle database, Express CDC still requires a user ID and password that have the authorities described in the PowerExchange ora_orad.sql file.

The following image shows a RAC with two member nodes, where each node has a database instance and an ASM instance and one node has the PowerExchange Express CDC capture process and PowerExchange Logger:



In this scenario, a tnsnames.ora file on RAC member 1 can specify the following ASM connection descriptor that allows connection to the ASM instance on either RAC member node:

```
ASMTst=
  (DESCRIPTION=
    (FAILOVER=ON)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP) (HOST=rclnxxrac21.informatica.com) (PORT = 1521))
      (ADDRESS=(PROTOCOL=TCP) (HOST=rclnxxrac22.informatica.com) (PORT = 1521))
    )
    (CONNECT_DATA=
      (SERVICE_NAME=+ASM)
    )
  )
```

The following PowerExchange files reside on RAC member 1, where Express CDC and the PowerExchange Logger run:

- CCT, CDEP, and CDCT files
- PowerExchange Logger log files
- PowerExchange Express CDC for Oracle configuration file
- A dbmover.cfg configuration file that includes the ORAD CAPI_CONNECTION, CAPX CAPI_CONNECTION, ORACLEID, and ORACLE_CAPTURE_TYPE statements

In the PowerExchange Express CDC for Oracle Configuration file, you must specify the READER statement with the ASM parameters and the RAC statement:

```
RAC MEMBERS=2;
READER
  MODE=ACTIVE
  ASM_CONNECT_STRING=tns_connect_string
  ASM_EPWD=encrypted_password|ASM_PASSWORD=password
  ASM_USERID=user_id
  other optional parameters;
```

Note: Specify either ASM_EPWD or ASM_PASSWORD but not both.

Gathering Information About the CDC Environment

To prepare for implementation, gather information about your Oracle CDC environment.

Ask your Oracle DBAs the following questions:

What is the Oracle database name?

Answer:

For which Oracle tables do you need to capture change data?

Answer:

Do any of the Oracle objects that will be involved in CDC conflict with PowerExchange Express CDC for Oracle restrictions?

Answer:

Is ARCHIVELOG mode and minimal global supplemental logging enabled for the Oracle source database? If not, can they be enabled?

Answer:

Do you have read access to the redo logs in your environment?

Answer:

If you do not have the authority to read the redo logs directly, can the archived redo log files be copied to a file system from which you can access them?

Answer:

Can you create a new Oracle user with the privileges that PowerExchange Express CDC for Oracle requires? What user name do you want to use?

Answer:

Are the redo logs in ASM-managed storage? If you plan to connect to an ASM instance to read redo logs, are you allowed to create a login user ID for ASM that has SYSDBA or SYSASM authority?

You must be able to connect to an ASM instance as well as to the database to read redo logs. The ASM login user ID must have SYSDBA authority, or SYSASM authority if you set the ASM_ASSYSASM parameter to Y in the READER statement in the PowerExchange Express CDC configuration file.

Answer:

Can you make your archived redo logs available for diagnostic use by Informatica Global Customer Support if an error or anomaly occurs during CDC processing?

Answer:

Do you need to capture changes from a RAC? How many member nodes does the RAC contain, including inactive nodes?

Answer:

What is the average amount of archive log that is created per hour during peak and non-peak periods for the Oracle database?

Answer:

What is the typical size of units of work (UOWs) for the source tables?

Answer:

Does the Oracle system have the capacity to run PowerExchange Express CDC for Oracle locally?

Answer:

Do you have unkeyed source tables for which you want to capture unique rowids?

Answer:

Do you need to capture changes for NUMBER columns that have a precision greater than 28 or an undefined length?

Answer:

Do you need to capture changes for FLOAT columns that have a precision greater than 15?

Answer:

Do you need to capture change data from an Oracle Data Guard logical or physical standby database?

Answer:

Do you need to capture change data from tablespaces that use Oracle Transparent Data Encryption (TDE)? If yes, what is the TDE password?

Answer:

Do you need to capture change data from Oracle index-organized tables?

Answer:

Do you need to capture change data from a pluggable database (PDB) in an Oracle multitenant environment?

Answer:

Do you need to capture Oracle direct-path operations?

Answer:

Do you need to capture change data for a cloud-based database deployed in Amazon RDS for Oracle?

Answer:

Express CDC Considerations

Before you configure PowerExchange Express CDC for Oracle, review the restrictions, operational and performance considerations, and supported datatypes table. This information can help you successfully configure and use this CDC solution.

Also, verify that a valid Oracle environment exists for the PowerExchange user. On Linux, and UNIX, the path to the Oracle client must be specified in the PATH and library path environment variables.

RELATED TOPICS:

- [“PowerExchange Express CDC for Oracle Restrictions” on page 139](#)
- [“Operational Considerations” on page 142](#)
- [“Performance Considerations” on page 148](#)
- [“Oracle Datatypes Supported for Express CDC” on page 140](#)
- [“ASM Considerations” on page 149](#)
- [“RAC Considerations” on page 148](#)
- [“Amazon RDS for Oracle Database Instances as Sources” on page 160](#)

PowerExchange Express CDC for Oracle Restrictions

The following restrictions apply to PowerExchange Express CDC for Oracle:

- PowerExchange Express CDC for Oracle cannot capture change data for the following Oracle objects:
 - Tablespace that use any type of encryption other than Oracle Advanced Security Transparent Data Encryption (TDE)
 - Tables in a sorted hash clusters
 - Columns that are TDE-encrypted
 - Virtual columns containing derived data, which Oracle does not log in the redo logs
 - Columns that have unsupported datatypes

You cannot include these columns in capture registrations. However, PowerExchange Express CDC for Oracle can capture change data for other columns in the same registered table.

- Oracle global temporary tables
These tables cannot be registered for change capture in the PowerExchange Navigator or with the DTLUCBRG utility. PowerExchange cannot capture change data from these tables because the table data is not available in the Oracle redo logs. The data is stored in temporary tablespaces for private use and deleted when the database session ends. If you try to register a global temporary table in the PowerExchange Navigator, the Navigator does not list the table for selection. Consequently, you cannot complete the registration process. If you try to register a global temporary table with the DTLUCBRG utility, the utility produces no registration for the table.
- PowerExchange Express CDC for Oracle does not capture CREATE TABLE...AS SELECT operations.
- PowerExchange Express CDC for Oracle can capture data that the SQL*Loader utility loaded into Oracle tables provided that the load method is Insert, Append, or Replace. Do not use Truncate. If you use Truncate, the SQL*Loader issues TRUNCATE TABLE DDL. PowerExchange cannot capture row deletions that result from TRUNCATE TABLE DDL.
- If a PowerExchange Express CDC source instance has redo logs on a RAW device and the logs are *not* managed by ASM, the PowerExchange Logger for Linux, UNIX, and Windows and database row tests fail when they try to process the logs.
- In multitenant environments, PowerExchange Express can capture change data from a single pluggable database (PDB) in a container database (CDB).
- If you use Oracle 19c or later and try to capture change data from Oracle indexed-organized tables (IOTs) in 4 KB tablespaces, data loss or corruption might occur.
- If a PowerExchange Express CDC for Oracle registration contains an Oracle source table name or one or more column names that are longer than 30 characters, Oracle suppresses supplemental logging for the entire table, and PowerExchange cannot capture changes for the table. This problem is caused by an Oracle restriction. In this situation, exclude the table from capture processing or rename the long table and column names to names of 30 characters or less.

Oracle Datatypes Supported for Express CDC

Verify that columns in the Oracle source tables from which you plan to capture change data have datatypes that PowerExchange Express CDC for Oracle supports.

Oracle does not log, or does not completely log, data from columns with certain datatypes in the Oracle redo logs. Consequently, PowerExchange cannot retrieve change data for these columns.

The following table identifies the Oracle datatypes that PowerExchange Express CDC for Oracle supports and does not support:

Datatype	Supported for CDC?	Comments
BFILE	No	Column data that has this datatype is not completely logged in the Oracle redo logs and cannot be captured.
BINARY_DOUBLE	Yes	-
BINARY_FLOAT	Yes	-
CHAR	Yes	-

Datatype	Supported for CDC?	Comments
DATE	Yes	<p>The date must be in the range Jan 1, 0001 A.D. to Dec 31, 9999 A.D. in the Gregorian calendar. Dates beyond 9999 A.D. cause an error.</p> <p>Also, PowerExchange does not support negative dates, such as -0001-12-20-00-00-00. If PowerExchange receives a negative date, it substitutes an absolute value for the date and tolerates the Oracle log record that contains the date.</p>
FLOAT	Yes	<p>If you set the ORACLE_UNHANDLED_NUMASCHAR statement to Y in the dbmover.cfg file, PowerExchange registration processing for Oracle CDC source tables converts FLOAT columns that have a precision greater than 15 to VARCHAR columns. This setting can prevent data loss.</p>
LOBs	Yes	<p>Datatypes supported for Oracle LOBs are BLOB, CLOB, and NCLOB. PowerExchange can capture change data from LOB columns and non-LOB columns in the same source table, within the maximum row size limit of 8 MB.</p>
LONG	No	Columns that have this datatype cannot be included in capture registrations.
LONG RAW	No	Columns that have this datatype cannot be included in capture registrations.
NCHAR	Yes	-
NUMBER	Yes	<p>PowerExchange handles NUMBER columns as follows:</p> <ul style="list-style-type: none"> - Numbers that have a precision value less than 10 and a scale of 0 are treated as INTEGER. - Numbers that have a defined precision and scale are treated as NUMCHAR. - Numbers that have an undefined precision and scale are treated as double-precision floating point numbers by default. <p>If you set the ORACLE_UNHANDLED_NUMASCHAR statement to Y in the dbmover.cfg file, PowerExchange registration processing for Oracle CDC source tables treats NUMBER columns that have a precision greater than 28 or an undefined length as variable length strings. This setting can prevent data loss.</p>
NVARCHAR2	Yes	-
RAW	Yes	-
ROWID	Yes	-
TIMESTAMP	Yes	<p>The date must be in the range Jan 1, 0001 A.D. to Dec 31, 9999 A.D. in the Gregorian calendar. Dates beyond 9999 A.D. cause an error.</p> <p>Also, PowerExchange does not support negative dates, such as -0001-12-20-00-00-00. If PowerExchange receives a negative date, it substitutes an absolute value for the date and tolerates the Oracle log record that contains the date.</p>

Datatype	Supported for CDC?	Comments
TIMESTAMP WITH TIME ZONE	Yes	<p>PowerExchange captures data with this datatype as a UTC timestamp. The date must be in the range Jan 1, 0001 A.D. to Dec 31, 9999 A.D. in the Gregorian calendar. Dates beyond 9999 A.D. cause an error.</p> <p>Also, PowerExchange does not support negative dates, such as -0001-12-20-00-00-00. If PowerExchange receives a negative date, it substitutes an absolute value for the date and tolerates the Oracle log record that contains the date.</p> <p>Note: PowerCenter does not support this datatype. If you use PowerCenter to materialize a target table from a source table that includes this datatype, manually override the datatype in Source Analyzer with the timestamp datatype. Also, edit the generated SQL select statement that PowerCenter sends to PowerExchange to use the sys_extract_utc() function. Syntax:</p> <pre>select sys_extract_utc(tmstmpwith_tz) from schema.source_table</pre>
TIMESTAMP WITH LOCAL TIME ZONE	Yes	<p>PowerExchange captures data with this datatype as a UTC timestamp. The date must be in the range Jan 1, 0001 A.D. to Dec 31, 9999 A.D. in the Gregorian calendar. Dates beyond 9999 A.D. cause an error.</p> <p>Also, PowerExchange does not support negative dates, such as -0001-12-20-00-00-00. If PowerExchange receives a negative date, it substitutes an absolute value for the date and tolerates the Oracle log record that contains the date.</p> <p>Note: PowerCenter does not support this datatype. If you use PowerCenter to materialize a target table from a source table that includes this datatype, manually override the datatype in Source Analyzer with the timestamp datatype. Also, edit the generated SQL select statement that PowerCenter sends to PowerExchange to use the sys_extract_utc() function. Syntax:</p> <pre>select sys_extract_utc(tmstmpwith_tz) from schema.source_table</pre>
UROWID	No	-
User-defined types (UDTs)	No	-
VARCHAR2	Yes	-
XML types	No	-

Note: PowerExchange Express CDC for Oracle also does not support virtual columns and columns that have user-defined types (UDTs). Because you cannot include these columns in capture registrations, PowerExchange does not capture change data for them. However, PowerExchange can capture change data for other columns in the same registered table.

Operational Considerations

Review the following operational considerations for PowerExchange Express CDC for Oracle:

- Use of the PowerExchange Logger for Linux, UNIX, and Windows is optional. However, Informatica strongly recommends that you use it to reduce or avoid CDC impacts on your system, such as increased database I/O and the need to retain archive logs longer than normal. You can run the PowerExchange Logger in batch or continuous mode.

- The maximum length of a row for which PowerExchange can capture and process change data is 128,000 bytes.
- The PowerExchange Express CDC for Oracle capture process and the Oracle source instance that contains the redo logs can run on machines that have different operating systems and architectures. You can use any combination of supported Linux, UNIX, and Windows operating systems.

To capture changes across heterogeneous platforms, the Oracle source tables must be registered on the PowerExchange Logger system.

- If the PowerExchange Express for Oracle capture process and the Oracle redo logs are located on different operating systems, one of which is Windows and the other is Linux or UNIX, PowerExchange can translate the forward or backward slash in the path to the redo logs so that it can process the logs locally. In this situation, PowerExchange issues the following message:

```
PWX-36113 ORAD Info Mbr 1: Server log file path_filename_on_source_server will be
processed locally as local_path_filename.
```

Note: The DIRSUB setting is still honored, and the slashes in its paths are translated appropriately.

- PowerExchange Express for Oracle can read Oracle redo logs that are in ASM-managed storage, on a standard file system, or on a network file system (NFS). However, if the logs are on an NFS and the NFS buffers become stale, Express CDC might fail with error message PWX-36171 when reading the active logs. To avoid this problem, use one of the following options:
 - Configure the Oracle database to write active redo logs to ASM-managed storage.
 - Disable PowerExchange Express CDC use of active redo logs by setting the READER MODE parameter to ARCHIVEONLY or ARCHIVECOPY in the pwxorad.cfg file.
 - When you configure the NFS mount point for active redo logs, disable read buffering and attribute caching.

Note: On Linux, PowerExchange Express CDC for Oracle cannot guarantee support for the combination of Oracle direct I/O writes and the reading of active redo logs over NFS. The Linux NFS server does not honor the request to open files by using direct I/O. The combination of direct I/O writes and buffered reads might lead to a corrupted page cache. If PowerExchange Express CDC detects a corrupted page cache, Express CDC fails with error message PWX-36171. Under some circumstances, duplexing the active redo logs might allow PowerExchange Express CDC to recover from the failure. To avoid the failure completely, you might need to disable Oracle direct I/O by setting the Oracle FILESYSTEMIO_OPTIONS parameter to none.

- If you have an Oracle database that runs on HP-UX, you can run the PowerExchange Listener, PowerExchange Logger, and PowerExchange Express CDC on a Linux machine to capture change data remotely. In this special cross-platform scenario, the Oracle redo logs must be ASM-managed or NFS-mounted. Because CDC processing is remote, performance might be degraded.
- Oracle ARCHIVELOG mode and minimal global supplemental logging must be enabled. Also, the registered Oracle source table columns for which you want to capture change data must be associated with an unconditional supplemental log group, also called an ALWAYS supplemental log group. With unconditional supplemental logging, Oracle logs before images of column data to redo logs whenever any column in a row is updated. PowerExchange Express CDC for Oracle requires before images to properly process updates.
- The PowerExchange Express CDC for Oracle log reader must have access to the Oracle redo log files. If the Oracle redo log files are not stored in ASM, the operating system user ID under which PowerExchange Express CDC for Oracle log reader runs must have the authority to read the redo logs or copies of the archived redo logs. If the redo logs are stored in ASM, the user ID must have either SYSDBA authority or SYSASM authority with the ASM_ASSYSASM parameter is set to Y in the READER statement in the PowerExchange Express CDC configuration file.
- PowerExchange Express CDC for Oracles reads the active and archived redo logs by default. If you want Express CDC to read only the archived redo logs, set the MODE parameter to ARCHIVEONLY in the

Express CDC configuration file. If you do not have the authority to read the active or archived redo logs, you can configure Express CDC to read copies of the archived logs that are located on a file system. In this case, set the MODE parameter to ARCHIVECOPY and run the PowerExchange Logger in batch mode.

- If a CDC problem occurs, Informatica Global Customer Support might need to request the Oracle archived redo logs from which changes are captured for diagnostic use.
- You must use the PowerExchange Client for PowerCenter (PWXPC) to integrate with PowerCenter. The PowerExchange ODBC driver does not support PowerExchange Express CDC for Oracle.
- If you use Oracle materialized views, PowerExchange can capture change data from the master tables that underlie those views. PowerExchange supports change capture for any type of materialized view. The view and its underlying table have a one-to-one correspondence and share the same name. If you issue DTLDESCRIBE tables from the **Database Row Test** dialog box in the PowerExchange Navigator, the results include a row for the materialized view and a row for the underlying table. The **Type** column indicates which row is for the materialized view and which row is for the table.
- PowerExchange Express CDC for Oracle uses the standard PowerExchange restart and recovery processing for relational sources. The format of PowerExchange Express CDC for Oracle restart tokens is different from that for any other data source type.

The default restart point for PowerExchange Express CDC for Oracle is one of the following points in the change stream:

- For CDC sessions that run in batch extraction mode or continuous extraction mode, the beginning of the oldest PowerExchange Logger log file that is recorded in the CDCT file.
 - For CDC sessions that run in real-time extraction mode, the beginning of last log sequence that was archived.
- If you capture change data only from archived redo logs and do not use the PowerExchange Logger, the Express CDC log reader uses one of the following default start points at initialization:
 - For non-RAC instances, the low SCN of the last available archive log.
 - For RAC instances, the highest low SCN of the last log archived across all active nodes.

If you capture change data only from archived redo logs and use the PowerExchange Logger, as recommended, the default restart point after a PowerExchange Logger cold start is the end of the last available archived log, also referred to as current end-of log (EOL), unless you set the RESTART_TOKEN and SEQUENCE_TOKEN parameters in the pwxcl.cfg file. In a RAC environment, the EOL is considered to be the lowest high SCN of the last log archived across all active nodes.

- PowerExchange Express CDC for Oracle can capture data from Oracle Exadata database machines, subject to the restrictions in [“PowerExchange Express CDC for Oracle Restrictions” on page 139](#).
- For tables that do not have row movement enabled, you can use the PowerExchange-generated DTL__CAPXROWID column in extraction maps and the OPTIONS ROWID=Y statement in the PowerExchange Express CDC for Oracle configuration file to include Oracle physical rowid values in change records. This feature is useful for processing rows in unkeyed tables during CDC extraction sessions.
- PowerExchange Express CDC for Oracle can capture Oracle direct-path operations except for tables that use Oracle Exadata Hybrid Columnar Compression (EHCC). To capture direct-path operations, you must set the SUPPORT_DIRECT_PATH_OPS parameter to Y in the OPTIONS statement of the PowerExchange Express CDC for Oracle configuration file.
- PowerExchange Express CDC for Oracle can capture conventional and direct-path DML changes from tables and table partitions and subpartitions that use Oracle Advanced Compression.
- PowerExchange Express CDC for Oracle can capture conventional DML changes from tables and table partitions and subpartitions that use Oracle Exadata Hybrid Columnar Compression (EHCC). However, Express CDC does *not* capture Oracle direct-path operations for objects that use EHCC.

- PowerExchange Express CDC for Oracle can capture changes from Oracle Data Guard logical and physical standby databases. PowerExchange Express CDC supports any configuration of primary and standby databases that Oracle Data Guard supports, including RAC databases in an ASM environment. The number of nodes on the primary and standby systems do not need to match. For example, the primary system can contain a RAC with multiple member instances, and the standby system can contain a single non-RAC instance. To handle database role transitions that involve a physical standby database, you might need to update some statements in the PowerExchange Express CDC configuration file. For more information, see [“Oracle Data Guard Physical Standby Databases as Sources” on page 155](#).
- If an Oracle RESETLOGS event occurs on a source database, PowerExchange Express CDC can continue change capture processing across the RESETLOGS boundary in the archived redo logs. A RESETLOGS event occurs in situations that require you to open the database with the RESETLOGS option, such as after a flashback database operation, incomplete point-in-time recovery, or point-in-time recovery with a backup control file. A RESETLOGS event can also occur transparently in a Data Guard environment with a physical standby database after a failover or after a switchover that is preceded by an incomplete recovery and followed by an ALTER DATABASE ACTIVATE PHYSICAL STANDBY DATABASE operation. A RESETLOGS operation archives the current online redo logs, resets the log sequence number to 1, creates a new database incarnation, creates a new timestamp and SCN for the online redo logs, and updates all the current data files with the new RESETLOGS SCN.

The PowerExchange Express CDC restart token includes the resetlogs ID to identify the database incarnation to use for restart processing. At initialization, PowerExchange Express CDC uses the resetlogs ID to check whether the database has undergone a RESETLOGS event. If a RESETLOGS event occurred, PowerExchange Express CDC verifies that the restart information and the last change data that was hardened to the PowerExchange Logger log files are valid and have not been orphaned by the event. PowerExchange Express CDC then continues capture processing.

- PowerExchange Express CDC for Oracle can capture changes from Oracle index-organized tables (IOTs).
- If you use the Oracle Multitenant option, which was introduced in Oracle 12c, PowerExchange Express CDC for Oracle can capture change data from a pluggable database (PDB) within an Oracle multitenant container database (CDB). PowerExchange Express CDC can capture data from only a single PDB at a time. For more information, see [“Oracle Multitenant Pluggable Databases as Sources” on page 159](#).
- PowerExchange Express CDC can capture change from an Oracle database in an Amazon Elastic Compute Cloud (EC2) environment. The Amazon EC2 instances must run on a 64-bit Red Hat Linux server. Express CDC does not support EC2 instances on 64-bit Windows servers. The configuration of PowerExchange, the Oracle database, the PowerCenter Integration Service, and Informatica domain is flexible. All of these applications can run on EC2 instances in the cloud, or some of them can run on premise. Any combination of these applications in the cloud and on premise is supported. To capture change data, PowerExchange Express CDC requires no special configuration tasks. As usual, ensure that Express CDC can access to the Oracle archived redo logs. Also, ensure that the Express CDC user has the same privileges as are required for on-premise CDC processing.
- PowerExchange Express CDC for Oracle can capture change data from online and archived redo logs for a cloud-based database instance that is deployed in an Amazon Relational Database Service (RDS) for Oracle environment. PowerExchange reads change data from the redo logs that are located in the ARCHIVELOG_DIR and ONLINELOG_DIR directories. For more information about creating these directories and performing other required configuration tasks, see [“Amazon RDS for Oracle Database Instances as Sources” on page 160](#).

Restriction: PowerExchange Express CDC cannot capture change data that was retrieved from TDE-encrypted tablespaces in an Amazon RDS Oracle database because the Oracle wallet is not available.

- If you perform an EXCHANGE PARTITION operation on an Oracle table, PowerExchange Express CDC does not capture the exchange operation or any rows it might generate. However, Express CDC does capture subsequent DML changes on the table or partition that was the target of the exchange operation, provided that you register it for CDC.

Handling Oracle LOB Data

For Oracle source tables, PowerExchange can process change data from BLOB, CLOB, and NCLOB columns. The maximum row size that PowerExchange can process is 8 MB. For tables with BLOB or CLOB datatypes that are defined with the Oracle attribute `ENABLE STORAGE IN ROW`, PowerExchange delivers up to 3964 bytes of data to PowerCenter workflows that use a PowerExchange Oracle connection. In Oracle, with the `ENABLE STORAGE IN ROW` attribute enabled, LOB data can be either stored fully inline in the row or out-of-line in the LOB file space. Oracle does not store partial LOB data in a row.

You can use the PowerExchange-generated `DTL__ST_columnname` control column with a PowerCenter Expression transformation and unconnected Lookup transformation to retrieve all of the LOB data from BLOB and CLOB columns. The session can then send the data to a target.

Note: Oracle has a maximum column size of 4000 bytes. For LOBs stored in-row, the column contains a 20 byte LOB locator, a 16 byte LOB node and data for a maximum of 3694 bytes. For CLOB data, this data is stored as a double-byte character set, which reduces the maximum size to 1847 characters, even if the column character set is single byte.

Using Expression and Lookup Transformations to Retrieve CLOB Data That is Not Fully Stored Inline

If you need to extract CLOB data that is stored fully or partially in an auxiliary table space, use an Expression transformation and an unconnected Lookup transformation in the PowerCenter CDC workflow. Also, use the PowerExchange-generated `DTL__ST_columnname` column in the extraction map.

Verifying That the `DTL__ST` Column Is in the Extraction Map

The `DTL__ST` column for a source CLOB column indicates whether or not PowerExchange was able to extract all of the data from the CLOB column so that the entire CLOB value is available in the change stream. If the `DTL__ST` column has the value `C` (complete), PowerExchange was able to extract all of the CLOB data because the data is stored fully inline in the base table space or is null. PowerExchange can then supply the entire CLOB value or null to a PowerCenter workflow that applies the data directly to the target. If the `DTL__ST` column has the value `I` (incomplete), PowerExchange was *not* able to extract all of the CLOB data because the CLOB data is stored fully or partially in the auxiliary table space. In this case, you must use a Lookup transformation in the PowerCenter workflow to retrieve all of the current CLOB data from the source table. CLOB data is stored in an auxiliary table space for either of the following reasons:

- The CLOB column was *not* defined with the `INLINE LENGTH(bytes)` clause. All of the CLOB data is stored in the auxiliary table space.
- The CLOB column was defined with the `INLINE LENGTH(bytes)` clause and the amount of CLOB data exceeds the inline length. In this case, the excess data is stored in the auxiliary table space.

By default, the `DTL__ST` column is selected for inclusion in extraction maps. To verify that the `DTL__ST` column is selected in the extraction map that you plan to import into PowerCenter to create the Source Qualifier, open the extraction map in the PowerExchange Navigator. Then right-click in the **Extraction Definition** window and select **Show Auto Generated Columns**.

Creating a PowerCenter CDC Workflow That Includes Expression and Lookup Transformations

When you create a PowerCenter workflow that processes incomplete CLOB data from a PowerExchange source, perform the following high-level steps:

1. In Source Analyzer, import the PowerExchange extraction map for the source to create the CDC source definition. You must select **CDC Datamaps** to list the extraction maps. This source definition will be used as the Source Qualifier in the mapping.

2. In Source Analyzer, import another source definition from PowerExchange but do not select **CDC Datamaps**. PowerExchange gets the source definition from the Oracle catalog for the Oracle database instance that you specify. This source definition will be used as the lookup source.
3. In Target Designer, create the target definition.
4. In PowerCenter Designer, add the CDC source definition and target definition to a mapping.
5. From the **Transformation** menu, create the Expression transformation and unconnected Lookup transformation. Then add the transformations to the mapping. Connect the Source Qualifier output ports to the Expression input ports, and connect the Expression output ports to the Target Definition input ports.

When you create the Expression transformation, use the following guidelines:

- Drag the required Source Qualifier columns to the Expression so that the ports are created with the correct attributes.
- On the **Expression Ports** tab, perform the following actions:
 - Rename the CLOB column port ("LOBVALUE" in the example) to *column_name_IN* (LOBVALUE_IN). Then clear the **O** check box for this port. Retain the **I** check box selection.
 - Add a *column_name_OUT* port ("LOBVALUE_OUT" in the example). Then clear the **I** check box for this port. Retain the **O** check box selection.
 - In the **Expression** field for *column_name_OUT* port ("LOBVALUE_OUT"), enter an expression that calls the lookup LKP() function when the DTL__ST_columnname value is I (incomplete) and the SQL change is not a DELETE. For example:

```
IIF( (DTL__ST_LOBVALUE='I') AND (DTL__CAPXACTION != 'D'),
:LKP.LKP_DSNOB_Q1CLOB_INLINE(KEYCOL), LOBVALUE_IN)
```

When you create the Lookup transformation, use the following guidelines:

- If the source table contains multiple LOB columns, create a separate Lookup transformation for each one.
- When you create the Lookup transformation, select **Source** in the **Location of the Look-up Table** field. Then make sure that you select the Oracle source, not the CDC source definition.
- In the Lookup object, delete all columns except those that are used for input or returned values. The lookup function accepts only a single return column for the CLOB.

Tip: Informatica recommends that you perform lookups with primary key columns instead of a ROWID column, unless you are sure that the ROWID column will not change after a REORG.
- In the **Edit Transformations** dialog box for the Lookup transformation, perform the following actions on the **Transformation Ports** tab:
 - Create an input key column port ("KEYCOL_IN" in the example) that has the same attributes as the table key column. Select the **I** check box.
 - For the key column port ("KEYCOL"), select the **L** check box so that it is used in the SQL statements passed to the Oracle lookup. Clear the **O** check box.
 - For the CLOB column port ("LOBVALUE"), select the **O**, **L**, and **R** check boxes. The **L** setting causes the column to be used in the SQL passed to the Oracle lookup. The **O** setting causes column data to be returned from Oracle. The **R** setting causes the column data that is returned from the lookup to be passed to the Expression transformation.
- On the **Condition** tab, define a lookup condition that specifies the key column in the lookup table ("KEYCOL") equals the input key column port (KEYCOL_IN).

For more information about Expression and Lookup transformations, see the *PowerCenter Transformation Guide*.

Performance Considerations

The following considerations pertain to PowerExchange Express CDC for Oracle performance:

- If CDC performance becomes degraded when the configured MEMOPS are exhausted and PowerExchange Express CDC for Oracle must create spill files to free up resources, perform the following steps:
 1. Ensure that the storage used for spill files is not the root cause of the performance degradation. Performance might be severely degraded if slow network storage is used for the spill files.
 2. Modify the following parameters in the OPTIONS statement:
 - SPILLMAX=2097151. Using fewer spill files is more efficient.
 - SPILLFILEBUFFSZ=1024. Increase the file buffering for spill files to decrease physical I/O.
 3. If the previous steps do not improve performance, consider increasing the MEMOPS parameter value in the OPTIONS statement of the PowerExchange Express CDC for Oracle configuration file to a level that is sufficient to process the change data volume in memory.
- If you use continuous extraction mode, minimize the size of the CDCT file. The CDCT file contains information about PowerExchange Logger log files. PowerExchange reads the CDCT file each time the interval that is specified in the FILEWAIT parameter of the CAPX CAPI_CONNECTION statement elapses. If a CDCT file is large, PowerExchange read operations can result in a high level of I/O activity, increased use of system resources, and increased extraction latency. To manage the CDCT file size, use the COND_CDCT_RET_P statement in the pwxcl.cfg configuration file for the PowerExchange Logger.

RAC Considerations

PowerExchange Express CDC for Oracle can capture changes from online and archived redo log files in an Oracle RAC environment.

The PowerExchange Express CDC capture process and PowerExchange Logger for Linux, UNIX, and Windows can run on a RAC member instance or on a server outside of the RAC. In both cases, PowerExchange must have read access to the Oracle online and archived redo logs. For greater resiliency, run the Express CDC capture process and PowerExchange Logger on a server outside of the RAC. With this configuration, if the RAC member node to which PowerExchange is connected fails, Express CDC continues running and tries to connect again. Oracle directs the connection request to another active RAC member that is referenced by an entry in the an Oracle tnsnames.ora file.

To configure PowerExchange for CDC in a RAC, you must define the RAC statement with the MEMBERS parameter in the PowerExchange Express CDC for Oracle configuration file. For the MEMBERS value, enter the maximum number of redo log threads that PowerExchange Express CDC can track for member instances in the RAC, including open and closed threads.

If you do not use ASM and you run the PowerExchange Express CDC capture process and PowerExchange Logger on a server outside of the RAC, you might also need to define the DIRSUB statement. Define the DIRSUB statement if the server where PowerExchange Express capture process runs uses a mount point to the directory with the online and archived redo logs that is different from the mount point that the RAC Oracle instance to which PowerExchange connects uses.

If you use the FAILOVER feature of Oracle, you can define a single entry in the tnsnames.ora file that covers multiple RAC member instances. Informatica recommends that you use the FAILOVER feature to prevent PowerExchange Express CDC for Oracle from failing if a RAC member node stops running. The following sample entry in the tnsnames.ora file has the FAILOVER option enabled and covers two RAC nodes:

```
ORA1A=
  (DESCRIPTION=
    (FAILOVER=ON)
    (ADDRESS_LIST=
      (ADDRESS=(PROTOCOL=TCP) (HOST=node1.informatica.com) (PORT=1521))
```

```

        (ADDRESS=(PROTOCOL=TCP) (HOST=node2.informatica.com) (PORT=1521))
    )
    (CONNECT_DATA=
        (SERVICE_NAME=ORA1A.informatica.com)
    )
)

```

If you use ASM, use the following equivalent entry in the tnsnames.ora file:

```

ASMAAny=
(DESCRIPTION=
  (FAILOVER=ON)
  (ADDRESS_LIST=
    (ADDRESS=(PROTOCOL=TCP) (HOST=node1.informatica.com) (PORT = 1521))
    (ADDRESS=(PROTOCOL=TCP) (HOST=node2.informatica.com) (PORT = 1521))
  )
  (CONNECT_DATA=
    (SERVICE_NAME=+ASM)
  )
)

```

Note: SERVICE_NAME must specify a net service name of +ASM.

Also, PowerExchange Express CDC assumes that, for all of the open and closed threads, each is associated with a RAC node. If the number of open and closed threads does not match the Oracle CLUSTER_DATABASE_INSTANCES parameter value, Express CDC issues the following error message:

```

PWX-36127 ORAD: Enabled thread count does not match instance count. Enabled threads =
<number_of_threads> : Cluster Instances = <number_of_instances>.

```

If you receive this error, disable the redundant threads by using the following SQL statement:

```

ALTER DATABASE DISABLE THREAD number_of_redundant_threads

```

ASM Considerations

PowerExchange Express CDC for Oracle can capture changes from active and archive redo log files that are managed by Oracle Automatic Storage Management (ASM).

Configure the PowerExchange Express CDC for Oracle reader to connect to ASM instance to read the redo logs. PowerExchange establishes connections to both the ASM instance and Oracle database. The PowerExchange Express CDC log reader connects to the ASM instance based on the ASM_CONNECT_STRING, ASM_USERID, ASM_EPWD or ASM_PASSWORD, and ASM_ASSYSASM parameters that you specify in the READER statement. In the ASM_USERID parameter, specify a user ID that has SYSDBA authority or SYSASM authority. To use SYSASM authority, you must set the ASM_ASSYSASM parameter to Y in the READER statement in the PowerExchange Express CDC configuration file. Because of the required authority level, usually the ASM user ID is different from the following user IDs:

- If you extract data to PowerExchange Logger for Linux, UNIX, and Windows log files, as recommended:
 - The USERID in the ORAD CAPI_CONNECTION statement in the dbmover.cfg file.
 - The CAPTURE_NODE_UID in the PowerExchange Logger configuration file.
- If you extract data directly to PowerCenter:
 - The **User Name** attribute on the PWX CDC Real Time application connection for the PowerCenter CDC session.

To improve CDC performance, you can run the Express CDC capture process and the PowerExchange Logger on the same node as the ASM instance. Alternatively, you can configure PowerExchange Express CDC to write chunks of redo log to a staging file on the ASM instance. The staging file can then be read by using standard I/O communication. For more information, see [“Using a Staging File to Optimize CDC Performance in an ASM Environment ” on page 150.](#)

Use the interprocess communication (IPC) protocol for the PowerExchange Express CDC connection to the ASM instance. Ask your Oracle DBA to configure the Oracle listener.ora and tnsnames.ora files for IPC connectivity.

Using a Staging File to Optimize CDC Performance in an ASM Environment

If the Oracle active and archived redo logs from which you capture change data are in an ASM environment, you can optionally write chunks of Oracle redo log to a staging file. This feature improves CDC performance, significantly increases data throughput, and reduces ASM CPU usage, without you having to implement dual logging of redo logs outside of ASM.

PowerExchange Express CDC uses one staging file for each active Oracle redo log thread.

To enable the use of staging files, perform the following actions:

1. Create a staging directory on the machine where the ASM instance to which PowerExchange Express CDC connects resides.
2. Configure the ASMSTAGING statement in the PowerExchange Express CDC for Oracle configuration file, `pwxorad.cfg`, to point to the staging directory.

