This software and documentation contain proprietary information of Informatica Corporation and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. 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 Corporation. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange and Informatica On Demand are trademarks or registered trademarks of Informatica Corporation in the United States and in jurisdictions throughout the world. All 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, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright © Genivia, Inc. All rights reserved. Copyright © 2007 Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved.

This product includes software developed by the Apache Software Foundation (http://www.apache.org/), and other software which is licensed under the Apache License, Version 2.0 (the "License"). You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software which was developed by Mozilla (http://www.mozilla.org/), software copyright The JBoss Group, LLC, all rights reserved; software copyright © 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under the GNU Lesser General Public License Agreement, which may be found at http://www.gnu.org/licenses/lgpl.html. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright © 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at http://www.openssl.org.

This product includes Curl software which is Copyright 1996-2007, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://curl.haxx.se/docs/copyright.html. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (©) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://www.dom4j.org/license.html.

The product includes software copyright © 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://svn.dojotoolkit.org/dojo/trunk/LICENSE.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://source.icu-project.org/repos/icu/icu/trunk/license.html.

This product includes software copyright © 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at http://www.gnu.org/software/kawa/Software-License.html.

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at http://www.opensource.org/licenses/mit-license.php.
This product includes software developed by Boost (http://www.boost.org/) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at http://www.boost.org/LICENSE_1_0.txt.

This product includes software copyright © 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at http://www.pcre.org/license.txt.

This product includes software copyright © 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at http://www.eclipse.org/org/documents/epl-v10.php.


This product includes software licensed under the Academic Free License (http://www.opensource.org/licenses/afl-3.0.php), the Common Development and Distribution License (http://www.opensource.org/licenses/cddl1.php) the Common Public License (http://www.opensource.org/licenses/cpl1.0.php) and the BSD License (http://www.opensource.org/licenses/bsd-license.php).

This product includes software copyright © 2003-2006 Joe WaInes, 2006-2007 XStream Committers. All rights reserved. Permissions and limitations regarding this software are subject to terms available at http://xstream.codehaus.org/license.html. This product includes software developed by the Indiana University Extreme! Lab. For further information please visit http://www.extreme.indiana.edu/.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,440,226; 6,789,096; 6,820,077; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,254,590; 7,281,001; 7,421,458; and 7,584,422, international Patents and other Patents Pending.

DISCLAIMER: Informatica Corporation provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of non-infringement, merchantability, or use for a particular purpose. Informatica Corporation does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

This Informatica product (the “Software”) includes certain drivers (the “DataDirect Drivers”) from DataDirect Technologies, an operating company of Progress Software Corporation (“DataDirect”) which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY, NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

Part Number: PWX-CCi-900-0001
# Table of Contents

Preface ...........................................................................  5  
Informatica Resources. .....................................................  5  
  Informatica Customer Portal. ...........................................  5  
  Informatica Documentation. ...........................................  5  
  Informatica Web Site. ...................................................  5  
  Informatica How-To Library. ..........................................  5  
  Informatica Knowledge Base. ........................................  5  
  Informatica Multimedia Knowledge Base. .........................  6  
  Informatica Global Customer Support. .............................  6  

Chapter 1: Change Data Capture Introduction .....................................  1  
  PowerExchange CDC Overview. ......................................  1  
  PowerExchange CDC Components. ................................  3  
  PowerExchange Condense Processing. .............................  3  
      Rollbacks or Backouts. ..............................................  4  
      Multiple Changes to the Same Record. ........................  4  
  Change Data Extraction. .............................................  4  
      Extraction Maps. ..................................................  4  
      Extraction Process. ...............................................  4  
  PowerExchange Integration with PowerCenter. ....................  5  

Chapter 2: PowerExchange Listener ..............................................  6  
  PowerExchange Listener Overview. ..................................  6  
  Configuring the PowerExchange Listener. .........................  6  
      Summary of DBMOVER Parameters for CDC. ..................  7  
      CAPI_CONNECTION Statements. .................................  7  
  Starting PowerExchange Listener. ..................................  15  
  Stopping the PowerExchange Listener. ............................  16  
  Stopping a PowerExchange Listener Task for an Extraction.  .  16  
  Displaying Active PowerExchange Listener Tasks. ...............  16  

Chapter 3: PowerExchange Condense ...........................................  17  
  PowerExchange Condense Overview. ...............................  17  
  PowerExchange Condense Tasks. ....................................  17  
      Controller Task. ..................................................  19  
      Condense Task and Condense Cycles. ........................  19  
      Command Handler Task. ...........................................  20  
      Diagnostic Task. ...................................................  20  
  Files Used by PowerExchange Condense. ..........................  20  

Table of Contents
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring CDC Sessions in PowerExchange.</td>
<td>86</td>
</tr>
<tr>
<td>Monitoring CDC Sessions in PowerCenter.</td>
<td>89</td>
</tr>
<tr>
<td>Tuning Change Data Extractions.</td>
<td>92</td>
</tr>
<tr>
<td>Using PowerExchange Parameters to Tune CDC Sessions.</td>
<td>93</td>
</tr>
<tr>
<td>Using Connection Options to Tune CDC Sessions</td>
<td>95</td>
</tr>
<tr>
<td>CDC Offload and Multithreaded Processing.</td>
<td>97</td>
</tr>
<tr>
<td>Planning for CDC Offload and Multithreaded Processing.</td>
<td>98</td>
</tr>
<tr>
<td>Enabling Offload and Multithreaded Processing for CDC Sessions.</td>
<td>99</td>
</tr>
<tr>
<td>Configuring PowerExchange to Capture Change Data on a Remote System.</td>
<td>100</td>
</tr>
<tr>
<td>Extracting Change Data Captured on a Remote System.</td>
<td>106</td>
</tr>
<tr>
<td>Configuration File Examples for CDC Offload Processing.</td>
<td>106</td>
</tr>
<tr>
<td>Index</td>
<td>109</td>
</tr>
</tbody>
</table>
Preface

This guide provides information for configuring and implementing DB2® for i5/OS® change data capture (CDC) environments.

This guide pertains to the PowerExchange for DB2 for i5/OS product.

Before implementing change data capture, verify that you have installed the required PowerExchange components.

Informatica Resources

Informatica Customer Portal

As an Informatica customer, you can access the Informatica Customer Portal site at http://my.informatica.com. The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, the Informatica Multimedia Knowledge Base, Informatica Documentation Center, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to the Informatica Documentation Center from http://my.informatica.com.

Informatica Web Site

You can access the Informatica corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at http://my.informatica.com. The How-To Library is a collection of resources to help you learn more about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.

Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at http://my.informatica.com. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products.
You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Multimedia Knowledge Base

As an Informatica customer, you can access the Informatica Multimedia Knowledge Base at http://my.informatica.com. The Multimedia Knowledge Base is a collection of instructional multimedia files that help you learn about common concepts and guide you through performing specific tasks. If you have questions, comments, or ideas about the Multimedia Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the WebSupport Service. WebSupport requires a user name and password. You can request a user name and password at http://my.informatica.com.

Use the following telephone numbers to contact Informatica Global Customer Support:

<table>
<thead>
<tr>
<th>North America / South America</th>
<th>Europe / Middle East / Africa</th>
<th>Asia / Australia</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Toll Free</strong></td>
<td><strong>Toll Free</strong></td>
<td><strong>Toll Free</strong></td>
</tr>
<tr>
<td>+1 877 463 2435</td>
<td>00 800 4632 4357</td>
<td>Australia: 1 800 151 830</td>
</tr>
<tr>
<td><strong>Standard Rate</strong></td>
<td><strong>Standard Rate</strong></td>
<td>Singapore: 001 800 4632 4357</td>
</tr>
<tr>
<td>Brazil: +55 11 3523 7761</td>
<td>Belgium: +32 15 281 702</td>
<td><strong>Standard Rate</strong></td>
</tr>
<tr>
<td>Mexico: +52 55 1168 9763</td>
<td>France: +33 1 41 38 92 26</td>
<td>India: +91 80 4112 5738</td>
</tr>
<tr>
<td>United States: +1 650 385 5800</td>
<td>Germany: +49 1805 702 702</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Netherlands: +31 306 022 797</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Spain and Portugal: +34 93 480 3760</td>
<td></td>
</tr>
<tr>
<td></td>
<td>United Kingdom: +44 1628 511 445</td>
<td></td>
</tr>
</tbody>
</table>
Chapter 1

Change Data Capture Introduction

This chapter includes the following topics:

♦ PowerExchange CDC Overview, 1
♦ PowerExchange CDC Components, 3
♦ PowerExchange Condense Processing, 3
♦ Change Data Extraction, 4
♦ PowerExchange Integration with PowerCenter, 5

PowerExchange CDC Overview

PowerExchange Change Data Capture (CDC) for DB2 for i5/OS captures change data for DB2 for i5/OS sources. PowerExchange CDC can extract changes in real-time directly from the DB2 journals. Alternatively, PowerExchange CDC can capture changes and place them in condense files. The use of full condense files has less impact on network performance because multiple changes are not written to the same record.

Companies often need to keep a large volume of data accurate and up to date. Only data that is current, complete, and accurate can provide a solid basis for valid business decisions. Continually repeating a bulk data load process is costly in terms of space, time, and human resources. PowerExchange CDC enables companies to capture additions, revisions, and deletions to source data and data structures and replicate those changes to data warehouses or other types of targets. If several changes are made to same record, PowerExchange CDC ensures that only the latest change is applied.

If you need to materialize or rematerialize a target entirely, you can use PowerExchange in conjunction with PowerCenter to move bulk data in a single operation.
The following figure shows a simple PowerExchange CDC implementation, in which change data is moved from system A to system B:

You can use change data capture in the following scenarios:

- Maintain a subset of the source data on a target, for example, a data mart that is used for decision-making.
- Synchronize a data source with a target to maintain exact, duplicate copies.
- Apply changes to a target as soon as possible after source data is updated to meet specific business or application requirements.
- Apply changes to a target at specific intervals based on a business requirements. This method accommodates the need for data cleansing and translation.

To extract captured change data, you can use a PowerCenter task. The changes can then be processed and implemented into the data warehouse using a "pull as required" method.

You can define a single extraction request that can be re-used each time the extraction process wants to retrieve the change data. By using a predefined extraction map, you can change the underlying capture registration without necessarily changing the extraction process.

**Note:** This approach might not always be feasible depending on the registration change. After the extraction map is defined, the extraction process can be re-run periodically or manually, as required, to request the latest set of captured changes. The emphasis of the extraction process lies with the requestor of the extraction process.

PowerExchange CDC provides additional features that are designed for the specific needs of data warehouses. You can create flexible and secure change data capture implementations to fulfill requests such as “give me all change data that was collected for a given file since the last time I asked for the change data.”

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

- Do issues exist that are related to the actuality of the target data warehouse?
- Should changes be made available in specific intervals?
- What are the backup and recovery requirements?

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

For information about bulk data movement, see the PowerExchange Bulk Data Movement Guide.
PowerExchange CDC Components

PowerExchange uses the following components for change data capture and replication:

**PowerExchange Listener**

The PowerExchange Listener performs the following tasks:

- Handles all tasks related to the use of capture registrations and extraction maps.
- Communicates with the PowerExchange Navigator GUI.
- Communicates with the PowerExchange PWXPC interface or ODBC driver layer.
- Runs extraction processes requested by clients.

**PowerExchange Condense**

PowerExchange Condense provides:

- A multi-task environment containing processes to ensure the correct collection of changes based on the defined registrations.
- Multiple processes to ensure the restart ability and the inter-process communication, condense process, and general service tasks.

**PowerExchange Navigator**

A graphical user interface from which you create and maintain capture registrations and extraction maps. You can also define data maps if you need to perform some column-level processing, such as adding user-defined columns and building expressions to populate them.

You must create a capture registration for each source table for capture change data. The PowerExchange Navigator automatically generates a corresponding extraction map.

A capture registration defines the following:

- Data source type, which is AS4 for DB2 for i5/OS
- The data to collect, including the database name, table name, and columns
- How to condense the collected changes
- How to provide the changes to the extraction process

If you add or change capture registrations when using PowerExchange Condense, you must restart the PowerExchange Condense job to activate the new or changed registrations.

---

PowerExchange Condense Processing

PowerExchange Condense performs the following tasks:

- Eliminates data that has been captured because jobs or transactions were backed out or rolled back.
- When full condense is selected, eliminates multiple changes to the same record and leaves only the latest image of the change data.

The condense process in PowerExchange Condense is triggered when any of the following events occur:

- Initialization completes.
- A PowerExchange CONDENSE command is issued through the SNDPWXCMD command on the i5/OS system.
A pwxcmd condense command is issued from a Linux, UNIX, or Windows system to the PowerExchange Condense process running on the i5/OS system.

The wait period in the NO_WAIT_DATA parameter of the CAPTPARM member elapses.

PowerExchange Condense periodically condenses change data based on specified parameters. PowerExchange extracts change data in the journal receivers based on the capture registrations that specify a condense type of Full or Part. After PowerExchange Condense completes writing changes to a condense file and closes the condense file, extraction of the changes begins.

Rollbacks or Backouts

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

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

Multiple Changes to the Same Record

With change data being collected at regular intervals, it is possible that several changes are captured for the same data row or record. However, for the purpose of updating the target, only the latest update is required.

PowerExchange uses only the latest update when you specify Full for the Condense parameter in a capture registration.

Change Data Extraction

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

Extraction Maps

When you create a capture registration, the PowerExchange Navigator generates an extraction map. An extraction map identifies the fields or columns in the registered table for which to extract change data. You can edit the default extraction map or add other extraction maps for the same capture registration. In PowerCenter, you can import extraction maps to define the source for the CDC session.

Extraction Process

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

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

- On first use of a extraction process in a PowerCenter task
- By using the DTLUAPPL utility
After an application name is used in an extraction process, it preserved as part of the audit trail and additional extraction processes are sensitive to it.

PowerExchange Integration with PowerCenter

PowerCenter provides transformation and data cleansing functions that you can use in change data replications. After capturing change data, use PowerCenter in conjunction with PowerExchange to extract and transform the change data and then apply it to one or more target tables or files.

To integrate PowerExchange with PowerCenter, use either the PowerExchange Client for PowerCenter (PWXPC) or the PowerExchange ODBC drivers in PowerCenter.

Tip: Informatica recommends that you use PWXPC. PWXPC provides more functionality, better performance, and better recovery and restart capabilities. This guide assumes that you use PWXPC.

For more information about PWXPC and the PowerExchange ODBC drivers, see PowerExchange Interfaces for PowerCenter.
This chapter includes the following topics:

- PowerExchange Listener Overview, 6
- Configuring the PowerExchange Listener, 6
- Starting PowerExchange Listener, 15
- Stopping the PowerExchange Listener, 16
- Stopping a PowerExchange Listener Task for an Extraction, 16
- Displaying Active PowerExchange Listener Tasks, 16

PowerExchange Listener Overview

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

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

The PowerExchange Listener interacts with the following PowerExchange CDC components:

- PowerExchange Navigator
- Other PowerExchange Listeners

Configuring the PowerExchange Listener

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

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

- Connect to journals for source RDBMS databases and objects.
- Determine the library to use for extraction maps.
Summary of DBMOVER Parameters for CDC

The following table describes the key dbmover.cfg statements and parameters that are required for DB2 for i5/OS CDC:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPI_CONNECTION</td>
<td>Specifies a named set of parameters that the PowerExchange Consumer API (CAPI) uses to connect to the change stream and control extraction processing. A CAPI connection is specific to a data source type. You can define up to eight CAPI_CONNECTION statements in a DBMOVER configuration file for the same data source type or different data source types. Use the CAPI_SRC_DFLT parameter to indicate a default CAPI_CONNECTION for a data source type. PowerExchange requires a connection statement for real-time extraction mode. For real-time extraction, PowerExchange has a specific type of CAPI_CONNECTION statement for each data source type.</td>
</tr>
<tr>
<td>CAPI_CONN_NAME</td>
<td>Specifies the CAPI_CONNECTION statement that PowerExchange uses by default when no CAPI connection override is supplied.</td>
</tr>
<tr>
<td>CPX_DIR</td>
<td>Specifies the library that stores extraction maps. PowerExchange creates this library during the installation process, and includes its name in the DBMOVER configuration file, when you run the CRTPWXENV command. Default is CPXLIB.</td>
</tr>
</tbody>
</table>

Related Topics:

* "CAPI_CONNECTION Statements" on page 7

CAPI_CONNECTION Statements

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

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

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

Note: PowerExchange does not require CAPI_CONNECTION statements in the dbmover.cfg on the PowerExchange Navigator machine. You can register tables for CDC and perform database row tests on that machine because the PowerExchange Navigator can communicate with the PowerExchange Listener on the source machine where CDC occurs.
AS4J CAPI_CONNECTION Statement

The AS4J CAPI_Connection statement specifies the Consumer API (CAPI) parameters that are needed for DB2 for i5/OS CDC sources.

### Data Sources:
- DB2 for i5/OS sources

### Related Statements:
- UOWC
- CAPI_CONNECTION

### Required:
- Yes for DB2 for i5/OS CDC

### Syntax:
```
CAPI_CONNECTION={
    DLLTRACE=trace_id,
    NAME=name,
    TRACE=trace,
    TYPE=(AS4J, ...),
    ALMCLRPFM={Y|N},
    ALMPARTIAL={Y|N},
    AS4JRNEXIT={Y|N},
    EOF={Y|N},
    INST=instance_name,
    JOURNAL=library/journal_name,
    LIBASUSER={Y|N},
    NOCUUOMSZ=ucw_size,
    POLWAIT=seconds,
    STOPIT={CONT|TERM,number_exceptions},
    UCWRSTNY={Y|N}
}
```

### Parameters:
Enter the following required and optional parameters and options, as needed:

**DLLTRACE=trace_id**
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=name**
Required. Unique user-defined name for this CAPI_CONNECTION statement. Maximum length is eight alphanumeric characters.

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

**TYPE=(AS4J, ...)**
Required. Type of CAPI_CONNECTION statement. For DB2 for i5/OS sources, this value must be AS4J.

**ALWCLRPFM= {Y|N}**
Controls whether DB2 for i5/OS CDC processing stops or continues when PowerExchange encounters changes that result from an i5/OS Clear Physical File Member (CLRPFM) command issued against a DB2 table registered for capture. PowerExchange cannot capture changes that result from a CLRPFM command.
Enter one of the following options:

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

Default is **N.**

**Warning:** If you set this option to Y, the data integrity of the CDC targets might be damaged. Specify this parameter only at the direction of Informatica Global Customer Support.

**ALWPARTIAL={Y|N}**

Controls whether PowerExchange processes journal receivers in partial status.

Enter one of the following options:

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

Default is **N.**

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

**AS4JRNEXIT={Y|N}**

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

Enter one of the following options:

- **N.** PowerExchange does not lock journal receivers that it is processing.
- **Y.** PowerExchange locks any journal receiver that it is processing so that the journal receiver cannot be deleted from the i5/OS system. PowerExchange records the journal receivers that it is processing in a lock file called PWXJRNLCKP in the CONDLIB library. When PowerExchange switches to the next journal receiver on the chain, PowerExchange removes the record for the previous journal receiver from the lock file.

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

Default is **N.**

**EOF={Y|N}**

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

- For CDC sessions that use real-time extraction mode, specify 0 for the Idle Time attribute of the PWX DB2i5OS CDC Real Time application connection.
- For PowerExchange Condense, specify 1 for the COLL_END_LOG statement in the CAPTPARM configuration member.
- For CDC sessions that use ODBC connections, specify 0 for the WAITTIME parameter in the ODBC data source.

Default is N.

\[ \text{INST=instance\_name} \]

Required. User-defined name for the source instance. This name must match the name that is specified in the Collection Identifier property of the registration group defined in the PowerExchange Navigator.

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

\[ \text{JOURNAL=library/journal\_name} \]

Required. Library name and journal name for the journal that contains change data for registered tables and from which PowerExchange extracts change data.

Optionally, you can override this journal specification by using one of the following methods:

- For real-time extraction CDC sessions, use the Journal Name on the PWX CDC application connection.
- For PowerExchange Condense, use the JRNL statement in the CAPTPARM configuration member.
- For CAPXRT database row tests, use the AS400 Journal/Library property in the CAPXRT Advanced Parameters dialog box in the PowerExchange Navigator.
- For real-time extraction CDC sessions that use ODBC connections, use the DTLJRNL parameter for the ODBC data source.

\[ \text{LIBASUSER={Y|N}} \]

Controls whether PowerExchange populates the DTL__CAPXUSER column of each change record with the user ID or the library name and file name.

Enter one of the following options:

- N. The DTL__CAPXUSER column contains the name of the user that made the change.
- Y. The DTL__CAPXUSER column contains the library and file name to which the change was made.

Default is N.

\[ \text{NOCCUOWSZ=uow\_size} \]

Size, in number of records, of the UOW that PowerExchange creates when reading change records from i5/OS journal receivers that were created without commitment control.

Valid values are 1 through 50000.

If you do not specify commitment control when making changes to DB2 for i5/OS tables, PowerExchange creates a UOW for each change record. Use this parameter to increase the size of the UOWs that PowerExchange creates. PowerExchange and PowerCenter process larger UOWs more efficiently. Larger UOWs also reduce commit activity on the targets of the change data.
Because this parameter affects all users of this CAPI_CONNECTION statement, Informatica recommends that you use the **Minimum Rows Per commit** attribute on the PWX DB2i5OS CDC Real Time application connection to specify UOW size values for CDC sessions.

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

**POLWAIT=seconds**

Time interval, in seconds, that PowerExchange waits after reaching the end of a journal receiver before checking for new change data in that journal receiver.

Valid values are 0 through 10.

Default is 10.

**STOPIT=((CONT|TERM),number_exceptions)**

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

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

The first positional parameter must be one of the following options:

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

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

Default is (CONT,5).

**UOWRSTANY={Y|N}**

Controls the point at which extraction session can be restarted.

Enter one of the following options:

- **N**. PowerExchange starts reading change data from the journal receiver at a start-UOW boundary. If the restart point is not a start-UOW boundary, PowerExchange fails the change data extraction to maintain change data integrity.
- **Y**. PowerExchange can start reading change data at any point in the journal receiver. PowerExchange discards change data for all in-flight UOWs and issues a PWX-06734 warning message when the end-UOW is encountered. In-flight UOWs are UOWs for which PowerExchange has not seen a start-UOW.

If you cannot select a restart point at which there are no in-flight UOWs, you might need to specify UOWRSTANY=Y to allow change date extractions to be restarted.

Default is N.

**Warning**: If you use this parameter, PowerExchange might skip change data during change data extraction operations, causing the data integrity of the targets of the change data to be compromised.

**UOWC CAPI_CONNECTION Statement**

The UOWC CAPI_CONNECTION statement specifies the Consumer API (CAPI) parameters needed for the UOW Cleanser.
In the change stream for some data sources, changes from multiple UOWs are intermingled. The UOW Cleanser reconstructs the intermingled changes read from the change stream into complete UOWs in chronological order based on end time.

### Data Sources:
- DB2 for i5/OS sources
- Oracle LogMiner CDC sources
- z/OS CDC sources

### Related Statements:
- AS4J CAPI_CONNECTION for i5/OS
- LRAP CAPI_CONNECTION for z/OS
- ORCL CAPI_CONNECTION

### Required:
Yes for the noted data sources

### Syntax:
```
CAPI_CONNECTION=
    [DLLTRACE=trace_id,]
    NAME=name,
    [TRACE=trace,]
    TYPE=(UOWC, ...)
    [CAPINAME=name,]
    [BLKSIZE=block_size,]
    [DATACLASS=data_class,]
    [MEMCACHE=cache_size,]
    [RSTRAUV=seconds,]
    [SPACEPRI=primary_space,]
    [STORCLASS=storage_class,]
    [UNIT=unit]
```

### Parameters:
Enter the following required and optional parameters and options, as needed:

**DLLTRACE=trace_id**
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=name**
Required. Unique user-defined name for this CAPI_CONNECTION statement. Maximum length is eight alphanumeric characters.

