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(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, 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 © Rouge 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/licenses.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/ licenses.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.dojo.org/license.html.

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/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.html.
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 Walnes, 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; 6,640,226; 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-CCz-900-0001
# Table of Contents

**Preface** .......................................................................................................................... x

Informatica Resources. .......................................................................................................... x
  Informatica Customer Portal. ................................................................................................. x
  Informatica Documentation. .................................................................................................... x
  Informatica Web Site. ............................................................................................................... xi
  Informatica How-To Library. .................................................................................................. xi
  Informatica Knowledge Base. ................................................................................................ xi
  Informatica Multimedia Knowledge Base. ........................................................................... xi
  Informatica Global Customer Support. .................................................................................. xi

**Part I: PowerExchange Change Data Capture Introduction** ............................................. 1

**Chapter 1: Change Data Capture Introduction** ................................................................. 2
  PowerExchange CDC Overview. .......................................................................................... 2
  PowerExchange Components for CDC. ................................................................................. 4
    PowerExchange Agent ......................................................................................................... 4
    PowerExchange Environmental Change Capture Routine (ECCR). .................................. 4
    PowerExchange Listener. ................................................................................................... 5
    PowerExchange Logger ...................................................................................................... 5
    PowerExchange Condense .................................................................................................. 5
    PowerExchange Navigator ................................................................................................. 6
  PowerExchange CDC for MVS Data Sources. .................................................................... 6
    Adabas Change Data Capture. ............................................................................................ 6
    Datacom Change Data Capture. .......................................................................................... 6
    DB2 for z/OS Change Data Capture. .................................................................................. 7
    IDMS Change Data Capture. .............................................................................................. 7
    IMS Change Data Capture. ................................................................................................. 8
    VSAM Change Data Capture. .............................................................................................. 8
  PowerExchange Integration with PowerCenter. ................................................................. 8
  CDC Implementation Summary. .......................................................................................... 9

**Part II: CDC Components Configuration and Management** ......................................... 11

**Chapter 2: PowerExchange Listener** .............................................................................. 12
  PowerExchange Listener Overview. .................................................................................. 12
  Configuring the PowerExchange Listener for CDC. ......................................................... 12
    Configuring the PowerExchange Listener JCL. ................................................................. 13
    Configuring CAPI_CONNECTION Statements. ................................................................. 13
  Managing the PowerExchange Listener. .......................................................................... 18
### Chapter 3: PowerExchange Agent 20

- PowerExchange Agent Overview...................................................... 20
- Running Multiple Instances of the PowerExchange Agent ......................... 21
- Configuring MVS for the PowerExchange Agent .................................... 21
  - Linkage Indexes and Common Data Spaces........................................ 21
  - Global Enqueue for PowerExchange CDC....................................... 22
- Configuring the PowerExchange Agent .............................................. 22
  - EDMSDIR Module Options.......................................................... 22
  - Configuring the EDMSDIR Module Options...................................... 24
  - Configuring AGENTCTL Parameters.............................................. 25
  - Configuring AGENTREP Parameters.............................................. 27
  - Customizing the PowerExchange Agent JCL.................................... 28
- PowerExchange Agent Message Log.................................................. 31
- Managing the PowerExchange Agent.................................................. 31
  - Starting the PowerExchange Agent............................................... 31
  - Stopping the PowerExchange Agent............................................... 32
  - Controlling the PowerExchange Agent.......................................... 32
  - Managing Capture Registration Caching...................................... 33
- Controlling Security for the PowerExchange Agent.................................. 34
  - Controlling Access to PowerExchange Agent Services........................ 34
  - Controlling Access to PowerExchange Agent Commands........................ 35
  - Controlling Access to PowerExchange Components........................... 35

### Chapter 4: PowerExchange Logger for MVS 37

- PowerExchange Logger for MVS Overview.......................................... 37
- Multiple Instances of the PowerExchange Logger for MVS.......................... 38
- Planning Considerations for the PowerExchange Logger for MVS.................... 39
  - XCF Groups............................................................................... 39
  - Recall of Archived Log Data Sets.............................................. 39
- Configuring the PowerExchange Logger for MVS...................................... 39
  - Configuring the EDMUPARM Module Options.................................... 40
  - Creating Active Log and Emergency Restart Data Sets.......................... 46
  - Customizing the PowerExchange Logger JCL.................................... 47
- Managing the PowerExchange Logger for MVS........................................ 48
  - Starting the PowerExchange Logger for MVS.................................... 48
  - Stopping the PowerExchange Logger for MVS.................................... 49
  - Controlling the PowerExchange Logger for MVS.................................. 49
  - Overriding Log-Read API Timed Defaults..................................... 49
  - Resolving In-Doubt Units of Work.............................................. 50
Part III: CDC Sources Configuration and Management ......................... 104

Chapter 6: Adabas Change Data Capture ........................................ 105
Introduction to Adabas Change Data Capture .................................. 105
Adabas Planning Considerations ............................................... 105
  Operational Considerations ............................................... 106
  Accessing Multiple Databases ........................................... 106
  PowerExchange CDC Component Relationships ............................ 106
Configuring Adabas for Change Data Capture ................................ 107
  Customizing the PowerExchange Adabas Exit 2 Sample .................. 107
Configuring the Adabas ECCR ............................................. 108
  Configuring the Adabas ECCR Parameters ............................... 108
  Configuring the Adabas ECCR JCL ...................................... 111
  Testing the Installation .................................................. 111
Managing Adabas Change Data Capture ...................................... 112
  Starting the Adabas ECCR ............................................... 112
  Stopping the Adabas ECCR ............................................. 113
  Using the DTLCCADW Utility ............................................ 113

Chapter 7: Batch VSAM Change Data Capture ................................ 114
Introduction to Batch VSAM Change Data Capture .......................... 114
  Relationships with Other PowerExchange Components .................. 114
  Batch VSAM ECCR Restrictions ......................................... 115
Configuring Batch VSAM Jobs for Change Data Capture ...................... 116
  Making the Batch VSAM ECCR Available to Batch Jobs ................. 117
  Activating and Loading the Batch VSAM ECCR Interface ................. 117
  Restoring VSAM Data Sets When Using the Batch VSAM ECCR ............ 118
Managing Batch VSAM Change Data Capture ................................ 118
  Controlling the Batch VSAM ECCR ..................................... 118
  Output from the Batch VSAM ECCR ..................................... 119
  Stopping Change Data Capture for VSAM Sources ....................... 120
  Stopping the Batch VSAM ECCR ....................................... 120
  Refreshing the Batch VSAM ECCR ..................................... 121
  Application Recovery Considerations .................................... 121
Managing VSAM Schema Changes ............................................. 121

Chapter 8: CICS/VSAM Change Data Capture .................................. 122
Introduction to CICS/VSAM Change Data Capture ......................... 122
Planning for CICS/VSAM Change Data Capture ........................... 122
  Prerequisites for CICS/VSAM CDC ..................................... 122
  Requirements and Restrictions for CICS/VSAM CDC ..................... 123
  CICS XFCFRIN and XFCFROUT Global Exits ............................. 123
## Table of Contents

**Chapter 13: IMS Log-Based Change Data Capture** ............................. 196

Introduction to IMS Change Data Capture. ........................................ 196

IMS Log-Based Change Data Capture. ........................................... 197

Relationships with Other PowerExchange Components. ......................... 198

Configuring IMS for Log-Based CDC. .......................................... 199

Specifying the EXIT Parameter in the DBD Source. ........................... 199

Configuring the IMS Log-Based ECCR. ............................................ 199

Configuring the IMS Log-Based ECCR Parameters. ........................... 200

Adding the PowerExchange LOAD Library to the APF List. ................... 204

Configuring the IMS Log-Based ECCR JCL. ................................... 204

Managing IMS Log-Based CDC. .................................................... 205

Starting the IMS Log-Based ECCR. ............................................. 205

Controlling the IMS Log-Based ECCR. ........................................... 205

Stopping Change Data Capture for IMS Sources. .............................. 206

Using the DTLCUIML Utility to Create Restart Points. ......................... 207

Refreshing the IMS Log-based ECCR. ........................................... 207

Managing IMS Schema Changes. .................................................... 207

**Chapter 14: IMS Synchronous Change Data Capture** ........................ 208

Introduction to IMS Change Data Capture. ........................................ 208

IMS Synchronous Change Data Capture. ........................................... 209

IMS CDC Operational Considerations. ............................................. 210

Relationships with Other PowerExchange Components. ......................... 211

Configuring the IMS Synchronous ECCR. ....................................... 212

Compatibility with BMC Software Products. ..................................... 212

Configuring IMS DBRC. ............................................................... 213

Configuring IMS Region JCL. ....................................................... 214

MVS LNKLST Concatenation ......................................................... 218

Activating the IMS Synchronous ECCR. ........................................... 218

Output from the IMS ECCR. .......................................................... 220
Chapter 17: Managing Change Data Extractions

Starting PowerCenter CDC Sessions
- Cold Start Processing
- Warm Start Processing
- Recovery Processing

Stopping PowerCenter CDC Sessions
- Stop Command Processing
- Terminating Conditions

Changing PowerCenter CDC Sessions
- Examples of Creating a Restart Point

Recovering PowerCenter CDC Sessions
- Example of Session Recovery

Chapter 18: Monitoring and Tuning Options

Monitoring Change Data Extractions
- Monitoring CDC Sessions in PowerExchange
- Monitoring CDC Sessions in PowerCenter

Tuning Change Data Extractions
- Using PowerExchange Parameters to Tune CDC Sessions
- Using Connection Options to Tune CDC Sessions

CDC Offload and Multithreaded Processing
- Planning for CDC Offload and Multithreaded Processing
- Enabling Offload and Multithreaded Processing for CDC Sessions
- Configuring PowerExchange to Capture Change Data on a Remote System
- Extracting Change Data Captured on a Remote System
- Configuration File Examples for CDC Offload Processing

Appendix A: CDC for z/OS Troubleshooting

CDC for z/OS Troubleshooting Overview
- Problems Related to Propagating Data
- Change Data Capture Components
- Gathering Operating Environment Information

Index
Preface

This guide describes how to configure, implement, and manage PowerExchange change data capture (CDC) environments on z/OS.