At CDC session runtime, if the available redo log data is less than 32 KB, PowerExchange Express CDC reads the data directly from the logs over the network. If the available redo log data is greater than 32 KB, PowerExchange Express CDC makes a call to ASM to write chunks of redo log to a staging file. The maximum chunk size is determined by the TARGETSIZE parameter in the ASMSTAGING statement. ASM generates the staging file in the staging directory using the PowerExchange file-naming convention `PWX_STG_dbid_thread`, for example, `PWX_STG_1497849897_0`. PowerExchange Express CDC can then use standard I/O communication to read change data from the staging file. Reading data from a staging file in a single operation is much more efficient than having to perform many SQL*NET read operations on the logs to extract the same amount of data from ASM.

At shutdown, cleanup processing deletes staging files from the staging directory.

If you plan to use staging files, review the following usage considerations:

- If PowerExchange Express CDC is configured to read data remotely from staging files, ensure that the staging directory is in NFS or other shared storage to which PowerExchange has read access.
- If PowerExchange Express CDC can read 32 KB of available redo log data in a single call to ASM, it will directly read the data from the logs instead of reading the data from a staging file, even if staging file use is configured.
- PowerExchange Express CDC does not read past the last known block that Oracle committed. New data is not read until the database is polled for its current state, which can result in a read delay. To control the polling interval, use the STATUSCHECKINTERVAL parameter in the READER statement.
- The PowerExchange Express CDC user requires read permissions on the staging file at minimum. Also, Informatica recommends providing the PowerExchange Express CDC user with write permissions on the staging file to enable Express CDC to perform cleanup processing and file resizing. If you do not grant write permissions, consider pre-allocating the staging file at the size specified by the TARGETSIZE parameter in the ASMSTAGING statement. Alternatively, you can add the Express CDC user to the Oracle DBA group.
- If PowerExchange Express CDC cannot access the staging file, it will issue a warning and fall back to regular ASM reads of the redo logs. For the first 5 log switches thereafter, PowerExchange Express CDC retries accessing the staging file. If all attempts fail, PowerExchange Express CDC permanently disables reading data from the staging file. In a RAC, this behavior pertains separately to each RAC member. If the workload is not balanced across RAC members, the staging file could be used for one member but not for other members.

- If you use ASM with a RAC, ensure that the amount of clustered storage specified in the TARGETSIZE parameter in the ASMSTAGING statement is available to each RAC member.

Checkpointing In-flight Transactions

Checkpointing periodically persists the state of in-flight transactions to disk or tables.

If a PowerExchange Express capture process stops and is then warm started, Express CDC uses the checkpoint information to resume the processing of outstanding transactions from the last recorded checkpoint. By using a checkpoint, Express CDC avoids time-consuming reprocessing of large amounts of redo log to rebuild the state of all transactions that were active when the process stopped.

You can store state information for in-flight transactions in file-system files or in three Oracle tables. For file-based storage, ensure that the user ID under which Express CDC runs is granted the appropriate permissions to create the files. For table-based storage, ensure that the database user has the appropriate permissions to create and update the tables. You can manually create the tables or allow Express CDC to automatically create them. On first initialization of a CDC process after checkpointing is enabled, Express CDC automatically creates the tables if they do not already exist. If you need to manually create the tables, you can use the DDL statements that documented here: [“DDL Statements for Manually Creating State Tables for Checkpointing” on page 152](#).

To configure checkpointing, you must specify the CHECKPOINT statement and STATESTORAGE statement in the PowerExchange Express CDC configuration file. If you want to encrypt the spill files that contain checkpointed data, you can optionally set the SPILLENCRYPTPASS or SPILLENCRYPTEPASS parameter in the OPTIONS statement.

Before you begin configuring checkpointing, review the following usage considerations, which might affect your configuration choices:

- Checkpointing occurs only for in-flight translations. Transactions cannot be checkpointed if they contain incomplete operations.
- DPL transactions are not checkpointed.
- All completed operations in a transaction must be spilled to disk before they can be checkpointed.
- Checkpointing is incompatible with and replaces the AGEOUT parameter in the OPTIONS statement. Unlike the AGEOUT parameter, checkpointing guarantees lossless restarts.
- Transactions older than the age specified by the COMPLETEDEVENTSAGE parameter can be forced to spill to disk to facilitate checkpointing.

Note: Transaction age is calculated as the difference between the transaction start time and the time of the last event that the transaction manager processed.

- You can disable the force spilling of very large transactions by using the MAXCHKPTSIZE parameter in the CHECKPOINT statement. This strategy might help reduce resource usage during large batch cycles.

Note: If resources become restricted, the transactions might still be spilled to disk, in which case they can be checkpointed then.

- Checkpoints are not written if they do not progress the restart point. A transaction that has data on stack will block checkpointing until the stack is cleared.
- Checkpoints are not taken until the PowerExchange Consumer API (CAPI) progresses the restart point beyond the checkpoint restart point. On an idle system, Informatica recommends that you set the COMPLETEDEVENTSAGE parameter in the CHECKPOINT statement to the same value as in the RSTRADV parameter in OPTIONS statement.
- To avoid the overhead of maintaining LOB data in the database, Express CDC always copies spill files to disk during checkpoint processing.

- If you configure frequent checkpointing or decrease the COMPLETEDEVENTSAGE value, performance might be degraded and the storage required for spill files might increase.
- PowerExchange Express CDC notifies the PowerExchange Logger when a checkpoint event occurs. The next read request that Express CDC receives serves as an acknowledgement from the PowerExchange Logger and triggers cleanup processing of expired checkpoint data.
- When a restart occurs, the checkpoint data is loaded and validated. The most recent, valid checkpoint with a restart point that is less than or equal to the PowerExchange Logger restart point is used to rebuild the state of all outstanding transactions at the time of that checkpoint. The actual restart point is substituted with the highest SCN of all transactions checkpointed.
- If a suitable, valid checkpoint is not available for a restart, the restart point reverts to the restart point of the PowerExchange Logger.
- When you cold start the PowerExchange Logger, the checkpoint data is cleared.
- When determining whether to encrypt the spill files, consider that checkpointing tends to increase the number and retention of spill files.

DDL Statements for Manually Creating State Tables for Checkpointing

Checkpointing requires three tables for state storage of transaction and checkpointing information if you are not using file-based storage.

On the first initialization of CDC after checkpointing is enabled, PowerExchange Express CDC for Oracle automatically creates the tables if they do not already exist. The database user must have the appropriate permissions for creating tables. If you need to manually create the tables, you must have the minimum permissions of SELECT and UPDATE on tables.

To manually create the tables with the correct schemas, use the following DDL statements:

```
CREATE TABLE [tableowner].[prefix]_CHECKPOINTBASE"
(
    "CHECKPOINTNUM" NUMBER(19,0) NOT NULL ENABLE,
    "CHECKPOINTNUMSEQ" NUMBER(10,0) NOT NULL ENABLE,
    "CHECKPOINTDATA" VARCHAR2(4000 BYTE) NOT NULL ENABLE,
    CONSTRAINT "ORCL_CHECKPOINTBASE_PK" PRIMARY KEY ("CHECKPOINTNUM",
"CHECKPOINTNUMSEQ")
);
CREATE TABLE [tableowner].[prefix]_CHECKPOINTUOWS
(
    "CHECKPOINTNUM" NUMBER(19,0) NOT NULL ENABLE,
    "STARTSCN" NUMBER(19,0) NOT NULL ENABLE,
    "XID" NUMBER(19,0) NOT NULL ENABLE,
    "CHECKPOINTNUMSEQ" NUMBER(10,0) NOT NULL ENABLE,
    "CHECKPOINTDATA" VARCHAR2(4000 BYTE) NOT NULL ENABLE,
    CONSTRAINT "ORCL_CHECKPOINTUOWS_PK" PRIMARY KEY ("CHECKPOINTNUM", "STARTSCN",
"XID", "CHECKPOINTNUMSEQ")
);
CREATE TABLE [tableowner].[prefix]_CHECKPOINTUOWSPILL
(
    "CHECKPOINTNUM" NUMBER(19,0) NOT NULL ENABLE,
    "STARTSCN" NUMBER(19,0) NOT NULL ENABLE,
    "XID" NUMBER(19,0) NOT NULL ENABLE,
    "FILEID" NUMBER(5,0) NOT NULL ENABLE,
    "CHECKPOINTNUMSEQ" NUMBER(10,0) NOT NULL ENABLE,
    "CHECKPOINTDATA" VARCHAR2(4000 BYTE) NOT NULL ENABLE,
    CONSTRAINT "ORCL_CHECKPOINTUOWSPILL_PK" PRIMARY KEY ("CHECKPOINTNUM", "STARTSCN",
"XID", "FILEID", "CHECKPOINTNUMSEQ")
);
```

In these statements:

- *tableowner* represents the value of the optional TABLEOWNER parameter in the STATESTORAGE statement in the Express CDC configuration file.
- *prefix* represents the ORACOLL parameter value in the ORAD CAPI_CONNECTION statement, unless you override that value by specifying the optional PREFIX parameter in the STATESTORAGE statement.

Dynamic Dictionary

PowerExchange Express CDC for Oracle supports a dynamic dictionary based on a snapshot of the Oracle catalog to accommodate DDL changes to source tables and table partitions for restart processing. With the dynamic dictionary, Express CDC can update dictionary information in memory for the source tables of CDC interest when DDL changes are committed. The changes, along with the SCN and sequence number, are persisted to an external stage table or file that records all changes of interest.

Use a dynamic dictionary in environments where DDL changes to source tables are likely to occur. If you use a static dictionary in this type of environment, data loss or corruption can occur after a CDC restart if tables were reorganized or renamed, tables or partitions were dropped, or table columns were added, dropped, or altered. By using a dynamic dictionary, you can avoid data loss and corruption after a restart.

To use a dynamic dictionary, perform the following high-level implementation steps:

1. Perform configuration tasks.
2. Ensure that an initial snapshot of the Oracle catalog populates the state table or file for the dynamic dictionary. The initial snapshot is automatically created when PowerExchange Express CDC first initializes after dynamic dictionary configuration.
3. Warm start PowerExchange Express CDC or cold start the PowerExchange Logger. PowerExchange Express CDC loads the in-memory dictionary from the external state storage during initialization of the warm start or cold start.

For more information, see the following sections.

Configuration Tasks

To use the dynamic dictionary, you must define both the `DICTIONARY` and `STATESTORAGE` statements in the PowerExchange Express CDC configuration file.

In the `DICTIONARY` statement, set the `MODE` parameter to `DYNAMIC`.

In the `STATESTORAGE` statement, define the storage that persists the dynamic dictionary information for each source table of CDC interest. You can configure Express CDC to store this information in a state table in the source database or in CCATALOG files in a file system directory. You can also specify a database user in the `STATESTORAGE` statement if the user requires permissions that are different from the user configured for normal access.

If you use file-based storage, the PowerExchange Express CDC user must have the appropriate permissions in the file system to create subdirectories and files. Ensure that the Express CDC user has read, write, and execute permissions on the base directory. The first time PowerExchange Express CDC starts, it creates one or more subdirectories under the base directory.

If you use a table to store dictionary state information, you can create the table in one of the following ways:

- Let PowerExchange Express CDC create the table. The first time Express CDC initializes after you enable dynamic dictionary use, it attempts to create the state table if the table does not already exist. If the Express CDC user needs permissions different from the user configured for normal access, specify a database user in the `STATESTORAGE` statement that has database permissions that allow creation of the state table on the Oracle source system.
- Manually create the table by using the sample DDL statements that are provided in [“DDL Statements for Manually Creating a State Table for the Dynamic Dictionary” on page 155](#). Ensure that the database user has `SELECT` and `UPDATE` permissions on the table.

Initial Snapshot

After dynamic dictionary use is configured, the first time PowerExchange Express CDC initializes, it checks for the state table or file directory. If the table or directory does not exist, Express CDC attempts to create it. If the Express CDC user does not have create table permissions, initialization will fail.

After the table or file directory exists, Express CDC takes an in-memory snapshot of the Oracle catalog and externalizes it to populate the state table or file, thereby creating an initial snapshot.

Incremental Snapshot

After the dynamic dictionary is in use, any tables added to the CCT interest list by means of registration or a rename-in operation, trigger an incremental snapshot during Express CDC initialization. Express CDC takes an in-memory snapshot of the current catalog using the new interest list and compares the snapshot to the old interest list. Any tables that do not match in the old interest list are provided to the dynamic dictionary as "futures". Operations in the log for the tables marked "futures" will get filtered out until the commit SCN of the transaction that contains them matches or exceeds the current SCN at the time the tables were added to the dictionary.

Warm Start Processing

For a warm start of CDC processing, PowerExchange Express CDC uses the SCN in the data dictionary to calculate how far back in the logs to start CDC processing so that no DDL changes for tables of interest are missed.

A warm start Instantiates an in-memory catalog from the state table or file instead of from the Oracle catalog. However, Express CDC still references the Oracle catalog to verify the supplemental logging status for all tables of CDC interest and to check if new tables have been added to the interest list. If tables have been added, an incremental snapshot is taken.

Cold Start Processing

When the PowerExchange Logger is cold started and the dynamic dictionary last restart SCN is less than the PowerExchange Logger restart SCN, the PowerExchange Logger uses the dynamic dictionary restart SCN to capture changes so that the dictionary does not contain gaps.

You can modify this behavior by setting `SNAPSHOTONCOLDSTART=Y` in the `DICTIONARY` statement. This setting causes a new snapshot of the current Oracle catalog to be taken, replacing the existing information in the dynamic dictionary.

Usage Considerations

- The structure of each source table of CDC interest must be consistent with its capture registration.
- If a source table that is no longer of CDC interest is removed from the CCT interest list by deleting its registration or by editing the Status or Condense option in its registration, the dynamic dictionary continues to track changes for it. If the table is then added back to the interest list, PowerExchange Express CDC does not need to take another snapshot of the table.
- Dynamic dictionary handles the following types of operations, which are not supported by a static dictionary:
 - *Online reorganization of tables.* Express CDC supports online reorg operations on tables of CDC interest without data loss or requiring a restart.
 - *Rename in.* If a table that is not in the dynamic dictionary is renamed to a name that matches the CCT interest list of source objects, Express CDC queries the Oracle catalog to get the table structure and persists that information to state storage. Express CDC then builds a new filter to include the table and compares it to the previous filter. If the table is marked as "unwanted" in the previous filter, CDC shuts down.
 - *Rename out.* If a table in the dynamic dictionary is renamed such that it no longer matches any entry in the CCT interest list, Express CDC stops capturing data for the table but continues processing.
 - *Rollup.* Dynamic dictionary automatically rolls up the base of the dictionary during initialization and once every 24 hours thereafter. All DDL operations for a table are merged into the base snapshot for that table, retaining only the amount of object history allowed by the retention period. During the rollup,

information about dropped tables and partitions is deleted from the dictionary. If the retention period is too short, information might be lost for these tables.

- *Table or partition drop.* A drop of a table or partition of CDC interest is recorded in the storage container with the commit SCN at the time the drop operation occurred. Since a warm start will rebuild the dictionary to the point-in-time of the restart SCN, the dictionary will still be aware of the object, even if the restart SCN is prior to the drop and no data will be lost.
- PowerExchange Express CDC issues a warning if a table in the CCT interest list has no entry in the Oracle catalog. If the table is subsequently created or renamed in, it is marked as of CDC interest and included in the table filter. For tables that are renamed in, Express CDC queries the Oracle catalog for the table structure and persists that information to state storage. The new filter is compared to the previous filter. If the table is marked as unwanted for CDC in the old filter, CDC shuts down to avoid potential data loss. In this case, you must warm start Express CDC to refresh the log parser filter so that it does not discard data that is now of CDC interest.
- If the PowerExchange Logger is cold started and the last SCN written to the dictionary is less than the Logger restart SCN, the dictionary SCN is used for the cold start. This behavior ensures that no operations are missing from the dictionary. To modify this behavior, you can set `SNAPSHOTONCOLDSTART=Y` to force a new snapshot of the current Oracle catalog to be taken, which replaces the existing dictionary contents.

DDL Statements for Manually Creating a State Table for the Dynamic Dictionary

The dynamic dictionary requires a table to store state information such as SCN and sequence number, for each source table of CDC interest, if you are not using file-based storage or do not want the table to be automatically created.

On the first initialization of CDC after dynamic dictionary use is enabled, PowerExchange Express CDC for Oracle automatically creates the table, called `CCATALOG`, if it does not already exist. The database user that is specified in the `STATESTORAGE` statement must have the appropriate permissions for creating tables.

Alternatively, you can create the table manually either by submitting DDL statements. If you manually create the table, the database user must have the minimum permissions of `SELECT` and `UPDATE` on the `CCATALOG` state table.

To manually create the table with the correct schema, use the following DDL statements:

```
CREATE TABLE [table_owner].[prefix]_CCATALOG
(
    "SCN" NUMBER(19,0) NOT NULL ENABLE,
    "ROWTYPE" NUMBER(3,0) NOT NULL ENABLE,
    "SEQ" NUMBER(10,0) NOT NULL ENABLE,
    "OP" VARCHAR2(4000 BYTE) NOT NULL ENABLE,
    PRIMARY KEY ("SCN", "ROWTYPE", "SEQ")
);
```

In these statements, the variables are:

- *table_owner* represents the value of the optional `TABLEOWNER` parameter in the `STATESTORAGE` statement in the Express CDC configuration file.
- *prefix* represents the `ORACOLL` parameter value in the `ORAD CAPI_CONNECTION` statement, unless you override that value by specifying the optional `PREFIX` parameter in the `STATESTORAGE` statement.

Oracle Data Guard Physical Standby Databases as Sources

PowerExchange Express CDC for Oracle can capture change data from Oracle Data Guard physical standby databases. PowerExchange monitors the standby and archived redo logs and the database SCN on the

standby system. As long as the database SCN is progressing, PowerExchange captures change data from the logs.

A PowerExchange CDC environment with a physical standby database source has the following characteristics:

- Informatica recommends that you install and run PowerExchange on the machine with the physical standby database. A PowerExchange installation is not required on the primary database machine.
- The physical standby database can use real-time apply with standby redo logs or apply data directly from the archived logs only.
- The physical standby database can be open with read-only access, or it can be not open, such as when started with the mount option.
- PowerExchange Express CDC supports any configuration of primary and standby databases that Oracle Data Guard supports, including databases in RACs that use ASM. The number of nodes on the primary and standby systems do not need to match.

To configure change capture from a physical standby database, you must complete the following configuration tasks:

- In the PowerExchange Navigator client, create a registration group and capture registrations for the Oracle source instance and tables on the primary system from which the changes originate.
- In the dbmover.cfg configuration file on the standby system, configure the following statements to point to the Oracle instance on the primary system:

- The ORACOLL parameter in the ORAD CAPI_CONNECTION statement
- The DFLTINST parameter in the CAPX CAPI_CONNECTION statement
- The first positional parameter, *collection_id*, in the ORACLEID statement

Important: In the ORACLEID statement, do *not* specify the fourth positional parameter *capture_connect_string*. This information is provided by the DATABASE or STANDBY statement in the pwxorad.cfg file.

- In the Express CDC pwxorad.cfg configuration file on the standby system, define the following statements if the standby database is open for read-only access:

- Define a DATABASE statement that provides connection information for the standby system.
- If the primary database is a RAC database, define a RAC statement. PowerExchange verifies the RAC MEMBERS parameter value against the number of threads on the standby or archived redo logs. The RAC MEMBERS value should be equal to the number of threads.

If the standby database is *not* open for read-only access, define the following parameters:

- Define a DATABASE statement that provides connection information for the primary system. PowerExchange requires this connection to access the Oracle data dictionary on the primary system.
- Define a STANDBY statement that provides connection information for the standby system. The user that you specify for the STANDBY connection must have SYSDBA authority to access the fixed views and monitor the Data Guard apply progress in a database that is not open.
- In the PowerExchange Logger pwxcl.cfg configuration file on the standby system, configure the DBID parameter to specify the Oracle collection ID that is defined in the registration group for the registered source tables on the primary system.

After CDC is running, the following operational considerations might apply:

- When the standby database is *not* open for read-only access, Oracle might not progress apply processing beyond the tip of the last archived log, even if more recent changes are available in the standby redo logs. To get near-real-time capture processing, add APPLYACTIVE=Y in the STANDBY statement. PowerExchange will then read change data from the standby redo logs up to the highest of the low SCN values in the standby logs for all threads.

- PowerExchange CDC captures data up to the current database SCN, or apply SCN, for a physical standby database. If Oracle stops applying data for some reason, for example, because of a log gap, PowerExchange change capture processing stalls and waits for the apply process to resume with message PWX-36098.

Considerations for Role Transitions of Physical Standby Databases

In an Oracle Data Guard environment, a physical standby database can transition to the primary role. Usually, the role transition occurs because of a failover or switchover. During the transition, all active connections to the physical standby database terminate.

To be able to resume CDC processing after the physical standby database transitions to the primary role, you may need to adjust some configuration parameters in the PowerExchange Express CDC configuration file on the original standby system for PowerExchange to process past the transition point. After the transition, you can adjust the parameters again for optimal performance in the new primary database environment. The following table describes these configuration settings by transition phase:

Statement > Parameter	Before Transition	During Transition	After Transition
RAC > MEMBERS	Specify the number of active threads on the primary database.	Specify the total number of active threads with unique thread IDs on <i>both</i> the standby database and primary database. For example, if the primary database is a two-node RAC database that uses thread IDs 1 and 2 and the standby database is a 3-node RAC database that uses thread IDs 2, 3, and 4, specify MEMBERS=4.	After the restart point has progressed beyond the transition point, edit the MEMBERS parameter value, as needed, for optimal performance of change data capture from the new primary database. Informatica recommends that you use the lowest value that is suitable for your environment to minimize the overhead of PowerExchange Express CDC thread tracking. For information about queries that you can run to determine the lowest suitable MEMBERS value, see "RAC Statement" on page 194 .
STANDBY	Applicable only to physical standby databases that are not open. Not applicable to physical standby databases open for read-only access.	Remove the STANDBY statement if present.	Ensure that the STANDBY statement is absent. This statement is not used for a primary database.

Statement > Parameter	Before Transition	During Transition	After Transition
DATABASE > CONNECT_STRING	<p>If the standby database is not open, define the connection string for the primary database. If a STANDBY statement is also specified, this connection string is used to retrieve table metadata.</p> <p>If the standby database is open, define the connection string for the standby database.</p>	Specify the connection string for the database that will have the primary role after the role transition.	Ensure that the CONNECT_STRING parameter defines the connection string for the new primary database.
OPTIONS ¹ > CONNRETRYMAX CONNRETRYWAIT	Set these connection resiliency parameters high enough to reconnect to the database after an unplanned role transition.	Ensure that the connection resiliency parameters are set high enough to reconnect to the database after the role transition.	If necessary, adjust the connection resiliency parameters for the new primary database environment.
<p>1. Informatica recommends that the connection resiliency parameters for capturing change data from a physical standby database be set high enough to reconnect to the database after a role transition so that the restart point in the redo logs can progress to the transition point. When connection resiliency allows the restart point to progress to the transition point, subsequent CDC warm starts need only the configuration settings described in the After Transition column for all of the statements and parameters.</p>			

How PowerExchange Express CDC reacts to the role transition depends on these configuration settings and whether CDC is active at the time of the role transition:

- If PowerExchange Express CDC is shut down before the role transition occurs, the configuration requirements for all of the statements and parameters in the **During Transition** column, except for the recommended CONNRETRYMAX and CONNRETRYWAIT settings, must be met before you perform a warm start in order to resume CDC processing.
- If PowerExchange Express CDC is active when the role transition occurs and the PowerExchange Express CDC configuration meets the requirements in the **During Transition** column, CDC processing continues without interruption.
- If PowerExchange Express CDC is active when the role transition occurs and the PowerExchange Express CDC configuration does *not* meet the requirements in the **During Transition** column, CDC processing either ends abnormally with an Oracle connection failure or shuts down when Express CDC detects the database role transition.

Oracle Multitenant Pluggable Databases as Sources

Oracle 12c introduced multitenant container databases. If you have an Oracle version that supports the Oracle Multitenant option, you can use PowerExchange Express CDC for Oracle to capture change data for source tables that reside in a pluggable database (PDB) within a multitenant container database (CDB).

The following CDC considerations apply to Oracle multitenant environments:

- PowerExchange Express CDC can capture change data for only a single PDB within a CDB at a time. If you want to capture change data for another PDB in the same CDB, you must configure a separate CDC environment. Informatica recommends that each CDC environment use a unique Oracle capture user ID, pwxorad.cfg file, PowerExchange Logger for Linux, UNIX, and Windows instance and pwxcl.cfg file, and capture registrations.
- If you move or clone the PDB for which PowerExchange Express CDC is capturing change data to another CDB, the PowerExchange Logger for Linux, UNIX, and Windows connection to the Oracle database is lost. You must cold start the PowerExchange Logger to prevent loss of change data.

Task Flow: Configuring Change Capture from an Oracle Multitenant Pluggable Database

To capture change data from a pluggable database (PDB) in an Oracle container database (CDB), perform the following tasks:

1. Verify that the PDB is open.
2. Verify that the CDB is running in ARCHIVELOG mode.
3. Add a PDB entry that includes the PDB service name to the tnsnames.ora file, if this entry does not already exist. For example:

```
PDB1234=
  (DESCRIPTION=
    (ADDRESS=(PROTOCOL=TCP) (HOST=host1) (PORT=1521))
    (CONNECT_DATA=
      (SERVER=DEDICATED)
      (SERVICE_NAME=pdb1234.informatica.com)))
```

4. Enable minimal global supplemental logging for the PDB or CDB. Use the following SQL statement in the ora_orad.sql file:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
COMMIT;
```

5. Recommended. Connect to the CDB\$ROOT container as a user who has the DBA role. Alternatively, log in to the PDB directly.

If you connect to CDB\$ROOT with the DBA role, execute the following SQL statement to switch to the source PDB:

```
ALTER SESSION SET CONTAINER=pdb_name
```

6. Create a unique Oracle user ID for CDC and grant the privileges that are required for change data capture from a PDB to this user.

Use the CREATE USER and GRANT statements in the ora_orad.sql file. This file uses the user name of "ORACPTL1." You can change this name if you want to use another user name.

Make sure that you issue the following GRANT statements that are required for PowerExchange Express CDC in Oracle multitenant environments:

```
GRANT CREATE SESSION TO "ORACPTL1";
GRANT SELECT on "PUBLIC"."V$PDBS" TO "ORACPTL1";
```

7. In the dbmover.cfg configuration file on the system where change capture occurs, add an ORACLEID statement that points to the name of the database that includes the PDB and the name of the PDB service entry in the tnsnames.ora file. For example:

```
ORACLEID=(PDB1234,ORADBNNAME,PDB1234,PDB1234)
```

In this example statement:

- The first positional parameter is a *collection_id* value that identifies the ORACLEID statement. In this case, it is also the PDB name.
- The second positional parameter is the name of the Oracle database that contains the PDB.
- The third positional parameter is the source database connect string, as defined in the tnsnames.ora file. The PowerExchange Navigator uses this connection string.
- The fourth positional parameter is the name of the PDB service entry in the tnsnames.ora file. The PowerExchange Logger for Linux, UNIX, and Windows uses this information for change data capture.

Also add an ORAD CAPI_CONNECTION statement that includes an ORACOLL value that matches the *collection_id* value in the ORACLEID statement.

8. In the PowerExchange Logger for Linux, UNIX, and Windows configuration file, pwxocl.cfg, ensure that the DBID parameter matches the *collection_id* value in the ORACLEID statement in the dbmover.cfg file.
9. Optional. If you specify the DATABASE statement in the PowerExchange Express CDC for Oracle configuration file, pwxorad.cfg, ensure that the CONNECT_STRING parameter points to the name of the PDB service entry in the tnsnames.ora file. Also, ensure that the USERID parameter specifies the Oracle user name that you defined for the change capture user.
10. In the PowerExchange Navigator, create a registration group. Then add a capture registration for each source table in the PDB to the registration group.

Note: The **Collection ID** value that you specify for the registration group must match the *collection_id* value in the ORACLEID statement in the dbmover.cfg file. For the PowerExchange Navigator to be able to connect to the source database to register the tables in the PDB, you must have specified the third positional parameter, *source_connect_string*, in the ORACLEID statement in the dbmover.cfg file.

Amazon RDS for Oracle Database Instances as Sources

PowerExchange Express CDC for Oracle can capture change data from online and archived redo logs for a database instance that is deployed in an Amazon Relational Database Service (RDS) for Oracle environment.

The redo logs must exist in ONLINELOG_DIR and ARCHIVELOG_DIR directories on the RDS file system. Express CDC runs on premises and reads change data from the redo logs.

Restriction: PowerExchange Express CDC for Oracle cannot capture change data that was retrieved from TDE-encrypted tablespaces in an Amazon RDS Oracle database because the Oracle wallet is not available.

For Amazon RDS limitations and requirements related to Oracle versions and operating systems and other Oracle features, see the Amazon RDS documentation.

To configure change data capture for an Amazon RDS Oracle database, perform the following tasks:

1. Create the ONLINELOG_DIR and ARCHIVELOG_DIR directories that will hold the online and archive redo logs, respectively, on the RDS file system. Use the following exec statements:

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```
2. Grant privileges to the Express CDC for Oracle user, ORACAPT1.

Important: You must log in to Amazon RDS under the master username to run the GRANT statements and procedures.

To grant the SELECT privilege, at minimum, on objects and system tables that are required for CDC processing, execute the following GRANT statements:

```
GRANT SELECT ON "PUBLIC"."V$ARCHIVED_LOG"          TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$DATABASE"              TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$LOG"                  TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$LOGFILE"              TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$TRANSPORTABLE_PLATFORM" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$THREAD"               TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$DATABASE_INCARNATION"   TO "ORACAPTL1";

GRANT SELECT ON "SYS"."DBA_LOG_GROUPS"             TO "ORACAPTL1";
GRANT SELECT ON "SYS"."DBA_LOG_GROUP_COLUMNS"      TO "ORACAPTL1";
GRANT SELECT ON "SYS"."DBA_TABLESPACES"           TO "ORACAPTL1";

GRANT SELECT ON "SYS"."OBJ$"                      TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TAB$"                      TO "ORACAPTL1";
GRANT SELECT ON "SYS"."IND$"                      TO "ORACAPTL1";
GRANT SELECT ON "SYS"."COL$"                      TO "ORACAPTL1";

GRANT SELECT ON "SYS"."PARTOBJ$"                 TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TABPART$"                 TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TABCOMPART$"              TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TABSUBPART$"              TO "ORACAPTL1";
COMMIT;
```

To grant the SELECT privilege on some additional objects, run the following Amazon RDS procedures:

```
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'DBA_USERS',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT',
p_grant_option => false);
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'ALL_TABLES',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT',
p_grant_option => false);
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'ALL_TAB_PARTITIONS',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT',
p_grant_option => false);
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$PARAMETER',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$SPPARAMETER',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;
--
begin
```

```

rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$STANDBY_LOG',
p_grantee => 'ORACAPT1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$VERSION',
p_grantee => 'ORACAPT1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'INDPART$',
p_grantee => 'ORACAPT1',
p_privilege => 'SELECT');
end;
--
-- To register sources and perform other tasks in the PowerExchange Navigator:
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$PARAMETER',
p_grantee => 'registration_user',
p_privilege => 'SELECT');
end;

```

To provide read access to the Amazon RDS online and archived redo logs, execute the following GRANT statements:

```

GRANT READ ON DIRECTORY ONLINELOG_DIR to "ORACAPT1";
GRANT READ ON DIRECTORY ARCHIVELOG_DIR to "ORACAPT1";

```

To perform certain tasks, execute the following GRANT statements, as needed:

```

-- To run PowerExchange Logger tasks and extract change data continuously:
GRANT CREATE SESSION TO "ORACAPT1";
--
-- To register sources and perform other tasks in the PowerExchange Navigator:
GRANT SELECT ON table TO "registration_user";          <-Repeat for each source table.
-- Or specify the following statement if your site security rules allow it:
-- GRANT SELECT ANY TABLE to registration_user
--
-- To run the SQL for creating supplemental log groups at the end of registration:
GRANT ALTER ANY TABLE TO "navigator_user";
-- Or give the SQL file to your DBA. The DBA can run the SQL in the file.
--

```

3. Define an appropriate retention time for the archived redo logs. Use the following exec statement:

```

exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention
days',number_of_days);

```

4. In the Amazon RDS console, set the backup retention period for the source database to a value greater than zero to enable automated backups of the database instance.

Note: This step enables ARCHIVELOG mode for the database.

5. Enable supplemental logging at the database level. Use the following exec statement:

```

exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');

```

6. Optionally, in the Amazon RDS console, you can create a parameter group and define the cache sizes of the default buffer pool. The default buffer pool holds buffers that use the primary database block size. Use the following DB_CACHE_SIZE parameter values:

- DB_2K_CACHE_SIZE
- DB_4K_CACHE_SIZE
- DB_16K_CACHE_SIZE

- DB_32K_CACHE_SIZE

Then select the parameter group for the source database.

7. Perform the usual PowerExchange configuration tasks, as described in [“Configuring PowerExchange for Express CDC” on page 169](#).

When you configure the PowerExchange Express CDC for Oracle configuration file, specify **RDS=Y** in the DATABASE statement to indicate that the source database is in an Amazon RDS for Oracle environment

Source Tablespaces That Use Transparent Data Encryption

PowerExchange Express CDC for Oracle can capture change data from Oracle tablespaces that are encrypted using Oracle Advanced Security Transparent Data Encryption (TDE).

To capture TDE-encrypted change data from tablespaces, you must perform the following tasks:

- In the PowerExchange Express CDC configuration file, enter the TDEKEYSTORE parameter in the DATABASE statement to specify the keystore type, location, and password so that PowerExvhange can access the encryption keys for encryption and decryption of data. The keystore can be in ASM, a file, or a hardware security module (HSM).
- Grant the following privilege to the ORACAPTL user to get information about the default wallet type, status, and location, when it is not specified in the TDEKEYSTORE parameter of the DATABASE statement in the PowerExchange Express CDC configuration file:

```
GRANT SELECT ON "PUBLIC"."V$ENCRYPTION_WALLET" TO "ORACAPTL";
```

- If you store the TDE wallet in a file system, ensure that PowerExchange Express CDC can access the file with Read file permissions. If you do NOT run PowerExchange on the Oracle database server, use NFS to mount the TDE wallet directory to the machine where PowerExchange is running, or copy the wallet to a file-system location that PowerExchange can access and specify the WALLETDIR option in the TDEKEYSTORE parameter in the DATABASE statement to point to that location.
- Verify that the Oracle TDE wallet is open in the database

Restriction: PowerExchange Express CDC cannot capture change data that was retrieved from TDE-encrypted tablespaces in an Amazon RDS Oracle database because the Oracle wallet is not available.

Limiting the Redo Logs From Which Express CDC Reads Changes

In the READER statement of the PowerExchange Express CDC configuration file, you can specify parameters that limit the archived log destinations and the active redo logs from which the Express CDC log reader reads change records. Express CDC uses these parameters to build a list of the redo logs to read.

For archived logs, Express CDC attempts to identify a primary copy and a secondary copy of each log that it must read. Express CDC identifies the primary and secondary archived log destinations based on the order of the archive destination numbers, where an archive destination number is the *n* value in an Oracle LOG_ARCHIVE_DEST_*n* parameter. The primary destination is the destination that 1) contains available logs that are suitable for PowerExchange use and 2) has the lower *n* value compared to any other destination with available archived logs. You can override this behavior by specifying the ARCHIVEDEST parameter in the READER statement of the Express CDC configuration file.

If you specify MODE=ACTIVE in the READER statement, you can use any of the following optional parameters to filter active redo logs or archived log destinations:

- **ACTIVELOGMASK.** Specifies a mask for selecting active redo logs for the PowerExchange Express CDC log reader when the Oracle instance uses multiplexing of redo logs.

- **ARCHIVEDEST.** Specifies a list of log destinations in which PowerExchange Express CDC looks for valid archived logs. If you create more than one copy of each archive log, use this parameter to indicate the primary and secondary log destinations.

For more information, see [“READER Statement” on page 195](#).

In both ACTIVE and ARCHIVEONLY modes, PowerExchange Express CDC uses the ARCHIVEDEST parameter to filter archived log destinations, as follows:

- If you specify one ARCHIVEDEST log destination, PowerExchange Express CDC uses only the log destination that is specified in the LOG_ARCHIVE_DEST_1 parameter to which the ARCHIVEDEST setting points. For example, if ARCHIVEDEST=(2), PowerExchange Express CDC uses the log destination specified in the LOG_ARCHIVE_DEST_2 parameter.

Note: Specifying one log destination for ARCHIVEDEST limits the resilience of the PowerExchange Express CDC log reader because the log reader then processes logs from only one archive log destination.

- If you specify two ARCHIVEDEST destinations, PowerExchange Express CDC first attempts to use the first ARCHIVEDEST specification. If the log reader cannot read the archived logs from this destination, PowerExchange Express CDC attempts to use the second ARCHIVEDEST log destination specification. For example, if you specify ARCHIVEDEST=(2,4), PowerExchange Express CDC first attempts to read the archived logs in the log destination that is specified in the LOG_ARCHIVE_DEST_2 parameter. If PowerExchange Express CDC cannot read the logs in this primary destination, it attempts to read the logs in the destination that is specified in the LOG_ARCHIVE_DEST_4 parameter.
- If you specify more than two ARCHIVEDEST destinations, PowerExchange queries v\$archived_log for the first two valid archive logs, based on the destinations and the order specified in ARCHIVEDEST. If the log reader cannot read the first log retrieved, it tries to read the log in the second destination retrieved. If it cannot read the log in the second destination retrieved, the process terminates. For example, if you specify ARCHIVEDEST=(7,6,5,4,3,1), the query might return archive logs in destinations 6 and 1. Therefore, the first log PowerExchange Express CDC tries to read is the log from destination 6. If PowerExchange Express CDC cannot read the log, it tries to read the log from destination 1. If PowerExchange Express CDC cannot read the log from destination 1, the process terminates.
- If you do not specify the ARCHIVEDEST parameter, PowerExchange Express CDC does not filter the archived log query by destination.

If you specify MODE=ARCHIVECOPY in the READER statement, the PowerExchange Express CDC log reader reads copies of the archived redo logs that are located on a file system. You must specify the DIR parameter to identify the base directory that includes the copies of the archived redo logs. To filter the subdirectories under the base directory that the log reader scans and the copies of the archived log files in these subdirectories, also specify the FILE parameter with masks for the subdirectory names, log file names, or both.

Implementation Task Flow

To configure and start a new PowerExchange Express CDC for Oracle environment, complete the following high-level tasks:

1. Configure the Oracle source for CDC.
Use the sample SQL in the ora_orad.sql file that PowerExchange provides. For more information, see [“Configuring Oracle for Express CDC” on page 165](#). For Amazon RDS for Oracle environments, also see [“Amazon RDS for Oracle Database Instances as Sources” on page 160](#).
2. Configure PowerExchange for CDC.
For more information, see [“Configuring PowerExchange for Express CDC” on page 169](#).

3. Configure the PowerExchange Logger for Linux, UNIX, and Windows.
Use of the PowerExchange Logger is optional but strongly recommended. For information about configuring and starting the PowerExchange Logger, see ["Configuring the PowerExchange Logger" on page 44](#) and ["Starting the PowerExchange Logger" on page 66](#).
4. Configure a CDC restart point.
For more information, see ["Creating Restart Tokens for Extractions" on page 283](#).
5. Materialize the target tables.
6. Start the PowerExchange Logger.
7. Create PowerCenter CDC workflows that include the Express CDC source tables.
Use an Oracle CDC application connection. For more information, see the *PowerExchange Interfaces for PowerCenter* publication.
8. Cold start the workflows.

Configuring Oracle for Express CDC

You must perform some configuration tasks in Oracle to prepare for PowerExchange Express CDC for Oracle.

If you have not configured the use of archived logs, enable ARCHIVELOG mode and specify a primary archive log destination. Then use the sample file, ora_orad.sql, that PowerExchange supplies in its installation directory to perform the other tasks. To run the SQL statements in the ora_orad.sql file, you must be assigned the role of DBA.

Complete the following Oracle configuration tasks:

1. Specify an archive log destination for PowerExchange Express CDC for Oracle use if one is not already defined.
2. Enable ARCHIVELOG mode if it is not already enabled.
3. If you enable ARCHIVELOG mode, stop and restart the Oracle database.
4. Create an Oracle user and grant user privileges.
5. If you use ASM and plan to connect to an ASM instance to read ASM-managed Oracle redo logs, create an ASM user that has SYSDBA or SYSASM authority.
6. Enable Oracle minimal global supplemental logging.

For configuration tasks specific to Oracle multitenant environments, see ["Task Flow: Configuring Change Capture from an Oracle Multitenant Pluggable Database" on page 159](#).

Specifying an Archive Log Destination

You must specify the archive log destination from which the Express CDC log reader selects archived redo log files.

Consult with your Oracle DBA. To configure the archived log destination, complete one of the following actions:

- Edit the init.ora file to specify the archive log destination and file-name format. For more information about the init.ora file, see the Oracle database administrator guide.

- Customize the appropriate server parameter file (spfile) to indicate the archived log destination, for example:

```
CONNECT SYS/sys_pwd AS SYSDBA;
ALTER SYSTEM SET
  log_archive_dest_1 = 'location=/oracle_path/arch'
  SCOPE=SPFILE;
```

- For Oracle databases deployed in an Amazon RDS for Oracle environment, create the directories ARCHIVELOG_DIR and ONLINELOG_DIR for the archived and online redo logs, respectively. Use the following exec statements:

```
exec rdsadmin.rdsadmin_master_util.create_archivelog_dir;
exec rdsadmin.rdsadmin_master_util.create_onlinelog_dir;
```

Also define an appropriate retention policy for the archived redo logs. Use the following exec statement:

```
exec rdsadmin.rdsadmin_util.set_configuration('archivelog retention
days', number_of_days);
```

Tip: The Express CDC for Oracle user will require read-only access to the online and archived redo log files in these directories.

Enabling ARCHIVELOG Mode

PowerExchange Express CDC for Oracle requires Oracle to be running in ARCHIVELOG mode.

By default, ARCHIVELOG mode is not enabled.

To enable ARCHIVELOG mode for Oracle databases not in an Amazon RDS environment, issue the following SQL statements:

```
SHUTDOWN IMMEDIATE;
STARTUP MOUNT;
ALTER DATABASE ARCHIVELOG;
ALTER DATABASE OPEN;
SHUTDOWN IMMEDIATE;
STARTUP;
```

Tip: Back up your database after each SHUTDOWN command.

To confirm that ARCHIVELOG mode is set for the database, enter the following SQL statement:

```
SELECT LOG_MODE FROM SYS.V$DATABASE;
```

For an Amazon RDS for Oracle databases, set the backup retention period to place the database in ARCHIVELOG mode and enable automated backups.

Creating an Oracle User and Granting User Privileges

To capture change data from Oracle redo logs, you must define a CDC user and assign specific Oracle system and object privileges to that user.

You can either use an existing user who has the required authority as the CDC user, or create a user and grant the required privileges to that user. In the ora_orad.sql file, PowerExchange provides sample SQL statements for creating an Oracle user and for granting the required privileges to that user.

Note: To execute GRANT statements and procedures for Amazon RDS sources, you must log in to the Amazon RDS console under the master username.

1. Log in to Oracle as a user who has the role of DBA.

2. If you have an Oracle source in a multitenant environment, issue the following SQL statement so that the Oracle user will be created in the pluggable database (PDB) from which change data will be captured instead of in the container database (CDB):

```
ALTER SESSION SET CONTAINER=pdb_name;
```

3. To create a user named ORACAPTL1, issue the following SQL statement:

```
CREATE USER "ORACAPTL1" PROFILE "DEFAULT"
  IDENTIFIED BY "oracaptl1"
  ACCOUNT UNLOCK;
COMMIT;
```

4. To retrieve the information that is required for correct capture processing, grant the SELECT privilege, at minimum, on certain objects and system tables to the ORACAPTL1 user:

```
GRANT SELECT ON "PUBLIC"."V$ARCHIVED_LOG" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$DATABASE" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$LOG" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$LOGFILE" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$TRANSPORTABLE_PLATFORM" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$THREAD" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$DATABASE_INCARNATION" TO "ORACAPTL1";
GRANT SELECT ON "PUBLIC"."V$PARAMETER" TO "ORACAPTL1"; <--Not for Amazon RDS
GRANT SELECT ON "PUBLIC"."V$SPPARAMETER" TO "ORACAPTL1"; <--Not for Amazon RDS
GRANT SELECT ON "PUBLIC"."V$STANDBY_LOG" TO "ORACAPTL1"; <--Not for Amazon RDS
GRANT SELECT ON "PUBLIC"."V$VERSION" TO "ORACAPTL1"; <--Not for Amazon RDS

GRANT SELECT ON "SYS"."ALL_TABLES" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."ALL_TAB_PARTITIONS" TO "ORACAPTL1";

GRANT SELECT ON "SYS"."DBA_LOG_GROUPS" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."DBA_LOG_GROUP_COLUMNS" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."DBA_USERS" TO "ORACAPTL1"; <--Not for Amazon RDS
GRANT SELECT ON "SYS"."DBA_TABLESPACES" TO "ORACAPTL1";

GRANT SELECT ON "SYS"."OBJ$" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TAB$" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."IND$" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."COL$" TO "ORACAPTL1";

GRANT SELECT ON "SYS"."PARTOBJ$" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TABPART$" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."INDPART$" TO "ORACAPTL1"; <--Not for Amazon RDS
GRANT SELECT ON "SYS"."TABCOMPART$" TO "ORACAPTL1";
GRANT SELECT ON "SYS"."TABSUBPART$" TO "ORACAPTL1";
--
-- To run PowerExchange Logger tasks and extract change data continuously:
GRANT CREATE SESSION TO "ORACAPTL1";
--
-- To extract change data from a multitenant container database: <--Not supported for Amazon RDS
GRANT CREATE SESSION TO "ORACAPTL1"; <--If not already specified
GRANT SELECT ON "PUBLIC"."V$PDBS" TO "ORACAPTL1";
--
-- To capture data from Oracle TDE-encrypted tablespaces: <--Not supported for Amazon RDS
GRANT SELECT ON "PUBLIC"."V$ENCRYPTION_WALLET" TO "ORACAPTL1";
--
-- To register sources and perform other tasks in the PowerExchange Navigator:
GRANT SELECT ON "PUBLIC"."V$PARAMETER" TO "registration_user"; <--Not for Amazon RDS
GRANT SELECT ON table TO "registration_user"; <--Repeat for each source table
-- Or specify the following statement if your site security rules allow it:
-- GRANT SELECT ANY TABLE TO registration_user
--
-- To run the SQL for creating supplemental log groups at the end of registration:
GRANT ALTER ANY TABLE TO "navigator_user";
-- Or give the SQL file to your DBA. The DBA can run the SQL in the file.
--
COMMIT;
```

For Amazon RDS for Oracle sources, issue the preceding grants except those that do *not* apply to Amazon RDS sources. Then, run the following procedures to issue additional grants:

```
-- To access online and archived redo logs in Amazon RDS for Oracle Database:
GRANT READ ON DIRECTORY ONLINELOG_DIR to "ORACAPTL1";
GRANT READ ON DIRECTORY ARCHIVELOG_DIR to "ORACAPTL1";
--
```

```

/* To grant the SELECT privilege on some objects, you must log in under the
master username and run the following Amazon RDS procedures:
*/
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'DBA_USERS',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT',
p_grant_option => 'false');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$PARAMETER',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$PARAMETER',
p_grantee => 'registration_user',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$SPPARAMETER',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$STANDBY_LOG',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'V_$VERSION',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;
--
begin
rdsadmin.rdsadmin_util.grant_sys_object(
p_obj_name => 'INDPART$',
p_grantee => 'ORACAPTL1',
p_privilege => 'SELECT');
end;

```

Attention: If you do not use ASM, the operating system user ID under which the PowerExchange Express CDC for Oracle capture process runs must also have the authority to read the Oracle online and archive redo logs. Otherwise, the Express CDC log reader cannot directly read the log files, and the CDC session ends with error message PWX-36140.

Creating an ASM User

If you use ASM and plan to connect to an ASM instance to retrieve change data from ASM-managed Oracle redo logs, you must configure an ASM login user ID that has SYSDBA or SYSASM authority.

For Oracle 11.1.0.7 or later ASM environments, create an ASM user and grant one of the following authority levels to that user:

- SYSDBA

- SYSASM

To use SYSASM, you must also set the ASM_ASSYSASM parameter to Y in the READER statement of the PowerExchange Express CDC for Oracle configuration file, pwxorad.cfg.

For more information about creating an ASM user, see the *Oracle Automatic Storage Management Administrator's Guide*.

Enabling Oracle Minimal Global Supplemental Logging

PowerExchange Express CDC for Oracle requires Oracle minimal global supplemental logging to properly handle chained rows.

To enable supplemental logging for an on-premises database, log in to the Oracle database and issue the following SQL statement, which is included in the sample ora_orad.sql script file:

```
ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;
COMMIT;
```

If you execute this statement when minimal global supplemental logging is already enabled for your database, the statement has no effect.

If you execute this statement while the Oracle database is open, Oracle waits for in-flight transactions to complete, which can affect database performance. This issue is likely to occur for databases that have a high level of user activity. To avoid this problem, you can close and re-open the database and then issue the statement manually.

To enable supplemental logging at the database level in an Amazon RDS for Oracle environment, execute the following exec statement:

```
exec rdsadmin.rdsadmin_util.alter_supplemental_logging('ADD');
```

Note: You must also define a supplemental log group for each Oracle source table. When you register an Oracle table in the PowerExchange Navigator, PowerExchange generates DDL for adding a supplemental log group for the table. Supplemental log groups cause Oracle to log full before- and after-images of the data that changed. PowerExchange requires these images.

Configuring PowerExchange for Express CDC

You must complete several tasks to configure PowerExchange for Express CDC for Oracle and to start change capture.

The specific tasks depend on your configuration. The following task flow assumes that you use the PowerExchange Logger, as recommended.

Important: Make sure that the system user ID under which the PowerExchange Express CDC for Oracle capture process runs has the authority to read the Oracle redo logs. Otherwise, the Express CDC log reader cannot directly read the log files, and the CDC session ends with error message PWX-36140.

1. Configure the dbmover configuration file on the PowerExchange system that initiates the CAPI to capture change data from the Oracle source database.
Include the following statements:
 - Required. CAPT_PATH and CAPT_XTRA statements.
 - Required. An ORAD CAPI_CONNECTION statement.

- Required. An ORACLEID statement.
- Recommended. An ORACLE_CAPTURE_TYPE statement.
- Recommended. A CAPX CAPI_CONNECTION statement, if you use the PowerExchange Logger and continuous extraction mode.
- Optional. An ORACLE_UNHANDLED_NUMASCHAR statement to treat NUMBER columns that have a precision greater than 28 or an undefined length as variable length strings and to treat FLOAT columns that have a precision greater than 15 as variable length strings. This behavior can prevent data loss. You must define this statement on the PowerExchange instance where the registration and extraction map information will be defined, before you create the registrations.

Notes: If you run PowerExchange Express CDC and the PowerExchange Listener, consumer API (CAPI), and PowerExchange Logger on the Oracle system, as in Configuration 1, define the CAPX CAPI_CONNECTION and ORACLE_UNHANDLED_NUMASCHAR statements in this dbmover configuration file.

If you capture change data from an Oracle Data Guard physical standby database, configure the ORACLEID and ORAD CAPI_CONNECTION in the DBMOVER configuration file on the standby system to point to the source instance on the primary system. If the primary system is a RAC environment, point to a single member in the RAC.

2. Customize the PowerExchange Express CDC for Oracle configuration file, which has the default file name of pwxorad.cfg, on the change capture system.

If you capture changes from RAC members, include the RAC statement.

If you use ASM, include the ASM parameters in the READER statement.

If the Oracle source database is in an Amazon RDS for Oracle environment, specify RDS=Y in the DATABASE statement..

Tip: To include rowid values in change records for tables that do not have Oracle row movement enabled, include the OPTIONS ROWID=Y statement. Provided that you use version 9.1.2 or later of the PowerExchange Navigator, PowerExchange can write rowid values to the generated DTL__CAPXROWID column in captured change records.

3. Customize the PowerExchange Logger configuration file, which has the default file name of pwxcl.cfg, on the system where the PowerExchange Logger runs.
For more information, see [“Customizing the PowerExchange Logger Configuration File” on page 45](#).

4. Configure a dbmover configuration file on any other PowerExchange instance in the CDC environment that needs to read capture registrations or extraction maps from disk.

If you use Configuration 2, include the following statements in the dbmover configuration file on the system with the separate PowerExchange Listener:

- Required. An ORACLE_CAPTURE_TYPE statement.
- Required. An ORACLEID statement.
- Recommended. A CAPX CAPI_CONNECTION statement if you run the PowerExchange Logger on the change capture system and use continuous extraction mode.
- Optional. An ORACLE_UNHANDLED_NUMASCHAR statement if you need PowerExchange to treat NUMBER columns that have a precision greater than 28 or an undefined length as variable length strings and to treat FLOAT columns that have a precision greater than 15 as variable length strings.

5. In the PowerExchange Navigator, create a registration group and a capture registration for each Oracle source table.

Make sure that you enter values in the following fields:

- In the **Type** list, select **ORACLE**.

- In the **Collection ID** box, enter a user-defined name for the Oracle instance. This value must match the collection ID in the ORACLEID statement in the dbmover.cfg file that is on the node specified in the **Location** field. If you use the PowerExchange Logger, this value also must match the DBID value in the PowerExchange Logger configuration file.
- Set the **Status** option to **Active**.
- In the **Condense** list, select **Part**. Informatica recommends this setting even if you do not use the PowerExchange Logger, because it allows you to implement the PowerExchange Logger later, if necessary, without changing the registration.
- In the **Supplement Log Group Name** box, enter a name for the Oracle supplemental log group.

Tip: If you want PowerExchange to run the DDL that it generates to create the supplemental log group at registration completion, select **Execute DDL Now**. However, Informatica recommends that you save the generated DDL to a file and give it to your Oracle DBA instead. The DBA can then use this DDL when migrating PowerExchange from a test or QA environment to the production environment.

After you click **Finish**, PowerExchange generates corresponding extraction maps.

For more information about registrations and extraction maps, see the *PowerExchange Navigator User Guide*.

6. Perform a database row test on each extraction map.
For more information, see the *PowerExchange Navigator User Guide*.
7. Configure a restart point.
For more information, see [“Creating Restart Tokens for Extractions” on page 283](#).
8. Materialize the target.
9. Start the PowerExchange Logger.

Next, configure PowerCenter CDC sessions. You can use batch extraction mode, continuous extraction mode, or real-time extraction mode. For more information, see *PowerExchange Interfaces for PowerCenter*.

RELATED TOPICS:

- [“Configuring the dbmover Configuration File ” on page 171](#)
- [“Customizing the PowerExchange Express CDC for Oracle Configuration File” on page 179](#)

Configuring the dbmover Configuration File

In the dbmover configuration file, define the statements that are required for PowerExchange Express CDC for Oracle. Include the optional statements as needed.

Define the following statements in the dbmover configuration file on the system where the capture registrations and CDC control files are stored, which is specified as the **Location** node in the registration group:

CAPT_PATH

Required. Path to the local directory on the system that contains the control files for CDC. These files are: the CCT file for capture registrations, the CDEP file for application names that are used for ODBC extractions, and the CDCT file for the PowerExchange Logger for Linux, UNIX, and Windows.

CAPT_XTRA

Required. Path to the local directory on a system that stores extraction maps for CDC.

ORACLE_CAPTURE_TYPE

Ensure that this statement specifies **D** to use PowerExchange Express CDC use for the PowerExchange installation.

If multiple systems are involved in CDC processing, for example, because you run a separate PowerExchange Listener to handle registrations and extraction maps, check the ORACLE_CAPTURE_TYPE setting in the dbmover.cfg file on each system.

ORACLEID

Required. The Oracle source instance, database, and connection information that is used for CDC.

If you capture change data from a pluggable database (PDB) in an Oracle multitenant environment, you must include the following parameter settings:

- In the second positional parameter, *oracle_db*, specify the name of the database that contains the PDB.
- In the third positional parameter, *source_connect_string*, specify the Oracle connection string, defined in TNS, that is used to connect to the Oracle database that contains the tables in the PDB. The PowerExchange Navigator requires this parameter value to access the source database when you create capture registrations or perform database row tests.
- In the fourth positional parameter, *capture_connect_string*, specify the name of the PDB service entry in the tnsnames.ora file.

ORACLE_UNHANDLED_NUMASCHAR

Optional. Controls whether capture registration processing for PowerExchange Express CDC sources handles NUMBER columns that have a precision greater than 28 or an undefined length as variable length strings and handles FLOAT columns that have a precision greater than 15 as variable length strings.

ORAD_CAPI_CONNECTION

Required. A named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and control PowerExchange Express CDC for Oracle processing for Oracle data sources. In this statement, you can include the PARMFILE parameter to point to the separate configuration file that contains statements and parameters specifically for PowerExchange Express CDC for Oracle. Include the PARMFILE parameter if you want to override the default file name of pwxorad.cfg.

For more information about the ORACLE_CAPTURE_TYPE, ORACLE_UNHANDLED_NUMASCHAR, and ORAD_CAPI_CONNECTION statements, see the following detailed descriptions. For more information about all other DBMOVER statements, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“Example dbmover Configuration File for the Oracle Change Capture System” on page 178](#)
- [“CAPI_CONNECTION - ORAD Statement” on page 175](#)
- [“CAPI_CONNECTION - CAPX Statement” on page 60](#)
- [“ORACLEID Statement” on page 172](#)
- [“ORACLE_UNHANDLED_NUMASCHAR Statement” on page 177](#)
- [“ORACLE_CAPTURE_TYPE Statement” on page 174](#)
- [“CAPT_PATH Statement” on page 31](#)
- [“CAPT_XTRA Statement” on page 31](#)

ORACLEID Statement

The ORACLEID statement specifies the Oracle source instance, database, and connection information for CDC.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Oracle CDC sources

Related Statements: CAPI_CONNECTION - ORAD

Required: Yes, for PowerExchange Express CDC for Oracle

Syntax:

```
ORACLEID=(collection_id
          ,oracle_db
          [,source_connect_string]
          [,capture_connect_string]
          [,fifth_positional_parameter]
          [,USEDATABASE])
```

Parameters:

collection_id

Required. User-defined identifier for this ORACLEID statement. This value must match the ORACOLL parameter value in the ORAD CAPI_CONNECTION statement, the collection ID in the registration group defined for the source tables, and the DBID value in the PowerExchange Logger pwxcl configuration file.

Maximum length is eight characters.

oracle_db

Required. Name of the Oracle database that contains the source tables that are registered for change data capture. If you use PowerExchange Express CDC for Oracle to capture change data from a pluggable database (PDB) in an Oracle multitenant environment, this value is the name of the database that contains the PDB.

source_connect_string

Optional. Oracle connection string, defined in TNS, that is used to connect to the Oracle database that contains the source tables. This connection string must be defined in the Oracle Client tnsnames.ora file on the system with the source database.

For PowerExchange Express CDC for Oracle, the source connection string is used only for PowerExchange Navigator access to the Oracle source database. Enter this parameter in the dbmover configuration file on the machine from which the PowerExchange Listener retrieves data for PowerExchange Navigator requests. If you plan to run a database row test on extraction maps for the source tables, also specify the *capture_connect_string* parameter.

Note: The source connection string is not used to transfer change data.

If this value is null and the Oracle source is *not* a PDB in a multitenant environment, the value of the ORACLE_SID environment variable is used by default. If you use PowerExchange Express CDC for Oracle to capture change data from a PDBs, you must enter a value in this parameter.

capture_connect_string

Optional. Oracle connection string, defined in TNS, that the PowerExchange Logger uses to connect to the Oracle database with the source tables for PowerExchange Express CDC. This connection string must be specified in the Oracle Client tnsnames.ora file that is used for connection to the Oracle source database. If you use PowerExchange Express CDC to capture change data from a PDB in an Oracle multitenant environment, specify the name of the PDB service entry in the tnsnames.ora file.

If this value is null and the Oracle source is *not* a PDB in a multitenant environment, the value of the ORACLE_SID environment variable is used by default.

If this value is null and the Oracle source is a PDB, PowerExchange cannot capture change data for the source. If you use PowerExchange Express CDC for Oracle to capture change data from PDBs, you must enter a value in this parameter.

Also, if you have multiple Oracle databases and capture changes from a database other than the default database, you must specify both the *source_connect_string* and *capture_connect_string* parameters.

Tip: If possible, bypass the use of SQL*Net to improve PowerExchange Logger performance, even if the PowerExchange Logger is running on the same machine as the Oracle source database. Set the following environment variables, whenever possible, to enable connection to the appropriate Oracle database without using the *capture_connect_string* parameter and SQL*Net:

- ORACLE_HOME
- ORACLE_SID
- PATH
- On Linux or UNIX, one of the following:
 - LD_LIBRARY_PATH
 - LIBPATH
 - SHLIB_PATH

fifth_positional_parameter

Not used. Add a comma as a placeholder if you specify the USEDDBNAME positional parameter, for example:

```
ORACLEID=(collection_id,oracle_db,src_connect_string,capture_connect_string,,USEDDBNAME)
```

USEDDBNAME

Optional. Specify this parameter only if you run the following SQL query on the V\$DATABASE view and the query returns different values for the NAME and DB_UNIQUE_NAME fields, including values that vary in case only such as ORAABC1 and oraabc1:

```
select name, db_unique_name from v$database;
```

In this situation, this parameter can prevent potential restart errors that are caused by the difference in the NAME and DB_UNIQUE_NAME values.

Tip: Alternatively, you can specify the DB_UNIQUE_NAME value in the second positional parameter, *oracle_db*.

Usage Notes:

- PowerExchange requires an ORACLEID statement for each Oracle database for which you want to capture and extract change data. You can define a maximum of 20 ORACLEID statements in a single dbmover configuration file.
- Define the ORACLEID statement in the dbmover configuration file on the system where the PowerExchange Logger runs, or if you plan to perform Oracle CDC without the PowerExchange Logger, on the system where your PowerExchange extractions run.

ORACLE_CAPTURE_TYPE Statement

The ORACLE_CAPTURE_TYPE statement specifies that the CDC type to use for the PowerExchange installation. The only available type is PowerExchange Express CDC for Oracle.

The ORACLE_CAPTURE_TYPE value must be consistent with the ORAD type of CAPL_CONNECTION statement that is defined on the system that initiates the connection to the Oracle system for change capture.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Oracle

Required: No

Syntax:

```
ORACLE_CAPTURE_TYPE=D
```

Valid Values:

- **D.** PowerExchange uses Express CDC for Oracle and the ORAD CAPI_CONNECTION statement.

No default value is provided. The only valid value is D.

Usage Notes:

- PowerExchange Express CDC for Oracle handles all character columns as variable length columns. This behavior affects PowerExchange column-level processing and the view of the CDC extraction map that is imported into PowerCenter. All systems that are involved in Oracle change capture processing must be aware of the Oracle CDC type that is in use.

On the PowerExchange system where the ORAD CAPI_CONNECTION statements are defined, the CAPI_CONNECTION type implicitly defines the CDC type, and the ORACLE_CAPTURE_TYPE statement is optional. However, if other systems are involved in CDC processing, for example, because you run a separate PowerExchange Listener or use offload processing, you can define the ORACLE_CAPTURE_TYPE statement in the dbmover.cfg file on each system.

- For the ORAD CAPI_CONNECTION statement, the ORACLE_CAPTURE_TYPE value must be D. Otherwise, PowerExchange issues an error message and ends abnormally.

CAPI_CONNECTION - ORAD Statement

The ORAD CAPI_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and control PowerExchange Express CDC for Oracle processing for Oracle data sources.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Oracle

Related Statements: ORACLEID, ORACLE_CAPTURE_TYPE

Required: Yes, for PowerExchange Express CDC for Oracle

Syntax:

```
CAPI_CONNECTION=( [DLLTRACE=trace_id]
                  ,NAME=capi_connection_name
                  ,TYPE=(ORAD
                        [,EPWD=database_encrypted_password]
                        ,ORACOLL=collection_id
                        [,PARMFILE=express_cdc_configuration_file]
                        [,PASSWORD=database_password]
                        [,USERID=database_user_id]
                        )
                  )
```

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.

TYPE=(ORAD, ...)

Required. Type of CAPI_CONNECTION statement. For PowerExchange Express CDC for Oracle sources, this value must be ORAD.

EPWD=database_encrypted_password

Optional. An encrypted password that PowerExchange uses to connect to the Oracle source database for PowerExchange Express CDC for Oracle. If you specify the USERID parameter in this statement, you must also specify either the EPWD or PASSWORD parameter.

This encrypted password overrides the EPWD parameter value in the DATABASE statement of the PowerExchange Express CDC for Oracle configuration file and the **Password** value in the CDC session connection attributes. To use one of these other encrypted passwords, do not include the EPWD parameter in the ORAD CAPI_CONNECTION statement.

ORACOLL=collection_id

Required. The collection identifier for the Oracle instance. This value must match the collection ID in the first positional parameter of an ORACLEID statement in the same dbmover.cfg file.

Usually, this value also matches the collection ID that you specify in the registration group for the Oracle instance. If you specify a different collection ID in the registration group, the registration collection ID overrides this ORACOLL value.

PARMFILE=path_and_filename

Optional. The path and file name for the PowerExchange Express CDC for Oracle configuration file, relative to the current working directory. You can use this parameter to override the default path and file name or to remind PowerExchange users of the default path and file name.

If this parameter is not specified, PowerExchange uses *pwd_home_directory\pwdorad.cfg* by default. The default path is the path in the PWX_HOME environment variable, or if this environment variable is not defined, the default path is the path to the PowerExchange bin directory. If the pwdorad.cfg file does not exist at the default location and a PARMFILE override is not defined, PowerExchange issues error messages PWX-09951 and PWX-00268 and change capture fails.

PASSWORD=database_password

Optional. A clear text password that PowerExchange uses to connect to the Oracle source database for PowerExchange Express CDC for Oracle. If you specify the USERID parameter in this statement, you must specify either the EPWD or PASSWORD parameter.

This clear-text password overrides the PASSWORD parameter in the DATABASE statement of the PowerExchange Express CDC for Oracle configuration file and the **Password** value in the CDC session connection attributes. To use one of these other passwords, do not include the PASSWORD parameter in the ORAD CAPI_CONNECTION statement.

USERID=database_user_id

Optional. A user ID that PowerExchange uses to connect to the Oracle source database for PowerExchange Express CDC for Oracle. If you specify the USERID parameter in this statement, you must also specify either the EPWD or PASSWORD parameter.

This user ID overrides the USERID parameter in the DATABASE statement of the PowerExchange Express CDC for Oracle configuration file and the **User Name** value in the CDC session connection attributes. To use one of these other user IDs, do not include the USERID parameter in the ORAD CAPI_CONNECTION statement.

Usage Notes:

- You can specify multiple ORAD CAPI_CONNECTION statements in the dbmover.cfg file to capture change data from more than one Oracle instance or to use different parameter settings for the same Oracle instance.
- Define the ORAD CAPI_CONNECTION and ORACLEID statements on the PowerExchange system that must connect to the Oracle source database for change data capture. Usually, the PowerExchange Logger for Linux, UNIX, and Windows runs on this system.
- The database user ID and password or encrypted password can be specified in multiple locations. If you do so, PowerExchange uses the following order of precedence:
 1. The USERID value and the EPWD or PASSWORD value that are specified in the ORAD CAPI_CONNECTION statement in the dbmover configuration file
 2. The USERID value and EPWD or PASSWORD value that are specified in the DATABASE statement in the PowerExchange Express CDC for Oracle configuration file, pwxorad.cfg.
 3. The **User Name** and **Password** values that are specified in the Oracle application connection attributes for the PowerCenter CDC session

ORACLE_UNHANDLED_NUMASCHAR Statement

The ORACLE_UNHANDLED_NUMASCHAR statement controls how PowerExchange handles some numeric Oracle source columns.

If you enter Y, PowerExchange converts the following Oracle numeric datatypes:

- NUMBER columns that have a precision greater than 28 or an undefined length are treated as variable-length strings instead of double-precision floating-point numbers.
- FLOAT columns that have a precision greater than 15 significant digits are treated as variable-length strings.

PowerExchange uses the ORACLE_UNHANDLED_NUMASCHAR setting when creating capture registrations.

This statement applies to PowerExchange Express CDC for Oracle sources. You can use this statement to override PowerExchange default processing of numeric data to prevent data loss in certain circumstances. To override default processing, you must specify this statement prior to creating capture registrations.

Operating Systems: Linux, UNIX, and Windows

Data Sources: Oracle

Required: No

Syntax:

```
ORACLE_UNHANDLED_NUMASCHAR={Y|N}
```

Valid Values:

- **N.** PowerExchange uses its default processing of Oracle NUMBER data. If you have NUMBER columns that have a precision greater than 28 or an undefined length, or if you have FLOAT columns with a precision greater than 15, change data loss might occur.
- **Y.** PowerExchange handles NUMBER and FLOAT data in a manner that prevents data loss.

Default is N.

Usage Notes:

- Enter this parameter with a value of Y before you create capture registrations for the Oracle source tables that contain the NUMBER or FLOAT columns. If you enter this parameter after the registrations exist, you must set the status of the registrations to History and then create the registrations again. Otherwise, change data loss might occur.
- Oracle allows columns that have the NUMBER datatype to have their precision and scale determined by the numeric data that is written to the columns. Oracle supports a maximum precision of 38 and an exponent of +/-127.

If you do not explicitly define the precision and scale for NUMBER columns from which change data is captured, the following default PowerExchange and PowerCenter processing of change data can result in loss of precision and change data:

- PowerExchange handles data in NUMBER columns that have an undefined length or a length greater than 100 bytes as double-precision floating-point numbers.
- PowerCenter allows a maximum precision of 28 for decimal numbers.

To prevent change data loss with this type of data, enter Y for this statement and then create your capture registrations. PowerExchange registration processing can then handle numbers that have a precision greater than 28 as variable-length strings.

If you are writing the data to an Oracle target and want to maintain the precision as a variable-length string, edit the target definition to modify the column datatype. Within a PowerCenter mapping, you can convert a variable-length string to a number either implicitly by connecting to a numeric port or explicitly by using expressions. To avoid loss of precision in implicit conversions, you might need to edit the mapping to pass the data as a string from source to target.

- PowerExchange supports the BINARY_DOUBLE and BINARY_FLOAT numeric datatypes by treating them as internal DOUBLE or FLOAT datatypes. However, PowerCenter converts BINARY_DOUBLE and BINARY_FLOAT datatypes to Oracle NUMBER(15) datatypes, which can result in arithmetic overflow and data loss.
- In columns with an Oracle numeric datatype, PowerExchange treats the value of infinity as 0.

Example dbmover Configuration File for the Oracle Change Capture System

This example dbmover configuration file contains the basic statements that are required on the Oracle system where PowerExchange Express CDC for Oracle initiates CAPI connections to Oracle for change capture.

The following dbmover configuration file corresponds to the example configuration, in which the Express CDC capture process and PowerExchange Logger run on the same system as the Oracle database:

```
LISTENER=(pwxlst1,TCPIP,2480)
NODE=(local,TCPIP,192.168.6.220,2480)
NODE=(pwxnode1,TCPIP,192.168.6.220,2480)
NODE=(oranode1,TCPIP,192.168.6.220,2480)
APPBUFFSIZE=256000
COLON=:
COMPRESS=N
CONSOLE_TRACE=N
DECPOINT=.
DEFAULTCHAR=*
DEFAULTDATE=19800101
MAXTASKS=60
MSGPREFIX=PWX
NEGSIGN=-
PIPE=|
POLLTIME=1000
TIMEOUTS=(300,600,600)
CAPT_PATH=/Informatica/PowerExchangeVR/capture
CAPT_XTRA=/Informatica/PowerExchangeVR/capture/extmaps
```