**TRACE=trace**
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=(UOWC, ...)**
Required. Type of CAPI_CONNECTION statement. For the UOW Cleanser, this value must be UOWC.

**BLKSIZE=block_size**
Block size, in bytes, for the sequential UOW spill files that the UOW Cleanser creates when the memory cache cannot hold all changes for a UOW.
Valid values and defaults vary by platform:
- For i5/OS CDC sources, enter a value from 8 through 32760. Default is 32760.
- For Oracle LogMiner CDC sources, enter a value from 8 through 65535. Default is 32768.
- For z/OS CDC sources, enter a value from 8 through 32760. Default is 18452.

**CAPINAME=** *name*

Required. Value from the NAME parameter in the related source-specific CAPI_CONNECTION statement.

The source-specific CAPI_CONNECTION is one of the following statement types:
- AS4J CAPI_CONNECTION statement for i5/OS CDC sources
- LRAP CAPI_CONNECTION statement for z/OS CDC sources
- ORCL CAPI_CONNECTION statement for Oracle LogMiner CDC sources

**DATACLAS=** *data_class*

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

**MEMCACHE=** *cache_size*

Memory cache size, in kilobytes, that PowerExchange allocates to reconstruct complete UOWs.

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

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

PowerExchange processes the change stream more efficiently if it does not need to use UOW spill files. In addition to degrading extraction performance, large numbers of UOW spill files can cause a disk space shortage.

**Important:** If the change stream contains only small UOWs, the default value might be sufficient. However, the default value is often too small to eliminate UOW spill files. Informatica recommends that you specify a larger value.

The location in which PowerExchange allocates the UOW spill files varies by operating system, as follows:
- For i5/OS, PowerExchange uses CRTPF command to create a physical file for UOW spill files. PowerExchange creates the UOW spill file names by using the C/C++ tmpnam() function.
- For Linux and UNIX, PowerExchange uses the current directory by default for UOW spill files. To use a different directory, specify the TMPDIR environment variable.
  PowerExchange creates the UOW spill file names by using the operating system tempnam function with a prefix of dtlq.
  **Note:** The UOW spill files are temporary files that are deleted when PowerExchange closes them. They are not visible in the directory while 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 creates the UOW spill file names by using the Windows _tempnam function with a prefix of dtlq.

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

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

  Valid values are 1 through 519720.

  **Warning:** Because PowerExchange allocates the cache size for each extraction operation, use caution when coding large values for MEMCACHE. Otherwise, many concurrent extraction sessions might cause memory constraints.

  Default is 1024, or 1 MB.

  **RSTRADV=nnnnn**

  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.

  The wait interval is reset to 0 when PowerExchange completes processing a UOW that includes changes of interest or returns an empty UOW because the wait interval expired without any changes of 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 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 RSTRADV is not specified, PowerExchange does not advance restart and sequence tokens for a registered source during periods when no changes of interest are received. In this case, when PowerExchange warm starts, it reads all changes, including those not of interest for CDC, from the restart point.

  For DB2 for i5/OS sources, Informatica recommends that you use this parameter only if the change records that are read from i5/OS journal receivers are created under commitment control. If the change records are created without commitment control, do not specify this parameter.

  Valid values are 0 through 86400. No default is provided.

  **Warning:** A value of 0 can degrade performance because PowerExchange returns an empty UOW after each UOW processed.

  **SPACEPRI=primary_space**

  On z/OS, the primary space value that the UOW Cleanser uses to allocate UOW spill files. The UOW Cleanser does not use secondary space. Instead, when a spill file becomes full, the UOW Cleanser allocates another spill file of the same size. The SPACETYP parameter specifies the space units for this value. Default is 50 cylinders.

  SMS ACS routines can override the UOW spill file size.

  Valid values are 1 through 2147483647.

  Default is 50 cylinders.

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

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

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

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

- The QMLTTHDACN system value is set to 1 or 2. For more information about the QMLTTHDACN system value, see the IBM i Information Center for i5/OS.

- The JOBD description includes the ALWMLTTHD(*YES) parameter to allow multiple threads.

To enter the SBMJOB command, use the following syntax:

```
SBMJOB CMD(CALL FGM(dtl1ib/DTLLST) PARM(’[CONFIG=library/file(myconfig_member)] [LICENSE=library/ file(mylicense_key_member)] node1’) JOB(job_name) JOBD(datalib/ DTLLIST) PRTDEV(*JOBD) OUTQ(*JOBD) CURLIB(*CRTDJFT) INLLIBL(*JOBD)
```

Where:

- `dtllib` is the name of the PowerExchange software library that was entered at installation.
- `node1` is the PowerExchange Listener node name that was specified in the LISTENER statement of the `datalib/ CFG(DBMOVER)` configuration member.
- `job_name` is the name of the PowerExchange Listener job or started task.
- `datalib` is the user-specified name for the PowerExchange data library that was entered at installation.

You can enter this SBMJOB command at the command line.

**Sample CAPI_CONNECTION Statements**

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

```
/* CAPI_CONN_NAME=DTECAPU */
/* CAPI_CONNECTION=(NAME=DTECAPU, */
/* TYPE=(UOWC,CAFINAME=DTIPAS4)) */
/* CAPI_CONNECTION=(NAME=DTIPAS4, */
/* TYPE=(AS4J,JOURNAL=/JLIB/JFILE,INST=FREDQ02,EOF=N, */
/* STOPIT=(COUNT=5),LIBASUSER=N,AS4JRNDT=X,N,PUWAIT=10)) */
CPX_DIR=cpxlib
```

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

```
STRREXXPRC SRCMEM(STARTLST) SRCPFIE(datalib/rexx)
```

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

**Stopping the PowerExchange Listener**

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

```
SNDLSTCMD datalib CLOSE [CLOSEOPT(Force)]
```

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

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

**Stopping a PowerExchange Listener Task for an Extraction**

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

```
SNDLSTCMD LSTMSGLIB(datalib) LSTCMD(STOPTASK) STOPTASK(application_name)
```

Where:

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

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

**Displaying Active PowerExchange Listener Tasks**

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

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

```
SNDLSTCMD datalib DISPLAY DISPLAYOPT(ACTIVE)
```

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

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

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

PowerExchange Condense captures changes from DB2 for i5/OS databases and makes them available at user-defined intervals. Use the full condense option to write only the net change over the specified interval to condense files. Alternatively, use the partial condense option to write changes in chronological order by end time, without eliminating any individual change made to a record.

PowerExchange Condense Tasks

When you start PowerExchange Condense, the main task called the controller initializes and then starts the following tasks:

- Command Handler, DTLCCMDH
- Condense, DTLCCNDS
- Dump Handler, DTLCDUMP
The following figure shows the PowerExchange Condense tasks:

![Diagram showing PowerExchange Condense tasks]

The following table describes the PowerExchange Condense tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller</td>
<td>Performs the startup and controls the running of the environment.</td>
</tr>
<tr>
<td>Command</td>
<td>Handles the operator communication and the processing of commands.</td>
</tr>
<tr>
<td>Condense</td>
<td>Periodically performs condense processing.</td>
</tr>
<tr>
<td>Diagnostic</td>
<td>Performs diagnostic options such as dumping internal control structures.</td>
</tr>
</tbody>
</table>

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

The following figure shows the records and files that the condense cycle uses:

![Diagram showing records and files used by the condense cycle]
Controller Task

This program controls the shared memory. This program is the first program to gain control during start up. It acquires various global memory structures and initializes them.

The program reads the Input Parameter File and formats the parameters into an internal format in shared memory. It rebuilds the address space from checkpoint files, if any exist. Otherwise, it confirms if a cold-start is needed through a message sent to a configurable message queue (or QSYSOPR if not specified) that requires a valid response.

It then submits the jobs for the sub-tasks (DTLCCMDH, DTLCCNDS and DTLCDUMP). The initialization of the tasks is synchronized by the controller waiting for initial is at ion events. This ensures that no task can proceed beyond its 'initial is at ion phase' until all tasks are ready to proceed.

Progress is reported with PowerExchange messages to the message log.

After all tasks have shut down, the controller shuts down and PowerExchange Condense ends. A return code of 0 is returned in the case of a normal shutdown. A return code of 8 is returned in any other circumstance.

Condense Task and Condense Cycles

A condense cycle is triggered when one of the following events occur:

- Initialization completes.
- A CONDENSE command is issued on the i5/OS system, or, on a Linux, UNIX, or Windows system, a pwxcmd condense command is issued to the PowerExchange Condense process running on the i5/OS system.
- The wait period expires. This value is set in the NO_WAIT_DATA parameter in the CFGCOND(CAPTPARM) configuration member of the PowerExchange Condense library.

A single condense process reads all data written to the journal receivers since the last condense process. The last condense process is defined using the Sequence and Restart tokens.

There are two types of condense processing for source tables when using PowerExchange Condense. Specify the type of condense process is specified in the capture registration using tone of the following options:

- Full. PowerExchange Condense creates full condense files containing only the last change to a source row.
- Part. PowerExchange Condense creates partial condense files containing all the changes for the source table.

In order to save disk space, records are combined into 32K blocks.

The Full option is only allowed for tables that have a primary index, or a unique key defined through DDS. Where several different tables are written to the same Full condense file, the record is sized at the largest.

In Part mode, the records on the partial condense file are sequenced in chronological order of when each UOW ended. The exception is where a DB2 for i5/OS table was updated outside of commit control in which case they are sequenced as each update occurred. To save disk space, the records on the partial condense file are blocked into 32K records. The records might appear incorrect if you access them through PowerExchange bulk data movement or through the i5/OS DSPPFM command.

PowerExchange Condense creates keyed file for full condense files. The key is made up of the key columns for the source table. The key columns uniquely identify the row. By using them as the key only the last update is kept for that specific row.

The Condense Control Table (CDCT) is used to control the number of files together with their start and end values. This table has an entry for each file and registration tag and their start and end times. During extraction processing, the PowerExchange Listener reads the CDCT file.

After the condense process is completed, a message is issued stating to what specific point in the log we have condensed.

Note: If a condense process fails, a message indicating the failure is written to the console.
Command Handler Task

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

Diagnostic Task

The diagnosis task processes DUMP commands entered by the user. It waits for messages from the Command Handler. When it is notified that there is a message in the communication queue, it processes all the messages that finds there and marks them as PROCESSED. The DUMP commands might have several arguments consisting of table names. For example:

GWA SRT CCT ERT CDCT MSG or ALL

When it finishes processing all the commands, the Diagnostic task waits for another event from the Command Handler.

Files Used by PowerExchange Condense

To run PowerExchange Condense, a variety of files must be present. Some of these files are allocated at installation, and other files are created dynamically by PowerExchange Condense. Also, some files are reused within a process whereas others are allocated, written to, and passed to other change data capture processes, such as the condense process.

The following table lists the files required for PowerExchange Condense:

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCT</td>
<td>Contains the actual registration information. It is used by both the PowerExchange Listener and PowerExchange Condense. The PowerExchange Listener job opens the file in read/write mode, while PowerExchange Condense only reads it.</td>
</tr>
<tr>
<td>CDCT</td>
<td>Contains the information about the condense files. It is opened by PowerExchange Condense in read/write mode to update the CDCT information. The PowerExchange Listener job reads this information as part of a data extract operation.</td>
</tr>
<tr>
<td>CDEP</td>
<td>Contains the information regarding each extraction process. It keeps an audit trail for the various extractions. It is used in the PowerExchange Listener as output to the extraction process and as input to the application group in PowerExchange Navigator.</td>
</tr>
<tr>
<td>CFG</td>
<td>This file is an existing parameter file in the DATALIB library for PowerExchange. Some of the parameters are also applicable to PowerExchange Condense.</td>
</tr>
<tr>
<td>CFGCOND</td>
<td>Contains the parameters to be used with PowerExchange Condense.</td>
</tr>
</tbody>
</table>
Multiple checkpoint files are specified in the PowerExchange Condense CAPTPARM parameters. These files are dynamically created and deleted as required by the PowerExchange Condense checkpoint process. A minimum of three files are required. The files are used in rotation. At any point in time when the PowerExchange Condense is successfully started, a specific checkpoint file contains only the information regarding one specific checkpoint. Choose the checkpoint interval with care to avoid unnecessary check pointing and unnecessary delete/define operations in the checkpoint process.

The PowerExchange message file. Contains prototypes for all messages used.

The PowerExchange license key file. Contains the license key for PowerExchange and the various options allowed.

The PowerExchange log file. Contains various messages reporting the current situation or events for PowerExchange Condense and PowerExchange Listener. This file is also used to contain tracing information. Because it is written to extensively, maintenance is required to ensure it does not become too large. It is recommended that you regularly clear the file using the CLRPFM command.

Separate members produced after each file switch.

PowerExchange Condense Progress Messages and Tracing

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

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

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

PWX-06111 Controller - All tasks initialization complete.

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

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

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

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHKPTVnn</td>
<td>Multiple checkpoint files are specified in the PowerExchange Condense CAPTPARM parameters. These files are dynamically created and deleted as required by the PowerExchange Condense checkpoint process. A minimum of three files are required. The files are used in rotation. At any point in time when the PowerExchange Condense is successfully started, a specific checkpoint file contains only the information regarding one specific checkpoint. Choose the checkpoint interval with care to avoid unnecessary check pointing and unnecessary delete/define operations in the checkpoint process.</td>
</tr>
<tr>
<td>DTLMSG</td>
<td>The PowerExchange message file. Contains prototypes for all messages used.</td>
</tr>
<tr>
<td>Full Condense File</td>
<td>condlib/FULL*.</td>
</tr>
<tr>
<td>LICENSE</td>
<td>The PowerExchange license key file. Contains the license key for PowerExchange and the various options allowed.</td>
</tr>
<tr>
<td>LOG</td>
<td>The PowerExchange log file. Contains various messages reporting the current situation or events for PowerExchange Condense and PowerExchange Listener. This file is also used to contain tracing information. Because it is written to extensively, maintenance is required to ensure it does not become too large. It is recommended that you regularly clear the file using the CLRPFM command.</td>
</tr>
<tr>
<td>Partial Condense File</td>
<td>condlib/PARTIAL. Separate members produced after each file switch.</td>
</tr>
</tbody>
</table>
Configuring PowerExchange Condense

This section covers the step that creates the database tables that are required for the PowerExchange Condense software. During this step, entries are added to the CAPTPARM and DBMOVER configuration members.

Customize these configuration files, as needed, prior to running PowerExchange Condense.

PowerExchange Condense Operational Modes

A PowerExchange Condense task can operate in batch or continuous mode. To set the operational mode, specify the COLL_END_LOG parameter in the dtlca.cfg file.

Use the following information to determine which operational mode to set:

**Batch Mode**

In batch mode, a single condense cycle runs and the program shuts down. For example, a single condense process might be inserted at appropriate points in an automated schedule, following update jobs.

**Continuous Mode**

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

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

Condense files become available for reading by the change extraction process only after a file switch. File switch processing closes the open files (if they contain data) and opens a new set of condense files for future changes. Only closed condense files can be processed by extractions.

This action occurs either automatically when one of the following events occurs:

- The file switch criteria defined by the FILE SWITCH CRIT and FILE SWITCH VAL parameters are met.
- A FILES STATS command or pwxcmd fileswitch command is issued.

**Note:** A file switch does not take place if no data is present in the current condense file. If this occurs, the next file switch attempt occurs when the criteria defined by the FILE SWITCH CRIT and FILE SWITCH VAL parameters are met. If there is still no data, then this cycle continues at the set intervals until data is available.

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

**Related Topics:**

- “PowerExchange Condense Configuration File” on page 23
PowerExchange Condense Message Log Files

PowerExchange Condense writes messages to the PowerExchange message log file. PowerExchange creates this file if it does not exist. On i5/OS, this file is the DTLLOG member of the LOG file in the PowerExchange dataLib library, by default. However, if you implemented alternative logging, PowerExchange Condense writes messages to alternative log files.

To implement alternative logging, you must specify the TRACING statement in the DBMOVER configuration member. PowerExchange can then create alternative log files in the current working directory. When an alternative log file becomes full, PowerExchange rotates to another alternative log file, which prevents out-of-space conditions. Also, with alternative logging, PowerExchange buffers messages and writes them to the alternative log files on disk based on a specific flush interval. This type of buffered logging can reduce I/O activity on the log files.

For more information about alternative logging, see the PowerExchange Reference Manual.

QPRINT File Size Limits

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

PowerExchange Condense Configuration File

This topic describes the PowerExchange Condense configuration file parameters.

PowerExchange provides sample configuration files for PowerExchange Condense in the CAPTPARM member in the following library files:

- dataLib/CFG
- condLib/CFGCOND

The parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPT_IMAGE</td>
<td>Data image type that PowerExchange Condense captures to condense files.</td>
<td>- AI. After images only.</td>
</tr>
<tr>
<td></td>
<td>PowerExchange Condense can capture after images only or both before and after images of the data. This image type must be consistent with the image type delivered to the target during extraction processing.</td>
<td>- BA. Both before and after images. Default is AI.</td>
</tr>
<tr>
<td></td>
<td>If you enter AI for this parameter, the following limitations apply:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- You cannot extract before images to the target.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- You cannot use DTL_BI columns in extraction maps.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- If you add DTL_CI columns to extraction maps, any Insert or Delete operations result in Null values in these columns. Informatica recommends that you specify BA so that you have the flexibility to use either AI or BA for the PowerCenter Image Type connection attribute for extraction processing.</td>
<td></td>
</tr>
<tr>
<td>CHKPT_NUM</td>
<td>Specifies the number of checkpoint data sets.</td>
<td>Minimum value is 3.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CHKPT_BASENAME</td>
<td>Specifies the library and file that contains checkpoint members. At installation, this value is set to: CONDLIB/CHKPT. At this location, PowerExchange creates checkpoint members named Vn, where n is a number from 0 to 2. For example: CONDLIB/CHKP(V.)</td>
<td></td>
</tr>
<tr>
<td>COLL_END_LOG</td>
<td>Indicates whether the PowerExchange Condense runs in continuous mode or batch mode. Options are: - 0. Use continuous mode. After each condense process, the system waits (for the number of minutes defined in parameter NO_DATA_WAIT) and then does another condense process. Use this setting for a 24 by 7 operation. - 1. Use batch mode. The system shuts down after a single condense process. For example, a single condense process might be scheduled following a particular batch update job.</td>
<td>- 0. Continuous mode. - 1. Batch mode.</td>
</tr>
<tr>
<td>CONDENSENAME</td>
<td>Optional. A name for the command-handling service for a PowerExchange Condense or PowerExchange Logger for Linux, UNIX, and Windows process to which pwxcmd commands will be issued. Syntax is: CONDENSENAME=service_name. This service name must match the service name that is specified in the associated SVCNODE statement in the dbmover.cfg file. The SVCNODE statement specifies the TCP/IP port on which this service listens for pwxcmd commands.</td>
<td>Maximum length is 64 characters. No default.</td>
</tr>
<tr>
<td>CONDENSE_SHUTDOWN_TIME</td>
<td>Specifies the maximum time period, in seconds, for the PowerExchange Condense (DTLCACON) to shut down normally after a SHUTDOWN or pwxcmd shutdown command is issued. This period depends on your environment. For example, the number of tables being condensed determine this setting.</td>
<td>A number from 0 through 2147483647. Default is 600 seconds.</td>
</tr>
<tr>
<td>COND_CDCT_RET_P</td>
<td>CDCT and condense files retention period in days. Files older than this period and their corresponding CDCT records are deleted during start-up, fileswitch, or shutdown processing. Ensure that an adequate number of days are specified so that the files can be processed by the Capture Extraction system before they are deleted.</td>
<td>Any number greater than 0.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>COND_DIR</td>
<td>Condense files directory.</td>
<td></td>
</tr>
<tr>
<td>DBID</td>
<td>When used in conjunction with DB_TYPE, defines selection criteria for registrations in the CCT file to be processed. This value must match the instance name specified for the registration group.</td>
<td>DBID instance name used on registrations that are created in the PowerExchange Navigator.</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Database type.</td>
<td>AS4</td>
</tr>
<tr>
<td>FILE_SWITCH_CRIT</td>
<td>When used in conjunction with FILE_SWITCH_VAL, defines the criteria for deciding when to do an automatic file switch. For example, a file switch might be done after every 100,000 records or every 30 minutes. If FILE_SWITCH_CRIT value is M but the condense file contains no data when the FILE_SWITCH_VAL interval is met, a file switch does not occur.</td>
<td>- M. Minutes. - R. Records.</td>
</tr>
<tr>
<td>FILE_SWITCH_VAL</td>
<td>When used in conjunction with FILE_SWITCH_CRIT, defines the number of units at which to automatically do a file switch. For example, a file switch might be done after every 100,000 records or every 30 minutes. If FILE_SWITCH_CRIT value is M but the condense file contains no data when the FILE_SWITCH_VAL interval is met, a file switch does not occur.</td>
<td>Any number greater than 0.</td>
</tr>
<tr>
<td>JRNL</td>
<td>The fully qualified library and journal name that is used instead of the journal that is specified in the PowerExchange registration. Example: JRNL=STQA/NEWJOURNAL</td>
<td></td>
</tr>
<tr>
<td>KEY_CHANGE_ALW</td>
<td>By default, PowerExchange relies on the data key not being changed after registration. You can change this behavior with this parameter. Options are: - N. If a key is changed, PowerExchange Condense fails. - Y. If a key is changed, PowerExchange Condense ignores the change to the key and continues processing.</td>
<td>Y or N Default is N.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>When running in condense continuous mode, it defines the number of minutes to wait on commands (manually entered through the Command Handler) before starting the next condense process. Typically, the system might wait for 60 minutes before starting the next condense process. If file switching is done on minutes criteria and FILE_SWITCH_VAL is smaller than NO_DATA_WAIT, then the wait period is reduced to the smaller of the two values. If the system is running in condense batch mode (COLL_END_LOG = 1), this parameter is ignored.</td>
<td>Any number greater than 0.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| NO_DATA_WAIT2 | Defines the number of seconds to cause reading from the Journal to stop. The completion of a condense process occurs when this period of seconds expires without data being provided by the journal. | Any number greater than 0. The optimum value for setting the parameter varies according to the loading of the system.  
- If the parameter is set too low, PowerExchange Condense might incorrectly report that no data exists. A delay can occur if a large unit of work is started. A large unit of work can contain several thousand rows.  
- If you set the value of this parameter too high, an excessive period of apparent inactivity elapses before control returns to the command handler and you can enter commands.  
Recommended value is 60 seconds. |
| OBJLOC       | If an object is locked, PowerExchange waits for 0 to 10 minutes. Then, if PowerExchange is still locked out, it sends a message to the specified i5/OS message queue to ask if a retry should be attempted.  
(WaitPeriod,MessageQueue)  
For example: OBJLOC=0,"LIBL/QSYSOPR" | A number from 0 through 10. |
| RESTART_TOKEN| Specifies the restart tokens for PowerExchange Condense. PowerExchange uses these restart token values during cold start processing.  
RESTART_TOKEN  
RESTART_TOKN2  
SEQUENCE_TOKEN  
SEQUENCE_TOKN2  
You can split the token into two tokens. Define SEQUENCE_TOKN2 or RESTART_TOKN2 to enable expansion and account for possible limitations of token length caused by limited column length in the configuration files.  
For example, the sequence token is 72-byte HEX, while a typical 80-column configuration file has 65 characters available.  
The two tokens could also be used for clarity and ease of reading in the configuration files.  
You can choose the location of the split between the two tokens. PowerExchange concatenates the two parts for processing. | |
| REG          | Identifies the capture registrations that PowerExchange is to condense.  
Note: Specify capture registrations in either the CAPTPARM parameters or in the default instance in DBMOVER configuration file, but not mixed between the two.  
If you do not specify any registrations, condense processing is as before, everything in the instance which is labelled for condense. | Example: REG=myreg |
Using Multiple Journals with PowerExchange Condense

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

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

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

Within each CONDLIB library, CAPTPARM member must be customized to point to the appropriate journal and CONDLIB library.