This guide applies to the PowerExchange CDC option for the following PowerExchange products:

- PowerExchange® for Adabas®
- PowerExchange for CA Datacom®
- PowerExchange for CA IDMS™
- PowerExchange for DB2® for z/OS®
- PowerExchange for IMS™
- PowerExchange for VSAM

In this guide, the term MVS refers to z/OS operating systems. The term DB2 refers to DB2 for z/OS.

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>
Part I: PowerExchange Change Data Capture Introduction

This part contains the following chapters:

- Change Data Capture Introduction, 2
CHAPTER 1

Change Data Capture Introduction

This chapter includes the following topics:
- PowerExchange CDC Overview, 2
- PowerExchange Components for CDC, 4
- PowerExchange CDC for MVS Data Sources, 6
- PowerExchange Integration with PowerCenter, 8
- CDC Implementation Summary, 9

PowerExchange CDC Overview

PowerExchange Change Data Capture (CDC) provides the ability to capture insert, update, and delete operations performed against z/OS data sources. When capturing changes, PowerExchange uses techniques that help minimize the impact on the performance and availability of the source table, database, or data set.

Sometimes, PowerExchange CDC captures changes in near real time by integrating with the transaction that performs the change. This technique is called synchronous change data capture. In other cases, PowerExchange CDC captures changes from the source database or source relational database logs. This technique is known as asynchronous or log-based change data capture.

PowerExchange can capture changes from the following z/OS data sources:
- Adabas files
- Datacom databases
- DB2 for z/OS tables
- IDMS databases
- IMS databases
- VSAM data sets

PowerExchange uses the following components for change data capture:

**PowerExchange Agent**

On a z/OS system, provides and verifies capture registration information for ECCRs.

**PowerExchange Condense**

Optionally creates condense files that contain a condensed version of the change data in the change stream.
**PowerExchange Environmental Change Capture Routine (ECCR)**

On a z/OS system, captures change data from a data source and passes the captured changes to the PowerExchange Logger for recording.

**PowerExchange Listener**

Manages data maps for nonrelational files and DB2 tables and capture registrations and extraction maps for all data sources. It also handles extraction requests for bulk data and change data.

**PowerExchange Logger**

On a z/OS system, receives captured change data from the ECCRs that are connected to it and stores the change data in log data sets.

**PowerExchange Navigator**

The graphical user interface that you use to define and manage data maps, capture registrations, and extraction maps for the data sources from which you want to extract bulk data or capture change data.

The PowerExchange Navigator runs on Windows. All of the other components run on z/OS.

The PowerExchange architecture is flexible enough to provide for many alternative configurations. You can configure PowerExchange to handle large volumes of change data using multiple instances of PowerExchange CDC components on a single z/OS system. You can capture change data from different source types to multiple PowerExchange Loggers.

The following figure shows an example configuration on a single z/OS system:

![PowerExchange Architecture Diagram](image)

This sample configuration contains the following components:

- Multiple ECCRs writing to a single PowerExchange Logger.
- Multiple instances of the PowerExchange Logger running simultaneously.
Multiple instances of PowerExchange Condense running simultaneously to extract changes from the logs of one PowerExchange Logger.

Multiple instances of the PowerExchange Listener running simultaneously and extracting changes from the logs of a PowerExchange Logger.

One PowerExchange Listener extracting changes from both the logs of a PowerExchange Logger and from condense files. To prevent data loss, the PowerExchange Logger provides dual logging for both the active and archive log data sets.

You can use PowerCenter to propagate the change data to one or more relational or nonrelational targets in your enterprise. PowerExchange CDC works in conjunction with PowerCenter to perform the following tasks:

• Capture change data for sources from which you want to propagate data
• Create an inventory of captured change data that you can use for auditing, recovery, and data propagation
• Provide data transformation capabilities that enable you to propagate changes that are captured from a database on one system to another type of database that is on another system

### PowerExchange Components for CDC

PowerExchange uses a number of components for change data capture. The PowerExchange Navigator runs on Windows. All of the other components run on MVS.

#### PowerExchange Agent

On an MVS system, the PowerExchange Agent provides and verifies capture registration information for ECCRs. The PowerExchange Agent provides capture registration information to the following ECCRs:

• DB2
• IMS Synchronous
• Batch VSAM
• CICS/VSAM

Other ECCRs read capture registrations directly from the CCT data set. For all of the ECCRs, the PowerExchange Agent verifies the capture registration information.

The PowerExchange Agent also manages global queues and data flow among various PowerExchange CDC components.

#### PowerExchange Environmental Change Capture Routine (ECCR)

On an MVS system, the ECCR captures change data from a data source and passes the captured changes to the PowerExchange Logger for recording.

PowerExchange provides an ECCR for each data source. The ECCR captures the changes to the source and passes the captured changes to the PowerExchange Logger for recording.

The mechanism that the ECCR uses to capture the changes depends on the data source. Some ECCRs capture changes synchronously as the changes are occurring. Other ECCRs capture changes asynchronously from database logs or CDC tables.
PowerExchange provides synchronous ECCR for the following sources:

- Datacom
- IMS
- Batch VSAM
- CICS/VSAM

PowerExchange provides asynchronous ECCRs for the following sources:

- Adabas
- Datacom
- DB2
- IDMS
- IMS

With the exception of Datacom, the asynchronous ECCRs are log-based. Datacom is a table-based ECCR.

**PowerExchange Listener**

The PowerExchange Listener manages data maps for nonrelational files and DB2 tables and capture registrations and extraction maps for all data sources. It also handles extraction requests for bulk data and change data.

The PowerExchange Listener maintains these definitions in the appropriate files:

- Data maps in the DATAMAPS file
- Capture registrations in the CCT file
- Extraction maps in the DTLCAMAP file

When you create and manage capture registrations and extraction maps, the PowerExchange Navigator communicates with the PowerExchange Listener on MVS. When you open a registration group or an extraction group, the PowerExchange Navigator communicates with the PowerExchange Listener to read the appropriate capture registrations or extraction maps. The PowerExchange Navigator uses the location specified in the registration and extraction group definitions to determine the PowerExchange Listener to contact.

**PowerExchange Logger**

On an MVS system, the PowerExchange Logger receives captured change data from the ECCRs that are connected to it and stores the change data in log data sets.

The PowerExchange Logger stores all change data that is captured by the ECCRs connected to it. The PowerExchange Logger provides the captured change data to real-time extractions or to a PowerExchange Condense job.

Change data is stored in the PowerExchange Logger active log data set. When the current active log data set is full, the PowerExchange Logger archives the change data to a sequential archive log data set. To prevent data loss, the PowerExchange Logger provides dual logging for both the active and archive log data sets.

**PowerExchange Condense**

PowerExchange Condense creates condense files that contain a condensed version of the changes that were captured by an ECCR and stored by the PowerExchange Logger. PowerExchange Condense processes changes for a single data source. You can run multiple PowerExchange Condense jobs.

When you create a capture registration, specify either full condense or partial condense. For full condense, PowerExchange creates VSAM condense files that contain all successful changes. Full condense processing
rationalizes all insert, update, and delete activity into the final image of the row or record. Transactional integrity is not maintained in full condense files.

For partial condense, PowerExchange creates sequential condense files that contain all successful changes. Transactional integrity is maintained in partial condense files.

When using PowerExchange Condense, you extract the change data from the condense files rather than from the PowerExchange Logger log data sets.

**PowerExchange Navigator**

The PowerExchange Navigator is the graphical user interface that you use to define and manage data maps, capture registrations, and extraction maps for the data sources from which you want to extract bulk data or capture change data.

PowerExchange uses capture registrations to determine what sources are eligible for CDC. You use the PowerExchange Navigator to create and manage capture registrations and extraction maps for change data capture sources. Extraction maps can be imported into PowerCenter for use in extracting the captured change data.

For more information about creating and managing capture registrations and extraction maps, see the *PowerExchange Navigator User Guide*.

**PowerExchange CDC for MVS Data Sources**

PowerExchange provides an Environmental Change Capture Routine (ECCR) for each data source. An ECCR captures changes from a data source and passes the captured change data to the PowerExchange Logger for logging.

**Restriction:** For any data source type, the maximum length of a record for which PowerExchange can capture and process change data is 32 KB. To determine if additional limits apply to the maximum length record for Datacom sources, see your CA Datacom documentation.

**Adabas Change Data Capture**

PowerExchange for Adabas CDC reads an Adabas Protection Log (PLOG) to capture change data. When Adabas switches to a new PLOG, PowerExchange for Adabas CDC records the new PLOG data set name in the PLOG catalog (PCAT).

The Adabas ECCR runs in a separate address space. It periodically checks the PCAT for new PLOGs from which to capture changes and passes any changes from those logs to the PowerExchange Logger for recording.

Each Adabas ECCR captures changes for a single Adabas database. If you have multiple Adabas databases, run an Adabas ECCR for each Adabas database.

**Datacom Change Data Capture**

PowerExchange for Datacom CDC can capture changes synchronously while the changes are occurring in the Datacom Multi-User Facility (MUF) address space. Alternatively, if you have Datacom Release 11 SP4 or later, you can capture changes asynchronously from Datacom CDC tables.
Table-Based CDC
PowerExchange for Datacom table-based CDC captures changes asynchronously from Datacom CDC tables. The table-based ECCR listens for changes to the CDC tables and writes the change data to the PowerExchange Logger.