```

LOGPATH=/Informatica/PowerExchangeVR/capture/logs
CODEPAGE=(utf-8,utf-8,utf-8)
/*
/* Define the ORACLE_CAPTURE_TYPE statement to explicitly define
/* the CDC type.
ORACLE_CAPTURE_TYPE=D
/*
/* Define an ORACLEID statement for each Oracle instance involved
/* in CDC.
ORACLEID=(ORAD1DB,ORAD1,ORAD1DB,ORAD1DB)
/*
/* An ORAD CAPI_CONNECTION statement is required for Oracle
/* Express CDC.
CAPI_CONNECTION=(NAME=CAPORA3,TYPE=(ORAD,ORACOLL=ORAD1DB,
PARMFILE=/Informatica/PowerExchangeVR/capture/pwxorad.cfg))
/*
/* A CAPX CAPI_CONNECTION statement is required if you use
/* the PowerExchange Logger and continuous extraction mode.
CAPI_CONNECTION=(NAME=oralcpx,TYPE=(CAPX,DFLTINST=ORAD1DB))

```

The ORAD CAPI_CONNECTION is required on the PowerExchange system that initiates the CAPI connection to Oracle. Include the PARMFILE parameter to point to the PowerExchange Express CDC for Oracle configuration file when you do not use the default file name or location.

The ORACLE_CAPTURE_TYPE statement is not required on this system because an ORAD CAPI_CONNECTION statement is present. However, Informatica recommends that you include ORACLE_CAPTURE_TYPE for consistency. The ORACLE_CAPTURE_TYPE setting must be **D** to be consistent with the CAPI_CONNECTION type of ORAD.

RELATED TOPICS:

- [“PowerExchange Express CDC for Oracle Architecture” on page 133](#)

Customizing the PowerExchange Express CDC for Oracle Configuration File

You specify PowerExchange Express CDC for Oracle parameters in a configuration file that is separate from the dbmover.cfg file.

PowerExchange supplies the sample pwxorad.cfg file in the directory that is specified in the PWX_HOME environment variable, or if that variable is not defined, in the PowerExchange bin directory. The sample file contains comments that describe the required and optional statements and keywords.

Copy the sample file and customize the copy. If you copy the file under another name or to another directory, you must include the PARMFILE parameter in the ORAD CAPI_CONNECTION statement in the dbmover.cfg file to point to the customized copy.

The following table summarizes the statements that you can define in the PowerExchange Express CDC for Oracle configuration file:

Statement	Required or Optional	Description
ASMSTAGING	Optional	Enables PowerExchange Express CDC for Oracle to make a single call to Oracle ASM to dump chunks of redo log to a staging file. PowerExchange can then extract change data from the staging file by using standard file I/O. This process improves CDC performance, increases data throughput, and reduces CPU usage in an ASM environment.
CHECKPOINT	Optional	Enables PowerExchange Express CDC for Oracle checkpoint processing of in-flight transactions.
DATABASE	Optional	Provides overrides for database connection information, including the capture connection string and database user ID and password. Provides the TDE keystore type, location, and password, which are required for capturing changes from TDE-encrypted tablespaces. Indicates whether the source database is in an Amazon RDS for Oracle environment.
DICTIONARY	Required	Indicates whether PowerExchange gets the Oracle data dictionary information that it uses to interpret the redo logs from a dynamic dictionary or static dictionary and includes options for each mode.
DIRSUB	Optional	Specifies a path prefix that PowerExchange Express CDC for Oracle substitutes for the original path prefix that the Oracle server uses to access active and archived redo logs. This substitute path is required when the PowerExchange Express CDC log reader runs on a system other than the Oracle server and uses a different path to access the redo log files.
OPTIONS	Optional	Provides the following options for CDC processing: <ul style="list-style-type: none"> - Number of log records staged in memory. - Maximum spill file size. - Prefix to a temporary file path for the spill files. - Period that the Express CDC waits for the next archived log to become available after the online redo log is overwritten. - Whether PowerExchange uses connection resiliency to retry killed Oracle sessions. - Whether Express CDC detects and removes multibyte characters that are not valid and appear at the end of a character field in an Oracle source table. - Whether PowerExchange captures Oracle rowid value and direct-path operations. - Whether Express CDC ends or continues capture processing when encountering a DROP PARTITION operation. - Whether Express CDC reports all DDL operations that it detects for registered source tables.
RAC	Optional	Specifies the maximum number of redo log threads that PowerExchange Express CDC can track for member instances in an Oracle RAC from which you capture changes.

Statement	Required or Optional	Description
READER	Required	Provides options for reading the redo logs.
STANDBY	Optional	Defines a connection to an Oracle physical standby database when the database is not open for read-only access. This statement applies only when the CDC source is an Oracle physical standby database.
STATESTORAGE	Optional	Defines the file-based or table-based storage of state information. For the dynamic dictionary, the storage persists DDL changes from the in-memory dictionary for the source tables of CDC interest. For checkpointing, the storage records the state of in-flight transactions.

In each statement that you include, you must specify at least one valid keyword. The end of an entire statement is indicated by a semicolon (;).

RELATED TOPICS:

- [“Example PowerExchange Express CDC for Oracle Configuration Files” on page 201](#)
- [“DATABASE Statement” on page 183](#)
- [“CHECKPOINT Statement” on page 182](#)
- [“DICTIONARY Statement” on page 186](#)
- [“DIRSUB Statement” on page 188](#)
- [“OPTIONS Statement” on page 188](#)
- [“RAC Statement” on page 194](#)
- [“READER Statement” on page 195](#)
- [“STANDBY Statement” on page 199](#)
- [“STATESTORAGE Statement” on page 200](#)

ASMSTAGING Statement

The ASMSTAGING statement enables the use of staging files in an Oracle ASM environment to hold chunks of Oracle redo log for CDC processing. You can optionally use staging files to improve CDC performance, increase data throughput, and reduce ASM CPU usage, without having to implement Oracle dual logging outside of ASM.

When staging is enabled, if the amount of available redo log is greater than 32 KB, PowerExchange Express CDC can make a single call to ASM to request that chunks of redo log be written to a staging file that ASM generates in the staging directory you created. ASM generates one staging file for each active redo log thread. PowerExchange Express CDC can then use standard I/O communication to read data from a staging file. This process is much more efficient than having to perform many SQL*NET read operations on the logs to extract the same amount of data from ASM.

Syntax:

```
ASMSTAGING
  STAGINGDIR="/path/directory"
  [LOCALDIR="/localmountpoint/directory"]
  [TARGETSIZE={50|number}]
;
```

Parameters:**STAGINGDIR**

Required. The path to the directory that you created for the staging files on the machine where the ASM instance to which PowerExchange connects resides. Enclose this value in double-quotation marks ("), for example:

```
"/oracle/staging"
```

To optimize ASM data throughput, ensure that the staging directory is on a fast storage device that is local to the ASM instance.

If PowerExchange Express CDC for Oracle is remotely accessing the staging directory, ensure the directory is in NFS storage or another type of shared storage.

LOCALDIR

Optional. The local NFS mount point for the staging directory. Specify this value only if it is different from the STAGINGDIR path. Enclose the value in double-quotation marks ("), for example:

```
"/localmountpoint/staging"
```

TARGETSIZE

Optional. The maximum size, in megabytes, of any chunk of redo log to be copied to the staging file. This value also determines the maximum file size.

CHECKPOINT Statement

The CHECKPOINT statement enables PowerExchange Express CDC for Oracle to write checkpoints for in-flight transactions during CDC processing. Then, if a CDC process terminates and you warm start it, PowerExchange can rebuild any long running transactions and resume processing from the last checkpoint, without having to reprocess large amounts of redo log.

Note: The CHECKPOINT statement is incompatible with the AGEOUTPERIOD parameter in OPTIONS statement. Do not specify both.

Syntax:

```
CHECKPOINT
  [CHKPTINTERVAL=minutes]
  [COMPLETEDEVENTSAGE=minutes]
  [COMPLETEDEVENTSDIR=directory]
  [MAXCHKPTSIZE=kilobytes]
  [RETENTIONPERIOD=minutes]
  [MINCHKPTSTOKEEP=number_of_checkpoints]
  [EXCEPTIONS=(REJECT|TOLERATE)]
;
```

Parameters:**CHKPTINTERVAL**

The interval, in minutes, that must elapse between taking checkpoints. Valid values are 1 through 1440. The default value is 5 minutes.

COMPLETEDEVENTSAGE

The interval, in minutes, that must elapse before a transaction with active operations can be forced to spill to disk. Only transactions that are spilled to disk can be checkpointed.

Valid values are 0 through 1440. A value of 0 means never force a transaction with active operations to spill to disk for checkpointing. The default value is 30.

COMPLETEDEVENTSDIR

The file system directory to which spilled active transaction data is copied during checkpoint processing. In the path, you can use double slashes (\\) or single slashes (\). To use single slashes, you must enclose the entire path in double quotation marks ("), for example:

```
=="C:\\Users\\jdoe\\resources\\capt\\CKPTOPS"
```

If the COMPLETEDEVENTSAGE parameter value is greater than 0, you must specify either this COMPLETEDEVENTSDIR parameter or the BASEDIRECTORY parameter in the STATESTORAGE statement. The default value is the BASEDIRECTORY value.

EXCEPTIONS

Controls whether to tolerate missing spill file data or to reject this type of exception. Tolerate this exception only if you can allow some data loss rather than regress the restart point.

Options are:

- **REJECT**
- **TOLERATE**

Default is REJECT.

MAXCHKPTSIZE

The maximum amount of data, in kilobytes, that can be forced to spill to disk during a single checkpoint. You can use this parameter to block very large transactions from forcibly spilling to disk, which might improve performance during large batch processing. Valid values are 1 through 1024*1024 (or 1 GB). The default value is 50*1024 (or 50 MB).

MINCHKPTSTOKEEP

The minimum number of checkpoints to keep. Valid values are 2 through 480. The default value is 3.

RETENTIONPERIOD

The period, in minutes, for which checkpoint data is retained. Valid values are 0 through 44640. The default value is 10080 (1 week).

DATABASE Statement

The DATABASE statement specifies overrides for the capture connection string and the database user ID and password. The statement also provides options for capturing change data from TDE-encrypted tablespaces and from database instances in an Amazon Relational Database Service (RDS) for Oracle environment.

The DATABASE statement is optional. If you specify it, include at least one of its optional parameters.

If you want to capture change data from an Oracle Data Guard physical standby database that is open for read-only access, configure the DATABASE statement to connect to the standby database instance. If the standby database is *not* open for read-only access, configure the DATABASE statement to connect to the primary database instance and also define the STANDBY statement.

Syntax:

```
DATABASE
[CONNECT_STRING=capture_connect_string]
[EPWD=database_encrypted_password|PASSWORD=database_password]
[RDS={N|Y}]
[TDEKEYSTORE=(TYPE=({ASM|FILE|HSM},parameters...))]
[USERID=database_user_id]
;
```

Parameters:

CONNECT_STRING

An Oracle connection string, defined in TNS, that the PowerExchange Express CDC for Oracle uses to connect to the Oracle database.

This connection string is overridden by the **Connect String** value, if specified, in the CDC session connection attributes.

EPWD

An encrypted password that PowerExchange uses to connect to the Oracle source database for PowerExchange Express CDC for Oracle. If you specify the USERID parameter in this statement, you must also specify either the EPWD or PASSWORD parameter.

This encrypted password overrides the **Password** value, if specified, in the CDC session connection attributes.

This encrypted password is overridden by the EPWD parameter value in the ORAD CAPI_CONNECTION statement, if specified.

PASSWORD

A clear text password that PowerExchange uses to connect to the Oracle source database for PowerExchange Express CDC for Oracle. If you specify the USERID parameter in this statement, you must also specify either the EPWD or PASSWORD parameter.

This password overrides the **Password** value, if specified, in the CDC session connection attributes.

This password is overridden by the PASSWORD parameter value in the ORAD CAPI_CONNECTION statement, if specified.

RDS

Indicates whether the Oracle source database is in an Amazon Service (RDS) for Oracle environment. Options are:

- **Y.** The source database instance is deployed in Amazon RDS for Oracle. PowerExchange Express CDC reads change data from the database redo logs that are in the ONLINELOG_DIR and ARCHIVELOG_DIR directories. You create these directories on the RDS file system.
- **N.** The source database instance is not deployed in Amazon RDS for Oracle.

Default is N.

TDEKEYSTORE

If the Oracle source uses Transparent Data Encryption (TDE) to encrypt source tablespaces, provide the location of the keystore that contains the encryption key required for the encryption and decryption of data. The keystore can be located in Automatic Storage Management (ASM), a file, or a hardware security module (HSM). Depending on the type of store, enter this parameter in one of the following ways:

- For an ASM store, the syntax is:

```
TDEKEYSTORE=(TYPE={ASM, WALLETDIR=path/directory,{PWD|EPWD}=wallet_password,  
[ASM_CONNECT_STRING=string,ASM_PASSWORD=asm_password,ASM_USERID=asm_userid,ASM_ASS  
YSASM=Y])
```

Where:

- TYPE must be the store type of ASM.
- WALLETDIR is the fully qualified path to the ASM wallet.
- {PWD|EPWD} is the password or encrypted password that is required to access the ASM wallet.

- ASM_CONNECT_STRING is the Oracle connection string, defined in TNS, that the PowerExchange Express CDC log reader uses to connect to the ASM instance.
- ASM_PASSWORD is a password for the user specified in the ASM_USERID parameter, which is used to connect to the ASM instance.
- ASM_USERID is an Oracle user ID that the PowerExchange Express CDC log reader uses to connect to the ASM instance, which has SYSDBA or SYSASM authority.
- ASM_ASSYSASM must be set to Y to have the PowerExchange Express CDC log reader to use a user ID that has SYSASM authority to connect to the ASM instance.

Note: If you do not define ASM_CONNECT_STRING in the TDEKEYSTORE parameter, you must define the ASM connection parameters in the READER statement.

Example:

```
TDEKEYSTORE=(TYPE=(ASM,WALLETDIR="+DATA2/ORCL18/wallet/tde/",PWD=mypwd,ASM_CONNECT_STRING=ORCLASM,ASM_PASSWORD=asmpwd,ASM_USERID=SYS,ASM_ASSYSASM=Y))
```

- For a file store, the syntax is:

```
TDEKEYSTORE=(TYPE=(FILE,WALLETDIR=path/directory,[PWD|EPWD]=password))
```

Where:

- TYPE must be the store type of FILE.
- WALLETDIR is the fully qualified path to the file system wallet. PowerExchange Express CDC will not attempt to extract the path from the database.
- {PWD|EPWD} is the password or encrypted password that is required to access the wallet.

Example:

```
TDEKEYSTORE=(TYPE=(FILE,WALLETDIR="E:\orcl19\wallet\tde",PWD=mypwd));
```

- For an HSM store, the syntax is:

```
TDEKEYSTORE=(TYPE=(HSM,HSMCLIENTLIB=path/module,[PWD|EPWD]=password))
```

Where:

- HSMCLIENTLIB is the fully qualified path and module name of the HSM client library. For information about setting up access to an HSM device or server, see the HSM vendor documentation. PowerExchange Express CDC uses the standard PKCS #11 interface to access HSM keys.
- {PWD|EPWD} is the password or encrypted password that is required to access the HSM device.

Example:

```
TDEKEYSTORE=(TYPE=(HSM,HSMCLIENTLIB="E:\orcl19\okvutil\lib\liborapkcs19.dll",PWD=null))
```

If you define multiple TDEKEYSTORE parameters, PowerExchange Express CDC searches them for the master key in the order in which they are defined.

USERID

A user ID that PowerExchange uses to connect to the Oracle source database for PowerExchange Express CDC for Oracle. If you specify the USERID parameter in this statement, you must also specify either the EPWD or PASSWORD parameter.

This user ID overrides the **User Name** value, if specified, in the CDC session connection attributes.

This user ID is overridden by the USERID parameter value in the ORAD CAPI_CONNECTION statement, if specified.

Usage Notes: You can specify the connection string in multiple locations. If you do so, PowerExchange uses the following order of precedence:

1. The **Connect String** value, if specified, in the CDC session connection attributes
2. The fourth positional parameter of the ORACLEID statement in the dbmover.cfg file
3. The CONNECT_STRING parameter value in the DATABASE statement of the PowerExchange Express CDC for Oracle configuration file
4. The ORACLE_SID environment variable

If you specify none of these values, PowerExchange passes Null values in the OCI call.

DICTIONARY Statement

The DICTIONARY statement indicates where PowerExchange Express CDC for Oracle gets the data dictionary information that it uses to interpret redo logs. This statement also controls how PowerExchange Express CDC for Oracle reacts when it encounters DDL changes in the redo log records.

The DICTIONARY statement is required.

Syntax:

```
DICTIONARY
  MODE={STATIC|DYNAMIC}
  [SOURCE=ONLINE]
  [EXCEPTIONS={FAIL|WARN}]
  [RETENTIONPERIOD=hours]
  [SNAPSHOTONCOLDSTART={Y|N}]
  [UPDATEINTERVAL=minutes]
;
```

Parameters:

MODE

Required. Indicates whether PowerExchange Express CDC for Oracle expects the data dictionary to remain the same or change after it is read into memory.

In STATIC mode, PowerExchange Express CDC expects the data dictionary to remain the same. If structural changes to source tables occur, the EXCEPTIONS parameter determines whether PowerExchange fails or issues a warning. However, PowerExchange Express CDC always tolerates the following types of DDL changes and continues CDC processing, regardless of the EXCEPTIONS setting:

- ALTER TABLE ADD statements for adding one or more columns to a table
- ALTER TABLE ADD PARTITION statements
- ALTER TABLE ADD CONSTRAINT statements
- CREATE USER statements
- ALTER USER statements
- DROP USER statements

In DYNAMIC mode, PowerExchange Express CDC updates an in-memory dynamic dictionary when DDL changes to tables of CDC interest are committed. Express CDC persists the dictionary information over time either in a state table in the source database or in CCATALOG files in a file system. The types of DDL changes stored include:

- Reorganizing, redefining, or renaming tables of CDC interest
- Adding or dropping tables or partitions of CDC interest
- Adding, dropping, or altering table columns

In an environment where DDL changes to source tables are likely to occur, a dynamic dictionary can help you avoid the data loss or corruption that can otherwise occur with a static dictionary after a CDC warm start or PowerExchange Logger cold start. For example, use the dynamic dictionary in Oracle environments that use the Oracle Online Reorganization and Redefinition feature.

The default value is STATIC.

SOURCE

Required when MODE=STATIC. Indicates where PowerExchange Express CDC for Oracle gets its data dictionary information. The only valid value is ONLINE, which causes Express CDC to get the data dictionary information from the Oracle online system when the change capture process initializes.

EXCEPTIONS

Optional. When MODE=STATIC, controls whether PowerExchange Express CDC for Oracle fails or continues with a warning message when it detects that a structural change was made to an Oracle table from which change data is captured.

This parameter does not apply to ADD PARTITION changes. PowerExchange Express CDC for Oracle tolerates ADD PARTITION changes.

Valid values are:

- **FAIL.** CDC processing ends abnormally.
- **WARN.** PowerExchange prints a warning message, and CDC processing continues. This option can result in loss of change data. Use this option only at the direction of Informatica Global Customer Support.

Default is FAIL.

RETENTIONPERIOD

Optional. When MODE=DYNAMIC, the period, in hours, to retain data in the dynamic data dictionary. Operations are rolled up into the data dictionary base snapshot at Express CDC initialization and once every 24 hours thereafter until this retention period elapses. Valid values are 1 through 744. The default value is 168 hours (or 1 week).

SNAPSHOTONCOLDSTART

Optional. When MODE=DYNAMIC, set this parameter to Y if you want to force a snapshot of the database to be taken of the current Oracle catalog when a PowerExchange Logger cold start is requested. The default value is N, which causes no snapshot to be taken.

If you use the default value of N and cold start the PowerExchange Logger when data exists in the dynamic dictionary and the cold start restart point SCN is greater than the dynamic dictionary SCN, the cold start uses the dictionary SCN as the restart point. This behavior ensures that the dictionary contains any changes that occur in the gap between the restart points.

UPDATEINTERVAL

Optional. When MODE=DYNAMIC, the interval, in minutes, that must elapse before PowerExchange Express CDC advances the SCN value in the data dictionary, during a period when no DDL changes of interest are occurring for a data source. When a restart occurs, PowerExchange Express CDC uses the data dictionary restart SCN when calculating how far back in the redo logs to start processing change records so that no DDL changes of interest are missed.

Valid values are 1 through 1440. The default value is 0, which causes the RSTRADV parameter value in the OPTIONS statement to be used. The actual restart SCN is the lesser of the PowerExchange Logger restart SCN and the dynamic dictionary restart SCN.

DIRSUB Statement

The DIRSUB statement specifies a path prefix that PowerExchange Express CDC substitutes for the original path prefix of the redo logs on the Oracle server. This substitute path is required when the PowerExchange Express CDC log reader runs on a system other than the Oracle server and uses a different mapping to access the redo log files.

You might need to add a DIRSUB statement in the following cases:

- The redo logs reside on shared disk.
- The redo logs have been copied to the system where Express CDC runs.
- The archived redo logs are accessed by using a different NFS mount.

Note: Do not use this statement if you use Oracle Automatic Storage Management (ASM) to manage the redo logs that PowerExchange Express CDC needs to access.

You can specify multiple DIRSUB statements. Each one must have a unique value for the original path prefix that Oracle server uses. PowerExchange compares the original path prefix against redo log directories and file names, starting with the longest string for a file. For the matching log files, the PowerExchange Express CDC log reader uses the corresponding substitute path prefix to access the redo logs.

Syntax:

```
DIRSUB SERVER="original_path_prefix",LOCAL="substitute_path_prefix";
```

If you want to include a trailing backslash in a Windows path, you must enter a double backslash (\\).

Parameters:

SERVER

Required. The original path prefix for the redo logs on the Oracle server. This value must be unique in each DIRSUB statement.

LOCAL

Required. The substitute path prefix that the PowerExchange Express CDC log reader uses to access redo logs. This value does not have to be unique across multiple DIRSUB statements.

Example: The Oracle server and PowerExchange Express CDC run on separate Linux systems. The redo logs reside in an NFS-mounted directory on shared disk. In the following DIRSUB statement, the path to the redo logs on the Oracle server is "/ora01/oraarchlogs/ORAB11," and the path that Express CDC uses to access the redo logs on shared disk is "oracle/oralogs/orab":

```
DIRSUB SERVER="/ora01/oraarchlogs/ORAB11",LOCAL="/oracle/oralogs/orab";
```

OPTIONS Statement

The OPTIONS statement specifies parameters for controlling CDC processing. The parameters control memory use, spill file size, capture of rowid values, long outstanding UOWs, and the wait period for advancing restart tokens when no change capture activity is occurring.

The OPTIONS statement is optional, and all of the parameters in the statement are optional. If you specify the OPTIONS statement, include at least one parameter.

Syntax:

```
OPTIONS
  [AGEOUTPERIOD=minutes]
  [BCE_SUBSTITUTE_YEAR=year_value]
  [BCE_YEAR_HANDLING=IGNORESIGN|SUBSTITUTE|PRE1040|FAIL]
  [CONNRETRYMAX=number]
  [CONNRETRYWAIT=seconds]
  [LARGEOPS=number_of_operations]
```

```

[LOGARCHIVEWAIT=seconds]
[MEMOPS=number_of_log_records]
[MONITOR_INTERVAL=minutes]
[PARTITION_DROP_FAIL={Y|N}]
[REPORTDDL={Y|N}]
[RETRYONKILLSESSION={Y|N}]
[ROWID={Y|N}]
[ROW_MOVEMENT_FAIL={Y|N}]
[RSTRADV=seconds]
[SPILLENCRYPTPASS=passphrase]
[SPILLENCRYPTPASS=encrypted_passphrase]
[SPILL_FILE_PREFIX=spill_file_name_prefix]
[SPILLFILEBUFFSZ=kilobytes]
[SPILLMAX=kilobytes]
[SUPPORT_DIRECT_PATH_OPS={Y|N}]
[TIME_STAMP_MODE={LOGTIME|COMMITTIME|BEGINTIME}]
[TRUNCINVALIDCHARS={Y|N}]
;

```

Parameters:

AGEOUTPERIOD

The age, in number of minutes, that an outstanding UOW that has no change records of CDC interest must reach before it is removed from the calculation of the CDC restart point. The age is calculated as the time difference between the start of the outstanding UOW and the start of the most recent UOW. This age-out processing occurs during the monitoring interval.

Use this parameter to prevent CDC failures that can occur if you shut down and then restart capture processing while the transaction is outstanding. After the restart, the archived redo log in which the outstanding UOW started might not be available, causing the Express CDC log reader to fail.

Valid values are 60 to 43200. By default, no value is specified and this parameter is disabled.

Oracle stores all time values in the log in local time. As a result, at the beginning or end of daylight saving time, a UOW might age out an hour late or an hour early.

Note: The AGEOUTPERIOD parameter is incompatible with the CHECKPOINT statement. Do not specify both.

BCE_SUBSTITUTE_YEAR

Specifies the substitute year value to use when the BCE_YEAR_HANDLING parameter is set to SUBSTITUTE. Valid values are 1 to 9999.

BCE_YEAR_HANDLING

Indicates how to handle a negative year in Before the Common Era (BCE) dates.

Options:

- **IGNORESIGN.** Default. Capture the year as a positive value.
- **SUBSTITUTE.** Substitute the year with value supplied by the BCE_SUBSTITUTE_YEAR parameter. If you specify BCE_YEAR_HANDLING=SUBSTITUTE you must specify a value for BCE_SUBSTITUTE_YEAR.
- **PRE1040.** Revert to the old behavior of reporting a negative value (-CCYY) as a positive value (YY00).
- **FAIL.** Report BCE dates as errors through message PWX-36200 and shut down the PowerExchange Logger for Linux, UNIX, and Windows.

Default is IGNORESIGN.

CONNRETRYMAX

The maximum number of times that PowerExchange tries to reconnect to an Oracle source database or ASM instance. Use this parameter in conjunction with the CONNRETRYWAIT parameter if you get multiple PWX-36086 messages followed by an Oracle error that indicates a dropped connection or if you set the RETRYONKILLSESSION parameter to Y.

Note: The CONNRETRYMAX, CONNRETRYWAIT, and RETRYONKILLSESSION parameters help improve connection resiliency.

Valid values are 0 to 86400. A value of 0 results in no retries. Default is 12.

CONNRETRYWAIT

The number seconds that PowerExchange waits between attempts to reconnect to an Oracle source database or ASM instance. Use this parameter in conjunction with CONNRETRYMAX parameter if you get multiple PWX-36086 messages followed by an Oracle error that indicates a dropped connection or if you want to tune connection retries for killed sessions. This parameter can help improve connection resiliency. Valid values are 1 to 300. Default is 5 seconds.

LARGEOPS

Overrides the default value that PowerExchange uses to identify transactions as large transactions for reporting purposes. Enter the number of log records that a transaction must process to be considered a large transaction. A log record might contain multiple DML operations or part of a single operation.

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 fifth of the MEMOPS value, rounded to the nearest thousand. For example, if the MEMOPS value is 5120, the default LARGEOPS value is 1000 (1,000,000 operations).

LOGARCHIVEWAIT

After an Oracle online redo log starts being overwritten, the number of seconds that PowerExchange Express CDC waits for the copy of the log to become available as a new archived redo log for change data capture processing. In an Oracle Data Guard environment, if Express CDC captures change data from a physical standby database, this parameter specifies the number of seconds that Express CDC waits for the next archived redo log to be transported from the primary database to the standby database.

Valid values are 0 through 86400. Default is 30. If you use any value less than the value of the STATUSCHECKINTERVAL parameter in the READER statement, Express CDC waits for the STATUSCHECKINTERVAL period.

MEMOPS

The maximum number of redo log records that contain DML operations that PowerExchange can stage in memory while processing Oracle transactions.

Valid values are 1000 through 1048576. Default is the greater of 5120 or the number of RAC members * 2 * 1024. For example, if you have three RAC members, the default is 6144.

MONITOR_INTERVAL

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.

PARTITION_DROP_FAIL

Controls whether PowerExchange Express CDC for Oracle ends with an error or continues processing when the log reader encounters an ALTER TABLE DROP PARTITION operation for a registered Oracle source table.

Options:

- **Y.** Express CDC processing ends with error message PWX-36332, which reports the log position of the DROP PARTITION operation that caused the failure.
- **N.** Express CDC ignores the DROP PARTITION operation and continues change capture processing. Message PWX-36390 reports that Express CDC encountered a DROP PARTITION operation for a source table.

Default is Y.

REPORTDDL

Controls whether PowerExchange Express CDC reports all of the DDL operations that it encounters in the Oracle redo logs for Oracle source tables with active capture registrations. Express CDC writes the following information for each DDL operation to a file that is generated in the directory from which Express CDC runs: the DDL statement, log position, owner number, DDL object number, and sequence number. The file naming conventions are:

- For RAC systems:

```
PWX_ORL_DDL_Dyyyymmdd_Thhmmss.MBRnode_sequence#.rpt
```

- For non-RAC systems:

```
PWX_ORL_DDL_Dyyyymmdd_Thhmmss.sequence#.rpt
```

In these file names, the *sequence#* is a generated number that starts from 0001 and that is incremented by 1 for each new file. A new file is generated every 20 MB of DDL change records.

Options are:

- **Y.** Generate the report of DDL operations for registered source tables.
- **N.** Do not generate the report of DDL operations.

Default is N.

RETRYONKILLSESSION

Controls whether PowerExchange Express CDC can detect when an Oracle KILL SESSION event occurs for a specific PowerExchange connection to an Oracle source instance and then retry the connection so that the Express CDC log reader and PowerExchange Logger for Linux, UNIX, and Windows process do not end abnormally. A KILL SESSION event occurs when a user issues the following SQL statement:

```
ALTER SYSTEM KILL SESSION 'sid,serial_number' [IMMEDIATE]
```

In this statement, the variable *sid* is the session ID and the variable *serial_number* is the session serial number, as shown in the V\$SESSION view.

Options:

- **N.** PowerExchange does not try to re-establish killed sessions for Express CDC processes. If a KILL SESSION event occurs, the PowerExchange Express CDC log reader and PowerExchange Logger end abnormally.

- **Y.** PowerExchange tries to re-establish killed sessions for Express CDC processes. Also set the CONNRETRYMAX parameter to a value greater than 0 to indicate the maximum number of times that PowerExchange retries the connection to the source instance.

Tip: The RETRYONKILLSESSION, CONNRETRYMAX, and CONNRETRYWAIT parameters help improve connection resiliency.

Default is N.

Important: Before setting this parameter to Y, consult with your Oracle database administrator to ensure that no unintended consequences occur.

ROWID

Controls whether Oracle physical rowid values are included in captured change records for tables that do not have Oracle row movement enabled. PowerExchange writes the rowid values to the PowerExchange-generated DTL__CAPXROWID column.

For example, you might want to use this parameter if you have unkeyed source tables on which you need to perform some processing that requires a unique row ID when extraction sessions run.

Valid values are:

- **N.** Do not capture rowid values. The DTL__CAPXROWID column contains null values.
- **Y.** Capture rowid values. The DTL__CAPXROWID column contains rowid values for tables that do not have row movement enabled.

Default is N.

ROW_MOVEMENT_FAIL

When ROWID=Y, controls whether PowerExchange Express CDC for Oracle processing fails or continues if PowerExchange detects that row movement is enabled for a source table.

You might want to continue processing if you do not need to capture rowid values for tables that have row movement enabled.

Valid values are:

- **Y.** CDC processing fails if PowerExchange encounters a table with row movement enabled.
- **N.** CDC processing continues. For any table that has row movement enabled, the DTL__CAPXROWID column contains null values.

Default is Y.

RSTRADV

The number of 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.

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.

Valid values are 0 through 86400. No default is provided.

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.

Note: 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.

SPILLENCRYPTPASS

A passphrase that enables spill file encryption. PowerExchange writes transactions to spill files when it does not have sufficient memory to store them. You can optionally encrypt the spill files to protect sensitive data by setting either this parameter or the SPILLENCRYPTEPASS parameter. The maximum passphrase length is 1024 bytes. No default value is provided.

Note: If you change the passphrase while checkpointing is enabled, the checkpointed spill files become unreadable. Therefore, change the passphrase only when checkpointing is disabled or if you plan to perform a cold start.

SPILLENCRYPTEPASS

An encrypted passphrase that enables spill file encryption. PowerExchange writes transactions to spill files when it does not have sufficient memory to store them. You can optionally encrypt the spill files to protect sensitive data by setting either this parameter or the SPILLENCRYPTPASS parameter. The maximum passphrase length is 1024 bytes. No default value is provided.

Note: If you change the passphrase while checkpointing is enabled, the checkpointed spill files become unreadable. Therefore, change the passphrase only when checkpointing is disabled or if you plan to perform a cold start.

SPILLFILEBUFFSZ

The block buffer, in kilobytes, that is allocated for spill file I/O. Increase this value to reduce physical I/O and improve performance if an increase in memory usage is acceptable.

Valid values are 0 through 1024. Default is 0, which causes the operating system to allocate the block buffer when the first write operation occurs. The size of the buffer then depends on the operating system.

SPILL_FILE_PREFIX

A file name prefix to a temporary file path for the spill files. If you want to specify the directory where the spill files reside, specify the fully qualified path to the temporary directory.

Linux or UNIX example:

```
SPILL_FILE_PREFIX="/tmp/spillfiles/"
```

Windows example:

```
SPILL_FILE_PREFIX="C:\tmp\spillfiles\xxx"
```

Default path is the TMP or TMPDIR environment variable value. If TMP or TMPDIR is not defined, the current directory is used.

SPILLMAX

The maximum size, in kilobytes, of a single PowerExchange Express CDC for Oracle spill file on UNIX. PowerExchange writes transactions to spill files when it does not have sufficient memory to store them. After PowerExchange processes all of the transactions in a spill file, the spill file is freed.

A large spill file retains disk space longer before being freed than a smaller-sized spill file.

Valid values are 1 through 2097151. Default is 10240.

SUPPORT_DIRECT_PATH_OPS

Controls whether Express CDC captures or ignores Oracle direct-path operations, such as direct-path Inserts, on registered source tables.

Options:

- **Y.** Enables the capture of direct-path operations.

Note: Express CDC does not capture direct-path operations for tables that use Oracle Exadata Hybrid Columnar Compression (EHCC).

- **N.** Disables the capture of direct-path operations. If the log reader encounters a redo log record for a direct-path operation, Express CDC ignores the direct-path operation with an informational message and continues.

Default is N.

TIME_STAMP_MODE

The type of timestamp that PowerExchange records in the generated DTL__CAPXTIMESTAMP column of each change record for a transaction. Usually, you specify this parameter only if you want to display the Oracle commit timestamp instead of the Oracle log timestamp.

Options are:

- **LOGTIME.** The timestamp that Oracle periodically writes to the Oracle archived logs.
- **COMMITTIME.** The timestamp of the transaction commit on the source database. Specify this option if you use the timestamp to calculate latency.
- **BEGINTIME.** The timestamp of the begin UOW log record.

Default is LOGTIME.

TRUNCINVALIDCHARS

Indicates whether to detect and remove invalid multibyte characters that appear at the end of a character field in an Oracle source table. The characters are invalid because they have been truncated. If you allow these invalid characters to be passed to a PowerCenter workflow that has an Oracle target, PowerCenter might corrupt subsequent columns in the target table when running in Unicode mode.

Valid values are:

- **Y.** Remove invalid multibyte characters from the source data. The invalid characters are not passed to PowerCenter or applied to the target database.
- **N.** Capture the invalid multibyte characters and pass them to PowerCenter. When the PowerCenter workflow tries to write the invalid characters to the target, PowerCenter might corrupt data in the subsequent columns in the target table.

Default is N.

RAC Statement

The RAC statement specifies the maximum number of active redo log threads with unique thread IDs that PowerExchange Express CDC for Oracle can track in the Oracle Real Application Cluster (RAC). Active threads include threads that have a status of open or closed. If the number of active threads is greater than this parameter value, CDC processing ends.

You must define this statement if you use PowerExchange Express CDC in an Oracle RAC environment.

Syntax:

```
RAC MEMBERS=number_of_threads;
```

Parameter:**MEMBERS**

The maximum number of active redo log threads in the RAC that PowerExchange Express CDC can track. For a Data Guard physical standby database that supports a primary database in a RAC environment, this value is the number of active threads for the primary database.

Valid values are 1 to 100. Default is 1.

Informatica recommends that you enter the lowest value that is suitable for your RAC environment to minimize the overhead of PowerExchange Express CDC tracking of threads. To determine the lowest value, you can run one of the following queries:

- For a RAC database, use one of the following queries:
 - If each open thread and instance in the v\$thread view is also identified in the v\$spparameter view, use:

```
select count(*) from
v$thread a,v$spparameter b
where a.status != 'DISABLED' and
b.name = 'thread' and
b.sid = a.instance and
b.value = TO_CHAR(a.thread#);
```

- If each open thread and instance in the v\$thread view is *not* identified in the v\$spparameter view, use:

```
select count(*) from
v$thread
where status != 'DISABLED';
```

- For a Data Guard physical standby database, use the following query:

```
select distinct(thread#)
from v$standby_log
where thread# != 0;
```

READER Statement

The READER statement provides parameters that the PowerExchange Express CDC log reader uses to read redo log files, issue log-read status messages, connect to an Oracle Automatic Storage Management (ASM) instance, and check the status of the Oracle server during periods of inactivity.

The READER statement is required.

Syntax:

```
READER
  MODE={ACTIVE|ARCHIVEONLY|ARCHIVECOPY}
  [ACTIVELOGMASK=mask]
  [ARCHIVEDEST=(log_destination1, log_destination2, log_destination3,...)]
  [ASM_ASSYSASM={Y|N}]
  [ASM_CONNECT_STRING=tns_connect_string]
  [ASM_EPWD=encrypted_password]
  [ASM_PASSWORD=password]
  [ASM_USERID=user_id]
  [DIR=base_directory_for_archived_log_copies]
  [FILE=mask_for_archived_log_copies]
  [READBUFFSIZE=kilobytes]
  [STATUSCHECKINTERVAL=hundredths_of_seconds]
  [STATUSREPORTINTERVAL=seconds]
;
```

Parameters:

MODE

Required. An option that indicates the source of and types of redo logs that the PowerExchange Express CDC log reader reads. Valid options are:

- **ACTIVE.** The PowerExchange Express CDC for Oracle log reader reads active and archived redo logs from the Oracle online system. With this option, you can run the PowerExchange for Linux, UNIX, and Windows in either continuous or batch mode. Optionally, you can use the `ACTIVELOGMASK` parameter to filter the active redo logs and use the `ARCHIVEDEST` parameter to limit the archived log destinations from which to read archived logs.
- **ARCHIVEONLY.** The PowerExchange Express CDC log reader only reads archived redo logs. Optionally, you can use the `ARCHIVEDEST` parameter to limit the archived log destinations from which to read archived logs. After reading all of the available archived logs, the log reader checks for additional archived logs to read based on the `STATUSCHECKINTERVAL` parameter.

In this mode, the log reader determines the current end of log (EOL) at initialization by using one of the following values:

- For non-RAC instances, the high SCN of the last log archived.
- For RAC instances, the lowest high SCN of the last log archived across all active nodes.

This option is a suitable choice when the PowerExchange Logger for Linux, UNIX, and Windows runs in batch mode and shuts down at EOL. In this case, you should coordinate the PowerExchange Logger runs with Oracle log file switches.

Note: If the PowerExchange Logger runs in continuous mode, this option can increase CDC latency because the log reader is idle for long periods while waiting for archived logs to become available.

- **ARCHIVECOPY.** The PowerExchange Express CDC log reader reads archived redo logs that have been copied to an alternate file system. Use this option in the following situations:
 - You do not have the authority to access the Oracle archived redo logs directly.
 - The archived redo logs are written to ASM, but you do not have access to ASM.
 - An aggressive archived log retention policy is in effect on the database server, which might cause the archived logs to not be retained long enough.

If you use this option, run the PowerExchange Logger in batch mode. If you run the PowerExchange Logger in continuous mode, the PowerExchange Logger might fail if it reads the archive copy before the copy is fully written.

You must implement a script to copy the archived redo logs from their primary location to the alternate location. To copy the archived logs, you can use any method that does not corrupt them, for example, FTP in binary mode. You must also specify the `DIR` parameter to indicate the name of base directory that the log reader scans for the copies of the archived logs. Optionally, you can use the `FILE` parameter to filter the copies of the archived logs that reside under the base directory.

Unlike the other `MODE` options, `ARCHIVECOPY` identifies candidate archived redo logs by scanning the file system directories. This process ignores the `ARCHIVEDEST` parameter and does not filter candidate logs by their `DELETED` status in `v$log`.

Default is `ACTIVE`.

ACTIVELOGMASK

Optional. When the `MODE` parameter is set to `ACTIVE`, this parameter specifies a mask for selecting active redo logs for the PowerExchange Express CDC log reader when the Oracle instance uses multiplexing of redo logs. The log reader compares the mask against the member names in an active redo log group to determine which log to read. In the mask, you can use the asterisk (*) wildcard to represent zero or more characters.

The mask can be up to 128 characters in length. The mask is case-sensitive on Linux or UNIX systems but not on Windows systems.

ARCHIVEDEST

Optional. Specifies a list of numbers from 1 to 10, separated by commas. Each number represents the *n* value in an Oracle LOG_ARCHIVE_DEST_*n* initialization parameter. If you create more than one copy of each archive log, this parameter indicates the primary and secondary log destinations from which PowerExchange Express CDC reads archived logs. You can specify up to 10 destinations. The order in which you specify the destinations in the list determines the order in which PowerExchange Express CDC searches for archive logs. PowerExchange Express CDC tries to process the first two valid logs it finds in the specified destinations. If PowerExchange cannot read the first two logs, the process terminates.

PowerExchange Express CDC uses the secondary destination when the primary destination becomes unavailable or when the logs at the primary destination are unreadable, for example, because they have been corrupted or deleted.

For example, the Oracle source database uses the Oracle parameters LOG_ARCHIVE_DEST_1, LOG_ARCHIVE_DEST_2, and LOG_ARCHIVE_DEST_3 to create copies of archived logs. If you want PowerExchange Express CDC to use LOG_ARCHIVE_DEST_2 as the primary destination and ARCHIVE_DEST_3 as the secondary destination, set ARCHIVEDEST=(2,3).

Note: Specifying one value for the ARCHIVEDEST parameter limits the resilience of the Express CDC log reader because the log reader then processes logs from only one archive log destination.

If you do not specify ARCHIVEDEST, PowerExchange Express CDC queries v\$aarchived_log for valid logs, favoring file system logs over ASM logs for performance reasons. Specifying ARCHIVEDEST provides more control over this process. For example, if the database writes archive logs to ASM and the file system, and you do not have access to the file system logs, you can use ARCHIVEDEST to specify the ASM location only.

ASM_ASSYSASM

Optional. If you want the PowerExchange Express CDC log reader to use a user ID that has SYSASM authority to connect to the ASM instance, set this parameter to Y. Then specify a user ID that has SYSASM authority in the ASM_USERID parameter. If you want to use a user ID that has SYSDBA authority, set this parameter to N. Default value is N.

ASM_CONNECT_STRING

Optional. In an Oracle ASM environment, the Oracle connection string, defined in TNS, that the PowerExchange Express CDC log reader uses to connect to the ASM instance that manages storage of active and archived redo logs for the source database.

Note: If you have TDE-encrypted source tablespaces and store the keystore in ASM, you must specify the ASM_CONNECT_STRING, ASM_USERID, and ASM_PASSWORD or ASM_EPWD parameter in this statement if you did not specify the ASM connection parameters in the TDEKEYSTORE parameter of the DATABASE statement.

ASM_EPWD

Optional. In an Oracle ASM environment, an encrypted password for the user that is specified in the ASM_USERID parameter. The PowerExchange Express CDC log reader uses this password and the ASM user ID to connect to the ASM instance that manages storage of active and archived redo logs for the source database. Define ASM_EPWD or ASM_PASSWORD but not both.

ASM_PASSWORD

Optional. In an Oracle ASM environment, a clear text password for the user that is specified in the ASM_USERID parameter. The PowerExchange Express CDC log reader uses this password and the ASM

user ID to connect to the ASM instance that manages storage of active and archived redo logs for the source database. Define ASM_EPWD or ASM_PASSWORD but not both.

ASM_USERID

Optional. In an Oracle ASM environment, an Oracle user ID that the PowerExchange Express CDC log reader uses to connect to the ASM instance that manages storage of active and archived redo logs for the source database. This user ID must have SYSDBA or SYSASM authority. To use SYSASM authority, set the ASM_ASSYSASM parameter to Y.

DIR

When MODE parameter is set to ARCHIVECOPY, this parameter is required. It specifies the name of the base directory that PowerExchange Express CDC log reader scans for the copies of the archived redo logs to read. To filter the copies of the logs that reside under this base directory, you can also specify the FILE parameter.

FILE

Optional. When the MODE parameter is set to ARCHIVECOPY, you can use this parameter to specify a mask for filtering the copies of the archived redo logs that the PowerExchange Express CDC log reader reads. PowerExchange matches the mask against the subdirectories and files under the base directory that is specified in the DIR parameter. Enter a mask for the subdirectory name, log file names, or both.

For example, the mask `/LOGS/*.DBF` causes the PowerExchange Express CDC log reader to scan the LOGS subdirectory under the base directory for all copies of the archived log files that have the file name extension of .DBF. If the subdirectories have different names and you want the log reader to scan all of the subdirectories for all copies of the archived logs, you can use the asterisk (*) wildcard as a subdirectory mask and file-name mask. For example, when `DIR=P:\oracle\orcl\archlogs` and the subdirectories are named after the log-copy date, the mask `**` could match the following copies of the archived logs:

```
P:\oracle\orcl\archlogs\2016-05-01\archlog1
P:\oracle\orcl\archlogs\2016-05-01\archlog2
P:\oracle\orcl\archlogs\2016-05-02\archlog10
P:\oracle\orcl\archlogs\2016-05-02\archlog11
```

READBUFFSIZE

Optional. The default buffer size, in kilobytes, that the PowerExchange Express CDC log reader uses to read a redo log. PowerExchange Express CDC for Oracle can automatically expand this buffer size, if necessary.

Valid values are 1 through 262144. Default is 10240.

STATUSCHECKINTERVAL

Optional. The time interval, in hundredths of seconds, that the PowerExchange Express CDC log reader waits at EOL when no additional data is available to read, before checking with Oracle on the following items:

- In ACTIVE mode, whether the active redo log file is still valid and available.
- In ARCHIVEONLY mode, if additional Oracle archive logs are available to read.
- Whether the number of data blocks that the log reader read matches the number of data blocks that Oracle wrote.

Valid values are 1 through 8640000. Default is 200.

Note: If PowerExchange issues message PXW-36171, the number of blocks read did not match the number of blocks written. In this case, try increasing the STATUSCHECKINTERVAL value so that any stale NFS read buffer has time to refresh and accept new data for the log reader to process.

STATUSREPORTINTERVAL

Optional. The frequency, in seconds, at which the PowerExchange Express CDC log reader issues message PWX-36151 to report log read progress.

Valid values are 1 through 86400. Default is 120.

STANDBY Statement

The STANDBY statement defines a connection to an Oracle physical standby database for change data capture when the database is *not* open for read-only access.

The STANDBY statement is optional. Use it only when the database is not open for read-only access. To access a database that is not open, you must have SYSDBA authority.

Syntax:

```
STANDBY
CONNECT_STRING=capture_connect_string
[APPLYACTIVE={N|Y}]
[EPWD=database_encrypted_password|PASSWORD=database_password]
[USERID=database_user_id]
;
```

Parameters:

APPLYACTIVE

Indicates whether PowerExchange Express CDC can process the standby redo logs on the standby system up to the highest low SCN when the Oracle does not progress apply processing beyond the tip of the last archived log, even though more recent changes are available in the standby logs. Set this parameter to Y if you want to perform near-real-time capture in this situation. Default is N.

CONNECT_STRING

An Oracle connection string, defined in TNS, that PowerExchange Express CDC for Oracle uses to connect to the Oracle physical standby database for change capture when the database is not open for read-only access.

EPWD

An encrypted password that PowerExchange Express CDC for Oracle uses to connect to the Oracle physical standby database for change capture.

You must specify either the EPWD or PASSWORD parameter, but do not specify both.

PASSWORD

A clear text password that PowerExchange Express CDC for Oracle uses to connect to the Oracle physical standby database for change capture.

You must specify either the PASSWORD or EPWD parameter, but do not specify both.

USERID

A user ID that PowerExchange Express CDC for Oracle uses to connect to the Oracle physical standby database for change capture. This user ID must have SYSDBA authority.

Usage Notes:

- To access the Oracle data dictionary on the primary system, PowerExchange Express CDC for Oracle uses the DATABASE statement, which points to the primary system when the database is not open for read-only access to the logs.

STATESTORAGE Statement

The STATESTORAGE statement defines the type of storage for state information and the connection string and credentials for accessing the storage. It is required to use the PowerExchange Express CDC checkpointing and dynamic dictionary features.

For checkpointing, the state information consists of the state of in-flight transactions at the time a CDC process terminates. After a restart, this information can be used to resume CDC processing from the last checkpoint, without having to reprocess lots of logs. For the dynamic dictionary, the state information is based on a snapshot of the Oracle catalog and is updated when DDL changes on source tables of CDC interest are committed, including create, rename, and reorganization operations and drops of tables or tables partitions, and column. With the data dictionary, CDC can correctly restart after DDL changes without data loss or corruption.

Syntax:

```
STATESTORAGE
  STORETYPE={Directory|Table}
  BASEDIRECTORY="path\\directory"
  [CONNECT_STRING=connect_string]
  [PREFIX=collection_id]
  [TABLEOWNER=user_id]
  [USERID=user_ID]
  [PASSWORD=password]
  [EPWD=encrypted_password]
;
```

Parameters:

STORETYPE

Required. The type of storage for state information. You can use a relational table or a file system directory. Options are:

- **Directory**
- **Table**

No default value is provided.

BASEDIRECTORY

Required. The base directory under which subdirectories that contain files or tables with state information are generated. Specify this parameter if the STORETYPE parameter is set to **Directory** or if you plan to enable checkpointing. In the path, you can use double slashes (\\) or single slashes (\). If you use single slashes, enclose the entire path in double quotation marks ("), for example:

```
BASEDIRECTORY="C:\Users\jdoe\resources\capt"
```

The Express CDC user must have read and write permissions on the directory. To create subdirectories, the user also requires the execute permission.

Note: For table-based checkpoint storage, you must specify either this BASEDIRECTORY parameter or the COMPLETEDEVENTSDIR parameter in the CHECKPOINT statement. Express CDC will copy the checkpointed spill files to this directory.

CONNECT_STRING

Optional. A connect string that is used to connect to the Oracle database where the state tables are created. The default value is the CONNECT_STRING value in the DATABASE statement.

PREFIX

Optional. A unique prefix for the names of the files or tables where state information is stored. Specify this prefix only if you want to override the default value, which is the collection ID specified in the ORACOLL parameter in the ORAD CAPI_CONNECTION statement. The ORACOLL parameter supplies a

unique value for each Express CDC instance that is running. If you use the optional PREFIX parameter, ensure that it also provides a unique value for the Oracle instance.

TABLEOWNER

Optional. The owner of the state tables. The default value is the USERID value in this statement, if specified, or the USERID value in the DATABASE statement.

USERID

Optional. A database user ID for accessing the state tables. The default value is the USERID value in the DATABASE statement.

PASSWORD

Optional. A password, which is used in conjunction with the user ID, to provide credentials for accessing the state tables. The default value is the PASSWORD value in the DATABASE statement. Do not also specify the EPWD parameter.

EPWD

Optional. An encrypted password, which is used in conjunction with the user ID, to provide credentials for accessing the state tables. The default value is the EPWD value in the DATABASE statement. Do not also specify the PASSWORD parameter.

Example PowerExchange Express CDC for Oracle Configuration Files

The minimum statements that you must specify in the PowerExchange Express CDC for Oracle configuration file vary by type of data dictionary. The following examples show the minimum statements for each dictionary type:

If you plan to use a static dictionary, specify at least the following statements:

```
DICTIONARY
MODE=STATIC
SOURCE=ONLINE;
READER
MODE=ACTIVE;
```

If you plan to use a dynamic dictionary with a state storage files in a file system, specify at least the following statements:

```
DICTIONARY
MODE=DYNAMIC;
STATESTORAGE
STORETYPE=DIRECTORY
BASEDIRECTIONS="C:\\path\\directory";
```

Managing PowerExchange Express CDC for Oracle

After CDC is running, you might need to perform some tasks to maintain and manage your PowerExchange Express CDC for Oracle environment occasionally.

These tasks include:

- Monitoring CDC processing
- Adding a capture registration
- Stopping CDC processing for a table
- Changing the structure of a table

Monitoring PowerExchange Express CDC for Oracle

To determine the progress of the Express CDC log reader in reading the redo logs, look for key messages in the PowerExchange message log file.

When a redo log switch occurs, PowerExchange Express CDC for Oracle issues messages such as the following example messages:

```
110324 125031 WIN32 2216 PWX-36145 ORAD Info: Low SCN 0x0000.05c2b16e.0000[96645486].  
Low SCN Time 03/24/2011 01:34:58.  
110324 125031 WIN32 2216 PWX-36146 ORAD Info: Next SCN 0x0000.05c32597.0000[96675223].  
Next SCN Time 03/24/2011 07:11:09.  
  
110324 125051 PWX-36144 ORAD Info: Reader processing active Log file \\s160020\fdrive  
\ORACLE\PRODUCT\ORADATA\ORCL\REDO03.LOG, SEQ 1797.  
110324 125051 PWX-36145 ORAD Info: Low SCN 0x0000.05c32597.0000[96675223]. Low SCN Time  
03/24/2011 07:11:09.
```

To determine how far behind the Express CDC log reader is in processing the redo logs, look at each PWX-36145 message and compare the time when the message was issued to the Low SCN Time in the message. Then compare this time interval across multiple PWX-36145 messages. The Low SCN represents the lowest SCN in the redo log, and the Next SCN is the SCN at the time the log was closed.

Use the example messages to perform the analysis:

- In the first PWX-36145 message, compare the message time of 12:50:31 to the Low SCN Time of 1:34:58. The difference is about 11 hours 15 minutes.
- In the second PWX-36145 message, compare the message time of 12:50:51 to the Low SCN Time of 7:11:09. The difference is about 5 hours 40 minutes.

In this case, the large reduction in the time interval indicates that the Express CDC log reader is catching up on reading change records from the redo logs.

Also, the PWX-36151 message, which is issued periodically, gives an indication of PowerExchange Express CDC for Oracle progress. When the Express CDC log reader is either catching up or falling behind, this message indicates that the log reader is reading an archived log, an active log, or the current log. After the log reader has caught up, this message indicates that the log reader is reading at the tip of the current log sequence. The following example PWX-36151 message indicates that the log reader has caught up with reading changes in the current log:

```
110324 145352 WIN32 7844 PWX-36151 ORAD Info: Reading at the tip of the current log  
sequence 1798, block 28475. Low SCN 0x0000.05c3bee9.0000[96714473], Low SCN time  
03/24/2011 14:53:48.
```

In this message, the Low SCN Time is the earliest time that the Oracle log writer recorded in the last span of the log that was read by the Express CDC log reader. To determine change capture latency, you can compare the Low SCN Time to the time when the message was issued, in a manner similar to the preceding analysis.

Adding Another Capture Registration

After PowerExchange Express CDC for Oracle is running, you might need to add a capture registration for another Oracle table.

1. In PowerExchange Navigator, create the capture registration.

Make sure that you include the following settings:

- In the **Condense** list, select **Part**.
- In the **Supplement Log Group Name** box, enter a name for the supplemental log group for the table. The PowerExchange Navigator generates DDL for creating the supplemental log group. If you select

Execute DDL now, the PowerExchange Navigator runs the DDL when you complete the registration. If you do not have the authority to run the DDL, ask your DBA to run it.

- In the **Status** list, select **Active**.
2. Run the DDL for a creating a supplemental log group for the table, if you did not enable the PowerExchange Navigator to do so at registration completion.
 3. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger and then warm start it.
The PowerExchange Logger begins capturing change data for the additional table.
 4. In PowerCenter, import the extraction map for the table to create a source definition.
 5. Add the source definition to a new mapping or to an existing mapping.
If you add the source definition to an existing mapping, you must the stop the workflow first.
 6. Start the workflow that processes the added source table.

Stopping CDC Processing for a Table

If you no longer need to capture change data for a table, you can stop PowerExchange Express CDC for Oracle processing for the table.

1. In PowerExchange Navigator, open the capture registration and set the **Status** option to **History**.
A capture registration that has a status of **History** cannot be activated again.
2. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger and then warm start it.
3. Drop the supplemental log group for the table by using the following SQL:

```
ALTER TABLE schema.table_name DROP SUPPLEMENTAL LOG GROUP
```

Oracle stops logging full before-images and after-images of column data.

Note: If you were to add the supplemental log group for the table again because you had to reinstate change capture, you would need to rematerialize the target.

Stopping CDC Processing Temporarily

You might need to temporarily stop PowerExchange Express CDC for Oracle processing to troubleshoot issues or to perform a maintenance task on the target database.

If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger to stop CDC processing for all source tables. Later, you can warm start the PowerExchange Logger to resume change capture processing without any change data loss. This method is preferable.

Changing the Structure of an Oracle Source Table

Occasionally, you might need to make DDL changes to a registered Oracle source table that add, alter, or drop columns from which PowerExchange Express CDC for Oracle captures changes. Learn how to switch to the new table definition in a manner that preserves access to previously captured data.

You do not need to do this task in the following situations:

- You selectively capture change data for a subset of columns, and the DDL changes do not affect any of these columns.

- You need to stop change data extraction processing for a column. In this case, remove the column from the extraction map and do not edit the capture registration. PowerExchange still captures change data for the column but does not extract it when CDC sessions run.

1. Stop data change activity (inserts, updates, and deletes) on the table.
2. Verify that any change data that was captured under the current table definition has completed extraction processing. Then stop all PowerCenter workflows that extract change data for the table.
3. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the Logger.
4. In the PowerExchange Navigator, open the original capture registration and set its status to **History**. PowerExchange does not capture change data based on capture registrations that have a status of **History** or **Inactive**.

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

5. Drop the supplemental log group for the table.
6. Make the DDL changes to the table.
7. In the PowerExchange Navigator, create a new capture registration for the table that reflects the DDL changes.

Make sure that you include these settings:

- In the **Condense** list, select **Part**.
 - In the **Supplement Log Group Name** box, enter a name for the supplemental log group that must be created for the table. The PowerExchange Navigator generates DDL for creating the supplemental log group. If you select **Execute DDL now**, the PowerExchange Navigator runs the DDL when you complete the registration. If you do not have the authority to run the DDL, you can ask your DBA to run it.
 - In the **Status** list, select **Active**.
8. If you shut down the PowerExchange Logger, warm start it.
The PowerExchange Logger begins capturing changes based on the new capture registration.
 9. Change the target table definition to reflect the source table changes, if necessary.
 10. In PowerCenter Designer, import the new extraction map for the altered source table to create a new source definition. Also, if you changed the target table, edit or re-create the target definition. Then, edit the mapping, if necessary.
 11. If necessary, rematerialize the target tables and then create new restart tokens.
 12. Allow change activity on the table to resume.
 13. Start the workflows again.
Extraction processing resumes.

Reporting DDL Operations for Registered Oracle Source Tables

You can configure PowerExchange Express CDC for Oracle to report the DDL operations that it encounters in the Oracle redo logs for Oracle source tables with active capture registrations.

To enable DDL reporting, you must specify the optional `REPORTDDL=Y` parameter in the `OPTIONS` statement of the `pwxorad.cfg` configuration file. For more information, see [“OPTIONS Statement” on page 188](#).

PowerExchange Express CDC reports all DDL operations that it detects in the Oracle redo logs for registered tables, such as `ALTER TABLE` operations that add, drop, or modify a column or that add or drop a partition. Express CDC writes the following information for each DDL operation to a generated file: the DDL statement,

log position, owner number, DDL object number, and sequence number. The file is generated in the directory from which Express CDC runs, which is usually the root PowerExchange installation directory. The file naming conventions are:

- For RAC systems:

```
PWX_ORL_DDL_Dyyyymmdd_Thhmmss.MBRnode_sequence#.rpt
```

- For non-RAC systems:

```
PWX_ORL_DDL_Dyyyymmdd_Thhmmss.sequence#.rpt
```

In these file names, *sequence#* is a generated number that starts from 0001 and that is incremented by 1 for each new file. A new file is generated every 20 MB of DDL change records.

Example report:

The following report shows two DDL operations on the same registered Oracle source table:

```
--DDL found at Location : redo log position SCN 0x0000.00ff2e7a.0001 (16723578) RBA
0x0007cd.0000509c.0010 (file: 1997)
-- Owner Number : 111, DDL Object 95944 Sequence 1 of total 1
-- DDL String :
alter table TSTV11.DDLTEST001 add COL03 varchar2(10) default NULL

--DDL found at Location : redo log position SCN 0x0000.00ff2e85.0001 (16723589) RBA
0x0007cd.000050a9.0010 (file: 1997)
-- Owner Number : 111, DDL Object 95944 Sequence 1 of total 1
-- DDL String :
alter table TSTV11.DDLTEST001 add COL04 varchar2(10) default 'xxxx'
```

Usage notes:

- The report of DDL operations is primarily for informational and diagnostic use. You can use it to determine the DDL operations that capture processing skipped. The reported DDL statements are not intended to be used directly to update targets. If you need make DDL changes to a registered source table, you should still follow the recommended procedure described in [“Changing the Structure of an Oracle Source Table” on page 203](#).
- If you generate the report in a RAC environment and no DDL operations occurred on one of the nodes, the report for that node is empty except for the lines "Starting DDL Report" and "Ending DDL Report."
- If you change the capture registration status from active to history or delete the registration for a source table, PowerExchange Express CDC stops reporting DDL operations for the table but continues reporting DDL operations for the other tables that have active registrations.
- If you have been using Express CDC to capture change data and want to start reporting DDL operations for registered source tables, define the REPORTDDL=Y parameter in the OPTIONS statement of the pwxorad.cfg file and then restart the PowerExchange Logger for Linux, UNIX, and Windows.

CHAPTER 8

PostgreSQL CDC

This chapter includes the following topics:

- [PostgreSQL CDC Overview, 206](#)
- [PostgreSQL CDC Considerations, 207](#)
- [PostgreSQL Datatypes Supported for CDC, 208](#)
- [Implementation Task Flow, 210](#)
- [Preparing PostgreSQL CDC Sources, 211](#)
- [Configuring PowerExchange for PostgreSQL CDC, 212](#)
- [Managing PostgreSQL CDC, 215](#)

PostgreSQL CDC Overview

PowerExchange can capture Insert, Update, Delete, and Truncate operations for PostgreSQL source tables. PowerExchange does not capture DDL changes other than Truncate operations.

The PostgreSQL database must have logical replication enabled. Also, each source table must have a primary key. Most PostgreSQL data types are supported.

PowerExchange retrieves change data from a PostgreSQL logical replication slot named `pw_x_repl`, which uses a PowerExchange-installed plugin named `pw_x_decode.dll` or `pw_x_decode.so`. The replication slot is generated for PowerExchange use when capture processing runs. PowerExchange stores the change data in the *replication store table* in the source database. PowerExchange automatically creates the replication store table if it does not already exist during capture processing. Alternatively, you can manually create the table by using the DDL statements in [“Manually Creating the Replication Store Table” on page 212](#).

PowerExchange requires the DataDirect ODBC driver for PostgreSQL to retrieve source metadata from the PostgreSQL database server. The PowerExchange installation delivers this ODBC driver in the `root_install_directory\ODBCv.r\Drivers` folder on Windows or `root_install_directory/ODBCv.r/lib` directory on Linux.

Use of the PowerExchange Logger for Linux, UNIX, and Windows is optional. If you use the PowerExchange Logger, the Logger extracts change data from the replication store table and stores the data in its log files.

When a CDC session runs, PowerExchange works with the PowerExchange Client for PowerCenter (PWXPC) to extract change records from the real-time change stream or from the PowerExchange Logger logs and to transmit the data to one or more targets.

RELATED TOPICS:

- [“Implementation Task Flow” on page 210](#)
- [“Preparing PostgreSQL CDC Sources” on page 211](#)
- [“Configuring PowerExchange for PostgreSQL CDC” on page 212](#)
- [“Managing PostgreSQL CDC” on page 215](#)

PostgreSQL CDC Considerations

Review the following operational considerations for PostgreSQL CDC.

- For registered PostgreSQL source tables, PowerExchange captures DML changes (inserts, updates, and deletes) but does not capture DDL changes.
- Each PostgreSQL source table must have a primary key before it can be registered for CDC.
- For PostgreSQL CDC, use of the PowerExchange Logger for Linux, UNIX, and Windows is optional.
- The PowerExchange Logger reads capture registrations for PostgreSQL sources only when it starts. If you change the registrations while the PowerExchange Logger is running, you can warm start the Logger and the changes will take effect.
- The replication store can grow large in size unless you take one of the following actions:
 - Specify `ENABLELWM=Y` in the PG `CAPL_CONNECTION` statement. With this setting, change data that has been hardened to PowerExchange Logger log files will be deleted from the replication store table automatically. To retain the table data longer, you can back up the replication store table.
 - Manually delete data from the replication store table by using the following SQL command:

```
DELETE FROM store_table_name [WHERE condition]
```

If you omit the WHERE clause, all table rows are deleted.
- You might need to take some action if the following DDL changes are made to registered source tables:
 - If you drop a registered source table, no action is required because no DML changes will be captured for the dropped table. However, you might want to remove the associated capture registration, which is no longer needed.
 - If you add columns to a source table, PowerExchange continues CDC processing but no data from the new columns is processed. To process data from these columns, you must re-create the capture registration for the source table and then warm start the CDC session.
 - If you delete columns that are included in the capture registration for a source table, PowerExchange issues an error message and the session stops. You must re-create the capture registration without selecting the columns and then warm start the CDC session.
- If you stop the CDC session, change data continues to be cached on the PostgreSQL server in the `pg_xlog` file. To avoid errors, make sure that the `pg_xlog` file does not become full.
- PowerExchange does not support EnterpriseDB (EDB) PostgreSQL.
- PowerExchange uses the DataDirect ODBC driver for PostgreSQL to connect to PostgreSQL source databases. You can define the `ODBC_CONN_PARAMS` statement in the `dbmover.cfg` file if you want to add ODBC parameters to the connection strings that are used on the servers you specify for the following processing:
 - Change data capture

- Creating, deleting, or modifying capture registrations from the PowerExchange Navigator or DBLUCBRG utility

For example, you might need to add ODBC parameters if your site policies require database connections to use SSL encryption, specific cryptographic protocols, or self-signed or third-party signed SSL certificates. For more information, see the *PowerExchange Reference Manual*.

- PostgreSQL CDC only returns accurate before-image values on UPDATE and DELETE operations for key columns. All non-key columns report a NULL value. To replicate the before-image (BI) and change-indicator (CI) values, alter the table by using an `ALTER TABLE <tblname> REPLICA IDENTITY FULL` statement.

PostgreSQL Datatypes Supported for CDC

Verify that the PostgreSQL columns for which you plan to capture change data have datatypes that PowerExchange supports for change data capture.

The following table shows the supported PostgreSQL datatypes:

Datatype	Supported for CDC?	Comments
abstime	Yes	PowerExchange treats this datatype as timestamp.
array types	Yes	PowerExchange treats these datatypes as varchar.
bigint	Yes	PowerExchange treats this datatype as num64.
bit	Yes	PowerExchange treats this datatype as char.
bit varying	Yes	PowerExchange treats this datatype as varchar.
boolean	Yes	PowerExchange treats this datatype as num8.
box	Yes	PowerExchange treats this datatype as varchar.
bytea - hex and escape formats	Yes	PowerExchange treats these datatypes as varbin.
character	Yes	PowerExchange treats this datatype as char.
character varying	Yes	PowerExchange treats this datatype as varchar.
cid	Yes	PowerExchange treats this datatype as num64.
cidr	Yes	PowerExchange treats this datatype as varchar.
circle	Yes	PowerExchange treats this datatype as varchar.
date	Yes	PowerExchange treats this datatype as varchar.
daterange	Yes	PowerExchange treats this datatype as varchar.

Datatype	Supported for CDC?	Comments
double precision	Yes	PowerExchange treats this datatype as double.
inet	Yes	PowerExchange treats this datatype as varchar.
int4range	Yes	PowerExchange treats this datatype as varchar.
int8range	Yes	PowerExchange treats this datatype as varchar.
integer	Yes	PowerExchange treats this datatype as num32
interval	Yes	PowerExchange treats this datatype as varchar.
json types	Yes	PowerExchange treats these datatypes as varchar.
line	Yes	PowerExchange treats this datatype as varchar.
lseg	Yes	PowerExchange treats this datatype as varchar.
macaddr and macaddr8	Yes	PowerExchange treats these datatypes as varchar.
money	Yes	PowerExchange treats this datatype as varchar.
name	Yes	PowerExchange treats this datatype as varchar.
numeric	Yes	PowerExchange treats this datatype as numchar, or If the precision is greater than 100 or the scale is greater than 50, PowerExchange treats this datatype as varchar.
numrange	Yes	PowerExchange treats this datatype as varchar.
oid	Yes	PowerExchange treats this datatype as num64.
path	Yes	PowerExchange treats this datatype as varchar.
point	Yes	PowerExchange treats this datatype as varchar.
polygon	Yes	PowerExchange treats this datatype as varchar.
pg_lsn	Yes	PowerExchange treats this datatype as varchar.
real	Yes	PowerExchange treats this datatype as float.
regclass	Yes	PowerExchange treats this datatype as varchar.
regconfig	Yes	PowerExchange treats this datatype as varchar.
regdictionary	Yes	PowerExchange treats this datatype as varchar.
regnamespace	Yes	PowerExchange treats this datatype as varchar.
regoper	Yes	PowerExchange treats this datatype as varchar.
regoperator	Yes	PowerExchange treats this datatype as varchar.

Datatype	Supported for CDC?	Comments
regproc	Yes	PowerExchange treats this datatype as varchar.
regprocedure	Yes	PowerExchange treats this datatype as varchar.
regrole	Yes	PowerExchange treats this datatype as varchar.
regtype	Yes	PowerExchange treats this datatype as varchar.
reltime	Yes	PowerExchange treats this datatype as varchar.
smallint	Yes	PowerExchange treats this datatype as num16.
text	Yes	PowerExchange treats this datatype as varchar.
time with time zone	Yes	PowerExchange treats this datatype as time.
time without time zone	Yes	PowerExchange treats this datatype as time.
timestamp with time zone	Yes	PowerExchange treats this datatype as varchar.
timestamp without time zone	Yes	PowerExchange treats this datatype as varchar.
tsquery	Yes	PowerExchange treats this datatype as varchar.
tsrange	Yes	PowerExchange treats this datatype as varchar.
tstzrange	Yes	PowerExchange treats this datatype as varchar.
tsvector	Yes	PowerExchange treats this datatype as varchar.
user-defined types, including enum	Yes	PowerExchange treats these datatypes as varchar.
uuid	Yes	PowerExchange treats this datatype as bin.
xid	Yes	PowerExchange treats this datatype as num64.
xml	Yes	PowerExchange treats this datatype as varchar.

Implementation Task Flow

To implement PostgreSQL CDC, you need to complete some tasks in PostgreSQL, PowerExchange, and PowerCenter. Use the following high-level task flow:

1. Prepare the PostgreSQL source environment. For more information, see [“Preparing PostgreSQL CDC Sources” on page 211](#).

2. Create capture registrations and extraction maps for the PostgreSQL source tables with either the PowerExchange Navigator or the DTLUCBRG utility. For more information, see the *PowerExchange Navigator User Guide* and *PowerExchange Utilities Guide*.
3. Add a PG CAPI_CONNECTION statement and other required statements to the DBMOVER configuration file on the system where the PowerExchange capture registrations and extraction maps are stored. See [“Configuring the dbmover Configuration File” on page 212](#).
4. If you use the PowerExchange Logger for Linux, UNIX, and Windows, configure the PowerExchange Logger. See [“Configuring the PowerExchange Logger” on page 44](#).
5. Configure a CDC restart point. See [“Creating Restart Tokens for Extractions” on page 283](#).
6. Materialize the target tables. Use any tool that you prefer. PowerExchange bulk data movement does not currently support PostgreSQL sources.
7. Start the PowerExchange Logger, if configured. See [“Cold Starting the PowerExchange Logger ” on page 70](#).
8. Create PowerCenter CDC workflows that include the PostgreSQL sources. Use a PostgreSQL CDC application connection. For more information, see the *PowerExchange Interfaces for PowerCenter* publication.
9. Cold start the CDC workflows.

Preparing PostgreSQL CDC Sources

To prepare a PostgreSQL source system for PowerExchange CDC, you must perform a few configuration tasks.

1. Create a PostgreSQL user role that allows PowerExchange to connect to the PostgreSQL database server in logical replication mode to create a replication slot. When you create the role, specify the LOGIN and REPLICATION attributes, for example:

```
CREATE ROLE pwx_role LOGIN REPLICATION;
```

To use this command, you must have the CREATEROLE or privilege or be a database superuser.

2. Ensure that the PostgreSQL postgresql.conf configuration file specifies the **wal_level=logical** parameter. This parameter determines how much information PostgreSQL writes to the Write-Ahead Log (WAL). The setting of **logical** adds information that is required to support logical decoding.
3. Install the pwx_decode.dll plug-in for creating the replication slot that PowerExchange uses, in one of the following ways:

- Copy the pwx_decode.dll plug-in from the PowerExchange root directory to the PostgreSQL lib directory.

When the PowerExchange capture process runs, PostgreSQL uses the pwx_decode.dll plugin to create the "pwx_repl" replication slot.

- If you do not have the authority to install the plug-in within the PostgreSQL lib directory, you can copy pwx_decode.dll to another location. You then must manually create the replication slot. To create the replication slot, use the following command, which points to the location of pwx_decode.dll:

```
SELECT slot_name FROM pg_create_logical_replication_slot('pwx_repl', 'path/
pwx_decode')
```

Note: PowerExchange automatically creates the corresponding replication store table that records the DML changes retrieved from the replication slot, if the table does not already exist. If you need to adjust

the replication store table for your source environment, you can manually create the table. See [“Manually Creating the Replication Store Table” on page 212](#).

Manually Creating the Replication Store Table

If you need to customize the replication store table to suit your environment, you can manually create the table instead of letting the PowerExchange capture process automatically generate it.

For example, you might want to customize the table to support table partitions.

To create the table, use the following sample SQL statements:

```
CREATE TABLE table_name(  
    id bigserial,  
    lsn pg_lsn NOT NULL,  
    xid xid NOT NULL,  
    data text NOT NULL);  
CREATE UNIQUE INDEX ON table_name (id);  
CREATE INDEX ON table_name (lsn);
```

Notes:

- Make sure that your customizations are compatible with the column definitions in the sample SQL.
- If you use a table name other than the default name of `public.pwx_repl`, enter the custom name in the `REPLSTORETBL` parameter in the `PG CAPI_CONNECTION` statement in the `dbmover.cfg` file.

Configuring PowerExchange for PostgreSQL CDC

You must complete several PowerExchange configuration tasks to prepare for PostgreSQL CDC. The tasks depend on whether you use the PowerExchange Logger for Linux, UNIX, and Windows and the type of extraction mode you plan to use.

Configuring the dbmover Configuration File

In the `dbmover` configuration file on the system where the capture registrations and CDC control files are stored, add the statements that are required for PostgreSQL CDC. This system is specified in the **Location** field of the registration group definition.

The following statements are also required:

PG CAPI CONNECTION

A named set of parameters that the Consumer API (CAPI) uses to connect to the change stream and control CDC processing for source tables on a PostgreSQL source database server.

CAPT_PATH

Path to the local directory on a Linux or Windows system that stores the control files for CDC.

These files include the CCT file for capture registrations, the CDEP file for application names that are used for ODBC extractions, and the CDCT file for the PowerExchange Logger for Linux, UNIX, and Windows.

CAPT_XTRA

Path to the local directory on a Linux or Windows system that stores the extraction maps for CDC.

If you plan to use continuous extraction mode, you must also include the following statement:

CAPX CAPI_CONNECTION

A named set of parameters that the CAPI uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files.

Also, Informatica recommends including the LOGPATH and TRACING statements to make finding messages easier. The LOGPATH statement defines a directory specifically for PowerExchange message log files, and the TRACING statement enables PowerExchange to create an alternative set of message log files for each PowerExchange process.

For more information about all DBMOVER statements, see the *PowerExchange Reference Manual*.

CAPI_CONNECTION - PG Statement

The PG 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 PostgreSQL sources.

Operating Systems: Windows

Data Sources: PostgreSQL

Required: Yes for PostgreSQL CDC

Syntax:

```
CAPI_CONNECTION=(NAME=capi_connection_name
                  [,DLLTRACE=trace_id]
                  ,TYPE=(PG
                        ,SERVER={database_server|localhost}[,port]
                        ,DATABASE=database_name
                        [,ENABLELWM={Y|N}]
                        [,FETCHLIMIT=maximum_rows_fetched|100]
                        [,ONDATA TRUNC={WARN|FAIL}]
                        [,ONTABLE TRUNC={WARN|FAIL}]
                        [,RECONNTRIES={reconnection_attempts|12}]
                        [,RECONNWAIT={seconds|5}]
                        [,REPLSTORETBL=schema.table_name|public.pwx_repl]
                        [,RSTRADV=seconds]
                        )
                  )
```

Parameters:

NAME=capi_connection_name

Required. A unique user-defined name for this CAPI_CONNECTION statement.

Maximum length is eight alphanumeric characters.

DLLTRACE=trace_ID

Optional. The 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.

TYPE=(PG, ...)

Required. The type of CAPI_CONNECTION statement. For PostgreSQL sources, this value must be PG.

SERVER={host_name|localhost}[,port_number]

Required. The host name or IP address of the server where the PostgreSQL source database runs. You can enter "localhost" if the database server and PowerExchange Listener run locally on the same machine.

If you want the source server to listen on a port number other than the default port, you can optionally append the port number to the server name.

DATABASE=*database_name*

Required. The name of the PostgreSQL source database on the specified server.

ENABLELWM={*N*|*Y*}

Optional. When you use the PowerExchange Logger for Linux, UNIX, and Windows, controls whether the PowerExchange consumer API (CAPI) capture process deletes data read from the replication store table after the data has been hardened to PowerExchange Logger log files. You can use this parameter to improve capture performance and to prevent the replication store table from growing too large when the PowerExchange Logger is in use.

Options are:

- **N.** Do not delete processed data from the replication store table after the data has been hardened to the PowerExchange Logger log files. With this option, the replication store table might become very large, causing CDC performance to be degraded.
- **Y.** Delete processed data from the replication store table after the data has been hardened to the PowerExchange Logger log files. After a log file switch, the PowerExchange Logger sends a low water marker (LWM) to the CAPI process to identify the last end UOW prior to the file switch. At the end of the next capture cycle, after the CAPI connection process has read to the end of the available data, the CAPI deletes all of the rows in the replication store table up to the LWM.

Default is N.

FETCHLIMIT

Optional. The maximum number of rows that can be fetched from the replication store table in response to a PowerExchange SELECT call. Valid values are 0 through 1000000. A value of 0 means no maximum limit is in effect.

Default is 100.

ONDATA TRUNC={*WARN*|*FAIL*}

Optional. Indicates whether PowerExchange issues a warning message and continues processing or ends abnormally when it needs to truncate data from PostgreSQL columns that are longer than 98,304 bytes.

Default is FAIL.

ONTABLE TRUNC={*WARN*|*FAIL*}

Optional. Indicates whether PowerExchange issues a warning message and continues processing or ends abnormally when it encounters a TRUNCATE TABLE record in the change stream.

Default is FAIL.

RECONNTRIES={*reconnection_attempts*|*12*}

Optional. The maximum number of times that PowerExchange tries to reconnect to the PostgreSQL database server after a connection is dropped while PowerExchange is retrieving data from the replication slot or reading data from the replication store table. If a connection retry is unsuccessful, PowerExchange waits for number of seconds specified in the RECONNWAIT parameter before trying to reconnect to the database server again. Use this parameter in conjunction with the RECONNWAIT parameter to improve connection resiliency.

Valid values are 0 through 2147483647. A value of 0 results in no connection retries. Default is 12.

RECONNWAIT={seconds|5}

Optional. The number of seconds that PowerExchange waits between attempts to reconnect to a PostgreSQL database server. Use this parameter in conjunction with the RECONNTRIES parameter to improve connection resiliency.

Valid values are 0 through 3600. A value of 0 results in no waiting between connection attempts. Default is 5.

REPLSTORETBL={schema.table_name|public.pwx_repl}

Optional. The name of the PowerExchange replication store table that contains the change data from the PostgreSQL logical replication slot. PowerExchange reads change data from this table.

If this value is not specified, the default name of public.pwx_repl is used.

RSTRADV=seconds

Optional. 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.

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

- PowerExchange finishes 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.

Valid values 0 through 86400. No default is provided. A value of 0 disables restart advance processing.

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.

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.

Managing PostgreSQL CDC

After CDC is running, you might need to perform some occasional tasks to maintain and manage your PostgreSQL CDC environment.

These tasks include:

- Stopping PostgreSQL CDC processing.
- Changing the structure of a source table.
- Adding a capture registration.

Stopping Change Data Capture for a PostgreSQL Table

You might want to stop change data capture for a PostgreSQL source table if the table has been dropped, change activity no longer occurs on the table, or the data in the table is no longer of interest for CDC processing.

1. In PowerExchange Navigator, open the capture registration and set the **Status** option to **History**.
A capture registration that has a status of **History** cannot be activated again.
2. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger and then warm start it.
This step refreshes the registration information that the PowerExchange Logger uses.
3. In PowerCenter, delete or update CDC workflows to make sure that no workflow processes the removed table.

Stopping PostgreSQL CDC Processing Temporarily

You might need to temporarily stop PostgreSQL CDC processing to troubleshoot issues or to perform a maintenance task on the target database.

If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger to stop CDC processing for all source tables. Later, you can warm start the PowerExchange Logger to resume change capture processing without any change data loss. This method is preferable.

Changing the Structure of a PostgreSQL Source Table

Occasionally, you might need to make DDL changes to a registered PostgreSQL source table that add, alter, or drop columns from which PowerExchange captures changes. You can switch to the new table definition in a manner that preserves access to previously captured data.

This topic describes the steps for properly switching to a new table definition.

Note: You do not need to perform these steps in the following situations:

- You selectively capture change data for a subset of columns, and the DDL changes do not affect any of these columns or their ordinal values.
- You need to stop change data extraction processing for a column. In this case, remove the column from the extraction map and do not edit the capture registration. PowerExchange still captures change data for the column but does not extract it when CDC sessions run.

1. Stop data change activity (inserts, updates, and deletes) on the table.
2. Verify that any change data that was captured under the current table definition has completed extraction processing. Then stop all PowerCenter workflows that extract change data for the table.
3. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the Logger.
4. In the PowerExchange Navigator, open the original capture registration and set its status to **History**.

PowerExchange does not capture change data based on capture registrations that have a status of **History** or **Inactive**.

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

5. Make the DDL changes to the table.

6. In the PowerExchange Navigator, create a new capture registration for the table that reflects the DDL changes.

Make sure that you include these settings:

- In the **Condense** list, select **Part**.
- In the **Status** list, select **Active**.

7. If you shut down the PowerExchange Logger, warm start it.

The PowerExchange Logger begins capturing changes based on the new capture registration.

8. Change the target table definition to reflect the source table changes, if necessary.
 9. In PowerCenter Designer, import the new extraction map for the altered source table to create a new source definition. Also, if you changed the target table, edit or re-create the target definition. Then, edit the mapping, if necessary.
 10. If necessary, rematerialize the target tables and then create new restart tokens.
 11. Allow change activity on the table to resume.
 12. Start the PowerCenter workflows again.
- Extraction processing resumes.

CHAPTER 9

SAP HANA CDC

This chapter includes the following topics:

- [SAP HANA CDC Overview, 218](#)
- [SAP HANA CDC Considerations, 218](#)
- [SAP HANA Datatypes Supported for CDC, 219](#)
- [SAP HANA CDC Log Tables, 220](#)
- [Implementation Task Flow, 220](#)
- [Preparing the SAP HANA Server for PowerExchange CDC, 221](#)
- [Preparing the SAP HANA Database for PowerExchange CDC, 223](#)
- [Configuring PowerExchange for SAP HANA CDC, 225](#)
- [Managing SAP HANA CDC, 227](#)

SAP HANA CDC Overview

PowerExchange CDC for SAP HANA captures inserts, updates, and deletes from SAP HANA tables on Windows or Linux. Because SAP HANA does not provide direct access to its log files, the capture process uses triggers on source tables to determine when to capture change data. Enter a short description of the concept here (required).

The PowerExchange Navigator can optionally create scripts for the PKLOG and PROCESSED log tables and trigger DDL when you create capture registrations for the source tables. If you do not use the Navigator to create the tables, PowerExchange creates the log tables and triggers when the extraction process runs.

The triggers write insert, update, and delete changes to the PKLOG transaction log table. These changes are read in table order and are then available for extraction to PowerCenter CDC sessions or PowerExchange CDC Publisher. PowerExchange provides only after-image records for SAP HANA CDC.

SAP HANA CDC Considerations

Review the following considerations for PowerExchange CDC for SAP HANA:

- SAP HANA CDC uses both the ODBC driver and JDBC driver for SAP HANA. You must install and configure these drivers separately from the PowerExchange installation. For more information, see [“Preparing the SAP HANA Server for PowerExchange CDC” on page 221](#).

- Each SAP HANA source table must contain a primary key before it can be registered for CDC. PowerExchange supports primary keys with up to six columns in each registered source table.
- You must create a separate capture registration for each SAP HANA source table.
- SAP HANA CDC requires the PowerExchange Logger for Linux, UNIX, and Windows.
- PowerExchange supports SAP HANA table names up to 120 characters. Tables with names longer than 120 characters cannot be registered for change data capture.

SAP HANA Datatypes Supported for CDC

Verify that the SAP HANA columns for which you plan to capture change data have datatypes that PowerExchange supports. PowerExchange CDC supports all native SAP HANA datatypes except spatial types.

The following table shows the SAP HANA datatypes that PowerExchange supports and does not support for CDC:

Datatype	Supported for CDC?	Comments
ALPHANUM(n)	Yes	
ARRAY	--	This datatype is not supported.
BIGINT	Yes	
BINARY(n)	Yes	
BINTEXT	No	This datatype is not supported.
BLOB	No	This datatype is not supported.
BOOLEAN	Yes	
CLOB	No	This datatype is not supported.
DATE	Yes	This datatype is only supported if precision and scale are explicitly specified.
DECIMAL(p,s)	Yes	
DOUBLE	Yes	
FLOAT(n)	Yes	
INTEGER	Yes	
NCLOB	No	This datatype is not supported.
NVARCHAR(n)	Yes	
REAL	Yes	

Datatype	Supported for CDC?	Comments
SECONDDATE	Yes	
SHORTTEXT(n)	Yes	
SMALLDECIMAL	Yes	
SMALLINT	Yes	
ST_POINT	No	This datatype is not supported.
ST_GEOMETRY	No	This datatype is not supported.
TEXT	No	This datatype is not supported.
TIME	Yes	
TIMESTAMP	Yes	
TINYINT	Yes	
VARBINARY(n)	Yes	
VARCHAR(n)	Yes	

SAP HANA CDC Log Tables

The extraction process uses the following two log tables to record change data:

- The PKLOG transaction log table contains the recorded inserts, updates, and deletes, and associated metadata.
- The PROCESSED log table contains the current maximum sequence number for the most recent extraction cycle.

Implementation Task Flow

To implement SAP HANA CDC, you need to complete some tasks in SAP HANA, PowerExchange, and PowerCenter. Use the following high-level task flow:

1. Create capture registrations for the SAP HANA source tables with either the PowerExchange Navigator or the DTLUCBRG utility. For more information, see the *PowerExchange Navigator User Guide* and *PowerExchange Utilities Guide*.
2. Install both an ODBC driver and a JDBC driver for use by the connection. For more information, see [“Preparing the SAP HANA Server for PowerExchange CDC” on page 221](#).

3. Prepare the SAP HANA database for data capture. For more information, see [“Preparing the SAP HANA Database for PowerExchange CDC” on page 223](#).
4. Configure the PowerExchange Logger.
5. Materialize the target tables. Use any tool that you prefer. PowerExchange bulk data movement does not currently support SAP HANA sources. In some cases, you might also need to materialize the target tables.
6. Start the PowerExchange Logger.
7. Create workflows or extraction tasks to apply the data to the target:
 - Create PowerCenter CDC workflows that include the SAP HANA sources. Use an SAP HANA CDC application connection. You must cold start the CDC workflows.
 - Configure a PowerExchange for CDC Publisher task to move the data to a Kafka message stream. For more information, see the *PowerExchange CDC Publisher User Guide*.

Preparing the SAP HANA Server for PowerExchange CDC

The configuration tasks for PowerExchange CDC depend on whether the SAP HANA source database resides on Windows or Linux:

- If the SAP HANA database is on Windows, download the ODBC driver from <https://developers.sap.com/tutorials/hxe-ua-install-hdb-client-windows.html>. You can specify the Data Source Name or configure the connection with the host name and port number.
- If the SAP HANA database is on Linux, download the ODBC driver from <https://developers.sap.com/tutorials/hxe-ua-install-hdb-client-linux.html>. You must specify the DSN on Linux, along with the host name and port number.
- Download and install the SAP HANA client and JDBC driver from <https://developers.sap.com/tutorials/hana-clients-install.html>.

For more information about using PowerExchange ODBC connections, see the *PowerExchange Reference Manual*.

Configuring ODBC on Linux

On Linux, use the DataDirect ODBC driver for SAP HANA or the unixODBC driver to establish the ODBC connection.

Notes:

- If you use the DataDirect driver, you must enter the database name when you create a capture registration.
- If you use the unixODBC driver, you must specify a DSN for the connection.

Create environment variables for the paths to `odbc.ini` and `odbcinst.ini` files and to the ODBC library, as follows:

```
ODBCINST=/dtl/installdirectory/ODBC/linux-x64/ODBC7.1/lib/odbcinst.ini
ODBCINI=/dtl/installdirectory/ODBC/linux-x64/ODBC7.1/lib/odbc.ini
```

The path for `ODBCHOME` variable depends on the installation of the ODBC driver on the Linux system.

The following example shows the path for the unixODBC driver:

```
ODBCHOME=/usr/lib64
```

The following example shows the path for the DataDirect driver:

```
export  
ODBCHOME=/dtl/installdirectory/ODBC/linux-x64/ODBC7.1
```

The following example shows a sample entry for the SAP HANA driver in the odbcinst.ini file:

```
[ODBC Drivers]  
HDBODBC=Installed  
[HDBODBC]  
DRIVER=/dtl/installdirectory/ODBC/linux-x64/ODBC7.1/lib/libodbcHDB.so  
DESCRIPTION=HANA Driver  
SETUP=/dtl/installdirectory/ODBC/linux-x64/ODBC7.1/lib/libodbcHDB.so  
CPTIMEOUT=0  
DriverManagerEncoding=UTF-8
```

The following example shows a sample entry in the odbc.ini file for the SAP HANA data source:

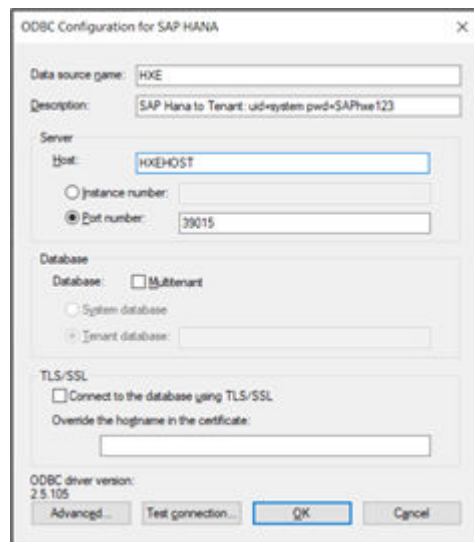
```
[HXEHOST]  
DRIVER=/dtl/installdirectory/ODBC/linux-x64/ODBC7.1/lib/libodbcHDB.so  
SERVERNODE=ip_address:port_number  
USER=user_name  
PASSWORD=password  
DATABASENAME=db_name  
DRIVERUNICODETYPE=1  
DESCRIPTION=description
```

Enter a user-defined **Data source name**, the **Server Host** name or IP address, and the **Port number** to use for the connection.

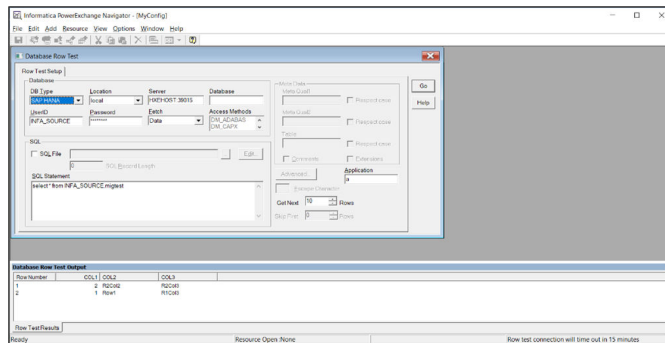
Configuring ODBC on Windows

Use the ODBC connection wizard in the PowerExchange Administrator to establish an ODBC connection to SAP HANA on Windows.

Enter a user-defined **Data source name**, the **Server Host** name or IP address, and the **Port number** to use for the connection.



Alternatively, you can enter the server host name or IP address from the PowerExchange Navigator.



Configuring JDBC on Windows or Linux

PowerExchange SAP HANA CDC requires JDBC access in addition to ODBC. PowerExchange installs the following files for the SAP HANA CDC:

- log4j-api-2.13.2.jar
- log4j-core-2.13.2.jar
- PWXHANACDC.jar

SAP HANA CDC also requires the access to the ngdbc.jar file, which is installed with the SAP HANA JDBC client. Use the following steps to give PowerExchange access to the JDBC client .jar file:

1. Copy the ngdbc.jar file to the directory where you installed PowerExchange. By default, the installation directory also includes the .jar files installed with PowerExchange.
2. By default, PowerExchange expects the ngdbc.jar file in the jars directory. If the file is named differently, either rename the file to ngdbc.jar or add a HANA_CLASSPATH statement in the dbmover.cfg file to point to all of the required .jar files. Use the separator appropriate for the environment between each file name. The following example shows the HANA_CLASSPATH statement on Windows:

```
HANA_CLASSPATH="D:/pwx/installdirectory/dbg/Win_x64/jars/PWXHanaCDC.jar;D:/pwx/
installdirectory/dbg/Win_x64/jars/log4j-api-2.13.2.jar;D:/pwx/installdirectory/dbg/
Win_x64/jars/log4j-core-2.13.2.jar;D:/pwx/installdirectory/dbg/Win_x64/jars/
ngdbc.jar"
```

The following example shows the HANA_CLASSPATH statement on Linux:

```
HANA_CLASSPATH="/dtl/installdirectory/pwx/hanalap/rel/linux-x64/jars/
PWXHanaCDC.jar:/dtl/installdirectory/pwx/hanalap/rel/linux-x64/jars/log4j-
api-2.13.2.jar:/dtl/installdirectory/pwx/hanalap/rel/linux-x64/jars/log4j-
core-2.13.2.jar:/dtl/installdirectory/pwx/hanalap/rel/linux-x64/jars/ngdbc.jar"
```

Note: PowerExchange looks first for dependent jar files in the HANA_CLASSPATH statement. If the HANA_CLASSPATH statement is not present, PowerExchange uses the path defined in the PWX_HOME environment variable. If you do not include the HANA_CLASSPATH statement or the PWX_HOME variable, PowerExchange uses a path relative to the location where you initiate the CDC process.

Preparing the SAP HANA Database for PowerExchange CDC

To use PowerExchange SAP HANA CDC, you must create a user for the process that has explicit permissions on the database, including permissions to create triggers on the database.

Creating a User for CDC

To create a user for SAP HANA CDC, connect to the SAP HANA database as an administrator and issue the following command:

```
CREATE USER INFA_CDC password "<passwd>" NO FORCE_FIRST_PASSWORD_CHANGE;
```

This command creates a new user in the database with the following default permissions:

- Read the data dictionary views, including TABLES.
- Create objects in the user's own schema.

Granting Additional Permissions

In addition to the default permissions, you must grant the SAP HANA CDC user INSERT permission on their schema to the target database and SELECT permission to the user on the source database schema and tables.

The following command allows the target database to insert the triggers required to capture data to the PKLOG table:

```
GRANT INSERT ON SCHEMA INFA_CDC TO database_name;
```

To grant permissions on the target database, you can either grant the permissions directly to the SAP HANA CDC user or create a user role and assign the SAP HANA CDC user to the role. Whichever method you choose, the user must have SELECT permissions on all tables in the source database that are registered for change data capture.

If you want to capture data from all or most tables in a database, issue the following command to grant access to all objects in the target database:

```
GRANT SELECT ON SCHEMA database_name TO INFA_CDC;  
GRANT TRIGGER ON SCHEMA database_name TO INFA_CDC;
```

If you want to capture data from just a few tables, you can limit access to only those tables. Issue the following command for each registered table:

```
GRANT SELECT ON database.table_name TO INFA_CDC;  
GRANT TRIGGER ON database.table_name TO INFA_CDC;
```

Manually Creating the Log Tables

When PowerExchange for SAP HANA CDC starts, it uses the SAP HANA CDC user ID to connect to the source database and checks for the PKLOG and PROCESSED tables. If these tables do not exist, PowerExchange creates them using the SAP HANA user's schema.

If you prefer, you can connect to the database with the SAP HANA CDC user ID and create the tables manually, with the following commands:

```
CREATE COLUMN TABLE PKLOG (...);  
CREATE COLUMN TABLE PROCESSED (...);
```

Manually Creating Triggers

PowerExchange queries each registered table to determine if the insert, update, and delete triggers exist. If a trigger does not exist for a table, PowerExchange creates it. Alternatively, you can create the triggers in the tables registered for change data capture manually. If you choose to create the triggers manually, the user ID connected to the source database must have WRITE permission for the PKLOG table.

Note: Because the SAP HANA CDC user has SELECT permissions on the SAP source tables, PowerExchange can select the list of trigger names, even without the TRIGGER permission. The list can be used to determine which triggers must be created, if you elect to create them manually.

If a table is removed from the list of registered tables, PowerExchange removes the trigger that it automatically created from that table. You can list the triggers to verify that PowerExchange removed the triggers from the removed source table or you can drop the triggers manually from the removed source table.

Configuring PowerExchange for SAP HANA CDC

You must complete several PowerExchange configuration tasks to prepare for SAP HANA CDC, including configuring the dbmover.cfg file on the system where you plan to capture change data.

Configuring the dbmover Configuration File

In the dbmover configuration file on the system where the capture registrations and CDC control files are stored, add the statements that are required for SAP HANA CDC. This system is specified in the **Location** field of the registration group definition..

CAPI_CONNECTION - HANA Statement

The HANA 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 SAP HANA sources.

Operating Systems: Windows and Linux

Data Sources: SAP HANA

Required: Yes for SAP HANA CDC

Syntax:

```
CAPI_CONNECTION=(NAME=capi_connection_name
                  [,DLLTRACE=trace_id]
                  ,TYPE=(HANA
                        ,SERVER=[server_name|ip_address]:port
                        ,DATABASE=database_name
                        [,BUFFERSIZE=bytes]
                        [,FETCHSIZE=records]
                        [,LOGCLEAR=days]
                        [,ONDATAATRUNC=[WARN|FAIL]]
                        [,SSL=[Y|N]]
                        [,TIMEOUT=milliseconds]
                        [,VMOPTS=options]
                        [,VMOPTSDEL=delimiter_character])
                  )
```

Parameters:

NAME=capi_connection_name

Required. A unique user-defined name for this CAPI_CONNECTION statement.

Maximum length is eight alphanumeric characters.

DLLTRACE=trace_ID

Optional. The 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.

TYPE=(HANA, ...)

Required. The type of CAPI_CONNECTION statement. For SAP HANA sources, this value must be HANA.

SERVER=[server_name|ip_address]:port

Required. The server name or IP address of the database server that contains the source tables.

The default port is 39015. If you use a different port number, you must specify the port in the SERVER parameter.

DATABASE=database_name

Required. The name of the SAP HANA database that contains the source tables for change data capture.

BUFFERSIZE =bytes

Optional. The maximum size of the buffer for CDC records retrieved from the SAP HANA source. When the maximum buffer size is reached, PowerExchange flushes the buffer and commits changes to the log tables. Valid values are 64000 (64 KB) to 4000000 (4 MB).

Default is 64000.

FETCHSIZE=records

Optional. The maximum number of records that can be returned from the SAP HANA source to the buffer at a time. Valid values are 100 to 10000.

Default is 1000.

LOGCLEAR=days

Optional. The time interval in days after which PowerExchange purges the PKLOG table entries. Valid values are 0 to 366. A value of 0 means that the PKLOG table entries are not purged.

Default is 14.

ONDATATRUNC=[WARN|FAIL]

Optional. Indicates whether PowerExchange issues a warning message and continues processing or ends abnormally when it encounters a column where data would be truncated. Valid values are WARN or FAIL.

Default is FAIL.

SSL=[Y|N]

Optional. Whether the connection to the SAP HANA source uses the SSL/TLS protocol. This parameter only affects connections that use the JDBC driver. Connections that use an ODBC driver on must specify the secure connection in the odbc.ini file.

Default is N.

TIMEOUT=milliseconds

Optional. The time interval in milliseconds after which the change cycle ends if PowerExchange does not receive a response from the source table. Valid values are 1000 to 60000.

Default is 10000.

VMOPTS=option;option;...

Optional. A set of one or more Java Virtual Machine (JVM) options, separated by semicolons. You can specify up to 20 options.

Default is a placeholder semicolon (;).

VMOPTSDEL=character

Optional. Changes the separator character for the VMOPTS parameter. Valid values are any separator character on the system where the options will be overridden.

Default is a semicolon (;).

Managing SAP HANA CDC

After CDC is running, you might need to perform some occasional tasks to maintain and manage your SAP HANA CDC environment.

These tasks include:

- Stopping SAP HANA CDC processing.
- Changing the structure of a source table.

Stopping Change Data Capture for a SAP HANA Source Table

You might want to stop change data capture for a SAP HANA source table if the table has been dropped, change activity no longer occurs on the table, or the data in the table is no longer of interest for CDC processing.

1. In PowerExchange Navigator, open the capture registration and set the **Status** option to **History**.
A capture registration that has a status of **History** cannot be activated again.
2. If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger and then warm start it.
This step refreshes the registration information that the PowerExchange Logger uses.
3. In PowerCenter, delete or update CDC workflows to make sure that no workflow processes the removed table.

Stopping SAP HANA CDC Processing Temporarily

You might need to temporarily stop SAP HANA CDC processing to troubleshoot issues or to perform a maintenance task on the target database.

If you use the PowerExchange Logger for Linux, UNIX, and Windows, shut down the PowerExchange Logger to stop CDC processing for all source tables. Later, you can warm start the PowerExchange Logger to resume change capture processing without any change data loss. This method is preferable.

CHAPTER 10

Remote Logging of Data

This chapter includes the following topics:

- [Remote Logging Overview, 228](#)
- [Requirements for Capture Registrations, 233](#)
- [Security Settings for Data from z/OS Sources, 233](#)
- [Configuration Tasks for Remote Logging, 234](#)
- [Example of Remote Logging from a z/OS Data Source, 240](#)
- [Example of Remote Logging from a Db2 for i Data Source, 242](#)
- [Example of Remote Logging from a PowerExchange Express CDC for Oracle Source, 244](#)

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 233](#)
- [“Configuration Tasks for Remote Logging” on page 234](#)
- [“Customizing the PowerExchange Logger Configuration File for Remote Logging” on page 234](#)
- [“Customizing the dbmover Configuration File on the System to Which Data Is Logged” on page 238](#)
- [“Customizing the dbmover Configuration File on the PowerCenter Integration Service System” on page 239](#)
- [“Example of Remote Logging from a PowerExchange Express CDC for Oracle Source” on page 244](#)
- [“Logging Data for IBM i or z/OS Sources to Remote PowerExchange Logger Logs” on page 44](#)

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:

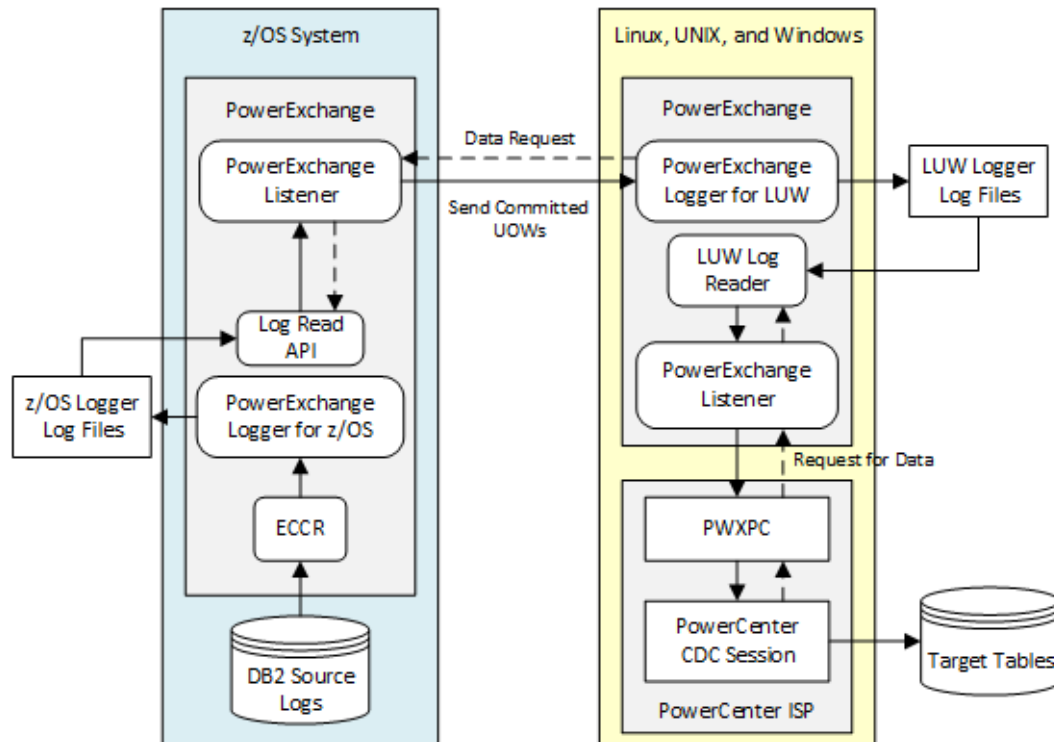
- Moves resource-intensive, column-level processing and UOW Cleanser processing off of the IBM i or z/OS system onto the Linux, UNIX, or Windows system where the PowerExchange Logger for Linux, UNIX, and Windows runs.
- Extracts change data from the Db2 for i (i5/OS) journal receivers or PowerExchange 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, `pwxccl.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 (PWXP) 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. PWXP makes the data available to the PowerCenter CDC session. Multiple PowerCenter CDC sessions can extract change data from the local PowerExchange Logger log files.

Remote Logging of Data from Linux, UNIX, or Windows Sources

You can use the PowerExchange Logger for Linux, UNIX, and Windows to extract change data from sources on a Linux, UNIX, or Windows system and log that data to another Linux, UNIX, or Windows system.

PowerCenter CDC sessions can then extract the change data from the log files on the PowerExchange Logger system.

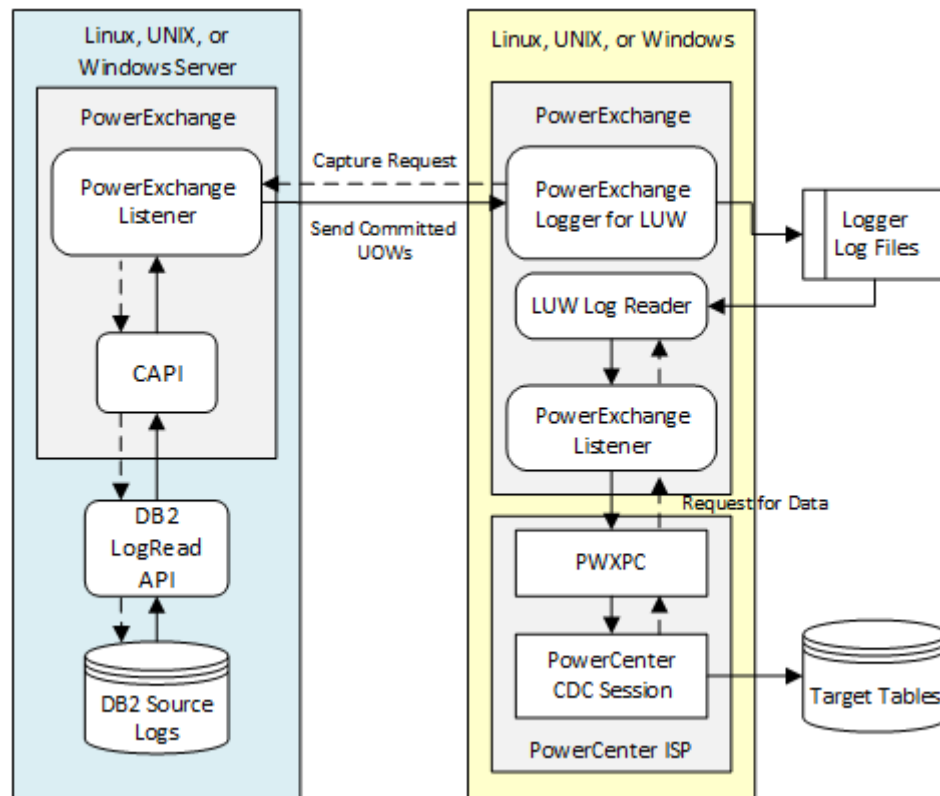
For Linux, UNIX, and Windows sources, the remote logging of data has the advantage of maximizing the performance of the capture process while minimizing the impact on the database server. Remote logging has the following benefits:

- Reduces disk space and CPU usage on the source database server by moving PowerExchange Logger processing and log files to another Linux, UNIX, or Windows system.
- Retains local access to the source database and database logs, which eliminates the latency of accessing the source across a network and avoids having to configure NFS to export the database logs.
- Retains object filtering and any internal UOW cleansing processing on the source system. Only the data of interest is sent across the network.
- Allows multiple PowerCenter sessions to access the same logged data without increasing overhead on the database server and, if the PowerExchange Logger runs on the same server as PowerCenter, without affecting network latency.

To configure this remote logging scenario, you must specify the **CAPTURE_NODE** statement in the PowerExchange Logger for Linux, UNIX, and Windows configuration file, `pwxccl.cfg`, on the system where the Logger 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 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, configure the PowerExchange Logger for Linux, UNIX, and Windows to extract Db2 change data from Db2 database logs on a Linux system and then log that data to PowerExchange Logger log files on the PowerCenter ISP system. The following image shows this type of configuration:



In this scenario, set the PowerExchange Logger CAPTURE_NODE statement to the node name of the PowerExchange Listener on the Db2 source system. Set the PowerCenter **Location** connection attribute to the node name of the PowerExchange Listener on the PowerCenter ISP machine where the PowerExchange Logger runs. Set the **Map Location** connection attribute to the node name of the PowerExchange Listener on the Db2 source system.

When the PowerExchange Logger sends a change data capture request, the PowerExchange Listener on the source system communicates with the CAPI to retrieve change data from the Db2 database logs by means of the Db2 LogRead API. The PowerExchange Listener sends only the committed UOWs for the objects of CDC interest to the PowerExchange Logger on the remote system. The PowerExchange Logger logs the data in its log files. When a PowerCenter CDC session requests change data for the tables of CDC interest, the PowerExchange Client for PowerCenter (PWXP) communicates with the PowerExchange Listener on the PowerExchange Logger system to get the change data from the local PowerExchange Logger log files.

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

Security Settings for Data from z/OS Sources

For the highest level of security for data from z/OS data sources, set the SECURITY option to 2 in the z/OS DBMOVER configuration member where the extraction maps are located. With this setting, PowerCenter CDC sessions are permitted to extract z/OS data from PowerExchange Logger for Linux, UNIX, and Windows log files only if their user credentials pass z/OS security checking.

When defining a PWXPC connection for the CDC sessions that extract data from the PowerExchange Logger log files, enter a valid z/OS user ID and password in the **Map Location User** and **Map Location Password** connection attributes. If the location of the log files is not local, enter the z/OS user ID and password in the **User Name** and **Password** connection attributes for use by the PowerExchange Listener on the Linux, UNIX, or Windows system where the log files reside.

For data extraction, these z/OS user credentials must have the following permissions:

- 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.

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

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 `pwxccl.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 `CAPL_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 `pwxccl.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 Remote Logging

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 where the PowerExchange Logger log files will reside.

PowerExchange provides a sample configuration file, named `pwxccl.cfg`, 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 needed 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 a z/OS or 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 a z/OS or 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 a z/OS or 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. If CAPTURE_NODE specifies a Linux, UNIX, or Windows node, enter a user ID that is valid for the data source type: <ul style="list-style-type: none"> - For Db2 sources on Linux, UNIX, or Windows, enter a valid operating system user ID that has Db2 DBADM or SYSADM authority. - For Microsoft SQL Server instances that use SQL Server Authentication, enter a database user ID that permits access to the SQL Server distribution database. For a SQL Server instances that use Windows Authentication, PowerExchange uses the user ID under which the PowerExchange Listener was started. In this case, do not specify this parameter unless you want to specify another user. - For MySQL sources on Linux or Windows, enter a database user ID that allows access to the MySQL binary logs. This user must have been granted the privileges that are required for MySQL CDC. For more information, see "Preparing MySQL Sources" on page 123. - For Oracle sources, if you use PowerExchange Express CDC for Oracle, enter the ORACAPTL user ID that you defined, which permits access to the Oracle online and archive redo logs. - For PostgreSQL sources, enter a database user ID that allows access to the replication store table in the source database and that has a user role with the REPLICATION attribute.
CONDENSENAME	Optional. A name for the command-handling service for a PowerExchange Logger for Linux, UNIX, and Windows process to which pwxcmd 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	<p>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.</p> <p>Enter a valid CAPI_CONNECTION name for the source type.</p> <p>Informatica recommends that you specify CONN_OVR because it is the only type of override that the PowerExchange Logger can use.</p>
DB_TYPE	<p>Required. The source database type. Options are:</p> <ul style="list-style-type: none"> - ADA for Adabas sources. - AS4 for Db2 for i (i5/OS) sources. - DB2 for Db2 for z/OS sources. - DCM for Datacom sources. - IDL for IDMS log-based CDC sources. - IMS for IMS sources. - MSS for Microsoft SQL Server sources. - MYS for MySQL sources. - ORA for Oracle sources. - PGS for PostgreSQL sources. - UDB for Db2 sources on Linux, UNIX, or Windows. - VSM for VSAM sources.

Parameter	Description
DBID	<p>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.</p> <p>This value must match the instance or database name that is displayed in the Resource Inspector of the PowerExchange Navigator for the registration group that contains the capture registrations.</p> <p>Enter one of the following values based on the source type:</p> <ul style="list-style-type: none"> - For Adabas, enter the Instance name that is displayed for the registration group. - For Datacom, enter the MUF Name value that is displayed for the registration group. Alternatively, if you use Datacom synchronous CDC, enter the value of the MUF parameter in the DTLINPUT data set specified in the MUF JCL. Or, if you use Datacom table-based CDC, enter the value of REG_MUF parameter in the ECCRDCMP member of the RUNLIB library. - For Db2 for i (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. - For Db2 on Linux, UNIX, or Windows, enter the Database name that is displayed for the registration group. - For Db2 for z/OS, enter the Instance name that is displayed for the registration group. This name should match the RN parameter value in the Db2 statement in the RUNLIB(REPDB2OP) member. - For IDMS Log-based CDC, enter the Logsid value that is displayed for the registration group. This value should match the LOGSID parameter value in the RUNLIB(ECCRIDLP) member. - For IMS, enter the IMSID value that is displayed for the registration group. For IMS log-based CDC, this value should match the first parameter value in the IMSID statement in the RUNLIB(CAPTIMS) member. - For Microsoft SQL Server, this value depends on whether you also specify the optional DISTSRV and DISTDB parameters in the PowerExchange Logger configuration file: <ul style="list-style-type: none"> - If you specify the optional DISTSRV and DISTDB parameters, enter a name that serves as the collection identifier for all of the registrations. This name must be one to eight characters in length and start with a letter. This name overrides the instance name that is associated with the individual registrations. - If you do not specify the DISTSRV and DISTDB parameters, enter the value that the PowerExchange Navigator generates and displays in the Instance field of the Resource Inspector for the registration group. - For MySQL, enter the Instance name that is displayed for the registration group in the Resource Inspector. - For Oracle, enter the Instance name that is displayed for the registration group. This value also should match the first positional parameter of the ORACLEID statement in the dbmover configuration file. - For PostgreSQL, enter the Instance name that is displayed for the registration group in the Resource Inspector. - For VSAM, enter the Instance name that is displayed for the registration group.
EXT_CAPT_MASK	<p>Required. An existing directory path and a unique prefix to be used for generating the PowerExchange Logger log files.</p>

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

For the PowerExchange Logger for Linux, UNIX, and Windows to log data from a remote Linux, UNIX, Windows, i5/OS, or z/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 needed 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.cfg.
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.
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. Use one of the following the statement types, as appropriate for the source from which you are remotely logging data: <ul style="list-style-type: none">- For Db2 for i (i5/OS) sources, use the AS4J and UOWC CAPI_CONNECTION statements.- For Db2 sources on Linux, UNIX, and Windows, use UDB CAPI_CONNECTION statement.- For Microsoft SQL Server sources, use the MSQL CAPI_CONNECTION statement.- For MySQL sources, use the MYSQL CAPI_CONNECTION statement.- For Express CDC for Oracle sources, use the ORAD CAPI_CONNECTION statement.- For PostgreSQL sources, use the PG CAPI_CONNECTION statement.- For z/OS sources, use the LRAP and UOWC CAPI_CONNECTION statements. Remove the z/OS-specific parameters from the UOWC statement.
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.

Connection Attribute	Value
Map Location User and Map Location Password	<p>Enter a user ID and password that can access the extraction maps.</p> <p>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.</p> <p>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:</p> <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 z/OS Data Source

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

The PowerExchange Logger for z/OS captures change data for registered Db2 for z/OS tables and logs that data to its log files on the z/OS system. The PowerExchange Logger for Linux, UNIX, and Windows reads data from the PowerExchange Logger for z/OS log files and relogs that data on the UNIX system. PowerCenter CDC sessions then extract change data from the PowerExchange Logger for Linux, UNIX, and Windows log files rather than from the log files on the z/OS source system.

You need the PowerExchange Logger for Linux, UNIX, and Windows to read change data for registered tables in the Db2 instance DSN9 and then relog that data to its log files on the remote UNIX system. To do so, you must customize a PowerExchange Logger for Linux, UNIX, and Windows configuration file on the UNIX system and dbmover configuration files on both the z/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 z/OS source system, ensure that the DBMOVER member in the RUNLIB library includes the following CAPI_CONNECTION statements:

```
LISTENER=(MVS02,TCPIP,2480)
/* UOW Cleanser
```



```

CAPI_CONNECTION=(NAME=MV2UOWC,TYPE=(UOWC,CAPINAME=M2_LRAP,RSTRADV=600,MEMCACHE=20480,
DATACLAS=UOWC))
/* Log Read API Connection
CAPI_CONNECTION=(NAME=MV2_LRAP,TYPE=(LRAP,LOG=MV2L,AGENT=MV2A))

```

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

```

/*
/* dbmover
/*
LISTENER=(unix1,TCPIP,2480)
NODE=(MVS02,TCPIP,prodms2,2480)
...
LOGPATH=/pwx/logs/mvscond
CAPT_XTRA=/pwx/capture/mvscond/camaps
CAPT_PATH=/pwx/capture/mvscond
/*
/* Source-specific CAPI Connection
CAPI_CONNECTION=(NAME=MV2UOWC,TYPE=(UOWC,CAPINAME=M2_LRAP,RSTRADV=600,MEMCACHE=20480)
)
CAPI_CONNECTION=(NAME=MV2_LRAP,TYPE=(LRAP,LOG=MV2L,AGENT=MV2A))
/*
/* CAPX CAPI Connection for continuous extraction
CAPI_CONNECTION=(NAME=CAPXDSN9,TYPE=(CAPX,DFLTINST=DSN9,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.

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

```

/*
/* pwxcl
/*
DBID=DSN9
DB_TYPE=DB2
CONN_OVR=MV2UOWC
CAPTURE_NODE=MVS02
PROMPT=Y
EXT_CAPT_MASK=/pwx/capture/mvscond/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 z/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 PowerExchange Listener on the UNIX system with the PowerExchange Logger log files, if you not using a "local" connection

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

```

NODE=(unix1,TCPIP,unix1,2480)
NODE=(MVS02,TCPIP,prodms2,2480)

```

6. Create a PowerCenter mapping, session, and workflow.

7. Configure a PWX DB2zOS CDC Real Time application connection for the 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 MVS02 to point to the location of the extraction maps, which the z/OS 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 CAPXDSN9 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.

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 journals 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 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=(i50S1,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=PRODDATA/
PRODJRN,INST=PROD2,EOF=N,STOPIT=(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=PRODDATA/
PRODURN,INST=PROD2,EOF=N,STOPIT=(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.

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

```

/*
/* pwxccl
/*
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.

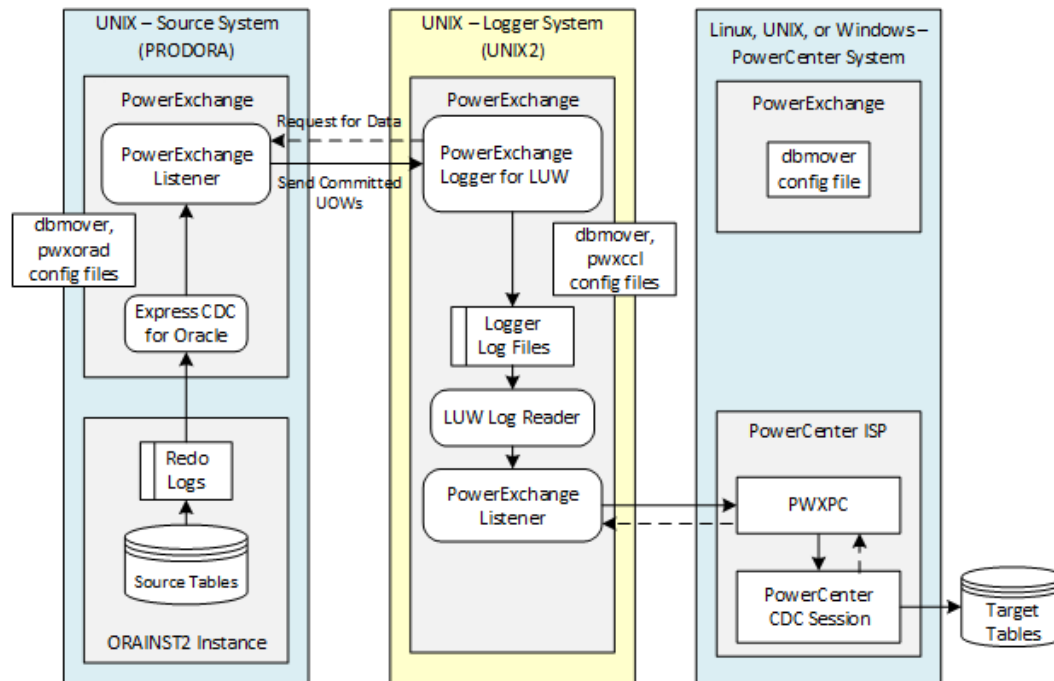
The session begins extracting change data from the PowerExchange Logger log files on the UNIX system.

Example of Remote Logging from a PowerExchange Express CDC for Oracle Source

In this example, you use a PowerExchange Logger for Linux, UNIX, and Windows instance on a UNIX system to capture change data from an Oracle instance on another UNIX system.

You want the PowerExchange Logger to capture change data from registered source tables in the Oracle ORAINST2 instance that runs on the PRODORA host and then log that data to its log files on a remote UNIX2 system. The system where the PowerExchange Logger runs is separate from the PowerCenter Integration Service system where you run CDC sessions. You must run a PowerExchange Listener on the source system and on the PowerExchange Logger system. You do not need to run a PowerExchange Listener on the PowerCenter Integration Service system.

The following image shows this shows the configuration:



First install PowerExchange on all three systems. Then perform the following steps to customize the PowerExchange Express CDC for Oracle, dbmover, and PowerExchange Logger configuration files and set the PWXPC connection attributes that PowerCenter CDC sessions require to extract change data from the PowerExchange Logger log files.

1. On the Oracle source system, perform the following configuration tasks:
 - a. Configure the PowerExchange Express CDC for Oracle configuration file. Use the sample pwxorad.cfg file that PowerExchange supplies in the directory that is specified in the PWX_HOME environment variable, or if that variable is not defined, in the PowerExchange bin directory. Include at least the following statements:

```

DICTIONARY
MODE=STATIC
SOURCE=ONLINE;
READER
MODE=ACTIVE;

```

- b. Ensure that the dbmover configuration file includes the following statements:

```

LISTENER=(unix1,TCPIP,2480)
ORACLE_CAPTURE_TYPE=D
CAPI_CONNECTION=(NAME=CAPIORA,TYPE=(ORAD,ORACOLL=COLINST2,PARMFILE=/Informatica/
PowerExchangeVR/capture/pwxorad.cfg))
ORACLEID=(COLINST2,ORAINST2)

```

2. On the PowerExchange Logger system where the Logger log files reside, perform the following tasks:

- a. Ensure that the dbmover configuration file includes the following statements:

```

LISTENER=(unix2,TCPIP,2480)
NODE=(unix1,TCPIP,PRODORA,2480)
...
LOGPATH=/pwx/logs/oracond
CAPT_XTRA=/pwx/capture/oracond/camaps
CAPT_PATH=/pwx/capture/oracond
/*

```

```

ORACLE_CAPTURE_TYPE=D
/* Source-specific CAPI Connection
CAPI_CONNECTION=(NAME=CAPIORA,TYPE=(ORAD,ORACOLL=COLINST2,PARMFILE=/Informatica/
PowerExchangeVR/capture/pwxorad.cfg))
/*
/* CAPX CAPI Connection for continuous extraction
CAPI_CONNECTION=(NAME=CAPXORA,TYPE=(CAPX,DFLTINST=COLINST2,FILEWAIT=60,RSTRADV=60
0))

```

Note: In the ORAD CAPI_CONNECTION statement, the PARMFILE parameter must point to the PowerExchange Express CDC for Oracle configuration file, pwxorad.cfg, that you created on the source system.

- b. On the PowerExchange Logger system, customize the PowerExchange Logger configuration file, pwxcl.cfg. For this example, include the following statements:

```

DBID=COLINST2
DB_TYPE=ORA
CONN_OVR=CAPIORA
CAPTURE_NODE=unix1
CAPTURE_NODE_UID=orauser
CAPTURE_NODE_EPWD=encrypted_password

```

Note: The value of the DBID parameter must match the value of the first positional *collection_id* parameter, COLINST2, in the ORACLEID statement on the source system. The CAPTURE_NODE parameter points to the source system node where the PowerExchange Listener processes capture requests.

3. Start the PowerExchange Logger and the PowerExchange Listener on the PowerExchange Logger system. Verify that the PowerExchange Listener is also running on the source system.
4. On the PowerCenter Integration Service system, add the following NODE statements in the dbmover configuration file:
 - A NODE statement that points to the PowerExchange Listener on the source system
 - A NODE statement that points to the PowerExchange Logger system

The following example shows these NODE statements:

```

NODE=(unix1,TCPIP,PRODORA,2480)
NODE=(unix2,TCPIP,unix2,2480)

```

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

For this example, set the following connection attributes:

- For the **Location** attribute, enter unix2 to point to the node where the PowerExchange Logger log files reside. CDC sessions will read data from this location.
- For the **Map Location** attribute, enter unix1 to point to the location of the extraction maps, which is the source system node.
- For the **Map Location User** attribute, enter a valid Oracle user ID.
- For the **Map Location Password** attribute, enter the password for the Oracle user ID.
- For the **CAPI Connection Name** attribute, enter CAPXORA to indicate the CAPX CAPI_CONNECTION statement to use.

7. Cold start the CDC session.

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

Part IV: Change Data Extraction

This part contains the following chapters:

- [Introduction to Change Data Extraction, 248](#)
- [Extracting Change Data, 262](#)
- [Managing Change Data Extractions, 289](#)

CHAPTER 11

Introduction to Change Data Extraction

This chapter includes the following topics:

- [Change Data Extraction Overview, 248](#)
- [Extraction Modes, 249](#)
- [PowerExchange-Generated Columns in Extraction Maps, 250](#)
- [Uses of BI and CI Fields in Extraction Maps, 256](#)
- [Restart Tokens and the Restart Token File, 257](#)
- [Multiple-Source Processing in CDC Sessions, 258](#)
- [Commit Processing with PWXPC, 259](#)
- [Tuning Options, 261](#)

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.

RELATED TOPICS:

- [“Extraction Modes” on page 249](#)
- [“Restart Tokens and the Restart Token File” on page 257](#)
- [“Multiple-Source Processing in CDC Sessions” on page 258](#)

- [“Commit Processing with PWXPC” on page 259](#)
- [“Tuning Options” on page 261](#)

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

Column	Description	Datatype	Length
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
DTL__CAPXRRN	<p>For Db2 for i sources only, provides the relative record number up to 2,147,483,647.</p> <p>DTL__CAPXRNN is deprecated but still supported in existing extraction maps created before PowerExchange 10.5.</p>	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

Column	Description	Datatype	Length
DTL__CAPXUSER	<p>Provides the user ID of the user who made the change to the data source, with the following exceptions:</p> <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 <code>UID:PLAN:CORR:CONN:CTYPE</code>. 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 <code>userid:psbname</code>. 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 322.</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_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_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 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. - null. 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. - null. 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

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:

- [“Recovery and Restart Processing for CDC Sessions” on page 277](#)
- [“Configuring the Restart Token File” on page 284](#)
- [“Creating Restart Tokens for Extractions” on page 283](#)

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.

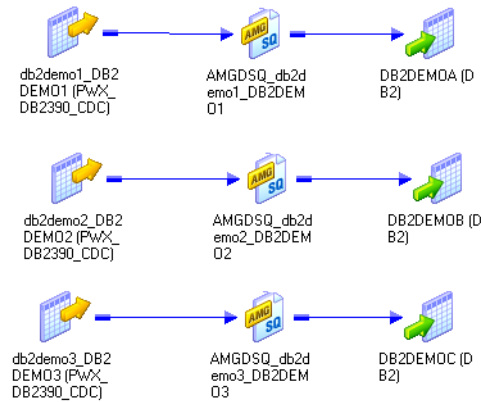
During change data extraction, PowerExchange processes all source definitions in the mapping that have the same source type. Do not include multiple data source types in the mapping. Otherwise, the CDC session fails with message PWXPC_10080.

For example, do not run a CDC session that contains a mapping with both Db2 and Oracle source definitions, even if changes for those sources are in the same change stream. Instead, create a mapping and session for all of the Db2 sources and a separate, unique mapping and session for all of the Oracle sources.

PowerExchange reads the change stream twice: once for the session with the Db2 sources, and once for the session with the Oracle sources.

The following figure shows an example mapping in PowerCenter Designer that includes three Db2 sources:

Mapping Designer



If you include this mapping in a session that uses a PWX DB2LUW 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. For example, the sessions might extract change data from the same PowerExchange Logger 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 272](#)
- [“Examples of Controlling Commit Processing” on page 275](#)

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.

RELATED TOPICS:

- [“CDC Offload Processing” on page 318](#)
- [“Multithreaded Processing” on page 320](#)
- [“Remote Logging Overview” on page 228](#)

CHAPTER 12

Extracting Change Data

This chapter includes the following topics:

- [Overview of Extracting Change Data, 262](#)
- [Security Considerations for Extracting z/OS Data, 263](#)
- [Task Flow for Extracting Change Data, 264](#)
- [Testing an Extraction Map, 264](#)
- [Configuring PowerCenter CDC Sessions, 266](#)
- [Recovery and Restart Processing for CDC Sessions, 277](#)
- [Creating Restart Tokens for Extractions, 283](#)
- [Displaying Restart Tokens, 283](#)
- [Configuring the Restart Token File, 284](#)

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.

RELATED TOPICS:

- [“Task Flow for Extracting Change Data” on page 264](#)
- [“Configuring PowerCenter CDC Sessions” on page 266](#)
- [“Recovery and Restart Processing for CDC Sessions” on page 277](#)
- [“Creating Restart Tokens for Extractions” on page 283](#)
- [“Configuring the Restart Token File” on page 284](#)

Security Considerations for Extracting z/OS Data

For the highest level of security for data from z/OS data sources, set the SECURITY option to 2 in the z/OS DBMOVER configuration member where the extraction maps are located. With this setting, a PowerCenter CDC session is permitted to extract data only if its user credentials pass z/OS security checking.

When you define a PWXPC connection for CDC sessions that extract data for a z/OS source from the z/OS system, you must enter a valid z/OS user ID and password in the **User Name** and **Password** connection attributes.

If you log captured z/OS data to remote PowerExchange Logger for Linux, UNIX, and Windows log files, when defining a PWXPC connection for CDC sessions that extract data from these log files, enter the z/OS user ID and password in the **Map Location User** and **Map Location Password** connection attributes. If the location of the PowerExchange Logger log files is not local, enter the z/OS user ID and password in the **User Name** and **Password** connection attributes for use by the PowerExchange Listener on the Linux, UNIX, or Windows system where the log files reside.

In all cases, the z/OS user credentials must be a valid z/OS user ID and password combination and have READ access to the PowerExchange data set that is defined in the DTLCAMAP DD statement of the PowerExchange Listener JCL.

To extract z/OS data from PowerExchange Logger from Linux, UNIX, and Windows log files, the z/OS user credentials also must have READ access to CAPX.CND.* resource profiles in the FACILITY class, which are managed by your z/OS security product.

For more information, see the *PowerExchange Reference Manual*.

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.
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.

Attribute Name	Attribute Location	Recommended Value for CDC	Description
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*.

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

Attribute	Value
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:

- [“Creating Restart Tokens for Extractions” on page 283](#)
- [“Configuring the Restart Token File” on page 284](#)

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.

The following table describes the default restart points for data sources on Linux, UNIX, or Windows, by source type and extraction mode:

Data Source	Batch and Continuous Extraction Modes	Real-time Extraction Mode
Db2	Oldest PowerExchange Logger log file that is recorded in the CDCT file.	Current log position at the time the PowerExchange capture catalog was created.
Microsoft SQL Server	Oldest PowerExchange Logger log file that is recorded in the CDCT file.	Oldest data available in the Publication database.
MySQL	Oldest PowerExchange Logger log file that is recorded in the CDCT file.	Oldest data available in the binary log. This position is the beginning of the binary log that has the log name that includes the lowest numeric suffix value. You can use the SHOW BINARY LOGS statement to identify this log.
Oracle	Oldest PowerExchange Logger log file that is recorded in the CDCT file.	Earliest available point in the change stream. For PowerExchange Express CDC for Oracle, the beginning of the most recent archive log.
PostgreSQL	Oldest PowerExchange Logger log file that is recorded in the CDCT file.	Oldest data available in the replication slot or replication store table.

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.

When generating restart tokens, consider the following points:

- 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.
- The length of restart tokens vary by source type.
- For Microsoft SQL Server, the sequence token represents the point from which PowerExchange starts reading change data from the SQL Server distribution database. The restart token includes the DBID of the distribution database and the name of the distribution server. PowerExchange verifies that the distribution database in the restart token matches the distribution database that is specified in the CAPI_CONNECTION statement.
- PWXPC can generate restart tokens when it starts extraction processing for a CDC session. PowerExchange also provides methods of generating restart tokens.

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 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 FUZZYRSTART 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 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: E2C2F2D3404000000000AEF4000000000

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 00000000000100000000000000000000000000000000. 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: E2C2F2D3404000000000AEF4000000000

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.

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.

Restriction: Use the CURRENT_RESTART option only for CDC sessions that use real-time extraction mode or continuous extraction mode.

```
<!-- my comments
```

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.

```
<!-- Restart Tokens for existing tables -->
Restart1=000000AD7756000000000000AD775600000000000000
Restart2=C1E4E2D3404000000AD5F2C00000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_001 -->
dldsn9.rrtb0001_RRTB_SRC_001=0000060D1DB2000000000000060D1DB20000000000000000
dldsn9.rrtb0001_RRTB_SRC_001=C1E4E2D340400000013FF3620000000000000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_002 -->
dldsn9.rrtb0002_RRTB_SRC_002=000000A3719500000000000000A3719500000000000000000
dldsn9.rrtb0002_RRTB_SRC_002=C1E4E2D3404000000968FC60000000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_004 -->
dldsn9.rrtb0004_RRTB_SRC_004=000006D84E7800000000000006D84E7800000000000000000
dldsn9.rrtb0004_RRTB_SRC_004=C1E4E2D340400000060D1E61000000000
```

```

=====
Session restart information:
=====
Extraction Map Name      Restart Token 1      Restart Token 2      Source
-----
didsn9.rrtb0001_RRTB_SRC_001 00000060D1DB2000000000000060D1DB200000000000000000 C1E4E2D340400000013FF36200000000 Restart file
didsn9.rrtb0002_RRTB_SRC_002 0000000A37195000000000000000A37195000000000000000 C1E4E2D34040000000968FC600000000 Restart file
didsn9.rrtb0003_RRTB_SRC_003 000000AD775600000000000000AD77560000000000000000 C1E4E2D34040000000AD5F2C00000000 Restart file (special override)
didsn9.rrtb0004_RRTB_SRC_004 0000006D84E7800000000000006D84E78000000000000000 C1E4E2D340400000006D1E6100000000 Restart file
didsn9.rrtb0005_RRTB_SRC_005 000000AD775600000000000000AD77560000000000000000 C1E4E2D34040000000AD5F2C00000000 Restart file (special override)
didsn9.rrtb0006_RRTB_SRC_006 000000AD775600000000000000AD77560000000000000000 C1E4E2D34040000000AD5F2C00000000 Restart file (special override)
didsn9.rrtb0007_RRTB_SRC_007 000000AD775600000000000000AD77560000000000000000 C1E4E2D34040000000AD5F2C00000000 Restart file (special override)
=====

```

For the sources to which PWXPC assigns the special override restart tokens, PWXPC writes “Restart file (special override)” in the Source column.

CHAPTER 13

Managing Change Data Extractions

This chapter includes the following topics:

- [Starting PowerCenter CDC Sessions, 289](#)
- [Stopping PowerCenter CDC Sessions, 291](#)
- [Changing PowerCenter CDC Sessions, 293](#)
- [Recovering PowerCenter CDC Sessions, 295](#)

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.

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 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 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 =<C1E4E2D3404000000DBF238200000000>
```

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=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
dldsn9.rrtb0004_RRTB_SRC_004=00000DBF240A0000000000000DBF240A0000000000000000
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	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0009_RRTB_SRC_009	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0005_RRTB_SRC_005	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0006_RRTB_SRC_006	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0008_RRTB_SRC_008	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0003_RRTB_SRC_003	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0002_RRTB_SRC_002	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0001_RRTB_SRC_001	00000FCA6584000000000000D2E004A00000000FFFFFFFF	C1E4E2D3404000000D21B1A500000000	GMD storage
dldsn8.rrtb0007_RRTB_SRC_007	00000FCA6584000000000000D2E004A00000000FFFFFFFF	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 [00000FCA6584000000000000D2E004A00000000FFFFFFFF] : Restart 2 [C1E4E2D3404000000D21B1A500000000]
to: Restart 1 [00000FCA6584000000000000D300D8000000000FFFFFFFF] : 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.

Part V: Monitoring and Tuning

This part contains the following chapters:

- [Monitoring CDC Sessions, 299](#)
- [Tuning CDC Sessions, 311](#)

CHAPTER 14

Monitoring CDC Sessions

This chapter includes the following topics:

- [Monitoring Overview, 299](#)
- [Monitoring CDC Sessions in PowerExchange, 299](#)
- [Monitoring CDC Sessions in PowerCenter, 307](#)

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 299](#)
- [“Monitoring CDC Sessions in PowerCenter” on page 307](#)

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 **PwrCntrSess** 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, PwrCntrSess=intserv1/workflow1/cdc_sess1,
Application=appl_name1, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=
PWX-00729 Userid=user1, Client=clientid, File=capture_registration_file_name, Table=table_name,
DB=database_name
PWX-00712 TaskId=2, Partner=10.10.10.02, Port=2480, PwrCntrSess=intserv2/workflow2/cdc_sess2,
Application=appl_name2, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=
PWX-00729 Userid=user1, 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 mcrs
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 mcrs
```

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 mcrs
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 mcrs
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*.

PowerExchange Logger for Linux, UNIX, and Windows Monitoring Statistics

You can use the PowerExchange Logger DL and DG commands or the `pwxcmd displaystats -tp {logger|groups}` command to publish monitoring statistics on demand for a PowerExchange Logger process and its tasks or for PowerExchange Logger group definitions. Also, you can configure the PowerExchange Logger to print monitoring statistics at a specific interval and when it shuts down.

Before you can publish monitoring statistics on demand, at shutdown, or at a specific interval, you must configure the `STATS=(MONITOR)` parameter in the PowerExchange Logger configuration file, `pwxccl.cfg`, to enable collection of the statistics. In this parameter, you can include the optional *interval* subparameter to publish the statistics at a regular interval.

To publish monitoring statistics on demand, use the following commands:

- Issue the DL and DG commands from the command line window on the Linux, UNIX, or Windows system where the PowerExchange Logger runs. The PowerExchange Logger must be running in the foreground.
- Issue the `pwxcmd displaystats -tp logger` or `pwxcmd displaystats -tp groups` command from a Linux, UNIX, or Windows system to the PowerExchange Logger on a remote system or the same system. You must use this method to issue the command to a PowerExchange Logger process that runs in background mode.
- Issue the PowerExchange Logger SHUTCOND or SHUTDOWN command. The Logger then publishes summary statistics when it stops.

For more information about the command syntax, see the *PowerExchange Command Reference*.

The statistics are displayed on screen and printed to the PowerExchange message log.

DL Logger Report

The DL and `pwxcmd displaystats -tp logger` commands produce statistics for the PowerExchange Logger process and its tasks. The following example report shows these statistics:

```
PWX-26011 Command handler received command "DS"
PWX-00723 Command <display L stats> succeeded
PWX-37130 PWXCCL pid = 7144 Writer status = Reading or waiting for source data
PWX-37134 CPU Time = 0:00:02.589616
PWX-37131 Memory (Current/Total/Maximum)
PWX-37132 Controller: (981/983/1849) KB Command Handler: (0/0/34) KB Writer: (5127/5147/5181)
KB
PWX-37135 Status 7144 Totals I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
PWX-37136 CurrFileOpened : 2015-08-11 13:20:39 I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
PWX-37137 Active Cycle : 2015-08-11 13:21:01 I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
```

This report contains the following fields:

- PWXCCL pid. The process ID of the PowerExchange Logger process.
- Writer status. The status of the PowerExchange Logger Writer subtask at the time the command was issued.
- CPU Time. The amount of CPU time used by the PowerExchange Logger since it started.
- Controller. The amount of memory in kilobytes that the PowerExchange Logger Controller has used.
- Command Handler. The amount of memory in kilobytes that the PowerExchange Logger Command Handler has used.
- Writer. The amount of memory in kilobytes that the PowerExchange Logger Writer subtask has used.
- Status. The process ID of the PowerExchange Logger process.
- I, U, D, C, and Total. Counts of inserts, updates, deletes, and commits that the PowerExchange processed, with the total for all of these operation types. These values are reported for the PowerExchange Logger process, the currently open log file, and the active logging cycle.
- CurrentFileOpened. The timestamp that indicates when the log file was opened.
- Active Cycle. The timestamp that indicates when the active logging cycle started.

DG Logger Group Definition Reports

The DG and `pwxcmd displaystats -tp groups` commands produce statistics for each PowerExchange Logger group definition that is definition. A *group definition* defines a set of PowerExchange Logger log files for a group of registered source tables. The following example report shows these statistics:

```
PWX-26011 Command handler received command "DG"
PWX-37138 Grp: dtld004 Regs=1 IUD=000000000000 C=000000000000 Unflushed=000000000000
PWX-37138 Grp: dtld003 Regs=2 IUD=000000000470 C=000000000028 Unflushed=000000000000
PWX-37138 Grp: dtld002 Regs=2 IUD=000000003276 C=000000000196 Unflushed=000000000000
```

This report contains the following fields:

- Grp. The name of the group definition.
- Regs. The number of capture registrations in the group.
- IUD. the total number of inserts, updates, and deletes processed for the group.
- C. The number of commits processed for the group.
- Unflushed. The number of change records for the group that have not yet been flushed to PowerExchange Logger log files on disk.

If no PowerExchange Logger groups are defined, the command reports the following monitoring statistics for the PowerExchange Logger, as if all of the registrations were in one group named "condenseO":

```
PWX-26011 Command handler received command "DG"
PWX-37138 Grp: c:\pwx\capture\condenseO Regs=5 IUD=000000032292 C=000000001931 Unflushed=000000034223
PWX-37139 FirstRec=2015-05-22 13:59:10.603648 Open file=c:\pwx\capture/
condenseO.CND.CP150707.T1816001
PWX-37140 BeginSeq =000000009DE60000000000000000088D800000000 BeginRstrrt
=D4C9C7D3404000000000037DA00000000
PWX-37141 LastSeq =00000158743800000000000000158728600000000
PWX-37142 CommitSeq=000001589B2400000000000001589B2400000000
CommitRstrrt=D4C9C7D3404000000000037DA00000000
```

- FirstRec. The timestamp of the first record in the open Logger log file.
- BeginSeq. The sequence token of the earliest record in the open Logger log file.
- BeginRstrrt. The restart token of the earliest record in the open Logger log file.
- LastSeq. The sequence token of the last change record in the Logger log file that is not followed by a commit record. This value should be greater than the CommitSeq value.
- CommitSeq. The sequence token of the last commit record in the Logger log file.
- CommitRstrrt. The restart token of the last commit record in the Logger log file.