For example, to run PowerExchange Condense for the first journal, configure the following parameters in condlib1/CFGCOND(CAPTPARM):

```plaintext
DBID=instance1 (This uses the journal in DBMover.)
CHKPT_BASENAME=condlib1/CHKPT
COND_DIR=condlib
```

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

```plaintext
DBID=instance2
JRNLLIBRARY/JOURNAL2 (The journal override.)
REG=(reg1)
REG=(reg2)
REG=(reg3, DB=library/file)
CHKPT_BASENAME=condlib2/CHKPT
COND_DIR=condlib2
```

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

```plaintext
DBID=instance3
JRNLLIBRARY/JOURNAL3 (The journal override.)
CHKPT_BASENAME=condlib3/CHKPT
COND_DIR=condlib3
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGNALLING</td>
<td>Options are:</td>
<td>- Y. Try to close down after error.</td>
</tr>
<tr>
<td></td>
<td>- Y. The system takes automatic action when certain abnormal ends occur, such as memory corruption. The system attempts to close down in an orderly manner.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- N. No automatic trapping of system errors occurs. Instead the operating system uses its default error handling, which is usually to report the offending program line and dump memory.</td>
<td>- N. Abort on error.</td>
</tr>
<tr>
<td>VERBOSE</td>
<td>Specifies whether PowerExchange Condense issues verbose or terse messages for frequent condense activity like cleanup, checkpoints, condense cycles, and file switch processing.</td>
<td>- Y. Verbose messaging.</td>
</tr>
<tr>
<td></td>
<td>- N. Terse messaging</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Default is Y.</td>
<td></td>
</tr>
</tbody>
</table>
Capture registrations can be specified in either the CAPTPARM member or the DBMOVER configuration member but not in both members. If no capture registrations are specified, all capture registrations that specify a Condense Type other than None are included.

Starting PowerExchange Condense

Before starting PowerExchange Condense, verify that the following tasks have been performed:

- Start the PowerExchange Listener.
- Verify that the required registrations have been added by the PowerExchange Navigator to the CCT file for the DB-TYPE and DBID being used in this run. If required, existing registrations can be disabled or deleted by the PowerExchange Navigator.
- Verify that the checkpoint files are in the desired state. If a cold start is required, then no checkpoint files should exist for the mask defined by CAPTPARM parameter CHKPT_BASENAME.

How a cold start behaves depends on the RESTART_TOKEN and SEQUENCE_TOKEN settings in the CAPTPARM parameters file:

- If the RESTART_TOKEN and SEQUENCE_TOKEN are not present in the CAPTPARM parameters then PowerExchange Condense starts from the current position in the journal.
- If the RESTART_TOKEN and SEQUENCE_TOKEN are present but set to zero then PowerExchange Condense starts from the beginning of the journal. This could mean that you would be re-processing data from previous capture runs.
- If the RESTART_TOKEN and SEQUENCE_TOKEN are present with valid values, PowerExchange Condense starts from that point.

The method of running PowerExchange Condense is similar to that of running the PowerExchange Listener. Use the following command syntax to run PowerExchange Condense on the same subsystem as the PowerExchange Listener:

```bash
SBKJOB CMD(CALL PGM(DTLCACON) PARM('cs-condlib/CFGCOND(CAPTPARM)')) JOB(jobname) JOBD(datalib/DTLIST)
JOBQ(*JOBD) PRTDEV(*JOBD) OUTQ(*JOBD) CURLIB(datalib) INLLIBL(*JOBD)
```

Where:

- `condlib` is name of the library that contains the PowerExchange Condense files.
- `jobname` is the job name.
- `datalib` is the user-specified name for the PowerExchange data library that was entered at installation.

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

- Command Handler (DTLCCMDH)
- Condense (DTLCNDS)
- Dump Handler (DTLCDUMP)

To run a job for each journal that you want to condense, specify the appropriate CONDLIB library name.

Note: You cannot start PowerExchange Condense by using the pwxcmd program.

Stopping PowerExchange Condense

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

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

Use the following syntax to issue the SHUTDOWN command:

```
SNDPWXCMD CMNDLRLIB(condlib) DTLCMD(SHUTDOWN)
```

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

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

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

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

Access the DTLLOG file by issuing the following command:

```
DSPPFM condlib/LOG (DTLOG)
```

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

SHUTCOND

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

Use the following syntax to issue the SHUTCOND command:

```
SNDPWXCMD CMNDLRLIB(condlib) DTLCMD(SHUTCOND)
```

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

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

Managing PowerExchange Condense

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

Displaying PowerExchange Condense Status

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

Use the following syntax to issue the DISPLAY STATUS command:

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

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

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

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

Use the following syntax to issue the FILESWITCH command:

```
SNOPWXCMD CMDHLRLIB(condlib) DTLCMD(FILESWITCH)
```

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

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

Backing Up PowerExchange Condense Output Files

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

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

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

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

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

Generating PowerExchange Condense Performance Statistics

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

The following types of statistics are reported:

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

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

The non-data timings indicate the amount of processing required to reach the restart point in the change stream after PowerExchange Condense issues the first read call.
CHAPTER 4

DB2 for i5/OS Change Data Capture

This chapter includes the following topics:
- DB2 for i5/OS Change Data Capture Overview, 31
- Planning for DB2 Change Data Capture, 32
- Configuring DB2 Journaling for Change Data Capture, 33
- Configuring PowerExchange for DB2 Change Data Capture, 39
- Managing DB2 Change Data Capture, 40

DB2 for i5/OS Change Data Capture Overview

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

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

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

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

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

RELATED TOPICS:
- “PowerExchange Condense” on page 17
Planning for DB2 Change Data Capture

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

DB2 CDC Prerequisite

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

DB2 CDC Restrictions

The following restrictions apply to DB2 CDC processing:

- You can register only physical files for change data capture.
- The maximum length of a DB2 row or column for which PowerExchange can capture changes is 32 KB.
- PowerExchange cannot process journals for registered tables that include minimized journal entries. Verify that you do not use minimized journal entries. Use the CRTJRN or CHGJRN command on the source i5/OS system to set MINENTDTA to "NONE.
- PowerExchange cannot capture change data that results from a Clear Physical File Member (CLRPFM) command on DB2 for i5/OS tables registered for change data capture. i5/OS records only a single journal entry for all delete operations that result from a CLRPFM command. A single journal entry is not sufficient for PowerExchange to properly replicate CLRPFM changes to CDC targets. By default, when PowerExchange encounters a CLRPFM journal entry, it stops CDC processing.
- If you use SQL fast delete processing, SQL DELETE operations are not written to the DB2 journals. To use DB2 for i5/OS CDC, you must prevent SQL fast delete processing by setting a query option in the QAQQINI file. By default, the SQL_FAST_DELETE_ROW_COUNT query option specifies "DEFAULT. You must change this default by setting the SQL_FAST_DELETE_ROW_COUNT parameter to "NONE. For more information about this parameter and the query option file, see the IBM DB2 for i5/OS documentation.

i5/OS Security Requirements

The following table describes the i5/OS system security requirements for the installation and execution of PowerExchange:

<table>
<thead>
<tr>
<th>File Name</th>
<th>PowerExchange Listener and PowerExchange Condense</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTHTSKLT</td>
<td>*USE</td>
</tr>
<tr>
<td>AUTHTSKSTP</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Used for PowerExchange task administration. Created with *EXCLUDE authority during the installation process. *USE authority should be granted only to user profiles that are to be permitted to list or stop PowerExchange subtasks.</td>
</tr>
<tr>
<td>CCT</td>
<td>*CHANGE</td>
</tr>
<tr>
<td>CDCT</td>
<td>*CHANGE</td>
</tr>
<tr>
<td>CDEP</td>
<td>*CHANGE</td>
</tr>
<tr>
<td>CFG</td>
<td>*USE</td>
</tr>
<tr>
<td>CFGCOND</td>
<td>*USE</td>
</tr>
</tbody>
</table>
Also, PowerExchange dynamically creates other files, such as those for capture registrations and data maps. Access to these files depends on the following settings:

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

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

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

### Journal and File Security

You must grant the appropriate authority for any journal and file objects that PowerExchange accesses to the user ID that runs the PowerExchange Listener and PowerExchange Condense.

The following table lists the objects and authority requirements:

<table>
<thead>
<tr>
<th>Object</th>
<th>Authority</th>
</tr>
</thead>
<tbody>
<tr>
<td>Journal Authority</td>
<td>*OBJEXIST</td>
</tr>
<tr>
<td>Journal Library Authority</td>
<td>*EXECUTE</td>
</tr>
<tr>
<td>Journal Receivers Authority</td>
<td>*USE</td>
</tr>
<tr>
<td>Journal Receivers Library Authority</td>
<td>*EXECUTE</td>
</tr>
<tr>
<td>File Authority</td>
<td>*USE</td>
</tr>
<tr>
<td>File Library Authority</td>
<td>*EXECUTE</td>
</tr>
</tbody>
</table>

---

**Configuring DB2 Journaling for Change Data Capture**

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

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

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

Where:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>library/file</td>
<td>Library and physical file for which changes are written to the journal.</td>
</tr>
<tr>
<td>jlib/jfile</td>
<td>The library and journal to which changes are written.</td>
</tr>
<tr>
<td>IMAGES(*BOTH)</td>
<td>Before and after images are written to the journal.</td>
</tr>
<tr>
<td>OMTJRNE(*OPNCLO)</td>
<td>Open and close entries are omitted from the journal.</td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**
- “Using Remote Journals with PowerExchange” on page 36

**Journal Receivers and DB2 CDC**

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

Ensure that any journal receivers required to extract change data are available for PowerExchange to read. You cannot delete a journal receiver until the change data has been extracted from the receiver and the restart tokens for CDC sessions or PowerExchange Condense jobs no longer reference the receiver.

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

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

**RELATED TOPICS:**
- “Determining the Journal Receiver Needed for Restart Processing” on page 41

**PowerExchange Journal Exit**

To use the PowerExchange journal exit, specify Y for the AS4JRNEXIT parameter on the AS4J CAPI_CONNECTION statement.
When you install PowerExchange and run the CRTPWXENV command, you must provide the following information:

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

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

```
WRKREGINF EXTPNT(QIBM_QJO_DLT_JRNRCV)
```

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

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

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

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

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

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

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

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

### PowerExchange Event Message Queue

You can specify an event message queue to which PowerExchange writes i5/OS-specific messages. To specify the queue, include the AS400EVENTMSGQ parameter in the DBMOVER configuration file.

If you also specify AS4JRNEXIT=Y on the AS4J CAPI_CONNECTION statement, PowerExchange writes the following message, DTL3001, to the queue after it completes change extraction processing of a journal receiver and when it reads the first journal entry in the next journal receiver on the chain:

```
Application extraction_name has finished processing entries from receiver
journal_receiver_name in library library_name.
```

PowerExchange writes this message for each PowerExchange Condense job or each change data extraction that runs in real-time extraction mode. As a result, you might see multiple DTL3001 messages on the event message queue for the same journal receiver.
Because PowerExchange writes DTL3001 messages for change data extractions that have successfully processed a journal receiver, you can use these messages to determine when it is safe to delete a journal receiver. First review the following considerations:

- Do not delete a journal receiver until all change data extraction operations have extracted change data from the receiver. If you use real-time extraction mode in CDC sessions, ensure that all of these sessions have processed a journal receiver.
- Do not delete a journal receiver until the restart tokens for all CDC sessions or PowerExchange Condense jobs that process that receiver no longer require the receiver for restart. PowerExchange might still require previous journal receivers to restart the change data extraction.
- PowerExchange does not remove messages from the event message queue. After you have deleted a journal receiver, you should remove the messages about that receiver from the message queue.

**Related Topics:**
- “Determining the Journal Receiver Needed for Restart Processing” on page 41

### Using Remote Journals with PowerExchange

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

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

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

You might want to use remote journals for change capture if your local production system has space constraints or high overhead. By running PowerExchange CDC processing on a remote system with higher availability, you can avoid degrading performance of the production system.
The following figure shows a simple PowerExchange CDC configuration in which PowerExchange reads change entries from remote journal receivers and the source tables reside only on the local system:

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

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

The remote system contains the following items:

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

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

Configuring DB2 Journaling for Change Data Capture       37
When you register DB2 source tables or perform database row tests from the PowerExchange Navigator, the PowerExchange Listener on the remote system can connect to the local system to read metadata for the DB2 source tables. The PowerExchange Listener uses the SQL package and views that are stored in the dtllib and datalib libraries on the local system to access the metadata. The capture registrations and the corresponding extraction maps are stored in the datalib and cpplib libraries, respectively, on the remote system.

**Note:** If you have copies of the source tables on the remote system and accept the default values for the RMTRDBDIRE and RMTSYSNAME parameters, the datalib and datalib libraries are not created on the local system when you run the CRTPWXENV command during installation.

### Considerations for Using Remote Journals

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

- When PowerExchange runs on the same system as the DB2 source tables, PowerExchange can validate that a registered source table is journaled to a specific journal. However, when PowerExchange runs on a remote system, PowerExchange cannot validate the association between a source table and a remote journal. In this case, you should verify that a source table is correctly associated with a remote journal. On the local system where the table resides, enter the WRKJRNA command and select the option to view remote journal information.

- When you create capture registrations for the DB2 source tables or perform database row tests in the PowerExchange Navigator, select the node name for the remote system in the **Location** field.

- You can install and run the PowerExchange Listener on the local system as well as on the remote system to access metadata for the DB2 source tables when you create capture registrations or perform database row tests. However, running the PowerExchange Listener on the local system is not necessary because the PowerExchange Listener on the remote system can access metadata for the source tables on the local system.

- PowerExchange supports i5/OS library redirection so that you can use names for the remote journals and journal libraries that are different from those of the associated local journals and journal receivers, if you prefer.

- The remote journals can be in a cascade configuration in which one remote journal writes to another remote journal, or in a broadcast configuration in which a local journal writes to multiple remote journals.

### DBMOVER Parameters for Remote Journal Use

On the remote system that contains the remote journal receivers, you must set a few parameters in the DBMOVER member to enable PowerExchange to capture changes from remote journal receivers and to read metadata from the DB2 source tables.

Configure the following DBMOVER statements and parameters in addition to those that are normally required for CDC:

- In the JOURNAL parameter of the AS4J CAPI_CONNECTION statement, enter the **library/journal_name** for the remote journal that contains entries for registered tables.

- (Optional) Edit the RMTSYSNAME and RMTRDBDIRE statements. RMTSYSNAME specifies the host name of the local system where the DB2 source tables, local journals, and local journal receivers reside. RMTRDBDIRE specifies the name of the DB2 database that contains the source tables on the local system. These parameters are automatically populated by the CRTPWXENV command during installation, but you can override the original values by using these DBMOVER parameters.

Also, on the Windows machine where the PowerExchange Navigator runs, you must define a NODE statement in the dbmover.cfg file, which enables the PowerExchange Listener on that machine to connect to the remote system where PowerExchange CDC processing occurs. An AS4J CAPI_CONNECTION statement for the source instance is not required.
Configuring PowerExchange for DB2 Change Data Capture

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

**RELATED TOPICS:**
- “PowerExchange Condense” on page 17

### Configuring PowerExchange CDC without PowerExchange Condense

Complete the following tasks to configure PowerExchange CDC if you plan not to use PowerExchange Condense:

1. Configure the DBMOVER configuration file. Include the following statements:
   - CPX_DIR
   - AS4J CAPI_CONNECTION
   - UOWC CAPI_CONNECTION
2. In the PowerExchange Navigator, create a capture registration for each source table. The PowerExchange Navigator generates a corresponding extraction map. Optionally, you can also create a data map if you want to perform field-level processing.
   
   **Tip:** Set the **Condense** option to **Part** or **Full** even though you do not plan to use PowerExchange Condense, 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 PowerExchange Condense.
3. Activate the capture registrations. Usually, this task is done after materializing the targets. For more information.

**Note:** You will need to configure your extraction sessions to run in real-time extraction mode.

### Configuring PowerExchange CDC with PowerExchange Condense

Complete the following tasks to configure PowerExchange CDC if you plan to use PowerExchange Condense.

To configure PowerExchange CDC with PowerExchange Condense:

1. Configure the DBMOVER configuration file. Define the following statements:
   - CPX_DIR
   - AS4J CAPI_CONNECTION
   - UOWC CAPI_CONNECTION
2. Configure the configuration file for PowerExchange Condense.
3. In the PowerExchange Navigator, create a capture registration for each DB2 source table. You must select **Part** or **Full** in the **Condense** drop-down list. The PowerExchange Navigator generates a corresponding extraction map.
   
   **Tip:** If capture registrations already exist for these tables, delete the existing registrations and extraction maps and create new ones.
4. Activate the capture registrations.
5. Start PowerExchange Condense.

**Note:** You must configure your change data extractions to run in batch extraction mode.
Managing DB2 Change Data Capture

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

Stopping DB2 CDC

This section describes different methods of stopping change data capture and their implications to change apply.

**Note:** When you stop the change data capture process without stopping updates to the source, you lose change data. You can avoid losing change data and rematerializing targets by stopping updates to the source instead of stopping the change data capture process.

The following table describes the methods of disabling change data capture at various levels in the DB2 environment:

<table>
<thead>
<tr>
<th>To Stop Capturing Changes for</th>
<th>Use This Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 table</td>
<td>Switch off journaling for the table.</td>
</tr>
<tr>
<td>DB2 environment</td>
<td>Stop the PowerExchange Condense job.</td>
</tr>
<tr>
<td>Any registered data object</td>
<td>Deactivate or delete the corresponding capture registration. Also, refresh the PowerExchange Condense process.</td>
</tr>
</tbody>
</table>

Additionally, if you issue a CLOSE or pwxcmd close command to stop a PowerExchange Listener, PowerExchange stops CDC sessions at the next unit of work (UOW) boundary.

**Related Topics:**
- “Stopping Journaling for DB2 for i5/OS Sources” on page 41
- “Stopping PowerExchange Condense” on page 28
- “Deleting or Changing the Status of DB2 Capture Registrations” on page 41

Altering DB2 Table Definitions

You might need to alter the structure of a table that is registered for change capture, for example, delete or alter columns.

To alter a table definition:
1. Stop journaling for the table.
2. Alter the table, as required.
3. Create a capture registration for the altered table.
4. Start journaling for the table.
Stopping Journaling for DB2 for i5/OS Sources

Use the following DB2 for i5/OS command to stop journaling for a table:

```bash
ENDJRNFPP FILE(STQA/AMFBL) JRN(STQA/STQJRN)
```

Where:
- FILE(STQA/AMFBL) indicates the journaled table.
- JRN(STQA/STQJRN) indicates the journal.

Deleting or Changing the Status of DB2 Capture Registrations

Use the PowerExchange Navigator to delete or change the status of PowerExchange capture registrations. You can change the status from Inactive to Active or from Active to History. After deleting or changing the status, restart PowerExchange Condense.

Determining the Journal Receiver Needed for Restart Processing

To restart CDC sessions at the proper location in the journal receiver chain, PowerExchange records information about the last journal entry that was successfully processed in the restart tokens for a CDC session. PowerExchange and PWXPC display the restart tokens in numerous messages, such as PWX-04565, PWX-09959, PWXPC_10081, PWXPC_10082, PWXPC_12102, PWXPC_12103, and PWXPC_12128. In addition, PWXPC writes the starting and ending restart tokens for a CDC session to files in the restart token file directory that is specified on the PWX CDC application connection.

The restart tokens consist of a pair of alphanumeric strings. The first token in the restart token pair is called the sequence token. The sequence token has a length of either 72 or 80 characters, depending upon whether PowerExchange or PWXPC displays the value, and contains the location of the last change record successfully processed by the CDC session. The second token in the restart token pair is called the restart token. The restart token has a length of 52 characters and contains the location in the change stream that is used to start reading change data when the CDC session is restarted.

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

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

```
0838F2000000
D3Kjm850as.as4rr001.A54_RRTB_SRC_001=D2D1D4F8F5F0C1E2513172FF043AA000000000000083F82513172FF043AA000000000000000
```

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