Synchronous CDC
PowerExchange for Datacom synchronous CDC captures changes while the changes are occurring in the Datacom Multi-User Facility (MUF) address space.

You can configure the Datacom synchronous ECCR to use the direct-log-write method. This method uses the following components:

Datacom Change Collector
- Runs in the Datacom MUF address space, captures changes as they occur, and passes them to the PowerExchange Logger for recording.

Datacom Change Controller
- Runs in a separate address space and manages the capture registrations for the Datacom Change Collector.

Informatica recommends the direct-log-write method because it has the following advantages:
- It reduces the latency between the time when the changes occur and the time when changes can be extracted.
- It reduces the operational complexity and system resource usage to capture change data.

For compatibility with earlier configurations of Datacom CDC, configure the Datacom synchronous ECCR to store the changes in a data space before they are passed to the PowerExchange Logger. This method uses the following components:

Datacom Change Collector
- Runs in the Datacom MUF address space, captures changes as they occur, and moves them into the dataspace created by the Datacom Change Controller.

Datacom Change Controller
- Runs in a separate address space and creates the dataspace into which the Datacom Change Collector moves the change data.

Datacom Log Feeder
- Runs in a separate address space and reads the captured change data from the data space created by the Datacom Change Controller. The Datacom Log Feeder passes the change data to the PowerExchange Logger for recording.

DB2 for z/OS Change Data Capture
PowerExchange for DB2 CDC uses the DB2 Instrumentation Facility Interface (IFI) to capture change data from DB2 logs. The DB2 ECCR runs in a separate address space and issues IFI 306 calls to DB2 to retrieve the changes. DB2 reads the DB2 logs and passes the data to the DB2 ECCR. The DB2 ECCR passes the change data to the PowerExchange Logger for recording.

A single DB2 ECCR can process change data for all DB2 subsystems in a DB2 data-sharing group.

IDMS Change Data Capture
PowerExchange for IDMS CDC can capture changes asynchronously from IDMS logs. For IDMS asynchronous change data capture, PowerExchange uses the IDMS log-based ECCR. The IDMS log-based ECCR runs in a
separate address space. It reads IDMS archive logs to capture change data. When IDMS archives an active journal, PowerExchange for IDMS CDC records the new archive log in the Log Catalog. The IDMS log-based ECCR periodically checks the Log Catalog for new archive logs from which to capture changes and passes any changes from those logs to the PowerExchange Logger for recording.

**IMS Change Data Capture**

PowerExchange for IMS CDC can captures changes synchronously in the IMS region or asynchronously from IMS logs.

The IMS synchronous ECCR runs in the IMS region. It captures changes as they occur and passes the changes to the PowerExchange Logger for recording. The IMS synchronous ECCR captures changes in the following IMS environments:

- DBCTL
- DB/DC
- Batch

The IMS log-based ECCR runs in a separate address space. It periodically checks the IMS RECON data sets for new system log data sets (SLDS) from which to capture changes and passes any changes from those logs to the PowerExchange Logger for recording.

**VSAM Change Data Capture**

PowerExchange for VSAM CDC synchronously captures change data for VSAM data sets from batch jobs and from CICS regions.

The Batch VSAM ECCR runs in the batch job address space. It captures changes as they occur using a VSAM JRNAD exit and passes the changes to the PowerExchange Logger for recording.

The CICS/VSAM ECCR runs in the CICS region. It captures changes as they occur using CICS global user exits (GLUE) and task-related user exits (TRUE) and passes the changes to the PowerExchange Logger for recording.

**PowerExchange Integration with PowerCenter**

PowerCenter provides transformation and data cleansing capabilities. After you capture change data, use PowerCenter in conjunction with PowerExchange to extract the change data and apply it to target tables or files.

PowerExchange Client for PowerCenter (PWXPC) provides integration between PowerExchange and PowerCenter. Using PWXPC, extract change data from both the PowerExchange Logger log data sets and condense files.

Alternatively, use PowerExchange ODBC drivers in PowerCenter to extract change data from PowerExchange. However, Informatica recommends using PWXPC instead of PowerExchange ODBC drivers because PWXPC has more functionality, better performance, and superior recovery and restart.
The following figure shows the data flow for changes made to MVS data sources. In this data flow the PowerExchange CDC components capture the changes and PowerCenter extracts the changes and applies them to the target:

For more information about using PowerCenter to extract change data from PowerExchange, see PowerExchange Interfaces for PowerCenter.

### CDC Implementation Summary

After you install PowerExchange, perform the following high-level steps to implement change data capture for a source and to extract the captured changes using PowerCenter. References to the appropriate sections of this book and to related PowerExchange and PowerCenter documentation are provided.

<table>
<thead>
<tr>
<th>Step</th>
<th>Task</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Configure the PowerExchange Listener.</td>
<td>- PowerExchange Bulk Data Movement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- “Configuring the PowerExchange Listener for CDC” on page 12</td>
</tr>
<tr>
<td>2</td>
<td>Start the PowerExchange Listener.</td>
<td>“Managing the PowerExchange Listener” on page 18</td>
</tr>
<tr>
<td>3</td>
<td>Configure the PowerExchange Agent.</td>
<td>“Configuring the PowerExchange Agent ” on page 22</td>
</tr>
<tr>
<td>4</td>
<td>Start the PowerExchange Agent.</td>
<td>“Managing the PowerExchange Agent” on page 31</td>
</tr>
<tr>
<td>Step</td>
<td>Task</td>
<td>References</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>5</td>
<td>Configure the PowerExchange Logger.</td>
<td>“Configuring the PowerExchange Logger for MVS” on page 39</td>
</tr>
<tr>
<td>6</td>
<td>Start the PowerExchange Logger.</td>
<td>“Managing Log and Restart Data Sets” on page 52</td>
</tr>
<tr>
<td>7</td>
<td>Configure the appropriate PowerExchange ECCR for the data source.</td>
<td>“CDC Sources Configuration and Management” on page 104</td>
</tr>
<tr>
<td></td>
<td><strong>Define data sources for CDC</strong></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Create a data map using the PowerExchange Navigator. This step is</td>
<td>PowerExchange Navigator Guide</td>
</tr>
<tr>
<td></td>
<td>required for nonrelational sources.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>For DB2 sources that require user-defined fields and expressions,</td>
<td>PowerExchange Navigator Guide</td>
</tr>
<tr>
<td></td>
<td>create a data map using the PowerExchange Navigator.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Define and activate capture registrations and extraction maps for</td>
<td>PowerExchange Navigator Guide</td>
</tr>
<tr>
<td></td>
<td>the data source using the PowerExchange Navigator.</td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Materialize targets and start capturing changes</strong></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Materialize the target from the source.</td>
<td>PowerExchange Bulk Data Movement Guide</td>
</tr>
<tr>
<td>12</td>
<td>Establish a starting point for the extraction.</td>
<td>“Change Data Extraction” on page 226</td>
</tr>
<tr>
<td>13</td>
<td>Start the ECCR.</td>
<td>“CDC Sources Configuration and Management” on page 104</td>
</tr>
<tr>
<td>14</td>
<td>(Optional) Configure PowerExchange Condense.</td>
<td>“Configuring PowerExchange Condense” on page 77</td>
</tr>
<tr>
<td>15</td>
<td>(Optional) Start PowerExchange Condense.</td>
<td>“Starting and Stopping PowerExchange Condense” on page 96</td>
</tr>
<tr>
<td></td>
<td><strong>Extract change data</strong></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Prepare and extract change data using PowerCenter.</td>
<td>- PowerExchange Interfaces for PowerCenter</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PowerCenter Designer Guide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- PowerCenter Workflow Basics Guide</td>
</tr>
</tbody>
</table>
Part II: CDC Components
Configuration and Management

This part contains the following chapters:

- PowerExchange Listener, 12
- PowerExchange Agent, 20
- PowerExchange Logger for MVS, 37
- PowerExchange Condense, 77
PowerExchange Listener Overview

The PowerExchange Listener is a component of PowerExchange CDC that provides services to other PowerExchange CDC components and to PowerExchange users. These services include:

- Storing and managing data maps, capture registrations, and extraction maps for MVS sources registered for CDC
- Providing new or modified capture registrations to the PowerExchange Agent
- Providing captured change data to PowerCenter extractions and to the PowerExchange Navigator database row tests

The PowerExchange Listener interacts with the following PowerExchange CDC components:

- PowerExchange Navigator
- PowerExchange Agent
- PowerExchange Logger

Configuring the PowerExchange Listener for CDC

The PowerExchange Listener manages capture registrations and extraction maps for change data capture sources. You also connect to the PowerExchange Listener to extract the captured change data.

Prior to using change data capture on MVS, configure the following:

- The PowerExchange Listener JCL on the MVS system where change data, capture registrations, and extraction maps reside
- The DBMOVER configuration parameters for the PowerExchange Listener on MVS
Configuring the PowerExchange Listener JCL

Change data capture requires additional DD statements in the PowerExchange Listener JCL. If you selected change data capture options during the installation process, PowerExchange customizes the PowerExchange Listener JCL to include these DD statements.

Verify that the PowerExchange Listener JCL is correct. If necessary, correct the JCL and recycle the PowerExchange Listener.