Summary Statistics at Logger Shutdown

To print summary monitoring statistics when the PowerExchange Logger shuts down, specify the STATS=(MONITOR) parameter in the pwxocl.cfg file, either with or without the *interval* subparameter. The Logger shuts down when it reaches the end of its batch run or when you issue a Logger SHUTCOND or SHUTDOWN command.

The following summary monitoring messages are included in the shutdown output:

```
PWX-00723 Command <Shutdown stats> succeeded
PWX-37130 PWXOCL pid = 9064 Writer status = Shutting down
PWX-37134 CPU Time = 0:00:00.686404
PWX-37131 Memory (Current/Total/Maximum)
PWX-37132 Controller: (476/477/1853) KB Command Handler: (476/477/1853) KB Writer: (0/0/0) KB
PWX-37105 Total Memory 16468 KB
PWX-37135 Status 9064 Totals I=000000001404 U=000000000000 D=000000001404
C=0000000000228 Total=0000000003036
PWX-37136 CurrFileOpened : 2016-08-19 10:37:47 I=000000000000 U=000000000000 D=000000000000
C=0000000000000 Total=0000000000000
PWX-37137 Active Cycle : 2016-08-19 10:37:47 I=000000001404 U=000000000000 D=000000001404
C=0000000000228 Total=0000000003036
```

Note: If you print summary statistics for a PowerExchange Logger that runs on a SUSE Linux version 11 machine, the PWX-37105 message incorrectly reports 0 KB as the total amount of memory that the PowerExchange Logger used. This problem does not occur if the Logger runs on a later SUSE Linux version.

Monitoring Interval Statistics

You can print the same monitoring statistics that are printed by the DL command at a specific interval if you specify the STATS=(MONITOR) parameter with the *interval* subparameter in the pwxocl.cfg file.

The following interval-based statistics are written to the PowerExchange message log:

```
PWX-37130 PWXOCL pid = 7144 Writer status = Reading or waiting for source data
PWX-37134 CPU Time = 0:00:02.589616
```

```

PWX-37131  Memory (Current/Total/Maximum)
PWX-37132  Controller: (981/983/1849) KB    Command Handler: (0/0/34) KB    Writer: (5127/5147/5181)
KB
PWX-37135  Status 7144                      Totals I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
PWX-37136  CurrFileOpened : 2015-08-11 13:20:39 I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
PWX-37137  Active Cycle : 2015-08-11 13:21:01 I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684

```

A subset of these monitoring statistics are printed on screen:

```

PWX-37132  Controller: (981/983/1849) KB    Command Handler: (0/0/34) KB    Writer: (5127/5147/5181)
KB
PWX-37135  Status 7144                      Totals I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
PWX-37136  CurrFileOpened : 2015-08-11 13:20:39 I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684
PWX-37137  Active Cycle : 2015-08-11 13:21:01 I=000000024344 U=000000000000 D=000000024336
C=000000004004 Total=000000052684

```

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.
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)	Number of change records read per second by PowerExchange during the current statistics interval. The value varies, based on the quantity of change data: <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. The following factors can increase this value: <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.

Performance Counter Field	Description
2 PowerCenter Processing Status:	Overall status of the CDC session, as indicated by one of the following values: <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)	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. The following factors can affect this rate: <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)	Mean number of change records per second for the rate displayed in 2.3 Commit Rate In The Current Interval . 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.
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.
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.

Performance Counter Field	Description
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 15

Tuning CDC Sessions

This chapter includes the following topics:

- [Tuning Overview, 311](#)
- [PowerExchange DBMOVER Statements for Tuning CDC Sessions, 312](#)
- [PowerCenter Connection Attributes and Session Properties, 315](#)
- [CDC Offload Processing, 318](#)
- [Multithreaded Processing, 320](#)

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 10, “Remote Logging of Data” on page 228](#).

RELATED TOPICS:

- [“PowerCenter Connection Attributes for Tuning CDC Sessions ” on page 315](#)
- [“PowerExchange DBMOVER Statements for Tuning CDC Sessions” on page 312](#)
- [“Tuning Commit Processing” on page 317](#)

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 319](#)
- [“Example of CDC Offload Processing with an Oracle Source” on page 320](#)
- [“Enabling Offload Processing for CDC Sessions” on page 319](#)

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 configuration file 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 configuration file on the PowerCenter Integration Service machine.

The following table identifies the CAPI_CONNECTION statement types to copy for each Linux, UNIX, and Windows source type:

Source Type	CAPI_CONNECTION Statement Types
Db2 on Linux, UNIX, or Windows	UDB
Microsoft SQL Server	MSQL
MySQL	MYSQL
Oracle	ORAD, for PowerExchange Express CDC for Oracle sources
PostgreSQL	PG

Example of CDC Offload Processing with an Oracle 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 an Oracle source. You are using the PowerExchange Express CDC for Oracle.

The source data remains on the Oracle system but all column-level and UOW Cleanser processing is offloaded to the PowerCenter Integration Service machine.

1. Copy the ORAD CAPI_CONNECTION statement from the dbmover configuration file on the Oracle source system to the dbmover configuration file on the PowerCenter Integration Service machine.

The PowerExchange Listener uses these statements to access change data from the specified Oracle instance.

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 ORAD CAPI_CONNECTION statement.
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, 322](#)

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 HDATE time stamp from the PLOG 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 LXX log. The LOCAL_TIME parameter in the ECCR configuration member, ECCRDMP, 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 ECCR captured the change data record. Each record in a UOW 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	<p>The time stamp type is controlled by the TIME_STAMP_MODE parameter setting in the OPTIONS statement of the Express CDC configuration file.</p> <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

- alternative logging [41, 59](#)
- Amazon RDS for Oracle sources
 - configuration steps [160](#)
 - creating redo log directories [165](#)
 - enabling ARCHIVELOG mode [166](#)
 - Express CDC user privileges for read access [166](#)
- Application Name attribute [270](#)
- application names [280](#)
- architectural diagrams
 - batch or continuous extraction processing [22](#)
 - real-time extraction processing [22](#)
- architecture
 - PowerExchange Express CDC for Oracle [133](#)
- architecture, PowerExchange CDC [22](#)
- archive log destination
 - for Express CDC for Oracle [165](#)
- ARCHIVELOG mode
 - enabling for PowerExchange Express CDC for Oracle [166](#)
- ASM
 - authority requirements for Oracle ASM users [168](#)
- ASM environments
 - Express CDC configuration considerations [149](#)
 - staging file use [150](#)
- ASMSTAGING statement
 - PowerExchange Express CDC for Oracle configuration file [181](#)

B

- batch extraction mode
 - use for terminating CDC sessions [292](#)
- before indicator (BI) fields
 - use cases [256](#)
- benefits
 - PowerExchange Express CDC for Oracle [133](#)
- binary log
 - changing the location or base name [131](#)

C

- CAPI Connection Name Override attribute [268](#)
- CAPI connection statements
 - CAPI_CONNECTION statement [27](#)
 - CAPI_SRC_DFLT statement [27](#)
 - CAPX parameters [60](#)
 - HANA parameters [225](#)
 - introduction [28](#)
 - MEMCACHE parameter [312](#)
 - MSQL parameters [107](#)
 - MYSQL parameters [125](#)
 - ORAD parameters [175](#)
 - PostgreSQL parameters [213](#)
 - RSTRADV parameter [312](#)

- CAPI connection statements (*continued*)
 - UDB parameters [89](#)
- CAPI_CONNECTION - CAPX statement
 - DBMOVER configuration file [60](#)
- CAPI_CONNECTION - HANA statement
 - DBMOVER configuration file [225](#)
- CAPI_CONNECTION - MSQL statement
 - DBMOVER configuration file [107](#)
- CAPI_CONNECTION - MYSQL statement
 - DBMOVER configuration file [125](#)
- CAPI_CONNECTION - ORAD statement
 - DBMOVER configuration file [175](#)
- CAPI_CONNECTION - PG statement
 - DBMOVER configuration file [213](#)
- CAPI_CONNECTION - UDB statement
 - DBMOVER configuration file [89](#)
- CAPI_SRC_DFLT statement
 - DBMOVER configuration file [29](#)
- CAPT_PATH statement
 - DBMOVER configuration file [31](#)
- CAPT_XTRA statement
 - DBMOVER configuration file [31](#)
- capture catalog table
 - creating [87](#)
 - DTLUCUDB SNAPSHOT command [87](#)
 - initializing the table [87](#)
- capture registrations
 - adding another registration for Express CDC for Oracle [202](#)
 - grouping in PowerExchange Logger group definition file [62](#)
 - settings for the PowerExchange Logger [44](#)
- CAPX CAPI_CONNECTION parameters
 - parameters and syntax [60](#)
- CDC
 - using Expression and Lookup Transformations to retrieve CLOB data [146](#)
- CDC sessions
 - adding sources with DTLUAPPL-generated CURRENT_RESTART tokens [294](#)
 - adding sources with special override CURRENT_RESTART tokens [294](#)
 - changing and restarting [293](#)
 - cold starting [290](#)
 - commit processing [259](#)
 - default restart points [281](#)
 - defining terminating conditions [292](#)
 - methods of starting [280](#)
 - monitoring in PowerCenter [307](#)
 - monitoring in PowerExchange [299](#)
 - multithreaded processing [320](#)
 - offload processing [318](#)
 - performance details in Workflow Monitor [308](#)
 - PowerExchange Logger for LUW logging of data from remote source [228](#)
 - processing of multiple source definitions [258](#)
 - recovering [295](#)
 - recovery example [296](#)

- CDC sessions (*continued*)
 - recovery start [291](#)
 - restart points for warm starts [282](#)
 - restart token file [285](#)
 - session and connection attributes for CDC [266](#)
 - start methods [289](#)
 - stop command processing [292](#)
 - stopping [291](#)
 - tuning [311](#)
 - tuning buffer memory [317](#)
 - tuning overview [261](#)
 - warm starting [290](#)
- CDCT file [38](#), [76](#), [79](#)
- change data capture (CDC)
 - architecture [22](#)
 - data source types [16](#)
 - DB2 for Linux, UNIX, and Windows CDC [81](#)
 - overview [14](#)
 - PowerExchange components [20](#)
 - SQL Server CDC [99](#)
 - task summary [25](#)
- change data extraction
 - connection attributes for Logger for LUW log files remote from the source [239](#)
 - creating restart tokens [283](#)
 - extraction modes [249](#)
 - monitoring in PowerCenter [307](#)
 - monitoring in PowerExchange [299](#)
 - multithreaded processing [320](#)
 - offload processing [318](#)
 - overview [15](#), [248](#)
 - overview of extracting change data [262](#)
 - task flow [264](#)
 - testing extraction maps [264](#)
 - tuning CDC sessions [311](#)
- change indicator (CI) fields [256](#)
- CHECKPOINT statement
 - PowerExchange Express CDC for Oracle configuration file [182](#)
- close command (pwxcmd) [34](#)
- closeforce command (pwxcmd) [34](#)
- cold starts
 - CDC sessions [290](#)
 - determining restart tokens [281](#)
- commit processing
 - commitment control attributes [272](#)
 - examples [275](#)
 - in CDC sessions [259](#)
 - target latency [272](#)
 - tuning [317](#)
- components
 - PowerExchange Express CDC for Oracle [133](#)
- components, PowerExchange
 - for CDC [20](#)
 - PowerExchange Listener [20](#), [27](#)
 - PowerExchange Logger [21](#)
 - PowerExchange Navigator [21](#)
- configuration in Oracle
 - enabling minimal global supplemental logging for Express CDC [169](#)
- configuration in PowerExchange
 - configuration tasks [169](#)
 - customizing the Express CDC configuration file [179](#)
 - example dbmover.cfg configuration file [178](#)
- configuration tasks
 - DB2 for Linux, UNIX, and Windows CDC [85](#)
 - PowerExchange Listener [27](#)
 - PowerExchange Logger [44](#)
 - SQL Server CDC [105](#)

- connection attributes
 - Application Name [270](#)
 - attributes to set for CDC [266](#)
 - CAPI Connection Name Override [268](#)
 - commitment control attributes [272](#)
 - Event Table [271](#)
 - Idle Time [268](#)
 - Image Type attribute [267](#)
 - PWX Latency in seconds [271](#)
 - restart control attributes [270](#)
 - RestartToken File Folder [270](#)
 - RestartToken File Name [270](#)
- continuous extraction mode [249](#)

D

- data maps
 - use in DB2 for Linux, UNIX, and Windows CDC [94](#)
- data sources, types [16](#)
- database row tests
 - testing data access with an extraction map [264](#)
- DATABASE statement
 - PowerExchange Express CDC for Oracle configuration file [183](#)
- datatypes
 - DB2 for Linux, UNIX, and Windows [83](#)
 - MySQL datatypes supported for CDC [121](#)
 - Oracle datatypes supported for Express CDC [140](#)
 - PostgreSQL datatypes supported for CDC [208](#)
 - SAP HANA datatypes supported for CDC [219](#)
 - SQL Server [101](#)
- DB2 for Linux, UNIX, and Windows CDC
 - changing a source table definition [95](#)
 - configuring in DB2 [85](#)
 - configuring in PowerExchange with the Logger [86](#)
 - configuring in PowerExchange without the Logger [86](#)
 - creating the capture catalog table [87](#)
 - datatypes supported [83](#)
 - dbmover.cfg parameters [88](#)
 - example dbmover.cfg statements [88](#)
 - IBM APARs [97](#)
 - initializing the capture catalog table [87](#)
 - overview [81](#)
 - planning [82](#)
 - planning considerations [84](#)
 - prerequisites [82](#)
 - stopping [95](#)
 - troubleshooting [97](#)
 - user authority requirement [82](#)
 - using a data map [94](#)
- DB2 partitioned databases
 - adding or dropping partitions [96](#)
 - reconfiguration tasks [96](#)
 - reconfiguring a database partition group [97](#)
- DB2 SQL1224 error [97](#)
- DB2CODEPAGE environment variable [85](#)
- DB2NOEXITLIST environment variable [85](#)
- DBMOVER configuration file
 - APPBUFSIZE statement [312](#)
 - CAPI_CONNECTION statements [27](#)
 - CAPI_SRC_DFLT statement [27](#)
 - CAPT_PATH parameter [59](#)
 - CAPT_PATH statement [27](#)
 - CAPT_XTRA statement [27](#)
 - DB2 for Linux, UNIX, and Windows CDC parameters [88](#)
 - DB2 for Linux, UNIX, and Windows example statements [88](#)
 - example for capture system [178](#)
 - general CDC parameters [27](#)

DBMOVER configuration file (*continued*)

- LOGPATH parameter [59](#)
- NODE and LISTENER buffer size parameters [312](#)
- PowerExchange Logger for LUW logging of remote source data [238](#), [239](#)
- PowerExchange Logger parameters [59](#)
- SQL Server CDC example statements [107](#)
- SQL Server CDC parameters [106](#)
- SVCNODE parameter [59](#)
- TRACE statement [312](#)
- TRACING parameter [59](#)
- types of CAPI connection statements for CDC [28](#)

DBMOVER statements

- CAPI_CONNECTION - CAPX [60](#)
- CAPI_CONNECTION - HANA [225](#)
- CAPI_CONNECTION - MSQL [107](#)
- CAPI_CONNECTION - MYSQL [125](#)
- CAPI_CONNECTION - ORAD [175](#)
- CAPI_CONNECTION - PG [213](#)
- CAPI_CONNECTION - UDB [89](#)
- CAPI_SRC_DFLT [29](#)
- CAPT_PATH [31](#)
- CAPT_XTRA [31](#)
- key statements for PowerExchange Express CDC for Oracle [171](#)
- ORACLE_CAPTURE_TYPE statement [174](#)
- ORACLE_UNHANDLED_NUMASCHAR statement [177](#)
- ORACLEID [172](#)

DDL catalog tables

- adding initial snapshots of source table definitions [129](#)
- creating for MySQL CDC [128](#)
- overview [117](#)

DDL reporting

- Express CDC for Oracle reporting of DDL operations [204](#)

detail.log [41](#)

diagrams

- batch or continuous extraction processing [22](#)
- real-time extraction processing [22](#)

DICTIONARY statement

- PowerExchange Express CDC for Oracle configuration file [186](#)

DIRSUB statement

- PowerExchange Express CDC for Oracle configuration file [188](#)

DISPLAY ACTIVE command [302](#)

DTL__BI_columnname column

- described [250](#)

DTL__CAPXACTION

- described [250](#)

DTL__CAPXCASDELIND

- described [250](#)

DTL__CAPXRESTART1 column

- described [250](#)
- displaying sequence token [283](#)

DTL__CAPXRESTART2 column

- described [250](#)
- displaying restart token [283](#)

DTL__CAPXROWID column

- described [250](#)

DTL__CAPXRRN column

- described [250](#)

DTL__CAPXTIMESTAMP column

- described [250](#)

DTL__CAPXTIMESTAMP field

- types of reported time stamps by data source [322](#)

DTL__CAPXUOW column

- described [250](#)

DTL__CAPXUSER column

- described [250](#)

DTL__CI_columnname column

- described [250](#)

DTL__ST column

- described [250](#)

DTLUAPPL utility

- displaying restart tokens in generated columns [283](#)

DTLUCUDB SNAPSHOT command [87](#)

DTLUTSK utility

- stopping CDC sessions [291](#)

dynamic dictionary

- DDL for manually creating a state table [155](#)

E

environment information checklist [137](#)

Event Table attribute [271](#)

event table processing

- guidelines for using [270](#)
- implementing [271](#)
- use for terminating CDC sessions [292](#)

Express CDC for Oracle

- Amazon RDS for Oracle sources [160](#)

extraction map columns, PowerExchange-generated

- DTL__BI_columnname [250](#)

- DTL__CAPXACTION [250](#)

- DTL__CAPXCASDELIND [250](#)

- DTL__CAPXRESTART1 [250](#)

- DTL__CAPXRESTART2 [250](#)

- DTL__CAPXROWID [250](#)

- DTL__CAPXRRN [250](#)

- DTL__CAPXTIMESTAMP [250](#)

- DTL__CAPXUOW [250](#)

- DTL__CAPXUSER [250](#)

- DTL__CI_columnname [250](#)

- DTL__ST [250](#)

extraction maps

- BI and CI fields [256](#)

- PowerExchange-generated columns [250](#)

extraction modes [249](#)

extraction of change data

- connection attributes for Logger for LUW log files remote from the source [239](#)
- creating restart tokens [283](#)
- extraction modes [249](#)
- monitoring in PowerCenter [307](#)
- monitoring in PowerExchange [299](#)
- multithreaded processing [320](#)
- offload processing [318](#)
- overview [248](#)
- overview of extracting change data [262](#)
- task flow [264](#)
- testing extraction maps [264](#)
- tuning CDC sessions [311](#)

F

file switches

- description [41](#)

- FILESWITCH command [71](#)

flush latency [271](#), [272](#)

G

group definition file

- configuring for PowerExchange Logger [62](#)

- example file [65](#)

- GROUP statement [63](#)

group definition file (*continued*)
REG statement [63](#)
SCHEMA statement [63](#)
statements and parameters [63](#)

H

HANA_CAPI_CONNECTION parameters
parameters and syntax [225](#)

I

Idle Time attribute
use for terminating CDC sessions [292](#)
Image Type attribute [267](#)
integration with PowerCenter [24](#)

L

listtask (pwxcmd) [34](#), [302](#)
LISTTASK command [302](#)
LOBs
Oracle [146](#)
lock files [40](#)
log files of PowerExchange Logger
file switches [41](#)
log files, PowerExchange Logger
maintaining with PWXUCDCT commands [76](#)
name format [39](#)

M

management and maintenance tasks
adding a capture registration [202](#)
changing the structure of a MySQL table [130](#)
changing the structure of a PostgreSQL table [216](#)
changing the structure of an Oracle table [203](#)
monitoring Express CDC for Oracle read progress [202](#)
overview for PowerExchange Express CDC for Oracle [201](#)
stopping CDC processing for a table [203](#)
stopping CDC processing temporarily [203](#)
stopping MySQL CDC processing temporarily [130](#)
stopping PostgreSQL CDC processing temporarily [216](#)
stopping SAP HANA CDC processing temporarily [227](#)
Maximum Rows Per commit attribute [272](#)
message log files [41](#)
Microsoft SQL Server CDC
changing a source table definition [113](#)
configuration tasks [105](#)
configuring in PowerExchange with the Logger [106](#)
configuring in PowerExchange without the Logger [105](#)
datatypes supported [101](#)
dbmover.cfg parameters [106](#)
example dbmover.cfg statements [107](#)
operational considerations [102](#)
overview [99](#)
planning [100](#)
prerequisites [100](#)
stopping [113](#)
user authority requirements [100](#)
minimal global supplemental logging
enabling for Express CDC for Oracle [169](#)
Minimum Row Per commit attribute [272](#)

monitoring CDC sessions
methods [299](#)
performance details in Workflow Monitor [308](#)
PowerCenter [307](#)
PowerCenter session log messages [307](#)
PowerExchange extraction statistics messages [300](#)
PowerExchange multithreaded processing statistics [301](#)
PowerExchange read progress messages [300](#)
viewing performance details in PowerCenter [310](#)
monitoring Express CDC for Oracle read progress [202](#)
MSQL CAPI_CONNECTION parameters
parameters and syntax [107](#)
multiple-source processing
in CDC sessions [258](#)
multithreaded processing
guidelines for usage [321](#)
overview [261](#), [320](#)
statistics messages [301](#)
MYSQL CAPI_CONNECTION parameters
parameters and syntax [125](#)
MySQL CDC
adding snapshots of source table definitions to the catalog [129](#)
binary log file use [117](#)
changing a source table definition [130](#)
changing the binary log location or base name [131](#)
configuring MySQL for CDC [123](#)
creating the DDL catalog tables [128](#)
datatypes supported [121](#)
DDL catalog overview [117](#)
implementation task flow [122](#)
maintaining and managing the CDC environment [129](#)
operational considerations [118](#)
overview [116](#)
stopping CDC processing for a table [129](#)
stopping CDC processing temporarily [130](#)
MySQL datatypes
supported for CDC [121](#)

O

offload processing
enabling for CDC sessions [319](#)
example of offload processing [320](#)
overview [261](#), [318](#)
rules and guidelines [319](#)
operational considerations
PowerExchange Express CDC for Oracle [142](#)
OPTIONS statement
PowerExchange Express CDC for Oracle configuration file [188](#)
Oracle configuration
creating an Oracle user and granting privileges [166](#)
enabling ARCHIVELOG mode for Express CDC for Oracle [166](#)
enabling minimal global supplemental logging for Express CDC [169](#)
specifying an archive log destination for Express CDC for Oracle [165](#)
Oracle Data Guard
capturing data from physical standby databases [156](#)
handling database role transitions in Express CDC [157](#)
Oracle datatypes
supported for Express CDC [140](#)
Oracle Express CDC
capturing data from physical standby databases [156](#)
DDL reporting for registered tables [204](#)
Oracle LOB data [146](#)
Oracle multitenant databases
configuration tasks for PowerExchange Express CDC [159](#)
using as PowerExchange Express CDC sources [159](#)

- Oracle user privileges
 - ASM user authority requirements [168](#)
 - required for CDC [166](#)
- ORACLE_CAPTURE_TYPE statement
 - DBMOVER configuration file [174](#)
- ORACLE_UNHANDLED_NUMASCHAR statement
 - DBMOVER configuration file [177](#)
- ORACLEID statement
 - DBMOVER configuration file [172](#)
- ORAD CAPI_CONNECTION parameters
 - parameters and syntax [175](#)
- output files, PowerExchange Logger
 - CDCT file [38](#)

P

- partitioned DB2 databases
 - adding or dropping partitions [96](#)
 - reconfiguring [96](#)
 - reconfiguring a database partition group [97](#)
- performance
 - CDC session performance details [310](#)
 - multithreaded processing [320](#)
- performance considerations
 - PowerExchange Express CDC for Oracle [148](#)
- PG CAPI_CONNECTION parameters
 - parameters and syntax [213](#)
- planning considerations
 - PowerExchange Express CDC for Oracle restrictions [139](#)
- PostgreSQL CDC
 - changing a source table definition [216](#)
 - datatypes supported [208](#)
 - implementation task flow [210](#)
 - maintaining and managing the CDC environment [215](#)
 - operational considerations [207](#)
 - overview [206](#)
 - preparing the PostgreSQL source for CDC [211](#)
 - stopping CDC processing for a table [216](#)
 - stopping CDC processing temporarily [216](#)
- PostgreSQL datatypes
 - supported for CDC [208](#)
- PowerCenter integration with PowerExchange [24](#)
- PowerExchange Client for PowerCenter (PWXPC) [24](#)
- PowerExchange components
 - for CDC [20](#)
 - PowerExchange Listener [20](#), [27](#)
 - PowerExchange Logger [21](#)
 - PowerExchange Navigator [21](#)
- PowerExchange Express CDC
 - database role transitions in a Data Guard environment [157](#)
- PowerExchange Express CDC for Oracle
 - ASM considerations [149](#)
 - benefits [133](#)
 - capturing data from physical standby databases [156](#)
 - changing a source table definition [203](#)
 - checkpointing in-flight transactions [151](#)
 - dbmover configuration file statements [171](#)
 - DDL for creating a state table for dynamic dictionary [155](#)
 - DDL reporting for registered tables [204](#)
 - gathering environment information [137](#)
 - improving performance in ASM RAC environment [150](#)
 - operational considerations [142](#)
 - Oracle datatypes supported [140](#)
 - Oracle multitenant databases as sources [159](#)
 - performance considerations [148](#)
 - restrictions [139](#)
 - specifying an Oracle archive log destination [165](#)
- PowerExchange Express CDC for Oracle (*continued*)
 - stopping CDC processing for a table [203](#)
 - stopping CDC processing temporarily [203](#)
 - tasks for capturing data from Oracle multitenant databases [159](#)
- PowerExchange Express CDC for Oracle configuration file
 - ASMSTAGING statement [181](#)
 - CHECKPOINT statement [182](#)
 - customization of statements and parameters [179](#)
 - DATABASE statement [183](#)
 - DICTIONARY statement [186](#)
 - DIRECTORY statement [188](#)
 - example configuration files [201](#)
 - OPTIONS statement [188](#)
 - RAC statement [194](#)
 - READER statement [195](#)
 - STANDBY statement [199](#)
 - STATESTORAGE statement [200](#)
- PowerExchange Listener
 - CLOSE command [34](#)
 - DISPLAY ACTIVE command [34](#), [302](#)
 - displaying active listener tasks [34](#)
 - LISTTASK command [302](#)
 - overview [27](#)
 - starting [32](#)
 - stopping [34](#)
 - STOPTASK command [34](#)
- PowerExchange Logger for Linux, UNIX, and Windows
 - backing up CDCT, checkpoint, and log files [78](#)
 - batch mode [43](#)
 - capture registrations for logging data from remote sources [239](#)
 - CDCT file [38](#)
 - cold starting [70](#)
 - CONDENSE command [71](#)
 - configuration parameters for logging from a remote source [234](#)
 - configuration tasks for remote logging [234](#)
 - configuring [44](#)
 - connection attributes for log files remote from the source [239](#)
 - continuous mode [42](#)
 - controlling [71](#)
 - DBMOVER statements for logging data from remote sources [238](#), [239](#)
 - dbmover.cfg parameters [59](#)
 - DG command [72](#)
 - DISPLAY CPU command [72](#)
 - DISPLAY EVENTS command [72](#)
 - DISPLAY MEMORY command [72](#)
 - DISPLAY RECORDS command [72](#)
 - DISPLAY STATUS command [72](#)
 - DL command [72](#)
 - example of logging data from a remote source [240](#), [242](#), [244](#)
 - FILESWITCH command [71](#)
 - group definition file [62](#)
 - lock files [40](#)
 - log file switches [41](#)
 - log files [39](#)
 - logging data from remote sources [228](#)
 - maintaining CDCT file and log files [76](#)
 - memory requirement on Linux and UNIX [43](#)
 - message log files [41](#)
 - monitoring [72](#)
 - monitoring statistics [304](#)
 - NFS access to log files [43](#)
 - operational modes [42](#)
 - output files [38](#)
 - overview [36](#)
 - pwxccl.cfg parameters [45](#)
 - regenerating the CDCT file after a failure [79](#)
 - required capture registration settings [44](#)

PowerExchange Logger for Linux, UNIX, and Windows (*continued*)

rules and guidelines for logging data from a remote source [233](#)

running in background mode on Linux or UNIX [43](#)

SHUTCOND command [71](#)

SHUTDOWN command [71](#)

start point in change stream [69](#)

starting [66](#)

stopping [71](#)

subtasks [38](#)

verbose messages [75](#)

PowerExchange-generated extraction map columns

DTL__BI_columnname [250](#)

DTL__CAPXACTION [250](#)

DTL__CAPXCASDELIND [250](#)

DTL__CAPXRESTART1 [250](#), [283](#)

DTL__CAPXRESTART2 [250](#), [283](#)

DTL__CAPXROWID [250](#)

DTL__CAPXRRN [250](#)

DTL__CAPXTIMESTAMP [250](#)

DTL__CAPXUOW [250](#)

DTL__CAPXUSER [250](#)

DTL__CL_columnname [250](#)

DTL__ST [250](#)

preparing for implementation

environment information checklist [137](#)

product overview

PowerExchange Express CDC for Oracle [132](#)

PWX Latency in seconds attribute [271](#)

pwxccl command

syntax [67](#)

pwxccl configuration file

COLL__END_LOG parameter [45](#)

pwxccl statement

parameters [68](#)

pwxccl.cfg

configuring [45](#)

example file [58](#)

parameters [45](#)

pwxccl.cfg configuration file

CAPT_IMAGE parameter [45](#)

CAPTURE_NODE parameter [45](#)

CAPTURE_NODE_EPWD parameter [45](#)

CAPTURE_NODE_PWD parameter [45](#)

CAPTURE_NODE_UID parameter [45](#)

COND_CDCT_RET_P parameter [45](#)

CONDENSE_SHUTDOWN_TIMEOUT parameter [45](#)

CONDENSENAME parameter [45](#)

CONN_OVR parameter [45](#)

DB_TYPE parameter [45](#)

DBID parameter [45](#)

DISTDB parameter [45](#)

DISTSVR parameter [45](#)

ENCRYPTEPWD parameter [45](#)

ENCRYPTOPT parameter [45](#)

ENCRYPTPWD parameter [45](#)

EXT_CAPT_MASK parameter [45](#)

FILE_FLUSH_VAL parameter [45](#)

FILE_SWITCH_CRIT parameter [45](#)

FILE_SWITCH_ON_CHECKPOINT parameter [45](#)

FILE_SWITCH_VAL parameter [45](#)

GROUPDEFS parameter [45](#)

LOGGER_DELETES_EXPIRED_CDCT_RECORDS parameter [45](#)

NO_DATA_WAIT parameter [45](#)

NO_DATA_WAIT2 parameter [45](#)

PROMPT parameter [45](#)

RESTART_TOKEN parameter [45](#)

SEQUENCE_TOKEN parameter [45](#)

SIGNALLING parameter [45](#)

pwxccl.cfg configuration file (*continued*)

STATS parameter [45](#)

VERBOSE parameter [45](#)

pwxcmd

close [34](#)

closeforce [34](#)

listtask [34](#)

listtask command [302](#)

pwxorad configuration file

ASMSTAGING statement [181](#)

CHECKPOINT statement [182](#)

DATABASE statement [183](#)

DICTIONARY statement [186](#)

DIRSUB statement [188](#)

OPTIONS statement [188](#)

RAC statement [194](#)

READER statement [195](#)

STANDBY statement [199](#)

STATESTORAGE statement [200](#)

PWXPC [24](#)

R

RAC environments

Express CDC considerations [148](#)

PowerExchange Express CDC for Oracle

RAC considerations [148](#)

RAC statement

PowerExchange Express CDC for Oracle configuration file [194](#)

READER statement

PowerExchange Express CDC for Oracle configuration file [195](#)

real-time extraction mode [249](#)

Real-Time Flush Latency in milli-seconds attribute [272](#)

recovery

CDC sessions [295](#)

example of session recovery processing [296](#)

PM_REC_STATE table [278](#), [279](#)

PM_RECOVERY table [278](#)

PM_TGT_RUN_ID table [278](#)

recovery information for nonrelational targets [279](#)

recovery state file for nonrelational targets [279](#)

recovery tables for relational targets [278](#)

recovery and restart processing [277](#)

recovery start

CDC sessions [291](#)

restart

default restart points [281](#)

methods of starting CDC sessions [289](#)

warm starting CDC sessions [290](#)

warm starts

CDC sessions [290](#)

restart control options

Application Name connection attribute [270](#)

RestartToken File Folder attribute [270](#)

RestartToken File Name attribute [270](#)

restart of CDC sessions

processing by start type [280](#)

restart token file

\$PMRootDir/Restart [284](#)

example file [288](#)

explicit override statements [285](#)

overview [257](#)

special override statement [287](#)

statement types [285](#)

syntax rules and guidelines [285](#)

restart tokens

creating for extraction sessions [283](#)

- restart tokens (*continued*)
 - determining for cold starts [281](#)
 - displaying in DTL_CAPXRESTART2 column [283](#)
 - overview [257](#)
 - recovery state file [279](#)
 - recovery state table [279](#)
- RestartToken File Folder attribute [270](#)
- RestartToken File Name attribute [270](#)
- restrictions
 - PowerExchange Express CDC for Oracle [139](#)
- row tests
 - testing data access with an extraction map [264](#)

S

- SAP HANA CDC
 - CDC Log Tables [220](#)
 - configuring SAP HANA for CDC [221](#)
 - datatypes supported [219](#)
 - implementation task flow [220](#)
 - maintaining and managing the CDC environment [227](#)
 - operational considerations [218](#)
 - overview [218](#)
 - stopping CDC processing for a table [227](#)
 - stopping CDC processing temporarily [227](#)
- SAP HANA datatypes
 - supported for CDC [219](#)
- security
 - extracting z/OS data during PowerCenter CDC sessions [263](#)
 - extracting z/OS data from remote PowerExchange Logger for LUW log files [233](#)
- sequence tokens
 - displaying in DTL_CAPXRESTART1 column [283](#)
- session attributes
 - attributes to set for CDC [266](#)
- SHOW_THREAD_PERF parameter [301](#)
- source RDBMSs [16](#)
- source table definitions
 - changing a DB2 table definition [95](#)
 - changing a MySQL table definition [130](#)
 - changing a PostgreSQL table definition [216](#)
 - changing a SQL Server table definition [113](#)
 - changing an Oracle table definition [203](#)
- SQL Server CDC
 - changing a source table definition [113](#)
 - configuration tasks [105](#)
 - configuring in PowerExchange with the Logger [106](#)
 - configuring in PowerExchange without the Logger [105](#)
 - datatypes supported [101](#)
 - dbmover.cfg parameters [106](#)
 - example dbmover.cfg statements [107](#)
 - extracting data for multiple publication databases [104](#)
 - overview [99](#)
 - planning [100](#)
 - prerequisites [100](#)
 - restrictions [102](#)

- SQL Server CDC (*continued*)
 - stopping [113](#)
 - user authority requirements [100](#)
- STANDBY statement
 - PowerExchange Express CDC for Oracle configuration file [199](#)
- STATESTORAGE statement
 - PowerExchange Express CDC for Oracle configuration file [200](#)
- stopping CDC processing
 - for a table [203](#)
 - temporarily [130](#), [203](#), [216](#), [227](#)
- STOPTASK command
 - stopping CDC sessions [291](#)
- supplemental logging
 - enabling for Express CDC for Oracle [169](#)

T

- target latency [272](#)
- task flow
 - CDC implementation [25](#)
 - extracting change data [264](#)
- terminating conditions
 - Idle Time attribute for CDC sessions [268](#)
- troubleshooting
 - DB2 for Linux, UNIX, and Windows CDC [97](#)
- tuning
 - overview of tuning options [261](#)
- tuning CDC sessions
 - APBUBFSIZE statement [312](#)
 - buffer memory [317](#)
 - CAPL_CONNECTION MEMCACHE parameter [312](#)
 - CAPL_CONNECTION RSTRADV parameter [312](#)
 - commit processing attributes [317](#)
 - DBMOVER tuning parameters [312](#)
 - methods [311](#)
 - NODE and LISTENER buffer size parameters [312](#)
 - PowerCenter CDC connection attributes [315](#)
 - TRACE statement [312](#)

U

- UDB CAPL_CONNECTION parameters
 - parameters and syntax [89](#)
- UOW Count attribute [272](#)
- use cases [256](#)
- user authority
 - DB2 for Linux, UNIX, and Windows CDC requirement [82](#)
 - SQL Server CDC requirement [100](#)

W

- warm starts
 - restart points used [282](#)