```
Column Position
---0-----1-----2----3----4-----5-----5
---5------5-----5------5------5------5------5
D2D1D4F8F5F0C1E2513172FF043AA000000000000083F80
```
In this example, the journal entry sequence number is 00000000000000000838. After converting 838 from hexadecimal to decimal, the journal entry sequence number that PowerExchange uses to start reading change data is 2104.

To determine the journal receiver PowerExchange uses to restart the CDC session, use the i5/OS DSPJRNRCVA command to display the journal receivers. The output display from the DSPJRNRCVA command includes the beginning and ending journal entry sequence number values for the journal receiver in the First sequence number and Last sequence number fields. You might need to display numerous journal receivers from the journal receiver chain before you find the one that contains the journal entry sequence value from the restart token.
CHAPTER 5

Introduction to Change Data Extraction

This chapter includes the following topics:

- Change Data Extraction Overview, 43
- Extraction Modes, 44
- PowerExchange-Generated Columns in Extraction Maps, 44
- Restart Tokens and the Restart Token File, 47
- Recovery and Restart Processing for CDC Sessions, 49
- Group Source Processing in PowerExchange, 54
- Commit Processing with PWXPC, 56
- Offload Processing, 61

Change Data Extraction Overview

Use PowerExchange in conjunction with PWXPC and PowerCenter to extract captured change data and write it to one or more targets. Review the topics in this chapter to learn key concepts about extraction processing so that you can configure CDC sessions to extract change data efficiently and to enable proper restart and recovery.

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

- For nonrelational data sources, import the extraction map from PowerExchange.
- For relational data sources, you can import either the metadata from the database or the extraction map from PowerExchange. If you import metadata from the database, 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.

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

RELATED TOPICS:

- “PowerExchange-Generated Columns in Extraction Maps” on page 44
Extraction Modes

You can use different modes to extract change data captured by PowerExchange. The extraction mode is determined by 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.

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

**Real-time extraction mode**

Continuously extracts change data directly from the DB2 for i5/OS journal receivers in near real time. Extraction processing continues until the CDC session is stopped or 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 that are closed at the time the session runs. After processing the condense files, the CDC session ends.

To implement this mode, configure the following items:
- In PowerCenter, configure a PWX CDC Change application connection for your data source type.
- In the PowerExchange Navigator, set the Condense option to Part or Full in your capture registrations.

**Continuous extraction mode.**

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

To implement this mode, configure the following items:
- On the remote Linux, UNIX, or Windows system, configure the PowerExchange Logger for Linux, UNIX, and Windows to log change data that was originally captured on i5/OS.
- In PowerCenter, configure a PWX CDC Real Time application connection for your data source type.
- In the PowerExchange Navigator, set the Condense option to Part in your capture registrations.

**RELATED TOPICS:**
- “Configuring PowerExchange to Capture Change Data on a Remote System” on page 100
- “Extracting Change Data Captured on a Remote System” on page 106

PowerExchange-Generated Columns in Extraction Maps

Besides the table columns defined in capture registrations, extraction maps include columns that PowerExchange generates. These PowerExchange-generated columns contain CDC-related information, such as the change type and timestamp.

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

When you perform 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 except the DTL__columnname_CNT and DTL__columnname_IND columns are selected in an extraction map. You must edit an extraction map to select these columns.

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

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Datatype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTL__CAPXRESTART1</td>
<td>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. The length of a sequence token varies by data source type, except on MVS where sequence tokens for all data source types have the same length. The value of DTL__CAPXRESTART1 is also known as the sequence token, which when combined with the restart token comprises the restart token pair. A sequence token for a change record is a strictly ascending and repeatable value.</td>
<td>VARBIN</td>
<td>255</td>
</tr>
<tr>
<td>DTL__CAPXRESTART2</td>
<td>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: - 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. 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. The value of DTL__CAPXRESTART2 is also known as the restart token, which when combined with the sequence token comprises the restart token pair.</td>
<td>VARBIN</td>
<td>255</td>
</tr>
<tr>
<td>DTL_CAPXRRN</td>
<td>For DB2 on i5/OS only, the relative record number.</td>
<td>DECIMAL</td>
<td>10</td>
</tr>
<tr>
<td>DTL__CAPXUOW</td>
<td>A binary value that represents the position in the change stream of the start of the UOW for the change record.</td>
<td>VARBIN</td>
<td>255</td>
</tr>
<tr>
<td>DTL__CAPXUSER</td>
<td>The user ID of the user that made the change to the data source, with the following exceptions: - DB2 for i5/OS: If you specify LIBASUSER=Y on the AS4J CAPI_CONNECTION statement, the value is the library and file name to which the change was made. - DB2 for z/OS: If you do not specify UIDFMT on the LRAP CAPI_CONNECTION, the value is the user ID of the user that made the change. Otherwise, the UIDFMT parameter determines the value. - Microsoft SQL Server. The value is null because Microsoft SQL Server does not record this information in the distribution database. - Oracle. The value might be null. If known, Oracle provides the user ID.</td>
<td>VARCHAR</td>
<td>255</td>
</tr>
<tr>
<td>Column</td>
<td>Description</td>
<td>Datatype</td>
<td>Length</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>DTL__CAPXTIMESTAMP</td>
<td>The timestamp for when the change was made to the data source, as recorded</td>
<td>CHAR</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>by the source DBMS in the following format:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- YYYYMDDhhmmsnnnnnnn</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Where:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- YYYYMDD is the date in year (YYYY), month (MM),</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>and day (DD) format.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- hhmmsnnnnnn is the time in hours (hh), minutes (mm),</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>seconds (ss), and microseconds (nnnnnn) format.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Oracle does not support microseconds in the timestamp.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTL__CAPXACTION</td>
<td>A single character that indicates the type of change operation. Valid</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>values are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- I. INSERT operation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- D. DELETE operation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- U. UPDATE operation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTL__CAPXASDELIND</td>
<td>For DB2 for z/OS sources only, a single character that indicates whether</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>DB2 has deleted the row because the table specifies the ON DELETE CASCADE</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>clause. Valid values are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Y. Indicates that DB2 deleted this row because of a cascade delete rule.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- N. Indicates that DB2 did not delete this row because of a</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>cascade delete rule.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTL__BI_columnname</td>
<td>For UPDATE operations, the value of the before image of the selected</td>
<td>Datatype of</td>
<td>Length of</td>
</tr>
<tr>
<td></td>
<td>column in the change record.</td>
<td>the source</td>
<td>the source</td>
</tr>
<tr>
<td></td>
<td></td>
<td>column</td>
<td>column</td>
</tr>
<tr>
<td>DTL__CI_columnname</td>
<td>For UPDATE operations, a single character that indicates whether the</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>selected column was changed. Valid values are:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Y. Indicates that the column changed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- N. Indicates that the column did not changed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Null value. Indicates an INSERT or DELETE operation.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTL__columnname_CNT</td>
<td>Binary count column. PowerExchange generates this column for variable</td>
<td>NUM32U</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>length columns of types VARCHAR and VARBIN to determine the length of</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the column during change data extraction processing.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> By default, binary count columns are not selected in an</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>extraction map. You must edit an extraction map to select these columns.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DTL__columnname_IND</td>
<td>Null indicator column. PowerExchange generates this column for nullable</td>
<td>BIN</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>columns to indicate the nullable value for the column.</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> By default, null indicator columns are not selected in an</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>extraction map. You must edit an extraction map to select these columns.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 a CDC session. For a new CDC session, you should generate restart token values that represent the point-in-time in the change stream where you materialized the targets. Each source in a CDC session can have unique values for its restart token pair in the restart token file.

A restart token pair matches the position in the change stream for a change record and has the following parts:

**Sequence token**

For each change record that PowerExchange reads from the change stream, a binary value that represents the change stream position of the end of the UOW for that change record followed by the position of the change record itself, with the following exceptions:

- For Microsoft SQL Server CDC, a binary value that represents the position of the change record in the distribution database.
- For change data extracted from full condense files on z/OS or i5/OS, a binary value that represents the full condense file and the position of the change record in that file.

A sequence token for a change record is a strictly ascending and repeatable value. 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.

**Restart token**

For each change record that PowerExchange reads from the change stream, a binary value that represents a position in the change stream that can be used to reconstruct the UOW state for that record, with the following exceptions:

- For Microsoft SQL Server CDC, a binary value that contains the DBID of the distribution database and the name of the distribution server.
- For change data extracted from full condense files on z/OS and i5/OS, a binary value that contains the instance name from the registration group for the capture registration.

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.

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.

PowerExchange uses these restart token values to determine the point from which to start reading change data from the change stream, with the following exceptions:

- For Microsoft SQL Server CDC, PowerExchange uses the sequence token value to determine the point from which to start reading change data from that distribution database, and the restart token value to verify that the distribution database is the same as the distribution database specified on the CAPI connection.
- For change data extracted from full condense files on z/OS or i5/OS, PowerExchange uses the sequence token value to determine the point from which to start reading change data from the condense files, and the restart token value to verify that the instance is the same as the instance recorded for the change record.

After determining the start point in the change stream for a CDC session, PowerExchange begins to read and pass change data to PWXPC. PWXPC uses the sequence token value for each source in the CDC session to determine the point at which to start providing the change data passed from PowerExchange to a specific source.
You should specify restart token values in the restart token file in the following situations:

- When creating a new CDC session, specify a restart token pair for each data source. Alternatively, you can use the special override statement to specify a restart token pair for some or all data sources.
- When adding a data source to an existing CDC session, specify a restart token pair for the new source.
- If you need to override token values for a data source that is defined in an existing CDC session, specify the override token values.

**Generating Restart Tokens**

Before you begin extracting change data, you must materialize the targets for the CDC session with data from the data sources. Usually, to perform this task, you run a bulk data movement session. After you materialize the targets and before you allow changes to be made to the data source again, you should generate restart tokens that represent the point-in-time in the change stream when the materialization occurred.

PWXPC can generate restart tokens when it starts to extract change data for a CDC session. Additionally, PowerExchange provides a number of methods to generate restart tokens. To generate restart tokens that match the current end of the change stream, use one of the following methods:

- In the PWXPC restart token file for the CDC session, specify CURRENT_RESTART on the RESTART1 and RESTART2 special override statements.
- In the PowerExchange Navigator, use the SELECT CURRENT_RESTART SQL statement when you perform a database row test.
- Run the DTLUAPPL utility with the GENERATE RSTTKN option.

If you use the DTLUAPPL utility or the PowerExchange Navigator to generate restart tokens, edit the restart token file that PWXPC uses to specify the token values before you start the CDC session.

**Restart Token File**

You can use the restart token file to provide restart tokens for a new CDC session, or for a source that you add to an existing CDC session. You can also use the restart token file to override restart tokens for sources in an existing CDC session.

Specify the name and location of the restart token file in the following attributes of the source PWX CDC application connection:

- RestartToken File Folder
- RestartToken File Name

When you run a CDC session, PWXPC reads the restart token file in the folder specified in the RestartToken File Folder attribute of the source CDC connection. If this folder does not exist and the RestartToken File Folder attribute contains the default value of $PMRootDir/Restart, PWXPC creates this folder. PWXPC does not create any other restart token folder name. PWXPC then verifies that the restart token file exists. If the file does not exist, PWXPC uses the name specified in the RestartToken File Name attribute to create an empty restart token file.

PWXPC stores restart tokens for CDC sessions at the following locations:

- For relational targets, in a state table in the target database
- For nonrelational targets, in a state file on the PowerCenter Integration Service machine

When you restart a CDC session, PWXPC reads the restart tokens for each source in the CDC session from the state table or file. PWXPC also reads the restart token file for the CDC session and overrides the restart tokens for any sources that have token values included in the file.
Recovery and Restart Processing for CDC Sessions

If you select **Resume from the last checkpoint** for the **Recovery Strategy** attribute in a CDC session that extracts change data from PowerExchange, PWXPC and PowerCenter provide recovery and restart processing for that session. In the event of a session failure, the PowerCenter Integration Service recovers the session state of operation, and PWXPC recovers the restart information.

PWXPC saves restart information for all sources in a CDC session. The restart information for CDC sessions, which includes 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:

- **Relational targets.** 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, which ensures that the applied data and the restart tokens are in-sync.
- **Nonrelational targets.** Recovery state file 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 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.

### PowerCenter Recovery Tables for Relational Targets

When the PowerCenter Integration Service runs a session that has a resume recovery strategy, it writes to recovery tables on the target database system. When the PowerCenter Integration Service recovers the session, it uses information in the recovery tables to determine where to begin loading data to target tables. PWXPC 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 privilege to the database user name 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 is 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 Integration Service stores the session state of operation in the shared location, $PMStorageDir, on the Integration Service machine. For nonrelational targets, the Integration Service also stores the target recovery status in a recovery state file in the shared location on the 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 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 Integration Service, uses the application name you specify as part of the key when it stores and retrieves the restart information for the CDC session. When you configure the PWX CDC application connection for each CDC session, specify a unique value in the `Application Name` attribute.

PWXPC appends the repository task instance ID for the CDC session to the `Application Name` value to create the `APPL_ID` value in the recovery state table and the `appl_id` portion in 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 session, changes to the CDC session such as adding and removing sources or targets affects restart processing. When you change the CDC session to add or remove sources and targets, you must use the restart token file to provide restart tokens and then cold start the CDC session.

Restart Processing for CDC Sessions

Each source in a CDC session has its own restart point. The method you use to start a CDC session controls how PWXPC determines the restart information for the sources in that session.

Use one of the following methods to start CDC sessions:

- **Cold start**. When you cold start a CDC session, PWXPC uses the restart token file to acquire restart tokens for all sources, does not read the state table or file, and makes no attempt to recover the session. The CDC session continues to run until stopped or interrupted.

- **Warm start**. When you warm start a CDC session, PWXPC reconciles the restart tokens for sources provided in the restart token file, if any, with any restart tokens that exist in the state tables or file. If necessary, PWXPC performs recovery processing. The session continues to run until stopped or interrupted.

- **Recovery start**. When you recover a CDC session, PWXPC reads the restart tokens from any applicable state tables and file. If necessary, PWXPC performs recovery processing. PWXPC then updates the restart token file with the restart tokens for each source in the CDC session, and the session ends.

Before you run a CDC session for the first time, you should create and populate the restart token file with restart tokens 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, you materialize a target table from a source and do not change the source data after materialization. To establish a starting extraction, or restart, point in the change stream, code a special override statement with the `CURRENT_RESTART` option in the restart token file that has the file name that you specified in the PWX CDC application connection in the CDC session. When you cold start
the CDC session, PWXPC requests that PowerExchange use the current end-point in the change stream as the extraction start point. After the CDC session starts, you can resume change activity to the sources.

If you cold start a CDC session and a restart token file does not exist, the PowerCenter Integration Service still runs the session. Because you did not provide any restart information, PWXPC passes null restart tokens for all sources to PowerExchange and indicates that the restart tokens for each source are NULL in message PWXPC_12060. PowerExchange then assigns the default restart point to each source.

**Warning:** If you use null restart tokens, the CDC session might not produce the correct results. When you cold start CDC sessions, provide valid restart tokens.

### Default Restart Points for Null Restart Tokens

The default restart points that PowerExchange uses when it receives null restart tokens vary by data source type. The following table describes the default restart points for null restart tokens, by data source type and extraction method:

<table>
<thead>
<tr>
<th>Data Source Type</th>
<th>Batch and Continuous Extraction Mode</th>
<th>Real-time Extraction Mode</th>
</tr>
</thead>
</table>
| All MVS sources      | Oldest condense file, as recorded in the CDCT.                                                          | Best available restart point as determined by the PowerExchange Logger for MVS, which is one of the following:  
- Oldest restart point for which an archive log is available  
- Current active log if there are no available archive logs. |
| DB2 for i5/OS        | Oldest condense file, as recorded in the CDCT.                                                          | Oldest journal receiver still attached on the journal receiver chain.                      |
| DB2 for Linux, UNIX, and Windows | Oldest PowerExchange Logger for Linux, UNIX, and Windows log file, as recorded in the CDCT. | Current log position at the time the PowerExchange capture catalog was created.             |
| Microsoft SQL Server | Oldest PowerExchange Logger for Linux, UNIX, and Windows log file, as recorded in the CDCT.             | Oldest data available in the Publication database.                                        |
| Oracle               | Oldest PowerExchange Logger for Linux, UNIX, and Windows log file, as recorded in the CDCT.             | Current Oracle catalog dump.                                                               |

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, 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 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.

---

**Group Source Processing in PowerExchange**

When you extract change data using PWX CDC application connections, PowerExchange uses group source processing for all source definitions that you include in a single mapping. With group source processing, PowerExchange reads data from the same physical source in a single pass. This processing enhances throughput and reduces resource consumption by eliminating multiple reads of the source data.

When you run a CDC session, PWXPC passes a source interest list that contains all of the sources. PowerExchange uses the source interest list to determine the sources for which to read data from the change stream. When PowerExchange encounters changes for a source in the interest list, it passes the change data to PWXPC. PWXPC then provides the change data to the appropriate source in the mapping.

If you use PWXPC connections for bulk data movement operations, PowerExchange uses group source processing for the following multiple-record, nonrelational data sources:
- IMS unload data sets
- Sequential data sets and flat files
- VSAM data sets

PowerExchange uses group source processing to read all records for a single multi-group source qualifier in a mapping. When you run a bulk data movement session, PWXPC passes PowerExchange the source data map information from the source definition metadata, which includes the data set or file name if available. If PWXPC does not pass the data set or file name, PowerExchange determines it from the PowerExchange data map. PowerExchange reads the data set or file and passes all of the data records to PWXPC. PWXPC then provides the data records to the appropriate source record type in the multi-group source qualifier.

---

**Using Group Source with Nonrelational Sources**

PowerExchange can use group source processing for some nonrelational data sources that support multiple record types in a single file.

A single mapping can contain one or more multi-record source definitions and single-record source definitions. If you use PWX NRDB Batch application connections, PWXPC creates a connection to PowerExchange for each source definition in the mapping and reads the source data.

For data sources with multiple record types, the PowerExchange data map defines a record and a table for each unique record type. The table represents the relational view of the related record.

For IMS, VSAM, and sequential or flat file data sources, you can use Designer to import data maps with multiple record types to create PowerCenter source definitions. If you want the source definition to represent only a single record type, import a single table from the data map. If you want the source definition to include all record types, import the data map as a multi-record data map.

To import the data map as a multi-record data map, select **Multi-Record Datamaps** in the **Import from PowerExchange** dialog box. If you import a multi-record data map, the source definition has a group for each
table in the data map. A group contains metadata for the fields in the table. If you import a single table from a multi-
record data map, the source definition has only a single group.

When you run a session that contains a mapping with source definitions for each table in a multi-record data map,
PowerExchange reads the data set or file once for each source definition. When you run a session that contains a
mapping with a single source definition for all records in a multi-record data map, PowerExchange uses group
source processing to read all of the records in the data set or file in a single pass.

For example, if you have a sequential file that contains three different record types, you can create a source
definition for each record type. Then create a mapping that contains the three source definitions. When you run a
session that contains the mapping, PowerExchange reads the sequential file three times.

Alternatively, if you import the data map as a multi-record data map and create a single multi-record source
definition, you can use this multi-record source definition in a mapping. When you run a session that contains this
mapping, PowerExchange reads the sequential file one time to extract the data.

When you import IMS data maps as multi-record data maps, you can use the source definitions only to process
IMS unload data sets. You cannot use multi-record IMS source definitions to read all segments from an IMS
database in a single pass. To perform bulk data movement operations on IMS databases, create mappings that
have a source definition for each segment in the IMS database.

Using Group Source with CDC Sources

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

To create source definitions in Designer that can be used to extract change data, import source metadata by using
one of the following methods:

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

**Restriction:** To read change data for nonrelational sources, you must import extraction maps from
PowerExchange.

Informatica recommends that you use extraction maps to create source definitions for all CDC sources. When you
create source definitions from extraction maps, the mapping and session creation process is simpler for the
following reasons:

- The source definition contains the extraction map name, which eliminates the need to provide it when you
  configure the session.
- The source definition contains the PowerExchange-defined CDC columns, which eliminates the need to add
  these columns to the source definition. The PowerExchange-defined columns include the change indicator and
  before image columns as well as the DTL__CAPX columns.

When you extract change data, PowerExchange uses group source processing for all source definitions in the
mapping. All source definitions must be for the same data source type, such as DB2, IMS, VSAM, or Oracle. Do
not include multiple data source types in the mapping. Otherwise, the session fails with message PWXPC_10080.

For example, you cannot run a CDC session that contains a mapping with both VSAM and IMS source definitions,
even though the change stream is the same. To extract change data for both IMS and VSAM data sources, create
unique a mapping and session for the VSAM sources and a separate, unique mapping and session for the IMS
sources. PowerExchange reads the change stream twice, once for the session with VSAM sources and once for
the session with IMS sources.
If you create a workflow that contains multiple CDC sessions, PowerExchange uses a connection for each session, even if the sessions extract change data from the same change stream, such as the PowerExchange Logger for MVS.

The following example mapping shows three DB2 sources, for which the source definitions were created from extraction maps:

![Mapping Diagram]

If you include this mapping in a session that uses a PWX DB2zOS CDC application connection, PowerExchange uses group source processing to read the change stream and extract the changes for all three source tables. PowerExchange extracts change data in chronological order, based on when the UOWs were completed. PowerExchange passes the change data to PWXPC, and PWXPC provides the changes to the appropriate source qualifier.

**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 commit properties and the commit type. Commit properties specify the commit interval and the number of UOWs or change records that you want to use as a basis for the commit. The commit type determines when the PowerCenter Integration Service commits data to the target.

By default, the **Commit Type** attribute on the session Properties tab specifies **Target**, which indicates target-based commit processing. For CDC sessions, the PowerCenter Integration Service always uses source-based commit processing, and PWXPC controls the timing of commit processing. When you run a CDC session that specifies target-based commit processing, the PowerCenter Integration Service automatically changes the commit type to source-based and writes message WRT_8226 in the session log.
PWXPC ignores the **Commit Interval** attribute. To control commit processing, configure attributes on the PWX CDC Change and Real Time application connections.

**RELATED TOPICS:**

♦ “Commitment Control Options” on page 71

### Controlling Commit Processing

To control commit processing, you can specify certain PWX CDC Real Time or Change application connection attributes.

The following table describes the connection attributes that control commit processing:

<table>
<thead>
<tr>
<th>Connection Attribute</th>
<th>Real Time or Change Connections</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows Per commit</td>
<td>Both</td>
<td>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 threshold is met. PWXPC does not wait for a UOW boundary to commit the change data. Default is 0, which means that PWXPC does not use maximum rows.</td>
</tr>
<tr>
<td>Minimum Rows Per commit</td>
<td>Real Time</td>
<td>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 value, PowerExchange skips commit records and passes only the change records to PWXPC. Default is 0, which means that PowerExchange does not use minimum rows.</td>
</tr>
<tr>
<td>Real-time Flush Latency in milliseconds</td>
<td>Real Time</td>
<td>Number of milliseconds that must pass before PWXPC flushes the data buffer to commit the change data to the targets. When this latency period expires, PWXPC continues to read the changes in the current UOW until the end of that UOW is reached. Then, PWXPC flushes the data buffer to commit the change data to the targets. Default is 0, which means that PWXPC uses 2,000 milliseconds.</td>
</tr>
<tr>
<td>UOW Count</td>
<td>Both</td>
<td>Number of UOWs that PWXPC processes before it flushes the data buffer to commit the change data to the targets. Default is 1.</td>
</tr>
</tbody>
</table>

You can specify values for all of these commitment control attributes. However, PWXPC commits change data only when one of the following values is met:

♦ **Maximum Rows Per commit**
♦ **Real-time Flush Latency in milliseconds**
♦ **UOW Count**

If you specify a value for the **Minimum Rows Per commit** attribute, this threshold must be met before a commit can occur. However, PWXPC flushes the data buffer to commit the change data to the targets only when **Maximum Rows Per commit**, **Real-time Flush Latency in milliseconds**, or **UOW Count** is met, whichever is first.

After PWXPC commits the change data, it resets the UOW count, the maximum and minimum rows, and the real-time flush latency timer. PWXPC continues to read change data. Whenever one of the commitment control values is met, PWXPC commits that 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]
```

**Restriction:** If you enable the Commit On End Of File attribute on the session Properties tab, duplicate data can occur in the targets because the Integration Service commits any remaining change data in the buffer to the targets. This final commit by the Integration Service occurs after the PWXPC CDC reader has committed all complete UOWs in the buffer, along with their restart tokens, to the targets. As a result, the final restart tokens might represent a point in the change stream that is earlier than final change data that the Integration Service commits to the targets. To prevent possible duplicate data when you restart CDC sessions, set the Commit Type attribute to Source and disable the Commit On End Of File attribute.

## Maximum and Minimum Rows per Commit

The Maximum Rows Per commit attribute controls the size of the UOWs written to the targets. The Minimum Rows Per commit attribute controls the size of the UOWs read from the change stream. You can use these attributes to mitigate the effects of processing very small or very large UOWs.

### Maximum Rows per Commit

If you have very large UOWs, you can use the Maximum Rows Per commit attribute to specify the maximum number of change records that PWXPC reads before it commits the change data to the targets. This attribute causes PWXPC to commit change data without waiting for a UOW boundary, which is called a subpacket commit. By using a subpacket commit for large UOWs, you can minimize storage use on the Integration Service machine and lock contention on the target databases.

**Warning:** Because PWXPC can commit 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.

Generally, you should use the maximum rows attribute only if you have large UOWs that cannot be processed without impacting either the Integration Service machine or the target databases. For example, if you have an application that makes 100,000 changes before it issues a commit, you can use the maximum rows attribute to commit the change data before PWXPC reads all 100,000 change records. When the maximum rows limit is met, PWXPC flushes the change data from the buffer on the Integration Service machine and commits the data to the targets. After the commit processing, the RDBMS can release the locks in the target databases for these change records and the Integration Service can reuse the buffer space for new change records.

### Minimum Rows per Commit

If your change data has many small UOWs, you can use the Minimum Rows Per commit attribute to create larger UOWs of a more uniform size. Use this attribute to specify the minimum number of change records that PowerExchange must pass to PWXPC before passing a commit record. Until the minimum rows value 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 to PWXPC and then resets the minimum rows counter.

Online transactions that run in transaction control systems such as CICS and IMS often commit after making only a few changes, which results in many, small UOWs in the change stream. PowerExchange and PWXPC can process fewer, larger UOWs more efficiently than many small UOWs. Therefore, if you use the minimum rows limit to increase the size of UOWs, you can improve CDC processing efficiency.

A minimum rows limit does not impact the relational integrity of the change data because PowerExchange does not create new commits points in the change stream data. PowerExchange simply skips some of the original commit records in the change stream.
Target Latency

Target latency is the total time that PWXPC uses to extract change data from the change stream and that the Integration Service uses to apply that data to the targets. If this processing occurs quickly, target latency is low.

The values you select for the commitment control attributes affect target latency. You must balance target latency requirements with resource consumption on the Integration Service machine and the target databases.

Lower target latency results in higher resource consumption because the Integration Service must flush the change data more frequently and the target databases must process more commit requests.

You can affect target latency by setting the commit control attributes.

The following default values can result in the lowest latency:

- 0 for Maximum Rows Per commit, which disables this option
- 0 for Minimum Rows Per commit, which disables this option
- 0 for Real-time Flush Latency in milli-seconds, which is equivalent to 2000 milliseconds or 2 seconds
- 1 for UOW Count

These values can decrease target latency because PWXPC commits changes after each UOW, or on UOW boundaries. However, these values also cause the highest resource consumption on the source system, the Integration Service machine, and the target databases. Alternatively, these values might decrease throughput because change data flushes too frequently for the Integration Service or the target databases to handle.

To lower resource consumption and potentially increase throughput for CDC sessions, specify a value greater than the default value for only one of the following attributes:

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

Disable the unused attributes.

Examples of Commit Processing

The following examples show how the commitment control attributes affect commit processing with PWXPC.

Subpacket Commit and UOW Count - Example

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 commitment control attributes have the following values:

- 300 for Maximum Rows Per commit
- 0 for Minimum Rows Per commit, which disables this attribute
- 0 for Real-time Flush Latency in milli-seconds, which is equivalent to 2 seconds
- 1 for UOW Count

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

### UOW Count and Time-Based Commits - Example

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 commitment control attributes have the following values:

- 0 for **Maximum Rows Per commit**, which disables this attribute
- 0 for **Minimum Rows Per commit**, which disables this attribute
- 5000 for **Real-time Flush Latency in milli-seconds**, which is equivalent to 5 seconds
- 1000 for **UOW Count**

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

### Minimum Rows and UOW Count - Example

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 commitment control attributes have the following values:

- 0 for **Maximum Rows Per commit**, which disables this attribute
- 100 for **Minimum Rows Per commit**
- -1 for **Real-time Flush Latency in milli-seconds**, which is disables this attribute
- 10 for **UOW Count**

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.

In this example, PWXPC commits change data after 1,000 change records, which is also after every 10 UOWs because each UOW contains 10 change records and the UOW Count is 10.

Offload Processing

You can use CDC offload processing and multithreaded processing to improve performance and efficiency of real-time CDC sessions.

You can use CDC offload processing to distribute processing to the PowerCenter Integration Service machine running the extraction, which reduces processing on the source system. You can also use CDC offload processing to copy change data to a remote system by using the PowerExchange Logger for LINUX, UNIX, and Windows.

You can use multithreaded processing to increase parallelism on the PowerCenter Integration Service machines.

CDC Offload Processing

When you extract change data, PowerExchange maps the captured data to the columns in the extraction map. PowerExchange also performs any data manipulation operations that you defined in the extraction map, such as populating change-indicator and before-image columns or running expressions. This column-level processing of change data occurs in the PowerExchange Listener and can be CPU-intensive.

By default, PowerExchange performs column-level processing on the system on which the changes are captured. For MVS, DB2 for i5/OS, and Oracle sources, PowerExchange also runs the UOW Cleanser to reconstruct complete UOWs from the change data in the change stream on the system.

To reduce the overhead of column-level and UOW Cleanser processing, you can use CDC offload processing. CDC offload processing moves the column-level and UOW Cleanser processing to the PowerCenter Integration Service machine running the extraction. CDC offload processing can also be used by the PowerExchange Logger for Linux, UNIX, and Windows to copy change data to PowerExchange Logger log files on a remote system. You can then extract the change data from the remote system rather than the original source system.

Use CDC offload processing to help increase concurrency and throughput and decrease costs in the following situations:

- You have insufficient resources on the machine where the change data resides to run the number of concurrent extraction sessions you require.
- You have insufficient resources on the machine where the change data resides to provide the necessary throughput you require.
- You have spare cycles on the PowerCenter Integration Service machine and those cycles are cheaper than the cycles on the machine on which the changes are captured.

Multithreaded Processing

If you use CDC offload processing for change data extractions, you can also use multithreaded processing, which might improve help improve throughput even more. By default, PowerExchange performs column-level processing on the change stream as a single thread. If you use multithreaded processing, PowerExchange might be able to extract changes faster and more efficiently by processing more than one UOW simultaneously.

PowerExchange multithreaded processing splits a UOW into multiple threads on the PowerCenter Integration Service machine. After the column-level processing completes, PowerExchange merges the threads and passes
the UOW to the PWXPC CDC reader for processing. Multithreaded processing works most efficiently when PowerExchange on the source machine is supplying data fast enough to take full advantage of the multiple threads on the PowerCenter Integration Service machine. If PowerExchange completely utilizes a single processor on the PowerCenter Integration Service machine, then multithreaded processing may provide increased throughput.
Extracting Change Data Overview

Use PowerExchange in conjunction with PWXPC and PowerCenter to extract captured change data and write the data to one or more targets. To extract change data that PowerExchange captures, you must import metadata for the CDC sources and the targets of the change data in Designer. After creating the source and target definitions in Designer, you must create a mapping and then an application connection, session, and workflow in Workflow Manager. 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 PowerExchange extraction maps.

Tip: Informatica recommends that you import the metadata from PowerExchange extraction maps instead of from database definitions. When you import extraction maps, the source definition contains all of the PowerExchange-generated CDC columns, such as the before image (BI) and change indicator (CI) columns. Additionally, PWXPC derives the extraction map name from the source definition so you do not need to code the extraction map name for each source in the session properties.

Before starting a CDC session, you should create restart tokens to define an extraction start point in the change stream. Restart tokens might also be required for resuming extraction processing in a recovery scenario.

To stop a CDC session using real-time extraction mode based on certain user-defined events, you can configure event table processing. Also, you can offload column-level extraction processing and any UOW Cleanser processing from the source system to the following remote locations:

- PowerCenter Integration Service machine
- A remote machine where the PowerExchange Logger for Linux, UNIX, and Windows runs

If you use offload processing with real-time extractions, you can also use multithreaded processing.
Task Flow for Extracting Change Data

Perform the following tasks in the PowerExchange Navigator, PowerCenter Designer, and PowerCenter Workflow Manager to configure and start extraction processing.

Before you begin, complete configuration of the data source and PowerExchange for CDC, 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 the 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, perform a database row test on the extraction map in 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 certain events, implement event table processing.

9. If you want to offload column-level extraction processing and UOW Cleanser processing from the source system to the PowerCenter Integration Service machine or PowerExchange Logger for Linux, UNIX, and Windows machine, configure offload processing. For real-time extractions, you can also configure multithreaded processing.

10. Start the CDC session.

**RELATED TOPICS:**
- “Configuring PowerCenter CDC Sessions” on page 66
- “Creating Restart Tokens for Extractions” on page 73
- “Displaying Restart Tokens” on page 74
- “Configuring the Restart Token File” on page 74
- “Testing Change Data Extraction” on page 64

Testing Change Data Extraction

Perform a database row test in the PowerExchange Navigator to ensure that PowerExchange can retrieve data when the extraction map is used in a CDC session.

A database row test verifies that:
- PowerExchange has captured change data for a data source defined in a capture registration.
- PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows has captured change data for a capture registration, if applicable.
- The extraction map properly maps the captured change data.
To test change data extraction:

1. In the Resource Explorer of the PowerExchange Navigator, open the extraction group that includes the extraction map that you want to test.
2. Open the extraction map.
3. Select the extraction map and click **File > Database Row Test**.
4. In the **Database Row Test** dialog box, enter or edit the following information:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB Type</td>
<td>An extraction mode indicator:</td>
</tr>
<tr>
<td></td>
<td>- <strong>CAPXRT</strong>. Real-time extraction mode or continuous extraction mode.</td>
</tr>
<tr>
<td></td>
<td>- <strong>CAPX</strong>. Batch extraction mode.</td>
</tr>
<tr>
<td>Location</td>
<td>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 file on the Windows machine from which you run the database row test.</td>
</tr>
<tr>
<td>UserID and Password</td>
<td>Optionally, a user ID and password that provides access to the source change data.</td>
</tr>
<tr>
<td>Application Name</td>
<td>At least one character to represent the application name. For a row test, a unique application name is not required. PowerExchange does not retain the value that you specify.</td>
</tr>
<tr>
<td>SQL Statement</td>
<td>A SQL SELECT statement that PowerExchange generates for the fields in the extraction map. You can edit this statement, if necessary.</td>
</tr>
</tbody>
</table>

   In the statement, a table is identified in the following format:

   \[ Schema.RegName_TableName \]

   Where:

   - **Schema** is schema 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 only extract change data after PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows has closed at least one condense or log file. Otherwise, PowerExchange displays no data in PowerExchange Navigator and writes the PWX-04520 message in the PowerExchange message log on the extraction system. PowerExchange also writes this message if no change data for the data source has been captured, condensed, or logged.

5. Click **Advanced**.

6. In the **CAPX Advanced Parameters** or **CAPXRT Advanced Parameters** dialog box, enter information, including the following:

   - If you use continuous extraction mode, enter the CAPX CAPI_CONNECTION name in the **CAPI Connection Name** field.
   - If you use the PowerExchange Logger for Linux, UNIX, and Windows to offload change data to system remote from the system on which it was captured, enter location of the extraction maps in the **Location** field.

7. Click **OK**.

8. Click **Go**.

The database row test returns each change from the extraction start point by column. The results include the PowerExchange-defined CDC columns, the DTL__ columns, which provide information such as the change type, change timestamp, and user ID of the user who made the change.
Configuring PowerCenter CDC Sessions

After you import metadata for CDC data sources and targets into PowerCenter, you can create a mapping and a CDC session to extract change data. Before running CDC sessions, you must configure numerous session and connection attributes.

Changing Default Values for Session and Connection Attributes

Certain PowerCenter session and application connection attributes have default values that are only appropriate for bulk data movement. You must change the values of these attributes for CDC sessions.

The following table summarizes these attributes and their recommended values:

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Location</th>
<th>Recommended Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit Type</td>
<td>Properties Tab</td>
<td>Source</td>
<td>Default is Target. The PowerCenter Integration Service automatically overrides it to Source. However, you cannot disable Commit On End Of File unless you change Commit Type to Source.</td>
</tr>
<tr>
<td>Commit On End Of File</td>
<td>Properties Tab</td>
<td>Disabled</td>
<td>Default is enabled. The PowerCenter Integration Service performs a commit when the session ends. This commit occurs after PWXPC commits the restart tokens, which can cause an out-of-sync condition between the restart tokens and the target data. As a result, duplicate data can occur when CDC sessions restart.</td>
</tr>
<tr>
<td>Recovery Strategy</td>
<td>Properties Tab</td>
<td>Resume from last checkpoint</td>
<td>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.</td>
</tr>
<tr>
<td>Stop on errors</td>
<td>Config Object Tab</td>
<td>1</td>
<td>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: - Key constraint violations - Loading nulls into a not null field - Database trigger responses If write errors occur, you might experience change data loss because PWXPC has advanced the restart tokens values. To maintain target data and restart token integrity, you must set this option to 1.</td>
</tr>
<tr>
<td>Application Name</td>
<td>Application Connection</td>
<td>Code a unique name for each CDC session.</td>
<td>Default is the first 20 characters of the WorkFlow Name. <strong>Warning:</strong> The default might not result in a unique name.</td>
</tr>
<tr>
<td>RestartToken File Folder</td>
<td>Application Connection</td>
<td>Default value</td>
<td>Use the default value of $PMRootDir/Restart, which PWXPC creates if it does not exist.</td>
</tr>
</tbody>
</table>
Configuring Application Connection Attributes

To extract change data, you must configure certain application connection attributes.

For a list of PWX CDC application connection attributes, see *PowerExchange Interfaces for PowerCenter*.

**Image Type**

For update operations, use the *Image Type* attribute to configure the format of the change data that a CDC session extracts.

Select one of the following options for the *Image Type* attribute:

- **AI**. After images only.
- **BA**. Before and after images.

Default is BA.

If you select **BA** for the *Image Type* attribute, PowerExchange provides the before-image (BI) and after-image (AI) data for the updated row as separate SQL operations:

- A DELETE with the before-image data
- An INSERT with the after-image data

**Note**: To select **BA** with batch or continuous extraction mode, you must configure PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows to log before and after images. Otherwise, you can only select after images.

If you select **AI** for the *Image Type* attribute, PowerExchange provides these changes as an UPDATE operation.

Because some relational databases do not allow updates to primary key columns, you cannot apply these changes.
as updates. If you configure BI columns for key columns, you can then use the Flexible Key Custom transformation to be change any UPDATE operations for key columns into a DELETE operation followed by an INSERT operation.

**Event Table Processing**

You can 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 of the changes for the day have been 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.

Event table processing has the following rules and guidelines:

- You can only use event table processing 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.

To implement event table processing:

1. Create an event table.
   
   The event table must be of the same source type and on the same machine as the change data that is extracted. For example, if you extract DB2 change data on MVS, 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 and extraction map for the event table.
   
   When you create a capture registration, the PowerExchange Navigator generates an extraction map.

3. In PowerCenter, create a CDC session, and specify the extraction map name in the **Event Table** attribute on the PWX CDC Real Time application connection.

4. When the defined event occurs, update the event table.
   
   When PowerExchange reads the update to the event table, PowerExchange places an end-of-file (EOF) into 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 currently in the pipeline to the targets and then ends the CDC session.

**CAPI Connection Name Override**

PowerExchange allows a maximum of eight CAPI_CONNECTION statements in the DBMOVER configuration file. You can use multiple CAPI_CONNECTION statements to extract changes from more than one data source type with a single PowerExchange Listener on a single machine. For example, you can extract changes for Oracle and DB2 for Linux, UNIX, and Windows through a single PowerExchange Listener by specifying multiple CAPI_CONNECTION statements in the dbmover.cfg file.

To specify the CAPI_CONNECTION statement that PowerExchange uses to extract change data in a CDC session, code the name in the **CAPI Connection Name Override** attribute.

You must code CAPI_CONNECTION statements on the system where the change data resides so that PowerExchange can extract change data for a data source type. If you use CDC offload processing, you must also code the CAPI_CONNECTION statements in the dbmover.cfg file on the PowerCenter Integration Service machine.
**Idle Time**

To indicate whether a real-time or continuous extraction mode CDC session should run continuously or shutdown after reaching the end-of-log (EOL), use the **Idle Time** attribute.

Enter one of the following values for the **Idle Time** attribute:

- **-1.** The CDC session runs continuously. PowerExchange returns end-of-file (EOF) only when the CDC session is manually stopped.
- **0.** After reaching EOL, PowerExchange returns EOF and the CDC session ends.
- **n.** After reaching EOL, PowerExchange waits for \( n \) seconds and, if no new change data of interest arrives, the CDC session ends. Otherwise, the CDC session continues until PowerExchange waits for \( n \) seconds without reading new change data of interest.

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 generally not static, and so the actual end-of-log is continually moving forward. After PowerExchange reaches EOL, it writes the PWX-09967 message in the PowerExchange message log.

Typically, real-time and continuous extraction mode CDC sessions use the default value of -1 for the **Idle Time** attribute. If necessary, you can manually stop a never-ending CDC session by using the PowerCenter Workflow Monitor, pmcmd commands, or the PowerExchange STOPTASK command.

Alternatively, you can set the **Idle Time** attribute to 0. After PowerExchange reaches 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 the 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 execute, 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 timing for the idle time begins.
2. If more data is in the change stream after 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** attribute is set to 0 and the CDC session ends.

If you set the **Idle Time** attribute to a low value, the CDC session might end before all available change data in the change stream has been read. If you want a CDC session to end periodically, Informatica recommends that you set the **Idle Time** attribute to 0 because active systems are rarely idle.

When a CDC session ends because either the idle time value has been reached or a PowerExchange STOPTASK command has been 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 never-ending CDC session with the PowerExchange STOPTASK command, PWXPC substitutes 86400 for the **idle_time** variable in the PWXPC_10072 message.
Note: If you specify values for the Reader Time Limit and Idle Time attributes, the PowerCenter Integration Service stops reading data from the source when the first one of these terminating conditions is reached. 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 Options

PWXPC uses the restart information to tell PowerExchange from which point to start reading the captured change data. To specify restart information, PWXPC provides options that you must configure for each CDC session.

The following table describes the restart attributes you must configure for CDC sessions:

<table>
<thead>
<tr>
<th>Connection Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Name</td>
<td>Application name for the CDC session. Specify a unique name for each CDC session. The application name is case sensitive and cannot exceed 20 characters. Default is the first 20 characters of the workflow name.</td>
</tr>
<tr>
<td>RestartToken File Folder</td>
<td>Directory name on the PowerCenter Integration Service machine that contains the restart token override file. Default is $PMRootDir/Restart.</td>
</tr>
<tr>
<td>RestartToken File Name</td>
<td>File name in the RestartToken File Folder that contains the restart token override file. PWXPC uses the contents of this file, if any, in conjunction with the state information to determine the restart point for the CDC session. Default is the Application Name, if specified, or the workflow name, if Application Name is not specified.</td>
</tr>
</tbody>
</table>

Informatica recommends that you specify a value for the Application Name attribute, because the default value might not result in a unique name. The values for Application Name and RestartToken File Name attributes must be unique for every CDC session. Non-unique values for either of these attributes can cause unpredictable results that include session failures and potential data loss.

PowerExchange Flush Latency

PowerExchange reads change data into a buffer on the source machine, or on the PowerCenter Integration Service machine if you use CDC offload processing. The PowerExchange Consumer API (CAPI) interface flushes the buffer that contains the data to PWXPC on the PowerCenter Integration Service machine for processing when one of the following conditions occurs:

- The buffer becomes full.
- The CAPI interface timeout, also called the PowerExchange flush latency, expires.
- A commit point occurs.

PowerExchange uses the flush latency value as the CAPI interface timeout value on the source machine, or on the PowerCenter Integration Service machine if you use CDC offload processing.

For CDC sessions that use real-time or continuous extraction mode, set the flush latency in the PWX Latency in seconds attribute of the PWX CDC Real Time application connection. For CDC sessions that use batch extraction mode, PowerExchange always uses two seconds for the flush latency.

Restriction: The value of PWX Latency in seconds impacts the speed with which a CDC session responds to a stop command from Workflow Monitor or pmcmd, because PWXPC must wait for PowerExchange to return control before it can handle the stop request. Informatica recommends that you use the default value of 2 seconds for the PWX Latency in seconds attribute.
PowerExchange writes the message PWX-09957 in the PowerExchange message log to reflect the CAPI interface timeout value set from the flush latency value. 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 to PWXPC, PWXPC provides the data to the appropriate sources in the CDC session for further processing and the PowerCenter Integration Service commits the data to the targets.

**Commitment Control Options**

PWXPC, in conjunction with PowerExchange and the PowerCenter Integration Service, controls the timing of commit processing for CDC sessions based on the values you code for the commitment control options.

To control commit processing, set one or more of the following connection attributes:

**Maximum Rows Per commit**

Maximum number of change records in a source UOW 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 the maximum rows limit is met. PWXPC does not wait for a UOW boundary to commit the change data. After the maximum rows limit is met, PWXPC issues a real-time flush to commit the change data and the restart tokens to the targets and writes the PWXPC_12128 message to the session log. PWXPC resets the maximum rows limit when a real-time flush occurs because either the maximum rows limit or UOW count is met or the real-time flush latency timer expires.

**Note:** The Maximum Rows Per commit attribute is a count of records within a UOW, unlike the UOW Count attribute that is a count of complete UOWs.

Default is 0, which means that PWXPC does not use maximum rows.

PWXPC uses the maximum rows limit to commit data before an end-UOW is received, a process also called sub-packet commit. If you specify either 0 or no value, commits occur only on UOW boundaries. Otherwise, PWXPC uses the value that you specify to commit change records between UOW boundaries.

**Warning:** 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.

The maximum rows limit is cumulative across all sources in the CDC session. PWXPC issues a real-time flush when the limit value is reached, regardless of the number of sources to which the changes were originally made.

Use a maximum rows limit when extremely large UOWs in the change stream might cause locking issues on the target database or resource issues on the node running the PowerCenter Integration Service. When you specify a low maximum rows limit, the session consumes more system resources on the PowerCenter Integration Service and target systems because PWXPC flushes data to the targets more frequently.

For example, a UOW contains 900 changes for one source followed by 100 changes for a second source and then 500 changes for the first source. If you set the maximum rows value to 1000, PWXPC issues the commit after reading 1,000 change records. In this example, the commit occurs after PWXPC processes the 100 changes for the second source.

**Minimum Rows Per commit**

For real-time or continuous extraction mode, minimum number of change records that PowerExchange reads from the change stream before it passes a commit record to PWXPC. 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 to PWXPC and then resets the minimum rows counter.

Default is 0, which means that PowerExchange does not use minimum rows.
If you specify a minimum rows limit, PowerExchange changes the number of change records in a UOW to match or exceed the limit. PWXPC does not commit change data to the targets when the minimum rows limit occurs. PWXPC only commits change data to the targets based on the values of the **Maximum Rows Per commit**, **Real-Time Flush Latency in milli-seconds**, and **UOW Count** attributes.

A minimum rows limit does not impact the relational integrity of the change data because PowerExchange does not create new commits points in the change stream data. It merely skips some of the original commit records in the change stream.

If your change data 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 making only a few changes, which results in many, small UOWs in the change stream. PowerExchange and PWXPC process fewer, larger 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.

**Real-Time Flush Latency in milli-seconds**

For real-time or continuous extraction mode, number of milliseconds that must pass before PWXPC flushes the data buffer to commit the 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 the change data and the restart tokens to the targets and writes the PWXPC_10082 message in the session log. PWXPC resets the flush latency interval when a real-time flush occurs because either the interval expires, or one of the UOW count or maximum row limit is met.

Enter one of the following values for the flush latency interval:

- `-1`. Disables data flushes based on time.
- `0 to 2000`. Interval set to 2000 milliseconds, or 2 seconds.
- `2000 to 86400`. Interval set to the specified value.

Default is 0, which means that PWXPC uses 2,000 milliseconds.

If you set the flush latency interval value is 0 or higher, 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 value, the faster you commit change data to the targets. Therefore, if you require the lowest possible latency for the apply of changes to the targets, specify a low value for the flush latency interval.

When you specify low flush latency intervals, the CDC session might consume more system resources on the PowerCenter Integration Service and target systems because PWXPC commits to the targets more frequently. When you choose the flush latency interval value, you must balance performance and resource consumption with latency requirements.

**UOW Count**

Number of complete UOWs that PWXPC reads from the change stream before flushing the change data to the targets. As PWXPC reads change data from PowerExchange and provides that data to the appropriate source in the CDC session, it counts the number of UOWs. After the UOW count value is reached, PWXPC issues a real-time flush to commit the change data and the restart tokens to the targets, and writes the PWXPC_10081 message in the session log. PWXPC resets the UOW count when a real-time flush occurs because the UOW count or maximum rows limit is met, or the flush latency interval expires.

Enter one of the following for the UOW count value:

- `-1 or 0`. PWXPC does not use the **UOW Count** attribute to control commit processing.
- `1 to 999999999`. PWXPC flushes change data after reading the number of UOWs specified by **UOW Count** attribute.

Default is 1.
The lower you set the value for the **UOW Count** attribute, the faster that PWXPC flushes change data to the targets. To achieve the lowest possible latency for applying change data to targets, set the **UOW Count** attribute to 1. However, the lowest possible latency for applying change data also results in the highest possible resource consumption on the PowerCenter Integration Service and the target systems.

Commit processing for CDC sessions is not controlled by a single commitment control attribute. The **Maximum Rows Per commit**, **Real-Time Flush Latency in milli-seconds**, and **UOW Count** values all result in a real-time flush of change data, which causes the data and restart tokens to be committed to the targets. When you choose values for the **UOW Count**, **Real-Time Flush Latency in milli-seconds**, and **Maximum Rows Per commit** attributes, balance performance and resource consumption with latency requirements.

**Warning:** You must ensure that the session properties **Commit Type** attribute specifies Source and that the **Commit at End of File** attribute is disabled. By default, the **Commit at End of File** attribute is enabled, which causes the PowerCenter Integration Service to write additional data to the targets after the CDC reader has committed the restart tokens and shut down. As a result, when you restart the CDC session, duplicate data might be written to the targets.

**RELATED TOPICS:**
- "Commit Processing with PWXPC" on page 56

---

### Creating Restart Tokens for Extractions

Before you extract change data, you must establish an extraction start point. An optimal extraction start point matches a time in the change stream that occurs after the target has been synchronized with the source but before any new changes occur for the source. Usually, this point is the end of the change stream because changes to the source are inhibited until the target is materialized and restart tokens are generated.

You can generate current restart tokens for the end of the change stream by using one of the following methods:

- **PWXPC restart token file.** Generate current restart tokens for CDC sessions that use real-time or continuous extraction mode by coding the `CURRENT_RESTART` option on the RESTART1 and RESTART2 special override statements in the PWXPC restart token file. When the session executes, PWXPC requests that PowerExchange provide restart tokens for the current end of the change stream, which PWXPC then uses as the extraction start point.

- **Database Row Test.** Generate current restart tokens for sources by performing a database row test in PowerExchange Navigator and coding a `SELECT CURRENT_RESTART` SQL statement.

- **DTLUAPPL utility.** Generate current restart tokens for sources by using the `GENERATE RSTKKN` option in the DTLUAPPL utility.

If you use a PowerExchange utility or the PowerExchange Navigator to generate restart tokens, edit the restart token file that PWXPC uses to specify the token values before you start the CDC session.
Displaying Restart Tokens

In the PowerExchange Navigator, you can perform a database row test on an extraction map to display the restart token pair for each row of change data. The database row test output includes the following columns for the token values:

- DTL__CAPXRESTART1 column for the sequence token
- DTL__CAPXRESTART2 column for the restart token

If you include the DTL__CAPXRESTART1 and DTL__CAPXRESTART2 columns in your 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 the messages PWX-04565 and PWX-09959, the sequence token is in the Sequence field and restart token is in the PowerExchange Logger field.
- In the 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 the messages PWXPC_10081, PWXPC_10082, and PWXPC_12128, the sequence token is the first token value and is followed by the restart token.

When you use the DTLUAPPL utility to generate restart tokens, 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 in the following attributes of the source PWX CDC application connection:

- **RestartToken File Folder.** Specify the directory that contains the restart token file. If the folder does not exist and the attribute contains the default value of $PMRootDir/Restart, PWXPC creates it. PWXPC does not create any other restart token folder name.
- **RestartToken File Name.** Specify the unique name of the restart token file. If you do not specify a value in this attribute, PWXPC uses the value of the Application Name, if available. Otherwise, PWXPC uses the name of the workflow. Because this name must be unique, Informatica recommends that you always code 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 the RestartToken File Name attribute to create an empty restart token file.

**Restriction:** The value of RestartToken File Name attribute in must be unique for every CDC session. Non-unique file names can cause unpredictable results, such as change data loss and session failures.

To locate the restart token file name for a CDC session, check the following places:

- For existing CDC sessions, message PWXPC_12057 in the session log contains the restart token file folder and the restart token file name.
- In Workflow Manager, the PWX CDC application connection associated with the source in the CDC session contains the restart token file name and folder location. If the restart token file name is not specified in the application connection, PWXPC uses the application name, if specified. Otherwise, PWXPC uses the workflow name.
Before you run a CDC session for the first time, configure the restart token file to specify the point in the change stream from which PowerExchange begins to extract change data. You can also configure the restart token file to add new sources to a CDC session or to restart change data extraction from a specific point in the change stream.

**Restart Token File Statements**

You can use the following types of statements in a the restart token file:

- **Comment**
- **Explicit override**. Specify a restart token pair for a specific source. You must provide the PowerExchange extraction map name.
- **Special override**. Specify a restart token pair for one or more sources. You can provide a specific restart token pair or request that PowerExchange use the current restart point.

**Restart Token File Statement Syntax**

For the comment statements, use the following syntax:

```
<!-- comment_text
```

For explicit override statements, use the following syntax:

```
extraction_map_name-sequence_token
extraction_map_name-restart_Token
```

For special override statements, use the following syntax:

```
RESTART1=(sequence_token|CURRENT_RESTART)
RESTART2=(restart_token|CURRENT_RESTART)
```

The following rules and guidelines apply:

- Statements can begin in any column.
- All statements are optional.
- Do not include blank lines between statements.
- Comment lines must begin with:

  `<!--`

- Per file, you can specify one or more explicit override statements and one special override statement.
- An explicit override statement for a source takes precedence over any special override statement.

**Comment Statements**

You can use the comment statement anywhere in the restart token file.

Comment statements must begin with:

```
<!--
```

**Explicit Override Statements**

Use the explicit override statement to specify the restart token pair for a specific source. Each source specification consists of a pair of restart tokens containing the source extraction map name with the restart token values. Define the source by specifying the extraction map name. A source can have multiple extraction maps and, therefore, multiple extraction map names.

You can code explicit override statements for one or more sources in a CDC session. Alternatively, you can use explicit override statements in conjunction with the special override statement to provide restart tokens for all sources in a CDC session.
When you warm start a CDC session, an explicit override statement for a source overrides the restart tokens stored in the state table or file for that source.

The explicit override statement has the following parameters:

```
extraction_map_name=restart1_token and extraction_map_name=restart2_token
```

The PowerExchange extraction map name and the sequence and restart tokens for the source.

- **extraction_map_name**
  - The extraction map name for the data source. To determine the extraction map name, check one of the following:
    - For CDC data map sources, the Schema Name Override and Map Name Override attributes in the session properties. These attributes override the schema and map names of the source extraction map.
    - For CDC data map sources, the Schema Name and Map Name values in the source Metadata Extensions in Designer.
    - For relational sources, the Extraction Map Name attribute in the session properties.

- **restart1_token**
  - The sequence token part of the restart token pair, which varies based on data source type.

- **restart2_token**
  - The restart token part of the restart token pair, which varies based on data source type.

**Special Override Statement**

Use the special override statement to specify or generate restart tokens for one or more sources. You must specify both the RESTART1 and RESTART2 parameters.

You can use the special override statement to provide restart tokens for all sources in a CDC session. Alternatively, you can use explicit override statements in conjunction with the special override statement to provide or override restart tokens for all sources in a CDC session.

When you warm start a CDC session, the special override statement overrides the restart tokens stored in the state table or file for all sources, except those sources specified in explicit override statements.

The special override statement has the following parameters:

```
RESTART1={restart1_token|CURRENT_RESTART} and RESTART2={restart2_token|CURRENT_RESTART}
```

The sequence token and restart token in the restart token pair or the current end of the change stream.

- **restart1_token**
  - The sequence token part of the restart token pair, which varies based on data source type.

- **restart2_token**
  - The restart token part of the restart token pair, which varies based on data source type.

- **CURRENT_RESTART**
  - PowerExchange generates current restart tokens. The PWXPC CDC reader opens a separate connection to PowerExchange to request generation of current restart tokens, and then provides the generated restart tokens to all applicable sources.
Restriction: You can only use CURRENT_RESTART for CDC sessions that use real-time and continuous extraction mode. You cannot use this option for CDC sessions that use batch extraction mode.

You can also generate current restart tokens in the Database Row Test dialog box in the PowerExchange Navigator.

Restart Token File - Example

In the example, a CDC session contains seven source tables. This restart token file specifies explicit override statements to provide the restart tokens for three sources and the special override statement to provide the restart tokens for the remainder of the source.

The restart token file contains the following statements:

```plaintext
<!-- Restart Tokens for existing tables -->
restart1=000000AD7756000000000000D7756000000000000
restart2=C1E4E2D3404000000D5F2C00000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_001 -->
did9.rrtb0001_RRTB_SRC_001=00000000D1DB0000000000D1DB2000000000000
did9.rrtb0001_RRTB_SRC_001=C1E4E2D3404000000D5F2C00000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_002 -->
did9.rrtb0002_RRTB_SRC_002=00000000A3719500000000000A3719500000000000
did9.rrtb0002_RRTB_SRC_002=C1E4E2D3404000000968FC60000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_004 -->
did9.rrtb0004_RRTB_SRC_004=00000000D84E780000000000000D84E7800000000000000
did9.rrtb0004_RRTB_SRC_004=C1E4E2D3404000000D1E6100000000
```

When you warm start the CDC session, PWXPC reads the restart token file to process any override statements for restart tokens. In this case, the restart token file overrides all restart tokens for all sources in the CDC session. After resolving the restart tokens for all sources, PWXPC writes message PWXPC_12060 to the session log with the following information:

```
Session restart information:
---------------------------------------------
<table>
<thead>
<tr>
<th>Restart Token 1</th>
<th>Restart Token 2</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000D1DB0000000000D1DB2000000000000</td>
<td>C1E4E2D3404000000D5F2C00000000</td>
<td>Restart file</td>
</tr>
<tr>
<td>00000000A3719500000000000A3719500000000000</td>
<td>C1E4E2D3404000000968FC60000000</td>
<td>Restart file</td>
</tr>
<tr>
<td>00000000D84E780000000000000D84E7800000000000000</td>
<td>C1E4E2D3404000000D1E6100000000</td>
<td>Restart file [special override]</td>
</tr>
<tr>
<td>00000000A3719500000000000A3719500000000000</td>
<td>C1E4E2D3404000000D1E6100000000</td>
<td>Restart file [special override]</td>
</tr>
<tr>
<td>00000000A3719500000000000A3719500000000000</td>
<td>C1E4E2D3404000000D1E6100000000</td>
<td>Restart file [special override]</td>
</tr>
<tr>
<td>00000000A3719500000000000A3719500000000000</td>
<td>C1E4E2D3404000000D1E6100000000</td>
<td>Restart file [special override]</td>
</tr>
<tr>
<td>00000000A3719500000000000A3719500000000000</td>
<td>C1E4E2D3404000000D1E6100000000</td>
<td>Restart file [special override]</td>
</tr>
</tbody>
</table>
```

PWXPC indicates the source of the restart token values for each source. For the sources that had explicit override statements in the restart token file, PWXPC writes “Restart file” in the Source column.

For the sources to which PWXPC assigns the special override restart tokens, PWXPC writes “Restart file (special override)” in the Source column.
Starting PowerCenter CDC Sessions

Use Workflow Manager, Workflow Monitor, or pmcmd to start a workflow or task for a CDC session. You can start the entire workflow, part of a workflow, or a task in the workflow. You can do a cold start, warm start, or recovery start. The method you use determines how PWXPC acquires the restart information.

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 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. To begin extracting change data again, either cold start or warm start the session.

**Cold Start Processing**

Cold start workflows and tasks by using the Cold Start command in Workflow Manager or Workflow Monitor. You can also 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 file and then writes the 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 commits the data and restart tokens to the targets. This processing continues until the session ends or is stopped.

**Warm Start Processing**

Warm start workflows and tasks by using the Start or Restart command in Workflow Manager or Workflow Monitor. You can also 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 flushes the data to those 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 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 is stopped.

Recovery Processing

Recover workflows and tasks by selecting the Recover command in Workflow Manager or Workflow Monitor. You can also use the pmcmd recoverworkflow command, or the starttask or startworkflow command with the recovery option.

You can use recovery to populate the restart token file with the restart tokens for all sources in a CDC session so that you can then cold start the CDC session or to ensure 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 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 flushes the data to those 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. You can also use pmcmd stoptask, stopworkflow, aborttask, or abortworkflow commands. In PowerExchange, issue the STOPTASK command or run the DTLUTSK utility.

Use one of the following methods to stop a running CDC session:

Stop

Use the Stop command in Workflow Monitor or the pmcmd stoptask or stopworkflow commands. After the PWXPC CDC reader and PowerCenter Integration Service process all of the data in the pipeline and shut down, the session ends.
STOPTASK

Use the PowerExchange STOPTASK command. You can run the STOPTASK command on the source system that is extracting the change data, from the PowerExchange Navigator, or by using pwxcmd or the DTLUTSK utility. When you issue the STOPTASK command, PowerExchange stops the extraction task in the PowerExchange Listener and passes an EOF to the PowerCenter Integration Service, which ends the session.

Abort

Use the Abort command in Workflow Monitor or the pmcmd aborttask or abortworkflow commands. When you abort a CDC session, the PowerCenter Integration Service waits 60 seconds to allow the readers and the writers time to process all of the data in the pipeline and shut down. If the PowerCenter Integration Service cannot finish processing and committing data within this timeout period, it kills the DTM process and ends the session.

Stop Command Processing

Stop CDC sessions and workflows by using the Stop command in Workflow Monitor or the pmcmd stopptask or stopworkflow command. You can also use the PowerExchange STOPTASK command.

After you issue a stop command in PowerCenter or PowerExchange, the following processing occurs:

1. If you use a PowerCenter stop command, the PowerCenter Integration Service requests PWXPC to stop. If you use a PowerExchange stop command, PowerExchange sends an EOF to PWXPC.
2. When PWXPC receives an EOF, it flushes any complete and uncommitted UOWs with the associated restart tokens to the targets. PWXPC then writes the messages PWXPC_12101 and PWXPC_12068 to the session log.
3. The PowerCenter Integration Service processes all of the 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

To stop a CDC session based on a user-defined event or at EOL, configure a termination condition in the session. A terminating condition determines when the PWXPC stops reading change data from the sources and ends the CDC session. After PWXPC reaches a terminating condition, it flushes the 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 session.

You can configure the following termination conditions for CDC sessions:

- **Event table processing.** If you specify an extraction map table in the Event Table attribute of the PWX CDC Real Time application connection, PowerExchange, after it reads a change record for the event table, passes EOF to PWXPC to end the CDC session.
- **Idle Time.** If you specify 0 for the Idle Time attribute on a PWX CDC Real Time application connection, PowerExchange, after it reaches EOL, passes EOF to PWXPC to end the CDC session.
- **Batch extraction mode.** If you use batch extraction mode by configuring a PWX CDC Change application connection, PowerExchange, after it reads all closed PowerExchange Condense condense files or PowerExchange Logger for Linux, UNIX, and Windows log files, passes PWXPC EOF to end the CDC session.
Changing PowerCenter CDC Sessions

You can add new sources and targets to an existing CDC sessions. Afterward, you must cold start the session. Because a cold start is required, you must also get the latest restart tokens for the original sources prior to restarting the session. To do so, you can perform a recovery.

To change a PowerCenter 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 new 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 Creating a Restart Point

The following examples show different methods of creating a restart point for a source table that is added to an existing CDC session. 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.

Adding a New Source and Use CURRENT_RESTART to Create Restart Tokens - Example

In this example, a new source table, RRTB_SRC_004, is added to an existing CDC session that contains three sources. The restart points for the existing sources are maintained. For the new source, the example uses the CURRENT_RESTART option in the restart token file to generate a restart token that represents the current end of the change stream.

To add a new source and use CURRENT_RESTART to create restart tokens:

1. To stop the workflow, select the Stop command in Workflow Monitor.
2. After the workflow stops, select the Recover Task command in Workflow Monitor to run a recovery session.

   PWXPC writes the following messages in the session log:

   ```
   PWXPC_12060 [INFO] [CDCRestart]
   +----------------------------------+
   | Session restart information:      |
   +----------------------------------+
   ```
In this example, a new source table, RRTB_SRC_004, is added to an existing CDC session containing three sources.

3. Edit the mapping, session, and workflow to add the new source, RRTB_SRC_004.

4. Edit the restart token file to specify the CURRENT_RESTART option for the new source.

The updated file appears as follows:

```plaintext
<!>-- existing sources

dld9.rtrb0001_RRTB_SRC_001=00000000d22df000000000000000000000000000

dld9.rtrb0001_RRTB_SRC_001=C1E4E2D340400000000D9C0000000

dld9.rtrb0002_RRTB_SRC_002-00000000d22df000000000000000000000000000

dld9.rtrb0002_RRTB_SRC_002=C1E4E2D340400000000D9C0000000

dld9.rtrb0003_RRTB_SRC_003=00000000d22df000000000000000000000000000

dld9.rtrb0003_RRTB_SRC_003=C1E4E2D340400000000D9C0000000

<!>-- new source

RESTART1=RESTART
RESTART2=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 new source, RRTB_SRC_004. PWXPC then passes the restart tokens to PowerExchange to begin change data extraction. Because the restart points for the other sources are earlier than the one just generated for RRTB_SRC_004, PWXPC does not pass any change data to this new source until the first change following its generated restart point is read.

Adding a New Source and Use DTLUAPPL to Create Restart Tokens - Example

In this example, a new source table, RRTB_SRC_004, is added to an existing CDC session containing three sources. The restart points for the existing sources are maintained. The DTLUAPPL utility is used to generate a restart token that represent the current end of the change stream.

To add a new source and use DTLUAPPL to create restart tokens:

1. To stop the workflow, select the Stop command in Workflow Monitor.

2. After the workflow stops, select the Recover Task command from Workflow Monitor to run a recovery session.

PWXPC writes the following messages in the session log:

```plaintext
PWXPC_12040 [INFO] [CDCRestart]

========================================================================
Session restart information:
========================================================================

<table>
<thead>
<tr>
<th>Extraction Map Name</th>
<th>Restart Token 1</th>
<th>Restart Token 2</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>dld9.rtrb0002_RRTB_SRC_002</td>
<td>00000000D22DF00000000000000000000000</td>
<td>C1E4E2D340400000000D9C0000000</td>
<td>GD storage</td>
</tr>
<tr>
<td>dld9.rtrb0001_RRTB_SRC_001</td>
<td>00000000D22DF00000000000000000000000</td>
<td>C1E4E2D340400000000D9C0000000</td>
<td>GD storage</td>
</tr>
<tr>
<td>dld9.rtrb0001_RRTB_SRC_001</td>
<td>00000000D22DF00000000000000000000000</td>
<td>C1E4E2D340400000000D9C0000000</td>
<td>GD storage</td>
</tr>
</tbody>
</table>

```

PWXPC also writes the restart tokens in the restart token file specified in the CDC application connection.

3. Edit the mapping, session, and workflow to add the new source, RRTB_SRC_004.

4. Run DTLUAPPL with RSTTKN GENERATE to generate restart tokens for the current end of the change stream. Use the following DTLUAPPL control cards:

```plaintext
mod APPL dummy DSN7 rsttkn generate
mod rtttkn rttb004
end appl dummy
print appl dummy
```

The PRINT command produces the following output:

```plaintext
Registration name=rtrt004.1> tag=<DB2DSN7:rttb0041>
Sequence=<000000DBF240A0000000000DBF240A000000000>
ReStart <C1E4E2D340400000000DBF2382000000000>
```

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 new source and its tokens.
The updated file contains the following lines:

```verbatim
<!-- existing sources
  dldsn9.rttbo001_RRTB_SRC_001=000000AD220F00000000000000AD220F000000000000000
  dldsn9.rttbo001_RRTB_SRC_001=C1E4E2D340400000000000D9C00000000
  dldsn9.rttbo002_RRTB_SRC_002=000000AD220F00000000000AD220F0000000000000
  dldsn9.rttbo002_RRTB_SRC_002=C1E4E2D34040000000000D9C00000000
  dldsn9.rttbo003_RRTB_SRC_003=000000AD220F00000000000AD220F0000000000000
  dldsn9.rttbo003_RRTB_SRC_003=C1E4E2D34040000000000D9C000000000
<!-- new source
  dldsn9.rttbo004_RRTB_SRC_004=000000DBF240A0000000000000DBF240A0000000000000
  dldsn9.rttbo004_RRTB_SRC_004=C1E4E2D34040000000000D9C000000000
```

6. **Cold start the session.**

PWXPC passes these restart tokens to PowerExchange to begin change data extraction. Because the restart points for the other sources are earlier than the one just generated for RRTB_SRC_004, PWXPC does not pass any change data to this new source until the first change following the generated restart point is read.

---

Recovering PowerCenter CDC Sessions

Use Workflow Manager, Workflow Monitor, or pmcmd to recover a workflow or task for a CDC session that fails. You can recover the entire workflow or a task in the workflow.

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, you should 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

In this example, a CDC session with relational targets is aborted in the Workflow Monitor. Then, the Restart Task command is issued from the Workflow Monitor to restart the CDC session.

When you warm start the session, PWXPC automatically performs a recovery, 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 or file and writes the message PWXPC_12060 in the session log. The PWXPC_12060 message records the restart tokens for the session and its sources, as shown in the following example:

PWXPC_12060 [INFO] [CDCRestart]

+----------------+----------------+----------------+----------------+----------------+
<table>
<thead>
<tr>
<th>Extraction Map</th>
<th>Restart Token 1</th>
<th>Restart Token 2</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>dian8.rtt00010</td>
<td>RRTB_SRC_004</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
<tr>
<td>dian8.rtt00020</td>
<td>RRTB_SRC_002</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
<tr>
<td>dian8.rtt00030</td>
<td>RRTB_SRC_003</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
<tr>
<td>dian8.rtt00090</td>
<td>RRTB_SRC_009</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
<tr>
<td>dian8.rtt00110</td>
<td>RRTB_SRC_001</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
<tr>
<td>dian8.rtt00050</td>
<td>RRTB_SRC_005</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
<tr>
<td>dian8.rtt00060</td>
<td>RRTB_SRC_006</td>
<td>000000FCA65840000000000000028D04A00000000FFFFFFFFFF</td>
<td>GND storage</td>
</tr>
</tbody>
</table>

If PWXPC detects that recovery is required, PWXPC writes the message PWXPC_12069 in the session log. This message usually includes the restart tokens for both the begin-UOW and the 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 file. However, if you specify a maximum rows threshold, 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 include “from” restart tokens that are the same as those displayed in the example PWXPC_12060 message:

PWXPC_12069 [INFO] [CDCRestart] Running in recovery node. Reader will resend the the oldest uncommitted UOW to resync targets:

from: Restart 1 [000000FCA658400000000000028D04A00000000FFFFFFFFFF] : Restart 2 [C1E4E4D04000000001B1A5000000000]

to:   Restart 1 [000000FCA658400000000000028D04A00000000FFFFFFFFFF] : Restart 2 [C1E4E4D04000000001B1A5000000000].

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 are 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 points defined by the two restart token values in the PWXPC_12069 message and then issues a commit for the data and the restart tokens. The PowerCenter Integration Service writes the flushed change data to the target tables and writes the restart tokens to the state table. Then the session ends.
Monitoring and Tuning Options

This chapter includes the following topics:

- Monitoring Change Data Extractions, 86
- Tuning Change Data Extractions, 92
- CDC Offload and Multithreaded Processing, 97

Monitoring Change Data Extractions

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 the PowerCenter Workflow Monitor.

Monitoring CDC Sessions in PowerExchange

In PowerExchange, you can use the following information to monitor the extraction of change data by CDC sessions:

- **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 command output.** Use the DISPLAY command with the ACTIVE option or the pwxcmd listtask command to display active CDC sessions.

Read Progress Messages

You can request that PowerExchange write messages that indicate read progress to the PowerExchange log file. If you select the Retrieve PWX log entries option on a PWX CDC application connection, PWXPC writes the progress messages in the session log.
To direct PowerExchange to write read progress messages, include the following parameters in the DBMOVER configuration file:

- **PRGIND.** Specify Y to have PowerExchange write PWX-04587 messages that indicate the number of records read for a CDC session. Default is N.
- **PRGINT.** Specify the number of records that PowerExchange reads before writing the PWX-04587 messages to the PowerExchange log file. Default is 250 records.

The PWX-04587 messages have the following format:

```
PWX-04587 int_server/workflow_name/session_name: Records read=num_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.
- **num_records** is the cumulative number of records read since the CDC session started.

For example, to direct PowerExchange to write read progress messages after 100 records, the DBMOVER configuration file contains the following parameters:

```
PRGIND=Y
PRGINT=100
```

When a CDC session that has a session name of \texttt{s\_cdc\_DB2\_SQL\_stats} runs, PowerExchange writes the following messages to the PowerExchange log file:

```
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 log file, each of these messages has a date and timestamp. You can 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 the following messages that contain statistical information about the session:

- **PWX-04578.** PowerExchange writes this message for each source in the CDC session. This 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 that CDC 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 the target, review the session log.

**Multithreaded Processing Statistics**

If you use CDC offload processing, you can also use multithreaded processing to attempt to increase throughput on the PowerCenter Integration Service machine where the offloaded processing runs.

To monitor the effectiveness of multithreaded processing, specify the following parameter in the DBMOVER configuration file on the PowerCenter Integration Service machine:

```
SHOW_THREAD_PERF=number_records
```

Number of change records that PowerExchange reads during a statistics reporting interval before writing the statistics messages PWX-31524 through PWX-31259 to the PowerExchange log file. If you select the
Retrieve PWX log entries option on the connection in the CDC session, PWXPC writes these messages in the session log.

You can use the information in the messages to tune multithreaded processing. For PowerExchange to write statistics messages for threads, you must specify 1 or above for Worker Threads on the connection. Otherwise, PowerExchange does not use multithreaded processing or produce statistics messages.

Valid values are from 10000 through 5000000.

The messages that PowerExchange writes during each statistics interval contain the following information:

- **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.

If the parsing and external processing times are higher than the I/O time, you might improve throughput by increasing the number of threads for the CDC session.

For the following example, SHOW_THREAD_PERF=10000 is specified in the DBMOVER configuration file. PowerExchange writes the following sample messages after 10,000 change records have been read and the next UOW boundary is reached:

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 I/O 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 I/O 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)

**DISPLAY ACTIVE Command Output**

Use the PowerExchange Listener DISPLAY ACTIVE command to display the CDC sessions that are active in the PowerExchange Listener.

Use the SNDLSTCMD command to issue the DISPLAY command with the ACTIVE option from the command line. Alternatively, issue the pwxcmd listtask command from a Linux, UNIX, or Windows system to a PowerExchange Listener running on an i5/OS system.
The command output includes the **PwrCntrSess** field. This field provides the PowerCenter session name in the following format:

integration_server_name/workflow_name/session_name

For example, if two active CDC sessions are active, the command produces the following output:

```
PWX-00711 Active tasks:
PWX-00712 jobname=jobid, Partner=10.10.10.01, Port=2480, PwrCntrSess=intserv1/workflow1/cdc_sess1, Application=appl_name1, Status=Active, AM=CAPXRT, Mode=Read, Process*, SessId=
PWX-00712 jobname=jobid, Partner=10.10.10.02, Port=2480, PwrCntrSess=intserv2/workflow2/cdc_sess2, Application=appl_name2, Status=Active, AM=CAPXRT, Mode=Read, Process*, SessId=
PWX-00713 2 active tasks
PWX-00709 0 Dormant TCRs
```

**Monitoring CDC Sessions in PowerCenter**

In PowerCenter, you can use the following information to monitor the progress of CDC sessions:

- **Session log messages.** PWXPC and PowerCenter write messages to the session log. You can use these messages to monitor the progress of a CDC session.
- **Performance details in Workflow Monitor.** If you configure a CDC session to report performance details, you can monitor the progress of the session in the Workflow Monitor.

For information about tuning CDC sessions and tuning buffer memory, 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 to commit the data to the targets, it writes one of the following messages to the session log, displaying 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 the 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 messages displays the source-based commit statistics.

**RELATED TOPICS:**

- "Using Connection Options to Tune CDC Sessions " on page 95
- "Tuning Commit Processing " on page 97
- "Viewing Performance Details in Workflow Monitor” on page 89

**Viewing Performance Details in Workflow Monitor**

Performance details include counters that you can use to assess the efficiency of a CDC session and change data extraction processing. The details include a single source qualifier that reflects group source processing for the change data.

From Workflow Monitor, you can view the details for the current CDC session while it is executing. If you notice degradation of CDC session performance, you can use the performance details to determine the bottleneck.
PWXPC does not store performance details in the repository so you cannot view previous performance details for CDC sessions.

**Note:** To view performance details for a CDC session that has ended, you must select performance details while the session is running. Otherwise, PWXPC does not display performance details.

To enable the collection of performance details, select **Collect performance data** on the Properties tab of the CDC session. During the execution of the CDC session, PWXPC refreshes the statistical information every ten seconds. If you have selected a resume recovery strategy in the CDC session, PWXPC displays data for all performance counter fields.

To view performance details in Workflow Monitor:

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 data source qualifier from the CDC session. The **Counter Value** column displays the PowerCenter node name.

3. To view performance details, select the data source qualifier. The following table describes the fields that PowerCenter displays in the **Performance Counter** column in the **Performance** area:

<table>
<thead>
<tr>
<th>Performance Counter Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 PowerExchange CDC Reader Status:</td>
<td>Current status of the PWXPC reader, as indicated by one of the following values:</td>
</tr>
<tr>
<td>1.1 Time Last Data Row Read</td>
<td>Time, in milliseconds, when PWXPC last received data from PowerExchange.</td>
</tr>
<tr>
<td>1.2 Data Rows In Current Interval</td>
<td>Number of change records received from PowerExchange during the current statistics interval.</td>
</tr>
<tr>
<td>1.3 End Packets In Current Interval</td>
<td>Number of UOWs received from PowerExchange during the current statistics interval.</td>
</tr>
<tr>
<td>1.4 Data Read Rate In Current Interval (rows/sec)</td>
<td>Number of change records read per second by PowerExchange during the current statistics interval. The value varies, depending on the quantity of change data being processed:</td>
</tr>
<tr>
<td>1.5 Mean Data Read Rate (rows/sec)</td>
<td>Mean number of change records that PowerExchange read per second, from the start of the CDC session.</td>
</tr>
</tbody>
</table>

The following factors can increase this value:
- Large network bandwidth
- CDC offload processing
- Multithreaded processing
<table>
<thead>
<tr>
<th>Performance Counter Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6 Max Data Read Rate (rows/sec)</td>
<td>Maximum number of change records that PowerExchange read per second during a statistics interval, from the start of the CDC session.</td>
</tr>
</tbody>
</table>
| 2 PowerCenter Processing Status: | Overall status of the CDC session, as indicated by one of the following values:  
- Idle. Waiting for change data.  
- Processing Data. Data is being processed.  
- Recovery Disabled. If a resume recovery strategy is not selected, the PWXPC CDC reader cannot obtain PowerCenter status information. |
| 2.1 Time Of Last Commit | Timestamp of the last commit to a target. |
| 2.2 Rows Processed To Commit In Current Interval | Number of change records flushed by the PWXPC reader 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 rate includes reading the UOW from PowerExchange and committing the change data to the targets.  
The following factors can influence this rate:  
- 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 in that it takes into account only the time when the session is actively processing data and 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, recorded 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 were issued by the PWXPC reader 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. |
### Performance Counter Field

<table>
<thead>
<tr>
<th>Field Description</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2 Timestamp On Last Target Commit</td>
<td>The capture timestamp, DTL__CAPXTIMESTAMP, from the last UOW committed to the target.</td>
</tr>
<tr>
<td>4 Totals</td>
<td></td>
</tr>
<tr>
<td>4.1 Elapsed Time</td>
<td>Total elapsed time for the CDC session.</td>
</tr>
<tr>
<td>4.2 Rows Read</td>
<td>Total number of change records read from PowerExchange.</td>
</tr>
<tr>
<td>4.3 End Packets Read</td>
<td>Total number of UOWs read.</td>
</tr>
<tr>
<td>4.4 Time in PowerExchange Processing</td>
<td>Total time of PowerExchange processing for the CDC session.</td>
</tr>
<tr>
<td>4.5 Rows Processed</td>
<td>Total number of change records processed through PowerCenter and committed to the targets.</td>
</tr>
<tr>
<td>4.6 Commits to Target</td>
<td>Total number of flushes that the PWXPC reader issued and that were committed to the targets.</td>
</tr>
<tr>
<td>4.7 TS on Last Commit minus TS at Commit (2.1-3.2)</td>
<td>Value that results 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.</td>
</tr>
</tbody>
</table>

### Tuning Change Data Extractions

You can use PowerExchange configuration parameters and connection options in PowerCenter to tune CDC sessions. In addition, you can use CDC offload and multithreaded processing to improve throughput by moving processing for change data to a different machine.

Use the following methods to tune CDC sessions:

- **Parameters and options.** You can tune sessions by using parameters and options that are specified in the DBMOVER configuration file and on PWX CDC connections.

- **CDC offload processing.** You can use CDC offload processing to distribute PowerExchange column-level processing for change data to the PowerCenter Integration Service machine that runs the CDC session. By distributing processing, you can reduce PowerExchange processing overhead on the system on which the change data resides. You can also use CDC offload processing with the PowerExchange Logger for Linux, UNIX, and Windows to capture change data on a different machine. CDC sessions can then extract change data from the PowerExchange Logger log files on that machine, rather than from the change stream on the original source machine.

- **Multithreaded processing.** If you use CDC offload processing, you can optionally use multithreaded processing to attempt to increase throughput. Multithreaded processing uses multiple threads on the PowerCenter Integration Service machine to perform the offloaded PowerExchange processing.
Using PowerExchange Parameters to Tune CDC Sessions

To tune your PowerExchange installation, you can customize the following parameters in the DBMOVER configuration file:

\[ \text{APPBUFSIZE= size} \]

Defines the maximum size, in bytes, of the buffer that PowerExchange uses to read or write data. This data buffer can exist on a source or target system.

If you are applying change data from the change stream on the source system to 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 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 1048576. Default is 128000.

If the target system is remote, enter the same APPBUFSIZE value in the DBMOVER configuration files on the source and target systems. Also, verify that the APPBUFSIZE value matches the TCPIPBUFSIZE value in the same DBMOVER configuration file. The TCPIPBUFSIZE parameter specifies the maximum size of the TCP/IP buffer.

If the APPBUFSIZE value is not optimal, PowerExchange writes the PWX-01295 message in the PowerExchange log file on the source system. This message includes a recommended minimum value.

\[ \text{COMPRESS= \{Y|N\}} \]

Defines whether PowerExchange uses its proprietary compression algorithm to compress data before it is sent to TCP/IP for transmission to the remote platform.

Default is Y.

PowerExchange uses the COMPRESS setting in the DBMOVER configuration file on the remote system that contacts the PowerExchange Listener. On the PWX CDC application connection, you can override the compression setting in the DBMOVER configuration file. If you enable compression, the CPU consumption of the PowerExchange Listener on the source system might increase.

To avoid unnecessary CPU consumption, set COMPRESS to N in the PowerExchange DBMOVER configuration file on the PowerCenter Integration Service machine.

\[ \text{CAPI\_CONNECTION= ( ..., MEMCACHE= cache\_value, ...)} \]

Amount of memory cache, in kilobytes, that is allocated to reconstruct complete UOWs. You can specify the MEMCACHE parameter on the following CAPI\_CONNECTION statement types:

- SQL
- UDB
- UOWC

PowerExchange keeps all changes in each UOW in cache until it processes the end-UOW record, which is the commit record. 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.

Valid values are from 1 through 519720. Default is 1024.

You might need to increase this value if you have large UOWs. PowerExchange processes a UOW more efficiently if all of the changes are cached in memory. If a UOW might be larger than 1024 KB in size, increase this parameter. For most environments, a value of 10240 (10 MBs) is a good starting value.
Tip: PowerExchange uses the MEMCACHE value to allocate cache memory to each connection for change data extractions. To prevent excessive memory use by a PowerExchange Listener, use a reasonable value for MEMCACHE based on your extraction processing needs and the number of CDC sessions that run concurrently.

**CAPI_CONNECTION=( ...,RSTRADV=rstr_secs, ...)**

Number of seconds that PowerExchange waits before advancing the restart tokens for a data source by returning an empty unit of work (UOW). You can specify the RSTRADV parameter on the following CAPI_CONNECTION statement types:

- MSQl
- UDB
- UOWC

Empty UOWs contain restart tokens only, without any data. PowerExchange uses the restart tokens to determine the start point in the change stream for change data extractions. The wait period for the RSTRADV value starts after a UOW for a data source is processed. PowerExchange resets the wait period after it reads the next UOW for that source or when it returns an empty UOW because the wait period expires.

For sources with low change activity, you can use the RSTSADV 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 RSTRADV=5 and changes are not made to the data source for five seconds, PowerExchange returns an empty UOW to advance the restart point for the data source.

Valid values are from 0 through 86400. If you do not specify RSTRADV, PowerExchange does not return empty UOWs to advance the restart point.

Consider the following issues when you set RSTRADV on CAPI_CONNECTION statements in the PowerExchange DBMOVER configuration file:

- A value of 0 adversely affects performance. PowerExchange returns an empty UOW with restart tokens to PWXPC after each UOW is processed.
- 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 its data buffer and commits restart tokens to the targets. Excessive flush activity can adversely affect performance on the PowerCenter Integration Service machine and target databases.

**LISTENER=(node_name,TCPIP,port,send_bufsize,receive_bufsize,send_msgsize,receive_msgsize, ...)**

Defines a port on which a PowerExchange Listener listens for local or remote connections. The positional parameters the send_bufsize, receive_bufsize, send_msgsize, and receive_msgsize define the send and receive buffer and message sizes. If you do not specify values for these parameters, PowerExchange uses the operating system defaults, which vary based on operating system.

To maximize throughput, consider increasing the send and receive buffer and message sizes on the LISTENER statement on the source system. Contact your network administration to determine the best values to use on your system.

Note: Do not specify values for the send and receive buffer and message sizes that exceed the TCP maximum receive buffer size.

**NODE=(node_name,TCPIP,hostname,port,send_bufsize,receive_bufsize,send_msgsize,receive_msgsize, ...)**

Defines a port the IP information that PowerExchange uses to communicate with a remote PowerExchange Listener. The positional parameters the send_bufsize, receive_bufsize, send_msgsize, and receive_msgsize define the send and receive buffer and message sizes. If you do not specify values for these parameters, PowerExchange uses the operating system defaults, which vary based on operating system.
To maximize throughput, consider increasing the send and receive buffer and message sizes on the NODE statement on the target system. Contact your network administration to determine the best values to use on your system.

**Note:** Do not specify values for the send and receive buffer and message sizes that exceed the TCP maximum receive buffer size.

**TRACE=(trace_id,trace_level,99)**

Defines PowerExchange diagnostic traces that Informatica Global Customer Support uses to solve problems with PowerExchange code.

TRACE statements can severely impact PowerExchange performance. You should use them only at the direction of Informatica Global Customer Support. To enhance performance, remove or comment out all TRACE statements in the DBMOVED configuration files on all systems.

**TCPIP_ASYNC={Y|N}**

Defines whether PowerExchange uses asynchronous network I/O when reading change data. If you specify Y, PowerExchange writes change data to network buffers and reads change data from the change stream asynchronously, which might improve throughput for CDC sessions.

Default is N.

**Restriction:** This parameter is not supported for AIX, i5/OS, or Windows.

**RELATED TOPICS:**

* "Using Connection Options to Tune CDC Sessions " on page 95

**Using Connection Options to Tune CDC Sessions**

In PowerCenter, you can customize options on the PWX CDC connections to tune CDC sessions. The following table describes the connection options that you can use to tune CDC sessions:

<table>
<thead>
<tr>
<th>Connection Option</th>
<th>Description</th>
<th>Tuning Suggestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>Select this option to compress source data during the PowerCenter session. Default is disabled.</td>
<td>Do not use compression.</td>
</tr>
<tr>
<td>Encryption Type</td>
<td>The type of data encryption that PowerExchange uses. Default is None.</td>
<td>Do not use encryption.</td>
</tr>
<tr>
<td>Image Type</td>
<td>Indicates whether PWXPC extracts after images (AI) only or both before and after images (BA) for the changes. Default is BA.</td>
<td>Set to AI.</td>
</tr>
<tr>
<td>Connection Option</td>
<td>Description</td>
<td>Tuning Suggestion</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>UOW Count</td>
<td>The number of UOWs that PWXPC reads from the source before it flushes the data buffer to commit the change data to the targets. Default is 1.</td>
<td>To improve efficiency on the PowerCenter Integration Service machine and the target databases, reduce commit processing.</td>
</tr>
<tr>
<td>Real-time Flush Latency in mill-seconds</td>
<td>The frequency, in milliseconds, with which PWXPC flushes the data buffer to commit the change data to the targets. Default is 0, which is equivalent to two seconds.</td>
<td>To improve efficiency on the PowerCenter Integration Service machine and the target databases, reduce commit processing.</td>
</tr>
<tr>
<td>PWX Latency in seconds</td>
<td>Select the maximum time, in seconds, that PowerExchange on the source platform waits for more change data before flushing data to PWXPC on the PowerCenter Integration Service platform. Default is 2.</td>
<td>Use the default value.</td>
</tr>
<tr>
<td>Maximum Rows Per commit</td>
<td>Maximum number of change records that PWXPC reads from the source before it flushes the data buffer to commit the change data to the targets. Default is 0, which means that PWXPC does not use maximum rows.</td>
<td>To improve efficiency on the PowerCenter Integration Service machine and the target databases, reduce commit processing.</td>
</tr>
<tr>
<td>Minimum Rows Per commit</td>
<td>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 does not use minimum rows.</td>
<td>If your UOWs contain only a few changes, select a larger value for this option to increase the size of the UOWs.</td>
</tr>
<tr>
<td>Offload Processing</td>
<td>Select this option to request CDC offload processing. Default is No.</td>
<td>For more information about offload processing, see “CDC Offload and Multithreaded Processing” on page 97.</td>
</tr>
<tr>
<td>Worker Threads</td>
<td>If you select Offload Processing, you can also set this option to have PowerExchange use multiple threads for the offloaded processing on the PowerCenter Integration Service machine. Enter the number of threads that you want PowerExchange to use. Valid values are from 1 through 64. Default is 0, which means that PowerExchange does not use multithreaded processing.</td>
<td>For more information about offload processing, see “CDC Offload and Multithreaded Processing” on page 97.</td>
</tr>
<tr>
<td>Array Size</td>
<td>If the Worker Threads value is greater than zero, the size of the storage array, in number of records, for the threads. Valid values are from 25 through 100000. Default is 25.</td>
<td>Use 25. <strong>Warning:</strong> If you specify a large value, have large records, or run many sessions that use multithreaded processing, you might experience memory shortages on the PowerCenter Integration Service machine.</td>
</tr>
</tbody>
</table>

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

**RELATED TOPICS:**
- “Tuning Commit Processing” on page 97
- “CDC Offload and Multithreaded Processing” on page 97
Tuning Commit Processing

If the PowerCenter session log for a CDC session contains groups of PWXPC flush messages followed by groups of source-based commit messages from PowerCenter, the CDC session might be reading change data faster than the data can be processed and written to the targets. To resolve this issue, you can adjust the values that you set for following commitment control options on the PWX CDC connection:

- **UOW Count.** If the session log contains mostly PWXPC_10081 flush messages, you might need to increase the value for this option.
- **Real-time Flush Latency in milli-seconds.** If the session log contains mostly PWXPC_10082 flush messages, you might need to increase the value for this option.
- **Maximum Rows Per commit.** If the session log contains mostly PWXPC_12128 flush messages, you might need to increase the value for this option.

PWXPC might also flush change data too frequently because the PWX CDC connection in the CDC session uses too many of the commitment control options. In this case, use a single option to control commit processing and disable the unused options.

If your change data has 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 UOWs of larger size 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.

The following additional factors can also affect the efficiency with which change data is applied to the targets:

- **Buffer Memory.** The DTM Buffer Size and Default Buffer Block Size values can impact the performance of the CDC session. If you have enabled the collection of performance details in the CDC session, review the difference between performance counters 4.5 Time in PowerExchange Processing and 4.6 Elapsed Time. If the elapsed time is much larger that the PowerExchange processing time, buffer memory constraints might exist.
- **Target database.** The performance of the target database can impact the performance of the CDC session. Contact your database administrator to ensure that access to the database is optimized.

CDC Offload and Multithreaded Processing

You can use CDC offload processing with the following types of change data extractions:

- CDC sessions that use real-time extraction mode
- PowerExchange Logger for Linux, UNIX, and Windows

When you use CDC offload processing with real-time extractions, the change data remains on the source system and PowerExchange moves the column-level processing to the PowerCenter Integration Service machine that runs the CDC session. For MVS, DB2 for i5/OS, and Oracle sources, PowerExchange also moves the UOW Cleanser processing to the PowerCenter Integration Service machine.

When you use CDC offload processing with the PowerExchange Logger for Linux, UNIX, and Windows, PowerExchange does the following processing:

- Reads the change data from the source system
- For MVS, DB2 for i5/OS, and Oracle sources, moves the UOW Cleanser processing to the machine on which the PowerExchange Logger is running

The PowerExchange Logger stores the change data in log files on the Linux, UNIX, or Windows machine. CDC sessions can then use continuous extraction mode to extract the change data from the PowerExchange Logger log files instead of from the source system.
You can use multithreaded processing for CDC sessions that select offload processing. By default, PowerExchange uses a single thread to process change data on the PowerCenter Integration Service machine. When you select multithreaded processing, PowerExchange uses multiple threads to process the change records in each UOW.

Planning for CDC Offload and Multithreaded Processing

Before you configure CDC offload and multithreaded processing, review the following considerations, requirements, and restrictions.

Restrictions and Requirements for CDC Offload Processing

When you use CDC offload processing, the following restrictions and requirements apply:

- You must configure CAPI_CONNECTION statements for the data source in the DBMOVER configuration file on the remote system. For real-time extraction mode, configure the CAPI_CONNECTION statements in the dbmover.cfg configuration file on the PowerCenter Integration Service machine. For the PowerExchange Logger for Linux, UNIX, and Windows, configure the CAPI_CONNECTION statements in the dbmover.cfg configuration file that the PowerExchange Logger uses.
- If you select the Idle Time option on the connection, you can only select values -1 or 0. PWXPC sets values larger than 0 to 0.
- PowerExchange does not invoke MVS RACF security authorization for change data extraction. Specifically, PowerExchange does not validate any CAPX.CND profiles.
- PowerExchange does not support CDC offload processing for capture registrations that have been created 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
- To store change data in the PowerExchange Logger log files, you must configure capture registrations for partial condense processing by selecting Part in the Condense list in the PowerExchange Navigator.
- The PowerExchange Logger for Linux, UNIX, and Windows cannot process capture registrations from MVS or i5/OS that are configured for full condense processing. You must either change these registrations to use partial condense processing or exclude them by using group definition files.
- Each PowerExchange Logger for Linux, UNIX, and Windows process must read all of the capture registrations that it uses from a single CCT file. Also, each PowerExchange Logger process must store the names of its log files in a unique CDCT file.
- PowerExchange does not support batch extraction mode for change data that is stored in PowerExchange Logger log files on a system that is remote from where the extraction maps reside. In this situation, you must use continuous extraction mode.
Considerations for Multithreaded Processing

In specific situations, multithreaded processing might improve performance for a CDC session. Before you configure multithreaded processing options, review the following considerations:

- Use multithreaded processing when the PWX reader thread of a CDC session uses 100% of a single CPU on a multi-CPU server on the PowerCenter Integration Service platform while processing change data. When a single CPU is consumed, spreading the PowerExchange processing across multiple threads improves throughput. Otherwise, additional threads do not improve throughput.

- If the network processing between the source and PowerCenter Integration Service machines is slow, try specifying 1 for the **Worker Threads** option to help improve throughput. When you specify one or more worker threads, PowerExchange overlaps network processing with the processing of the change data on the PowerCenter Integration Service machine.

- For optimal performance, the value for the **Worker Threads** option should not exceed the number of installed or available processors on the PowerCenter Integration Service machine.

### Enabling Offload and Multithreaded Processing for CDC Sessions

To use CDC offload processing and multithreaded processing, you must configure connection options in the CDC session and **CAPI_CONNECTION** statements in the PowerExchange DBMOVER configuration file.

To enable offload and multithreaded processing for CDC sessions:

1. Configure the following options on the PWX CDC Real Time application connection for the CDC session:

   - **Location**
     The node name of the system on which the change data resides. This node name must be the name of a **NODE** statement in the **dbmover.cfg** configuration file on the PowerCenter Integration Service machine.

   - **Offload Processing**
     Specifies whether to use CDC offload processing to move PowerExchange processing for the change data from the source system to the PowerCenter Integration Service machine.
     The options are:
     - **No**
     - **Yes**
     - **Auto**. PowerExchange determines whether to use offload processing.
     Default is **No**.

   - **Worker Threads**
     When you select CDC offload processing, specifies the number of threads that PowerExchange uses on the PowerCenter Integration Service machine to process change data. You must also enter a value for the **Array Size**.
     Default is 0.

   - **Array Size**
     If the **Worker Threads** value is greater than zero, the size of the storage array, in number of records, for the threads.
     Default is 25.
CAPI Connection Name

The name of the source CAPI_CONNECTION statement in the dbmover.cfg on the PowerCenter Integration Service machine.

2. Copy the CAPI_CONNECTION statements from the DBMOVER configuration file on the source system to the dbmover.cfg configuration file on the PowerCenter Integration Service machine. For MVS sources, remove all MVS-specific parameters from the UOWC CAPI_CONNECTION statement.

Use the following table to select the correct CAPI_CONNECTION statement types to configure, based on source type:

<table>
<thead>
<tr>
<th>CDC Source Type</th>
<th>CAPI_CONNECTION Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 for i5/OS</td>
<td>AS4J and UOWC</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>UDB</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>MSQL</td>
</tr>
<tr>
<td>MVS sources</td>
<td>LRAP and UOWC</td>
</tr>
<tr>
<td>Oracle</td>
<td>ORCL and UOWC</td>
</tr>
</tbody>
</table>

Configuring PowerExchange to Capture Change Data on a Remote System

You can use CDC offload processing with the PowerExchange Logger for Linux, UNIX, and Windows to capture change data from source systems other than the system where the PowerExchange Logger runs. With CDC offload processing, a PowerExchange Logger for Linux, UNIX, and Windows can capture change data from i5/OS and MVS systems as well as from other Linux, UNIX, or Windows systems.

CDC sessions use continuous extraction mode to extract the change data from the PowerExchange Logger log files instead of from the source system.

You must first install PowerExchange on the remote Linux, UNIX, or Windows system.

Before you start a PowerExchange Logger for Linux, UNIX, and Windows process on a remote system, configure the pwxccl.cfg and the dbmover.cfg configuration files on that system. When you use CDC offload processing, each PowerExchange Logger must have unique pwxccl.cfg and dbmover.cfg configuration files.

To extract the change data from the PowerExchange Logger on the remote system, you must also configure and start a PowerExchange Listener on that system. The dbmover.cfg file that the PowerExchange Listener uses must specify the same CAPT_PATH value as the dbmover.cfg file that the PowerExchange Logger uses. Alternatively, you can use the same dbmover.cfg file for the PowerExchange Logger and the PowerExchange Listener.

The following steps describe how to configure a PowerExchange Logger and PowerExchange Listener to offload change data from source systems and capture that data to PowerExchange Logger log files on Linux, UNIX, or Windows.

**Related Topics:**

- “Extracting Change Data Captured on a Remote System” on page 106
Configuring pwxccl.cfg

Configure the pwxccl.cfg configuration file for the PowerExchange Logger on the remote system where the
PowerExchange Logger will run.

PowerExchange provides a sample pwxccl.cfg file in the PowerExchange installation directory, which you can
copy and then edit. For CDC offload processing, customize the following parameters:

**CAPTURE_NODE**

Specifies the node name of the system on which the change data was originally captured.

This node name must match the node name in a NODE statement in the dbmover.cfg configuration file that
the PowerExchange Logger uses.

**CAPTURE_NODE_EPWD**

Specifies an encrypted password for the CAPTURE_NODE_UID user ID.

If you specify CAPTURE_NODE_UID, you must specify a password for that user ID by using either
CAPTURE_NODE_EPWD or CAPTURE_NODE_PWD. If you specify CAPTURE_NODE_EPWD, do not also
specify CAPTURE_NODE_PWD.

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

**CAPTURE_NODE_PWD**

Specifies a clear text password for the CAPTURE_NODE_UID user ID.

If you specify CAPTURE_NODE_UID, you must specify a password for that user ID by using either
CAPTURE_NODE_EPWD or CAPTURE_NODE_PWD. If you specify CAPTURE_NODE_PWD, do not also
specify CAPTURE_NODE_EPWD.

**CAPTURE_NODE_UID**

Specifies a user ID that permits PowerExchange to read 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 the CAPTURE_NODE is an z/OS or i5/OS system with a SECURITY setting of 1 or 2, you must specify a
valid operating system user ID. If the SECURITY setting is 2, PowerExchange uses the specified user ID to
control access to capture registrations and change data. However, if the SECURITY setting is 1,
PowerExchange uses the user ID under which the PowerExchange Listener job runs.

If the CAPTURE_NODE is an z/OS or i5/OS system with 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 the CAPTURE_NODE is a Linux, UNIX, or Windows system, specify a user ID that is valid for the data
source type:

- For a DB2 for Linux, UNIX, or Windows source, enter a valid operating system user ID that has DB2
  DBADM or SYSADM authority.
- For an Oracle source, enter a database user ID that permits access to Oracle redo logs and Oracle
  LogMiner.
- For a SQL Server instance that uses SQL Server Authentication, enter a database user ID that permits
  access to the SQL Server distribution database. For a SQL Server instance that uses 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.
CHKPT_BASENAME

Specifies an existing path and base file name to use for generating the PowerExchange Logger checkpoint files.

CONDENSENAME

Optional. Specifies a name for the command-handling service for a PowerExchange Condense process to which pwxcmd commands will be issued. Issue pwxcmd commands from a Linux, UNIX, or Windows system to a PowerExchange Condense process running on an i5/OS system.

Specify the same service name in the associated SVCNODE statement in the DBMOVER configuration file.

CONN_OVR

Specifies the name of the CAPI_CONNECTION statement in the dbmover.cfg file that the PowerExchange Logger uses. This CAPI_CONNECTION statement defines the connection to the change stream for the data source type.

For data sources that include UOW Cleanser (UOWC) CAPI_CONNECTION statements, specify the name of this statement. For all other data sources, specify the CAPI_CONNECTION name for the data source type.

DB_TYPE

Specifies the data source type.

Use the following table to select the correct DB_TYPE to configure, based on source type:

<table>
<thead>
<tr>
<th>CDC Source Type</th>
<th>DB_TYPE Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adabas</td>
<td>ADA</td>
</tr>
<tr>
<td>Datacom</td>
<td>DCM</td>
</tr>
<tr>
<td>DB2 for i5/OS</td>
<td>AS4</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>UDB</td>
</tr>
<tr>
<td>DB2 for z/OS</td>
<td>DB2</td>
</tr>
<tr>
<td>IDMS log-based</td>
<td>IDL</td>
</tr>
<tr>
<td>IMS</td>
<td>IMS</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>MSS</td>
</tr>
<tr>
<td>Oracle</td>
<td>ORA</td>
</tr>
<tr>
<td>VSAM</td>
<td>VSM</td>
</tr>
</tbody>
</table>

DBID

Specifies the source collection identifier that you defined in the registration group. The PowerExchange Navigator displays this value in the Resource Inspector when you open the registration group. When used with DB_TYPE, it defines selection criteria for capture registrations in the CCT file.
Use the following table to select the correct DBID value, based on source type:

<table>
<thead>
<tr>
<th>CDC Source Type</th>
<th>DBID Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adabas</td>
<td>The Instance name that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td>Datacom</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The MUF Name value that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- For Datacom synchronous CDC, the MUF parameter value in the DTLINPUT data set specified in the MUF JCL.</td>
</tr>
<tr>
<td></td>
<td>- For Datacom table-based CDC, the REG_MUF parameter value in the ECCRDCMP member of the RUNLIB library.</td>
</tr>
<tr>
<td>DB2 for i5/OS</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The Instance name that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- The INST parameter value in the AS4J CAPI_CONNECTION statement in the DBMOVER member of the CFG file.</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>The Database name that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td>DB2 for z/OS</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The Instance name that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- The RN parameter value from the DB2 statement in the REPDB2OP member of the RUNLIB library.</td>
</tr>
<tr>
<td>IDMS Log-based</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The Logsid value that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- The LOGSID parameter value in the ECCRIDLP member of the RUNLIB library.</td>
</tr>
<tr>
<td>IMS</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The IMSID value that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- For IMS log-based CDC, the first parameter of the IMSID statement in the CAPTIMS member of the RUNLIB library.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>The Instance name that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
<tr>
<td>Oracle</td>
<td>ORCL and UOWC</td>
</tr>
<tr>
<td>VSAM</td>
<td>The Instance name that you defined for the registration group in the PowerExchange Navigator.</td>
</tr>
</tbody>
</table>

**EPWD**

A deprecated parameter. Use CAPTURE_NODE_EPWD instead. If you specify both CAPTURE_NODE_EPWD and EPWD, CAPTURE_NODE_EPWD takes precedence.

**EXT_CAPT_MASK**

Specifies a path and unique prefix that PowerExchange uses to generate the PowerExchange Logger log files.
PWD

A deprecated parameter. Use CAPTURE_NODE_PWD instead. If you specify both CAPTURE_NODE_PWD and PWD, CAPTURE_NODE_PWD takes precedence.

RESTART_TOKEN and SEQUENCE_TOKEN

Optionally, specifies a restart point for starting change data processing when you cold start the PowerExchange Logger.

The format of the restart tokens varies based on data source type. If specified, the restart token format must match the format required by the DB_TYPE specified. If you do not specify these parameters, the PowerExchange Logger uses the end of the change stream as the restart point when you cold start the PowerExchange Logger.

UID

A deprecated parameter. Use CAPTURE_NODE_UID instead. If you specify both CAPTURE_NODE_UID and UID, CAPTURE_NODE_UID takes precedence.

For more information about the pwxccl.cfg parameters, see the PowerExchange CDC Guide for Linux, UNIX, and Windows.

Configuring dbmover.cfg on the PowerExchange Logger Machine

On the remote system where the PowerExchange Logger will run, configure the dbmover.cfg file that the PowerExchange Logger and PowerExchange Listener will use.

Note: Unless the change data is captured on the PowerCenter Integration Service machine, you must run a PowerExchange Listener so CDC sessions can extract the offloaded change data.

The dbmover.cfg file that the PowerExchange Listener uses must specify the same CAPT_PATH value as the dbmover.cfg that the PowerExchange Logger uses. Alternatively, you can use the same dbmover.cfg configuration file for the PowerExchange Logger and PowerExchange Listener. This step assumes that you use the same dbmover.cfg file.

PowerExchange provides a sample dbmover.cfg file in the PowerExchange installation directory, which you can copy and then edit. For CDC offload processing, set the following parameters:

CAPT_PATH

Specifies the path to the directory where the CDCT file resides. The CDCT file contains information about the PowerExchange Logger log files, such as file names and number of records.

Each PowerExchange Logger that uses CDC offload processing to capture change data requires its own CDCT file.

CAPX CAPI_CONNECTION

Specifies parameters for continuous extraction of change data from PowerExchange Logger log files. In continuous extraction mode, extractions run in near real time and read the data in the PowerExchange Logger log files as the change stream.

In the DFLTINST parameter of the CAPX CAPI_CONNECTION, specify the DBID value from the PowerExchange Logger pwxccl.cfg configuration file.

LOGPATH

Specifies the path to the PowerExchange log files that contain PowerExchange Logger messages.

NODE

Specifies the TCP/IP connection information for a PowerExchange Listener.
Configure a NODE statement for the system on which the change data was originally captured. Specify the node name for this statement in the CAPTURE_NODE parameter of the PowerExchange Logger pwxccl.cfg configuration file.

**Source-specific CAPI_CONNECTION**

Specifies CAPI parameters that are specific to the data source type and that PowerExchange uses to connect to the change stream.

Copy the CAPI_CONNECTION statements from the DBMOVER configuration file on the source system where the change data resides. Use the following table to select the correct CAPI_CONNECTION statement types to configure, based on source type:

<table>
<thead>
<tr>
<th>CDC Source Type</th>
<th>CAPI_CONNECTION Statements</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 for i5/OS</td>
<td>AS4J and UOWC</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>UDB</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>MSQL</td>
</tr>
<tr>
<td>z/OS sources</td>
<td>LRAP and UOWC</td>
</tr>
<tr>
<td>Oracle</td>
<td>ORCL and UOWC</td>
</tr>
</tbody>
</table>

For z/OS sources, remove z/OS-specific parameters from the UOWC CAPI_CONNECTION statement.

**SVCNODE**

Optional. Specifies the TCP/IP port on which a command-handling service for a PowerExchange Listener or PowerExchange Condense process listens for pwxcmd commands.

**TRACING**

Optional. Enables alternative logging. By using alternative logging, you can separate PowerExchange Logger messages from other PowerExchange messages.

### Configuring dbmover.cfg on the PowerCenter Integration Service Machine

In the dbmover.cfg configuration file on the PowerCenter Integration Service machine, add a NODE statement for the PowerExchange Listeners that run on the following systems:

- The system where the change data was originally captured and where the capture registrations reside
- The system where the change data is stored in PowerExchange Logger for Linux, UNIX, and Windows log files

### Configuring Capture Registrations for the PowerExchange Logger

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

If capture registrations do not specify **Part** for the **Condense** option, delete the capture registrations and corresponding extraction maps. Then create the capture registrations again. PowerExchange generates corresponding extraction maps. You can edit the PowerExchange-generated extraction maps or create additional ones.

**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 pwxccl.cfg configuration file. With the AI setting, the PowerExchange Logger captures after images only.
Consequently, PowerExchange cannot populate BI columns with before images. Also, with this setting, PowerExchange writes Nulls to CI columns for any INSERT or DELETE operations.

**Starting the PowerExchange Logger and PowerExchange Listener**

Start the PowerExchange Logger and PowerExchange Listener on the remote system that will capture the change data.

**Note**: If the remote system also runs the PowerCenter Integration Service, you can use local mode to extract the data instead of a PowerExchange Listener.

**Extracting Change Data Captured on a Remote System**

After you have captured change data on a remote system in the PowerExchange Logger for Linux, UNIX, and Windows log files, you can use continuous extraction mode to extract the change data in a CDC session. In the CDC session, select the appropriate PWX CDC Real Time connection for the source type. For example, if you captured change data for a DB2 for i5/OS source to PowerExchange Logger log files on a remote system, use a PWX DB2i5OS CDC Real Time connection to extract the data.

Customize the following connection options to extract offloaded change data:

- **Location**. Specify the node name for the PowerExchange Listener that runs on the remote system where the change data was stored in PowerExchange Logger log files.

- **Map Location**. Specify the node name for the PowerExchange Listener that runs on the source system where the change data was originally captured. The PowerExchange Listener on the original source system stores the capture registrations.

- **Map Location User and Map Location Password**. Specify a user ID and password that can access the capture registrations for the change data.
  
  If the PowerExchange Listener on the source system is running on MVS or i5/OS and is configured with security, specify a valid operating system user ID. You do not need to specify this parameter if the PowerExchange Listener is running without security.

  If the PowerExchange Listener on the data source system is running on Linux, UNIX, or Windows, specify a valid database user ID.

- **CAPI Connection Name Override**. Specify the name of the CAPX CAPI_CONNECTION in the dbmover.cfg configuration file used by the PowerExchange Listener on the remote system where the change data is stored in PowerExchange Logger log files.

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

**Configuration File Examples for CDC Offload Processing**

The following examples show the configuration required for CDC offload processing.

**Extracting Change Data from i5/OS Using Offload Processing - Example**

In this example, a CDC session that uses real-time connections to extract change data from an i5/OS source is changed to use CDC offload processing. The source change data remains on i5/OS but all column-level and UOW Cleanser processing is moved to the PowerCenter Integration Service machine.

The i5/OS system has the following CAPI_CONNECTION statements in the DBMOVER member in the RUNLIB library that the PowerExchange Listener uses to read change data:

```
CAPI_CONNECTION=(NAME=I5OUWC,
                TYPE=(OUWC,CAPINAME=I5_AS43,RSTRADV=600,MEMCACHE=20480))
```
To extract change data from i5/OS using CDC offload processing:

1. Configure the dbmover.cfg configuration file on the PowerCenter Integration Service machine for CDC offload processing.

   Copy the UOWC and AS4J CAPI_CONNECTION statements from the DBMOVER member on i5/OS to the dbmover.cfg configuration file on the PowerCenter Integration Service machine. In this example, the following CAPI_CONNECTION statements are copied into the dbmover.cfg:

   ```
   CAPI_CONNECTION=(NAME=i5UOWC,
   TYPE=(UOWC,CAPINAME=i5_AS4J,RSTRADV=600,MEMCACHE=20480))
   CAPI_Connection=(NAME=i5_AS4J,
   TYPE=(AS4J,JOURNAL=PRODDATA/PRODJRN,INST=PROD,EOF=N,
   STOPIT=(CONT=5),LIBASUSER=Y))
   ```

2. Stop the CDC session.

3. Update the following options on the PWX CDC Real Time application connection in the CDC session:
   - Select **Yes** for the **Offload Processing** option.
   - In the **CAPI Connection Name** option, specify the name of the UOWC CAPI_CONNECTION statement. In this example, the name is i5UOWC.

4. Restart the CDC session.

### Capturing and Extracting Change Data from i5/OS on UNIX - Example

In this example, change data for DB2 for i5/OS sources is captured on a UNIX machine by the PowerExchange Logger for Linux, UNIX, and Windows. The DB2 for i5/OS sources are registered for capture on the i5/OS system. A CDC session then extracts the change data for the DB2 sources from PowerExchange Logger log files on the UNIX machine, rather than from the i5/OS system where the change data was originally captured.

The i5/OS system has the following CAPI_CONNECTION statements in the DBMOVER member in the `datalib/CFG` library that the PowerExchange Listener uses to read change data:

```
CAPI_Connection=(NAME=i5UOWC,
   TYPE=(UOWC,CAPINAME=i5_AS4J,RSTRADV=600,MEMCACHE=20480))
CAPI_Connection=(NAME=i5_AS4J,
   TYPE=(AS4J,JOURNAL=PRODDATA/PRODJRN,INST=PROD,EOF=N,
   STOPIT=(CONT=5),LIBASUSER=Y))
```

The instance name used to register the DB2 tables for capture on the i5/OS system is called PROD.

The following procedure assumes that PowerExchange is installed and configured on the UNIX system where the PowerExchange Logger for Linux, UNIX, and Windows will run.

To capture and extract change data from i5/OS on UNIX:

1. Configure the PowerExchange Logger for Linux, UNIX, and Windows on the UNIX system by performing the following steps:
   - Configure `pwxccl.cfg`.
   - Configure `dbmover.cfg` on the PowerExchange Logger machine.

   In this example, the `dbmover.cfg` has the following parameters:

   ```
   /*
   /* dbmover.cfg
   */
   LISTENER=(unix1,TCPIP,2480)
   NODE=(i5OS2,TCPIP,prod2,2480)
   ...
   logpath=/pwx/logs/i5oscond
   CAPT_XTRA=/pwx/capture/i5oscond/camaps
   CAPT_PATH=/pwx/capture/i5oscond
   ```
On the PowerCenter Integration Service machine, customize the following statements:

- NODE statement to point to the PowerExchange Listener on the UNIX system
- NODE statement to point to the PowerExchange Listener on the i5/OS system

In this example, the following statements are added to the dbmover.cfg on the PowerCenter Integration Service machine:

```
NODE=(unix1,TCPIP,unix1,2480)
NODE=(i5OS2,TCPIP,prod2,2480)
```

4. Create and configure the PowerCenter mapping, session, and workflow to extract the change data.

5. To extract the change data from the UNIX system, configure a PWX DB2i5OS CDC Real Time application connection in the CDC session.

   In this example, specify the following options to point to the UNIX system for the change data, the i5/OS system for the extraction maps, and the CAPX CAPI_CONNECTION to use continuous extraction mode:

   - For the Location option, specify unix1.
   - For the Map Location option, specify i5OS2.
   - For the CAPI Connection Name option, specify CAPXPROD.

6. Cold start the CDC session to extract the change data from the PowerExchange Logger log files on the UNIX system.

**Related Topics:**

- "Configuring pwxcl.cfg " on page 101
- "Configuring dbmover.cfg on the PowerExchange Logger Machine" on page 104
INDEX

A
altering table definitions 40
APPBUFSIZE
  configuration parameter for PowerExchange Listener 93
application name
  configuring for CDC sessions 70
application names 51
AS4J CAPI_CONNECTION parameters
  parameters and syntax 8

B
backing up PowerExchange Condense files 30

C
CAPI connection statements
  AS4J parameters 8
  UOWC parameters 12
capture scenarios 1
CDC data map
  extraction map 75
CDC sessions
  commit processing 56
  default restart points 52
  methods of starting 51, 78
  offload processing 61
  recovery example 84
  restart points for warm starts 53
  restart token file 75
  stopping 80
change data capture
  architecture 17
  configuration 22
Change Data Capture
  overview 1
close (pwxcmd) 16
closeforce (pwxcmd) 16
command handler task
  PowerExchange Condense 20
commit processing
  configuring for CDC sessions 71
  controlling with connection attributes 57
  examples 59
  in CDC sessions 56
  minimum and maximum rows per commit 58
  target latency 59
condense batch mode 22
condense continuous mode 22
condensing data
  PowerExchange Condense 19
controller task
  PowerExchange Condense 19

D
DBMOVER statements and parameters
  RMTRDBDIRE 38
  RMTSYSNAME 38
diagnostic task
  PowerExchange Condense 20
  DISPLAY ACTIVE command 88
  DISPLAY STATUS command 29
displaystatus (pwxcmd) 29
DTL__CAPXRESTART1
  sequence token 74
DTL__CAPXRESTART2
  restart token 74
DTLUAPPL
  displaying restart tokens 74
DTLUTSK utility 80

e
extraction map columns, PowerExchange-generated
  DTL__BI_columnname 44
  DTL__CAPXACTION 44
  DTL__CAPXASDELIND 44
  DTL__CAPXRESTART1 44
  DTL__CAPXRESTART2 44
  DTL__CAPXRRN 44
  DTL__CAPXTIMESTAMP 44
  DTL__CAPXUOW 44
  DTL__CAPXUSER 44
  DTL__CI_columnname 44
extraction maps
  PowerExchange-generated columns 44
  extraction process 4

F
fileswitch (pwxcmd) 30

G
group source
  description 54
  processing CDC data for multiple source definitions 55

I
idle time
  configuring for a CDC session 69
  description 69
**J**

journals, remote 36

**L**

listtask (pwxicmd) 16, 86, 88

**M**

maximum row count
configuring for a CDC session 71
minimum row count
configuring for a CDC session 71
multiple journal condensing
PowerExchange Condense 20
multithreaded processing
overview 61

**O**

offload processing
overview 61

**P**

$PMRootDir/Restart 74
PowerExchange Condense
backing up output files 30
cmd handler task 20
cntroller task 19
diagnostic task 20
displaying status of 29
progress messages and tracing 21
shutting down 28
PowerExchange Condense configuration file
CAPT_IMAGE parameter 23
CHKPT_BASENAME parameter 23
CHKPT_NUM parameter 23
COLL_END_LOG parameter 23
COND_CDCRT_RET_P parameter 23
CONDENSE_SHUTDOWN_TIMEOUT parameter 23
CONDENSENAME parameter 23
DB_TYPE parameter 23
DBID parameter 23
FILE_SWITCH_CRIT parameter 23
FILE_SWITCH_VAL parameter 23
JRNL parameter 23
KEY_CHANGE_ALW parameter 23
NO_DATA_WAIT parameter 23
NO_DATA_WAIT2 parameter 23
OBJLOC parameter 23
REG parameter 23
RESTART_TOKEN parameter 23
SEQUENCE_TOKEN parameter 23
SIGNALLING parameter 23
VERBOISE parameter 23
PowerExchange Listener
DISPLAY ACTIVE command 88
PowerExchange-generated extraction map columns
DTL__BI_columnname 44
DTL__CAPXACTION 44
DTL__CAPXCASDELIND 44
DTL__CAPXRESTART1 44

**R**

real-time flush latency
configuring for a CDC session 71
recovery
example 84
PM_RECV_STATE table 49, 50
PM_RECOVERY table 49
PM_TGT_RUN_ID table 49
recovery information for nonrelational targets 50
recovery state file for nonrelational targets 51
recovery tables for relational targets 49
remote journals 36
restart
$PMRootDir/Restart 70, 74
application name 70
default restart points 52
earliest restart points 52
methods of starting CDC sessions 51, 78
null restart tokens 52
restart token file 48, 70
restart token file folder 70
RESTART1 76
RESTART2 76
restart points
defaults 52
earliest 52
restart token
DTL__CAPXRESTART2 74
restart token file
example 77
explicit override 75
overview 47
special override 76
syntax 75
restart tokens
displaying with DTLUAPPL 74
DTL__CAPXRESTART1 74
DTL__CAPXRESTART2 74
null 52
overview 47
recovery state file 51
recovery state table 50
RMTRDBDIRE statement 38
RMTSYSNAME statement 38
index