The following table lists the DD statements required for CDC:

<table>
<thead>
<tr>
<th>DD Statement Name</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLAMCPR</td>
<td>Yes</td>
<td>This DD statement points to the CCT VSAM data set, which contains the capture registrations.</td>
</tr>
<tr>
<td>DTLCACDC</td>
<td>No</td>
<td>This DD statement points to the CDCT VSAM data set, which contains condense file information. This DD statement is only necessary if you are using PowerExchange Condense.</td>
</tr>
<tr>
<td>DTLCACDE</td>
<td>Yes</td>
<td>This DD statement points to the CDEP VSAM data set, which contains the application names. This DD statement is necessary to perform database row tests from the PowerExchange Navigator and if extracting data using PowerExchange ODBC connections in PowerCenter.</td>
</tr>
<tr>
<td>DTLCAMAP</td>
<td>Yes</td>
<td>This DD statement points to the DTLCAMAP VSAM data set, which contains the extraction maps.</td>
</tr>
<tr>
<td>EDMPARMS</td>
<td>Yes</td>
<td>This DD statement points to the USERLIB library, which contains the EDMSDIR module options used to connect to the appropriate PowerExchange Agent and Logger.</td>
</tr>
</tbody>
</table>

Configuring CAPI_CONNECTION Statements

To extract captured change data using real-time extraction mode, the PowerExchange Listener invokes the Log-Read API to connect to the PowerExchange Logger.

Change the DBMOVER configuration parameters used by the PowerExchange Listener on the MVS system where the change data is stored to include UOW Cleanser and Log-Read API CAPI_CONNECTION statements. Recycle the PowerExchange Listener to activate the changes in the DBMOVER configuration parameters.

LRAP CAPI_CONNECTION Parameters

The LRAP CAPI_CONNECTION statement specifies the Consumer API (CAPI) parameters needed for the Log Read API (LRAPI) component of the PowerExchange Logger for MVS. The LRAPI connects to the PowerExchange Logger to read change data for the address space that is extracting that data, such as the PowerExchange Listener address space.
Syntax:

```plaintext
CAPI_CONNECTION=(
    [DLLTRACE=trace_id,]
    NAME=name,
    [TRACE=trace,]
    TYPE=[LRAP,
    AGENT=agent_id,
    [EOF={Y|N},]
    LOG=logger_id,
    [UIDFMT={ALL|CONN|CORR|CTYPE|PLAN|UID}],
}
)
```

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=[LRAP, ... ]**

Required. Type of CAPI_CONNECTION statement. For the LRAPI, this value must be LRAP.

**AGENT=agent_id**

Required. PowerExchange Agent ID, which must match the value specified in the AGENTID parameter of the EDMMSDIR module. PowerExchange reads the EDMMSDIR module from the EDMPARMS DD statement, or, if not specified, from the STEPLIB or JOBLIB DD statement. Maximum length is four alphanumeric characters.

**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 EOL is reached.
- **Y**: PowerExchange stops change data extractions when EOL is reached.

Because this parameter affects all users of the LRAP 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.

**LOG=logger_id**

Required. PowerExchange Logger ID, which must match the value specified in the LOGGER parameter of the EDMSDIR module.

Maximum length is four alphanumeric characters.

**UIDFMT=[ALL|CONN|CORR|CTYPE|PLAN|UID]**

For DB2 for z/OS data sources, controls the data that PowerExchange returns in the DTL__CAPXUSER field.

Enter one of the following options:

- **ALL.** Requests the information for all of the other options. PowerExchange provides this information in a colon-delimited list in the following format:
  
  UID:PLAN:CORR:CONN:CTYPE

- **CONN.** DB2 connection identifier when the change was made.

- **CORR.** DB2 correlation identifier when the change was made.

- **CTYPE.** DB2 connection type when the change was made.

- **PLAN.** DB2 plan name used when the change was made.

- **UID.** User ID that made the change.

*Restriction:* You can specify only one option. If you need more than one option, specify ALL.

Default is UID.

**UOWC CAPI_CONNECTION Parameters**

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.

<table>
<thead>
<tr>
<th>Data Sources:</th>
<th>Required: Yes for the noted data sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 for i5/OS sources</td>
<td></td>
</tr>
<tr>
<td>Oracle LogMiner CDC sources</td>
<td></td>
</tr>
<tr>
<td>z/OS CDC sources</td>
<td></td>
</tr>
<tr>
<td>Related Statements:</td>
<td></td>
</tr>
<tr>
<td>AS4J</td>
<td></td>
</tr>
<tr>
<td>CAPI_CONNECTION for i5/OS</td>
<td></td>
</tr>
<tr>
<td>LRAP</td>
<td></td>
</tr>
<tr>
<td>Related Statements:</td>
<td></td>
</tr>
<tr>
<td>CAPI_CONNECTION for z/OS</td>
<td></td>
</tr>
<tr>
<td>ORCL</td>
<td></td>
</tr>
<tr>
<td>Related Statements:</td>
<td></td>
</tr>
<tr>
<td>CAPI_CONNECTION for Oracle</td>
<td></td>
</tr>
</tbody>
</table>

**Syntax:**

```plaintext
CAPI_CONNECTION=(
    [DILLTRACE=trace_id,]
    [TRACETYPE=name,]
    [TYPE=UOWC,]
    [CAPthere=trace,]
    NAME=name,
)
```

**Configuring the PowerExchange Listener for CDC** 15
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 z/OS CDC sources, enter a value from 8 through 32760. Default is 18452.
- 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.

**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.Thhmmss.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=nnnnnn

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.

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.

**SPACETYPE={BLK|TRK|CYL}**

On z/OS, the type of space units that the UOW Cleanser uses to allocate UOW spill files.

Enter one of the following options:

- **BLK.** Use blocks.
- **CYL.** Use cylinders.
- **TRK.** Use tracks.

Default is BLK.

**STORCLAS=storage_class**

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

**UNIT=unit**

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

---

**Managing the PowerExchange Listener**

You can control certain aspects of PowerExchange Listener processing by using commands.
Starting the PowerExchange Listener

To start the PowerExchange Listener, issue the MVS START command with the name of the started task. For example:

```
START listener
```

Start the PowerExchange Listener prior to starting any other PowerExchange CDC component address spaces including the PowerExchange Agent.

You can also run the PowerExchange Listener as a batch job. However, because it is a long-running task, using a MVS started task is more appropriate.

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

Stopping the PowerExchange Listener

To stop the PowerExchange Listener, issue one of the following commands:

- The z/OS MODIFY command with CLOSE or CLOSE FORCE. Use the following syntax:
  
  ```
  F listener_task_name,CLOSE
  
  - CLOSE causes the PowerExchange Listener to stop after all user subtasks complete, including bulk data movement subtasks and CDC subtasks.
  
  - CLOSE FORCE causes the PowerExchange Listener to wait 30 seconds for active tasks to complete and then stops any remaining active tasks before shutting down.
  
- The z/OS STOP (P) command. This command has the same effect as the CLOSE FORCE command.

- The pwxcmd close or closeforce command. Issue these commands from a Linux, UNIX, or Windows system to a PowerExchange Listener running on a z/OS system.

Controlling PowerExchange Listener Tasks

You can use commands to display information about and stop PowerExchange Listener tasks. Enter PowerExchange Listener commands by using the z/OS MODIFY (F) command, as follows:

```
F listener_task_name,COMMAND
```

The following table briefly describes the PowerExchange Listener commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISTTASK</td>
<td>Lists all active PowerExchange Listener tasks.</td>
</tr>
<tr>
<td>STOPTASK</td>
<td>Stops a specified PowerExchange Listener task.</td>
</tr>
</tbody>
</table>

You can also issue pwxcmd listtask and stoptask commands from a Linux, UNIX, or Windows system to a PowerExchange Listener running on a z/OS system.

The pwxcmd listtask command and the LISTTASK command entered with the z/OS MODIFY command results in the same output.
PowerExchange Agent Overview

The PowerExchange Agent is a component of PowerExchange CDC that provides services to other PowerExchange CDC components and runs in a separate address space. You can control certain aspects of PowerExchange Agent processing by issuing commands from the MVS system console.

The PowerExchange Agent has the following characteristics:

- The PowerExchange Agent must run as a MVS started task. It cannot run as a batch job.
- The PowerExchange Agent interacts with the following PowerExchange CDC components:
  - PowerExchange Listener
  - Environmental Change Capture Routines (ECCR)
  - PowerExchange Logger
- The PowerExchange Agent connects to a single PowerExchange Listener.
- Start the PowerExchange Agent after the PowerExchange Listener. By default, the PowerExchange Agent acquires capture registration information from the PowerExchange Listener.
- Start the PowerExchange Agent prior to the PowerExchange Logger and any other PowerExchange CDC components. The PowerExchange Agent provides services to other PowerExchange CDC components. These services include:
  - Getting and managing global queues for other PowerExchange CDC components
  - Getting new or modified capture registrations from the PowerExchange Listener
  - Managing data flow between PowerExchange CDC components in different address spaces
  - Managing requests from ECCRs for capture registration information
- Providing access to authorized users
- Providing a common message log

- The PowerExchange Agent uses the AgentID you specify in the EDMSCTL parameters to create a MVS subsystem. You use the AgentID to communicate with the PowerExchange Agent address space.

Running Multiple Instances of the PowerExchange Agent

You can run multiple instances of the PowerExchange Agent simultaneously on a single MVS system. For example, you may want to run separate instances for your test and production environments. Ideally, you should create a separate test and production environment and use a different PowerExchange Agent for each environment.

Use the following rules and guidelines when you run multiple instances of the PowerExchange Agent:

- A PowerExchange Agent can only use one PowerExchange repository.
- Multiple PowerExchange Agents can share the same PowerExchange repository.
- The Batch VSAM ECCR status specified in AGENTCTL parameters affects all PowerExchange Agents on a single MVS system. If you activate or deactivate the Batch VSAM ECCR for one PowerExchange Agent, the status changes for all PowerExchange Agents on the same MVS system.
- The AgentID specified in the AGENCTL parameters is defined as an MVS subsystem. To use the same AgentID for different PowerExchange Agents, each PowerExchange Agent must reside on a different MVS system.

Configuring MVS for the PowerExchange Agent

To optimize the MVS configuration for the PowerExchange Agent, consider increasing the following:

- Number of linkage indexes.
- Number of common data spaces.

Linkage Indexes and Common Data Spaces

You may want to increase the values of the NSYSLX and MAXCAD parameters in the EASYSxx member in SYS1.PARMLIB based on the following considerations:

- Each PowerExchange Agent requires two linkage index entries. During warm start processing, the PowerExchange Agent reuses the linkage index entries. During cold start processing, two new linkage index entries are used. Consider increasing the NSYSLX parameter of the EASYSxx member in SYS1.PARMLIB.

- Each PowerExchange Agent uses one common data space. If you use the SHUTDOWN command with the COMPLETELY option to stop the PowerExchange Agent, PowerExchange CDC deletes the data space. However, if you do not specify the COMPLETELY option, the data space persists.

When you restart the PowerExchange Agent, the agent reuses the data space if it exists, unless you are performing a cold start. Consider increasing the MAXCAD parameter of the EASYSxx member in SYS1.PARMLIB to enable increased usage of common data spaces.

If you change either parameter, you must IPL the MVS system for the changes to take affect.
Global Enqueue for PowerExchange CDC

PowerExchange CDC issues only SYSTEMS-level enqueues for serializing across multiple MVS systems. To use PowerExchange CDC on multiple MVS systems, you must make use of a global serialization product such as GRS or MIM to propagate these enqueues.

If you are using a cross-system serialization product that requires you to specifically define the enqueues that need to be propagated globally, you need to know the QNAMEs issued by PowerExchange CDC.

**Note:** The DB2 ECCR uses a SYSTEMS-level enqueue to prevent multiple instances of the same ECCR running. The QNAME is DB2CAPT. The RNAME is an eight-byte field, the NAME= value from the DB2 ECCR REPL2CTL control file statement CA. The SYSTEMS enqueue exists for the life of the ECCR execution.

You might need to include this information in the options for your cross-system serialization product to ensure these enqueues are properly handled.

Configuring the PowerExchange Agent

The PowerExchange Agent has several options and parameters that control unique aspects of its operation:

- EDMSDIR module options
- AGENTCTL parameters
- AGENTREP parameters

When you install PowerExchange, these options and parameters are configured with defaults and values you provide in the MVS Installation Assistant. Prior to starting any PowerExchange CDC components, review the PowerExchange Agent options and parameters to ensure they are appropriate for your installation.

EDMSDIR Module Options

You set default configuration options for the EDMSDIR module during the installation of PowerExchange.

The XICDC600 member of the RUNLIB library contains the EDMSDIR module options created during installation process as well as the necessary JCL to assemble and link-edit the EDMSDIR module.

The installation process places the EDMSDIR module into the USERLIB library.

When you execute the SETUPCC1 member of the RUNLIB library, the USERLIB data set is created.

The EDMSDIR module options are in effect for any PowerExchange CDC component that points to the USERLIB library containing the EDMSDIR module.
The following table describes the EDMSDIR module options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default Value</th>
<th>Valid Values</th>
</tr>
</thead>
</table>
| AGENTID | Specifies the name of the default PowerExchange Agent | EDMA | - Four characters, beginning with a letter, #, @, or $  
- A value that does not conflict with an existing MVS subsystem  
**Note:** The value of AGENTID and the LOGGER cannot be the same. |
| CCERR | Specifies what action to take when a DB2, IMS synchronous, batch VSAM, or CICS/VSAM ECCR is unable to capture changes for a data source. | CONT | - CONT stops capture but lets the job continue; any changes to the data resource are not captured. If a /STOP subsys is issued from IMS and CCERR=CONT, work continues, but the data to be captured is not logged.  
- ABEND abnormally ends the job; the transaction does not update the resource. If CCERR=ABEND, the BMP or MPP terminates abnormally, but the control region continues to function.  
**Notes:**  
- With a value of ABEND, if the CICS/VSAM ECCR encounters a serious error, or abnormally ends during initialization, the ECCR immediately terminates the CICS region to prevent loss of data.  
- If the PowerExchange Logger fails or is shut down, which means the ECCR can not pass any updates to the Logger, the CICS/VSAM ECCR causes the transaction performing the updates to be abended with abend code ASP7 at the transaction syncpoint. Thus, no updates occur to files that are registered for capture, and no data is lost.  
- Similarly, if the registration status of a file cannot be determined when the file is being opened, the CICS/VSAM ECCR abends any transaction performing updates to the files to be abended, typically with abend code ASP7 at the transaction syncpoint. For example, this situation can occur when the PowerExchange Agent is down or repository access through the PowerExchange Agent has been stopped. Again, no updates occur to files that might be registered for capture, and no data is lost. |
| CENTURY | Specifies whether to include the century when the PowerExchange CDC components display the date | Y | - Y displays the century.  
- N displays the date without the century. |
| DATE | Specifies the date format that the PowerExchange CDC components display, for example, in messages. | (MDY,/| The first value indicates the order of the date elements:  
- YMD indicates YY/MM/DD  
- MDY indicates MM/DD/YY  
- DMY indicates DD/MM/YY  
The second value is the date separator. The separator can be any character. |
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default Value</th>
<th>Valid Values</th>
</tr>
</thead>
</table>
| ESLLIB   | Specifies the data sets to be concatenated to existing DFSESL DD statements in your IMS dependent region or IMS control region. This option is required for IMS synchronous ECCR online environments. If a DFSESL DD statement does not already exist in your dependent region or control region, PowerExchange allocates one for you. For more information about the DFSESL DD statement, see the IBM IMS installation procedures. | N/A           | - Specify the appropriate data set or sets, enclosed within parentheses.  
- If you specify multiple data sets with this parameter, separate them with commas.  
- Specify up to five data sets.                                                                 |
| LOGGER   | Specifies the name of the default PowerExchange Logger. You can specify only one instance of the PowerExchange Logger with this parameter. Consequently, if you use multiple PowerExchange Loggers you must have a separate EDMSDIR for each instance of the PowerExchange Logger. Because you cannot rename EDMSDIR, you must allocate a separate user library, your.USERLIB, for each copy of EDMSDIR. | EDML          | - Four characters, beginning with a letter, #, @, or $  
- A value that does not conflict with an existing MVS subsystem  
Note: The value of LOGGER and AGENTID cannot be the same. |
| LOGRGPR  | Specifies whether the PowerExchange Logger is configured for Post-Log Merge | N             | - Y specifies the Post-Log Merge configuration.  
- N specifies that the Post-Log Merge feature is not used.                                                                                   |
| SYSOUT   | Specifies the default SYSOUT class that any dynamically allocated SYSOUT data sets use. | Any valid SYSOUT class. |                                                                                                                                                   |
| TIME     | Specifies the time format that the PowerExchange CDC components display, for example, in messages. | (24,:)        | The first value indicates the hour format:  
- 24 indicates a 24-hour format, as in military time.  
- 12 indicates a 12-hour format.  
The second value is the time separator. The separator can be any character.                                                              |

**Related Topics:**

- “Using Post-Log Merge” on page 69

**Configuring the EDMSDIR Module Options**

You can modify the values for these EDMSDIR module options after installation by performing the following procedure.
To configure the EDMSDIRM module options:

1. Customize and run the JCL in member XICDC600 of the RUNLIB library.
2. Stop any of the following PowerExchange CDC components that specify the USERLIB library containing the EDMSDIR module:
   - PowerExchange Listener
   - PowerExchange Agent
   - PowerExchange Logger
   - Environmental Change Capture Routines (ECCRs)
   - PowerExchange Condense jobs
3. Restart the PowerExchange CDC components that you just stopped.

**RELATED TOPICS:**

- “EDMSDIR Module Options” on page 22

**Configuring AGENTCTL Parameters**

The PowerExchange installation process generates the AGENTCTL parameters member containing the PowerExchange Agent AGENTCTL parameters. The EDMSCTL DD statement in the PowerExchange Agent JCL points to the AGENTCTL parameters.

You can modify the AGENTCTL parameters after installation by changing the AGENTCTL member in the RUNLIB library. If you do not have an AGENTCTL member in the RUNLIB library, you can determine the correct member by checking the EDMSCTL DD statement in the PowerExchange Agent JCL.

**Note:** You must restart the PowerExchange Agent for changes to AGENTCTL to take effect.

The following table describes the parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgentID</td>
<td>Required. The name of the PowerExchange Agent. You can use the same AgentID for different PowerExchange Agents, provided they are on different MVS systems. The value that you specify must match the value of the AGENTID parameter in the EDMSDIR module.</td>
<td>EDMA</td>
<td>- Four characters, beginning with a letter, #, @, or $. - A value that does not conflict with an existing MVS subsystem.</td>
</tr>
<tr>
<td>CCVACTIVE</td>
<td>Optional. Specifies whether to activate the Batch VSAM ECCR during the startup of the PowerExchange Agent.</td>
<td>No</td>
<td>- Yes. The Batch VSAM ECCR is activated during startup. - No. The Batch VSAM ECCR is not be activated during startup.</td>
</tr>
<tr>
<td>CmdAuthCheck</td>
<td>Optional. Specifies whether to check authorization by issuing a RACROUTE authorization macro when a PowerExchange Agent command is issued.</td>
<td>No</td>
<td>- Yes. The PowerExchange Agent checks authorization. - No. The PowerExchange Agent does not check authorization.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Default Value</td>
<td>Valid Values</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| CmdPrefix     | Optional. The MVS command prefix to use for all PowerExchange Agent commands.  | Value of AgentID parameter. | One to eight characters, beginning with a letter or a symbol. Valid symbols are: 
< > ( + | & | ' | = | - | / | % | _ | > ? | : | # | @ | | |
A value that does not conflict with existing MVS or PowerExchange Agent commands. |
| InitAuthCheck | Optional. Whether to check authorization by issuing a RACROUTE authorization macro whenever anyone makes a request to initialize a PowerExchange Agent service. | No              | - Yes. The PowerExchange Agent checks authorization.  
- No. The PowerExchange Agent does not check authorization. |
| LogBuffLimit  | Optional. The amount of data space storage to allocate as an integration area for EDMSLOG messages. You can specify the size of the space in terms of number of messages, with each message allowing 216 bytes. The message log is stored in a data space and uses no common storage. | 2000            | A number from 1000 through 10000. |
| LogClass      | Required. The EDMSLOG SYSOUT class.                                         | Any valid SYSOUT class. |  |
| LogHold       | Optional. Specifies whether the EDMSLOG SYSOUT data is allocated with HOLD=YES. | No              | - Yes. The data is held.  
- No. The data is not held. |
| LogLimit      | Optional. The EDMSLOG line limit. When the PowerExchange Agent determines that the limit has been reached, the agent allocates a new log. | 10000           | A number from 5000 through 100000. |
| Refreshsscvt  | Optional. Causes the system to build a new SSCVT. The parameter specifies the existing SSCVT address that you wish to refresh because the existing SSCVT address is no longer usable. Use this parameter if all of the following are true: 
- You received message PWXEDM172020E.  
- The STARTUP parameter is set to COLD.  
- You do not need to IPL as a result of the failure. | An 8-character hexadecimal address that you obtain from message PWXEDM172020E |  |
| RepositoryDSN | Required. The PowerExchange Agent repository data set name of either the AGENTREP data set or CCT data set. | A valid cataloged data set name. |  |
| RepositoryMode | Required. The type of repository being read.                               |                 |  |
### Configuring AGENTREP Parameters

The AGENTREP data set, created during the installation of PowerExchange, specifies PowerExchange Agent parameters related to control of the capture registration subtask.

**Note:** The AGENTREP data set is created as a sequential data set and should not be changed to a PDS member.

The AGENTREP data set name is specified in the RepositoryDSN parameter in the AGENTCTL parameters. For example:

```
RepositoryDSN=hlq.AGENTREP
```

The *hlq* variable is the PowerExchange high-level qualifier specified in the MVS Installation Assistant at installation time.

You can also specify data set name of the PowerExchange CCT data set in the RepositoryDSN parameter. For example:

```
RepositoryDSN=hlqvs.CCT
```

The *hlqvs* is the PowerExchange high-level qualifier for VSAM specified in the MVS Installation Assistance at installation time.

**Tip:** For improved performance and resource usage, Informatica recommends using the AGENTREP data set rather than the CCT data set as the PowerExchange Agent repository.

- When you use the AGENTREP data set as the PowerExchange Agent repository, the PowerExchange Agent only retrieves the capture registrations from the PowerExchange Listener, during each registration update interval, when there are changes.
- When you use the CCT data set as the PowerExchange Agent repository, the PowerExchange Agent must read the entire CCT during each registration update interval to determine if there are any changes. This activity results in unnecessary I/O activity and CPU overhead in the PowerExchange Agent address space.

---

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Startup</td>
<td>Optional. Whether, during startup, the PowerExchange Agent creates a new data space or uses an existing data space, if one exists.</td>
<td>- Warm. Ruses an existing data space if one exists. - Cold. Create a new data space.</td>
<td></td>
</tr>
<tr>
<td>TaskLimit</td>
<td>Optional. The amount of data space storage used as an integration area for concurrent PowerExchange Agent tasks. This limit is specified in terms of the maximum number of concurrent task control blocks (TCBs) that can request services from the PowerExchange Agent, allowing 128 bytes per control block.</td>
<td>500</td>
<td>A number from 150 through 1500.</td>
</tr>
</tbody>
</table>
The following table shows the AGENTREP Agent Commands:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackToBackDelay</td>
<td>0</td>
<td>Determines the minimum time interval between update notifications. Can be used to cut down or eliminate the number of registration change messages displayed environments where repositories are modified frequently. The default is 0, which does not suppress any messages. Where messages are suppressed the Repository Display command can be used to display latest change information.</td>
</tr>
<tr>
<td>Cache1</td>
<td>None</td>
<td>Copy 1 of the sequential cache data set.</td>
</tr>
<tr>
<td>Cache2</td>
<td>None</td>
<td>Copy 2 of the sequential cache data set.</td>
</tr>
<tr>
<td>Location</td>
<td>None</td>
<td>The name of the PowerExchange Listener retrieved from the PowerExchange configuration member.</td>
</tr>
<tr>
<td>RestartInterval</td>
<td>60</td>
<td>Interval specified in number of UpdateIntervals stating the frequency for the agent subtask, which interrogates the PowerExchange Listener for capture registration changes, to be restarted. Restarting effectively frees memory, which has been allocated to the TCP/IP layer.</td>
</tr>
<tr>
<td>UpdateInterval</td>
<td>1</td>
<td>Frequency in minutes of checking for PowerExchange updates. Messages appear in the agent when PowerExchange checks for changes.</td>
</tr>
</tbody>
</table>

Customizing the PowerExchange Agent JCL

The PowerExchange Agent must run as a started task. After you have customized the JCL for your installation, you must copy the JCL to a system procedure library (PROCLIB) for started tasks.

PowerExchange provides sample JCL for the PowerExchange Agent. Cleanup job XIZZZ998 in the RUNLIB library, which runs as a part of the PowerExchange installation process, moves the PowerExchange Agent JCL to the PowerExchange PROCLIB library.

The member name in the PROCLIB library is the value you specify in the Agent / Logger Prefix field in the MVS Installation Assistant followed by the letter A. For example, by default the Agent / Logger Prefix field is PWX. Therefore, the default member name for the PowerExchange Agent JCL in the PROCLIB library is PWXA.
The following table describes the JCL Statements and JCL parameters for the PowerExchange Agent:

<table>
<thead>
<tr>
<th>JCL Statements and Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The PGM= of the EXEC statement must specify the PowerExchange Agent module EDMSTART.</td>
</tr>
<tr>
<td>STARTUP</td>
<td>The STARTUP symbolic parameter that determines whether the PowerExchange Agent should start WARM or COLD. If you start the agent without any parameters, the agent starts with all of the options you specified during installation. The STARTUP parameter also enables you to override the option you specified during installation for starting the agent as either WARM or COLD. In a WARM start, the PowerExchange Agent uses an existing agent environment, assuming that one exists. Conversely, in a COLD start, the agent creates a new agent environment and starts as if for the first time. Use the following syntax to start the PowerExchange Agent with all of the options you chose during installation: <code>START agent_proc_name</code> The variable <code>agent_proc_name</code> refers to the name you assigned to the PowerExchange Agent procedure at the time of installation. Use the following syntax to start the PowerExchange Agent with all of the options you chose during installation, except for the option of whether to start COLD or WARM: `START agent_proc_name,STARTUP=(COLD</td>
</tr>
<tr>
<td>STEPLIB or JOBLIB DD</td>
<td>Include the PowerExchange load libraries, hlq.LOAD and hlq.LOADLIB. This statement is required even if you specify the load library in the LNKLS1 concatenation. The PowerExchange Agent loads some modules from the STEPLIB or JOBLIB.</td>
</tr>
<tr>
<td>EDMPARMS DD</td>
<td>The name of the user library, YOUR.USERLIB, that contains the EDMSDIR options module associated with the PowerExchange Agent. If you do not include an EDMPARMS DD statement, or if you specify a library that does not contain the options module, PowerExchange uses the STEPLIB concatenation to obtain the configuration options.</td>
</tr>
<tr>
<td>EDMSCTL DD</td>
<td>The data set containing the PowerExchange Agent's startup parameters. Informatica Corporation recommends that you also include the FREE=CLOSE statement so that this data set is deallocated after it is read.</td>
</tr>
<tr>
<td>SYSPRINT DD</td>
<td>The output data set for MVS system messages.</td>
</tr>
</tbody>
</table>

Sample JCL Procedure for the PowerExchange Agent

PowerExchange provides sample JCL for the PowerExchange Agent in the RUNLIB library created during the installation. The install process customizes this JCL with the values specified in the MVS Installation Assistant.

The sample PowerExchange Agent PROC is provided in member AGENTSTP, which is copied to the PROCLIB library with a member name consisting of the Agent /Logger Prefix value followed by A.

The following sample shows JCL for the PowerExchange Agent:

```plaintext
//FWXA PROC  STARTUP=WARM,HLQ=YOUR.INSTALL.HLQ,
//  RUNLIB=YOUR.INSTALL.HLQ.RUNLIB,
//  LOGGER=PRXL
/* PowerExchange Agent
/*
/** POSSIBLE VALUES FOR STARTUP= ARE WARM AND COLD
/** CAUTION = USE "COLD" START ONLY FOR PROBLEM RESOLUTION
/**
/** FWXAGENT EXEC  PGM=EDMSTART,PARM='STARTUP=\$STARTUP',
/**  TIME=NOLIMIT,
/**  ACCT=XXXX
/** STEPLIB   DD DISP=SHR,DSN=\$HLQ..LOADLIB
```
Sample Messages from Starting the PowerExchange Agent

The following sample text shows the PowerExchange Agent startup messages:

FWXEDM172021 EDMSINFO: ChangeDataMove. Version 2.4.04. Release date: 20031015
FWXEDM172008 EDMSINFO: EDMAgent Configuration Parameters:
FWXEDM172010 EDMSINFO: AgentID=PWXA
FWXEDM172010 EDMSINFO: LogClass=*
FWXEDM172010 EDMSINFO: LogHold=NO
FWXEDM172010 EDMSINFO: LogLimit=5000
FWXEDM172010 EDMSINFO: LogBuffLimit=2000
FWXEDM172010 EDMSINFO: TaskLimit=500
FWXEDM172010 EDMSINFO: LSNPort=0
FWXEDM172010 EDMSINFO: CmdPrefix=PWXA
FWXEDM172010 EDMSINFO: RepositoryDSN=EDMSR.DETAIL.V811.AGENTREP
FWXEDM172010 EDMSINFO: RepositoryMode=Detail
FWXEDM172010 EDMSINFO: InitAuthCheck=No
FWXEDM172010 EDMSINFO: CmdAuthCheck=No
FWXEDM172010 EDMSINFO: CCVActive=YES
FWXEDM172010 EDMSINFO: SysplexLogDays=0
FWXEDM172010 EDMSINFO: STARTUP=ARM
FWXEDM172010 EDMSINFO: ServiceModule=EDMSDUMY
FWXEDM172010 EDMSINFO: ServiceModule=EDMSQIO
FWXEDM172010 EDMSINFO: ServiceModule=EDMCTIQ
FWXEDM172010 EDMSINFO: DelOldPmod=0
FWXEDM172010 EDMSINFO: EMAgentTrace=off
FWXEDM172010 EDMSINFO: TRACEOPTIONS=NONE
FWXEDM172010 EDMSINFO: PATROLIM=NO
FWXEDM172010 EDMSINFO: PKDGASPACEIN=100
FWXEDM172010 EDMSINFO: PKDGASPACEINMAX=500
FWXEDM172010 EDMSINFO: MSGPREFIX=PWX
FWXEDM172024 EDMSINFO: New SSCVT built for EDMAgent PWXA. Addr=00C16328
FWXEDM172024 EDMSINFO: New SAST built for EDMAgent PWXA. Addr=00C16210
FWXEDM172064 EDMSINFO: EDMAgent dataspace created. Name=00001EDM,STOKEN=80001F0100000056,Blocks=234
FWXEDM172069 EDMSINFO: Subtask ATTACHED. Module=EDMCCCV0,TaskID=CCV,RC=0
FWXEDM172071 EDMSINFO: Subtask initialization completed. TaskID=CCV
FWXEDM172069 EDMSINFO: Subtask ATTACHED. Module=EDMDIS0,TaskID=DIS,RC=0
FWXEDM172023 EDMSINFO: Active= 1, Inactive= 0. PWXA
FWXEDM172071 EDMSINFO: Subtask initialization completed. TaskID=DIS
FWXEDM172069 EDMSINFO: Subtask ATTACHED. Module=EDMREP0,TaskID=REP,RC=0
FWXEDM172071 EDMSINFO: Subtask initialization completed. TaskID=REP
FWXEDM172069 EDMSINFO: Subtask ATTACHED. Module=EDMSDISP0,TaskID=DISP,RC=0
FWXEDM172071 EDMSINFO: Subtask initialization completed. TaskID=DISP
FWXEDM172069 EDMSINFO: Subtask ATTACHED. Module=EDMSLOG0,TaskID=LOG,RC=0
FWXEDM172071 EDMSINFO: Subtask initialization completed. TaskID=LOG
FWXEDM172256 EDMSINFO: EDMAgent PWXA has completed initialization
FWXEDM172072 EDMSINFO: Log file EDMSLOG OPENed. LogClass=*,LogLimit=5000,LogHold=NO
FWXEDM181223 DTERRIOM: PWX-00607 DTERDI VRM 8.1.1 Build V811_B09 started.
FWXEDM172076 EDMSINFO: Repository file CLOSED
FWXEDM181207 DTERRIOM: Repository Configuration Parameters (EDMSR.DETAIL.V811.AGENTREP):
FWXEDM181206 DTERRIOM: Location=mod1
FWXEDM181206 DTERRIOM: Cache=EDMSR.DETAIL.V810.C1.CACHE
FWXEDM181206 DTERRIOM: RestartInterval=60
FWXEDM181206 DTERRIOM: UpdateInterval=1
FWXEDM181206 DTERRIOM: BackToBackDelay=0
FWXEDM181212 DTERRIOM: Using cached capture registrations (20060721162905)
PowerExchange Agent Message Log

The PowerExchange Agent message log, EDMSLOG, is a SYSOUT data set that contains messages from the PowerExchange Agent and all PowerExchange CDC components that interact with the PowerExchange Agent. You can configure parameters that control aspects of the message log including its size.

**Note:** The PowerExchange Agent closes the current log and allocates a new log when it reaches the message log line limit specified in AGENTCTL parameter LogLimit.

The PowerExchange Agent allocates data space storage that acts as an integration area or buffer to the message log. This storage is allocated based on the LogBuffLimit AGENTCTL parameter. The PowerExchange Agent writes to EDMSLOG any messages sent to the integration area.

If you stop the PowerExchange Agent, the other PowerExchange CDC components continue to write messages to the integration area. When you restart the PowerExchange Agent, it checks for any messages written to this data space and writes them to the EDMSLOG.

**Warning:** If you stop the PowerExchange Agent and the messages written to the data space exceed the value of the LogBuffLimit parameter, additional messages overwrite those at the beginning of the allocated data space, resulting in missed messages. A message in the next EDMSLOG indicates the number of messages that were missed.

**RELATED TOPICS:**
- “Configuring AGENTCTL Parameters” on page 25

Managing the PowerExchange Agent

You can control certain aspects of PowerExchange Agent processing by using MVS commands.

Starting the PowerExchange Agent

To start the PowerExchange Agent, issue the MVS START command with the name of the started task. For example:

```
START PWXA
```

Start the PowerExchange Agent after you start the PowerExchange Listener but prior to starting any other PowerExchange CDC component address spaces.

**Cold or Warm Startup**

If you select Warm start, the PowerExchange Agent uses an existing data space if one exists. If a data space does not exist, the PowerExchange Agent creates one.

If you select Cold start, the PowerExchange Agent creates a new data space and starts as if for the first time. Use this value only if the PowerExchange Agent does not start using the Warm start.

**Warning:** Regularly using Cold start for the PowerExchange Agent can lead to exhaustion of the non-system linkage indexes, or the limit for SCOPE=COMMON data spaces, or both.
The number of non-system linkage indexes is specified in the NSYSLX parameter in the MVS EASY$xx PARMLIB member. The SCOPE=COMMON data space limit is specified in the MAXCAD parameter in the MVS EASY$xx PARMLIB member.

### Stopping the PowerExchange Agent

PowerExchange Agent commands use the MVS command prefix defined by the CmdPrefix statement in the PowerExchange AGENTCTL parameters. To stop the PowerExchange Agent, use the PowerExchange Agent CmdPrefix value followed by SHUTDOWN or SHUTDOWN COMPLETELY. For example:

```
PWXA SHUTDOWN
```

Only use SHUTDOWN COMPLETELY if removing PowerExchange from the system.

### Controlling the PowerExchange Agent

You can use commands to control PowerExchange Agent processing. PowerExchange Agent commands use the MVS command prefix defined by the CmdPrefix statement in the PowerExchange Agent configuration parameters.

The following table briefly describes the PowerExchange Agent commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>DISPLAY LOCKS displays any PowerExchange Agent locks and their owners.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>DISPLAY JOBS displays all MVS TCBs registered to the PowerExchange Agent for its services.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>DISPLAY MODULES displays all modules that the PowerExchange Agent loads.</td>
</tr>
<tr>
<td>DISPLAY</td>
<td>DISPLAY GBLQDSNS displays all global circular queues that are allocated.</td>
</tr>
<tr>
<td>DRAIN</td>
<td>Ensures that all tasks using the PowerExchange Agent are completed and no longer in the system. You must issue this command before issuing the SHUTDOWN COMPLETELY command.</td>
</tr>
<tr>
<td>LOGCLOSE</td>
<td>Closes the PowerExchange Agent message log, EDMSLOG SYSOUT data set.</td>
</tr>
<tr>
<td>LOGOPEN</td>
<td>Opens a new PowerExchange Agent message log, EDMSLOG SYSOUT data set, if one is not currently open.</td>
</tr>
<tr>
<td>LOGSPIN</td>
<td>Performs a LOGCLOSE operation and subsequent LOGOPEN operation.</td>
</tr>
<tr>
<td>REPCLOSE</td>
<td>Deallocates the current PowerExchange repository data set.</td>
</tr>
<tr>
<td>REPOPEN</td>
<td>Allocates the current PowerExchange repository data set if it has been deallocated by either the REPCLOSE or REPOSITORYDSN commands.</td>
</tr>
<tr>
<td>REPOSITORYDSN</td>
<td>Deallocates the current PowerExchange repository data set and allocates the data set specified on the command.</td>
</tr>
<tr>
<td>REPSTATUS</td>
<td>Displays the current status of the PowerExchange repository.</td>
</tr>
<tr>
<td>RESUME</td>
<td>Enables tasks to access the PowerExchange Agent following a DRAIN command.</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>SHUTDOWN stops the PowerExchange Agent address space.</td>
</tr>
<tr>
<td></td>
<td>SHUTDOWN COMPLETELY shuts down the PowerExchange Agent and removes its data spaces from the system.</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>START DIS</td>
<td>START DIS starts the DIS subtask, which processes DISPLAY commands.</td>
</tr>
<tr>
<td>START LOG</td>
<td>START LOG starts the LOG subtask, which writes data from the PowerExchange Agent data space to the EDMSLOG SYSOUT data set.</td>
</tr>
<tr>
<td>START REP</td>
<td>START REP starts the REP subtask, which retrieves PowerExchange repository information.</td>
</tr>
<tr>
<td>STOP DIS</td>
<td>STOP DIS stops the DIS subtask, which processes DISPLAY commands.</td>
</tr>
<tr>
<td>STOP LOG</td>
<td>STOP LOG stops the LOG subtask, which writes data from the PowerExchange Agent data space to the EDMSLOG SYSOUT data set.</td>
</tr>
<tr>
<td>STOP REP</td>
<td>STOP REP stops the REP subtask, which retrieves PowerExchange repository information.</td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**

- “PowerExchange Agent Message Log” on page 31

### Managing Capture Registration Caching

The PowerExchange Agent caches capture registrations in-storage. Caching capture registrations in-storage enables the PowerExchange Agent to respond as quickly as possible to registration check requests from ECCRs.

By default, the PowerExchange Agent obtains new capture registrations from the PowerExchange Listener and stores the capture registrations in two sequential cache data sets. During startup, the PowerExchange Agent reads the cache data sets to populate the in-storage cache of capture registrations. Then the PowerExchange Agent contacts the PowerExchange Listener and requests all capture registrations. The PowerExchange Agent adds new capture registrations to the in-storage cache and to the cache data sets.

If the PowerExchange Listener is temporarily unavailable for any reason when a real-time system is started, this could cause a problem. The mechanism designed to resolve such a problem involves the use of locally held information stored in two physical sequential data sets to provide resilience. These data sets are refreshed at an interval determined when the installation is configured. You can alter the frequency by changing the UpdateInterval parameter. After any new registrations have been successfully saved into the cache data sets the agent uses them to answer capture queries. If there is any problem obtaining or saving new registrations, the current registrations continue to be used.

### Creating the Cache Data Sets

You create the PowerExchange Agent cache datasets during the MVS installation by running the job in member SETUPCC1 of the RUNLIB library.

Use the following DCB attributes for the cache data sets:

- Record length (LRECL) of 254
- Record format (RECFM) of VB
- Data set organization (DSORG) of PS
- Any valid block size. The SETUPCC1 job specifies BLKSIZE=18452, which results in three records per 3390 track.
Repository Display Command

Use the PowerExchange Agent REPSSTATUS command to display the status of the repository. The PowerExchange Agent displays messages indicating the name and type of repository and the name of the cache data sets. For example:

PWXEDM172078I EDMSREP0: REPSSTATUS command accepted by EDM Agent AUSA
PWXEDM181216I DTERION : Repository status follows:
PWXEDM181217I DTERION : PMX-10052 last refresh attempt Tue Jan 22 15:23:39 2008
PWXEDM181217I DTERION : PMX-10053 current change identifier 20080122152344
PWXEDM181217I DTERION : PMX-10055 configuration type repository AUSQA.PMX.AGENTREP
PWXEDM181217I DTERION : PMX-10057 location model
PWXEDM181217I DTERION : PMX-10058 cache (1) AUSQA.PMX.C1.CACHE
PWXEDM181217I DTERION : PMX-10058 cache (2) AUSQA.PMX.C2.CACHE
PWXEDM181217I DTERION : PMX-10063 memory usage: below the line 3%, above the line 0%
PWXEDM181218I DTERION : End of repository status

If the cache data sets are not specified in the AGENTREP parameters, the REPSSTATUS command displays <NONE> for the data set names.

Tip: Informatica recommends using cache data sets to prevent possible loss of change data in situations where the PowerExchange Listener is temporarily unavailable.

Adding or Repairing Cache Data Sets

During normal operation, the PowerExchange Agent caches capture registrations in virtual storage. Because registrations are already in storage, you can temporarily disable the PowerExchange Agent repository to add or repair one or more cache data sets.

To add or repair cache data sets:

1. Close the PowerExchange Agent repository using the REPCLOSE command.
2. Repair the datasets as required. Placing the datasets on separate disk storage spindles adds some resilience.
3. Open the PowerExchange Agent repository using the REPOEN command.

Controlling Security for the PowerExchange Agent

You might need to change the access that you assigned to the PowerExchange Agent services and commands at installation.

Controlling Access to PowerExchange Agent Services

You can restrict access to PowerExchange Agent services.

The hlq.SAMPLIB contains sample commands for the most common mainframe security products. The member #SECURTY directs you to the specific member for the type of security product for your system.

Any job that requests PowerExchange Agent services must be granted read access to this resource. The agent_ID variable is the AgentID specified in the AGENTCTL member and the default options module EDMSDIR.

Note: In the following procedure, replace the variable hlq with the high-level qualifier that you chose when installing PowerExchange.

Enter the prerequisites here (optional).
To control access to PowerExchange Agent services:

1. In the hlq.RUNLIB library, locate the AGENTCTL member and verify that the value of the InitAuthCheck parameter is YES.

2. Define the RACF resource profile, or an equivalent security system, named BMCEDM.agent_IDREGISTER in class FACILITY.

   Defining this resource to RACF, or an equivalent security system, with UACC (READ) effectively disables registration security for PowerExchange Agent services. All RACROUTE macros that the agent issues are successful.

   You can also disable registration security with the InitAuthCheck configuration parameter. Set its value to NO to disable security checking.

### Controlling Access to PowerExchange Agent Commands

Use this procedure to restrict access to PowerExchange Agent commands.

Any user who needs to use PowerExchange Agent commands requires read access to this resource. The agent_ID variable is the AgentID specified in the AGENTCTL member and in the EDMSDIR default options module.

**Note:** In the following procedure, replace the variable hlq with the high-level qualifier that you chose when installing PowerExchange.

To control access to PowerExchange Agent commands:

1. In the hlq.RUNLIB library, locate the AGENTCTL member and verify that the value of the CmdAuthCheck parameter is YES.

2. Define the RACF resource profile, or an equivalent security system, called BMCEDM.agent_ID.COMMAND.* in class FACILITY.

   You can define control for individual agent commands by replacing the asterisk (*) with the command name.

   For example, the following FACILITY class resource profile only protects the SHUTDOWN command for AgentID AG01:

   ```
   BMCEDM.AG01.COMMAND.SHUTDOWN
   ```

   Defining this resource to RACF or an equivalent security system with UACC (READ) effectively disables security for PowerExchange Agent commands. All RACROUTE macros that the agent issues are successful.

   You can also disable command security with the CmdAuthCheck configuration parameter. Set its value to NO to disable security checking.

### Controlling Access to PowerExchange Components

Some PowerExchange components must have system authorization to run. You can limit access to these components through RACF or an equivalent security product. Use this procedure to limit this access.

To control access to PowerExchange components:

1. Get the startup procedure names for the following components:
   - PowerExchange Logger
   - PowerExchange Agent
   - Any PowerExchange component running as a started task

2. Use one of the following methods to provide user authorization for each component:
   - Add the procedure names to the RACF-started procedures table (ICHRIN03), or its equivalent.
   - Create a RACF profile for each procedure name and use the class STARTED.
This step associates a user ID and group ID with the started tasks. This association provides authorized access to any data set that the tasks use and enables PowerExchange components to pass the authorization-checking process. For more information about the RACF-started procedures table or STARTED class profiles, see the IBM documentation for RACF or an equivalent security product.
CHAPTER 4

PowerExchange Logger for MVS

This chapter includes the following topics:

- PowerExchange Logger for MVS Overview, 37
- Planning Considerations for the PowerExchange Logger for MVS, 39
- Configuring the PowerExchange Logger for MVS, 39
- Managing the PowerExchange Logger for MVS, 48
- Monitoring the PowerExchange Logger for MVS, 51
- Managing Log and Restart Data Sets, 52
- Using Post-Log Merge, 69

PowerExchange Logger for MVS Overview

The PowerExchange Logger stores all change data captured by connected ECCRs and provides captured change data to real-time mode extractions and to PowerExchange Condense.

The PowerExchange Logger prepares to write data to log files when it receives a message from an ECCR. The PowerExchange Logger retrieves logged data when it receives a request from an log reader that specifies a relative byte address (RBA) as the starting point for data transfer.

When you use real-time extraction mode to read change data, the PowerExchange Listener passes a Resource Interest List that contains the EDMNAMEs of the capture registrations in the extraction process to the PowerExchange Logger. The PowerExchange Logger uses this list to filter out change records for EDMNAMEs that are not included in the extraction process, which reduces the resource consumption of the log read process in the PowerExchange Listener.

The IBM Cross-System Coupling Facility (XCF) controls the connection from other components to the PowerExchange Logger. The number of log readers that can request data from the PowerExchange Logger is limited to the maximum number of members that can join an XCF group. The maximum members in an XCF group is MVS release dependent and controlled through the XCF MAXMEMBER specification used when defining the SYSPLEX Couple data sets.
In addition to illustrating data flow, the following figure also shows PowerExchange Logger control flow.

You can control the PowerExchange Logger by running batch change utility procedures that perform the following functions:

- Set system parameters in the EDMUPARM module.
- Modify the restart data set to manage active and archive logs.

You can also issue interactive commands to the PowerExchange Logger.

**Multiple Instances of the PowerExchange Logger for MVS**

You can run multiple instances of the PowerExchange Logger simultaneously in a single PowerExchange system. The number of instances that you use depends on your performance needs and your data-management processes.

For example, you might want to use separate instances of the PowerExchange Logger to capture changes from different branch offices of an organization.

The following situations are possible reasons for using multiple instances of the PowerExchange Logger:

- High volume of data
- Multiple environments. Although not required, you may want to dedicate a separate PowerExchange Logger for each data-resource type. For example, one for IMS and one for VSAM.
- Application requirements

Up to 50 PowerExchange Loggers can attach to a PowerExchange Agent. The value of the TaskLimit parameter in the AGENTCTL parameters limits the number of PowerExchange Loggers that can attach to a PowerExchange Agent. Each PowerExchange Logger requires a minimum of 12 tasks, and uses additional tasks for log readers and archive processes.
Restriction: A Post-Log Merge group can be comprised of a maximum of eight PowerExchange Loggers.

Planning Considerations for the PowerExchange Logger for MVS

Read the following planning considerations before configuring the PowerExchange Logger for MVS.

XCF Groups

To optimize the MVS configuration for the PowerExchange Logger, consider increasing the number of cross-coupling facility (XCF) groups.

PowerExchange uses IBM Cross-System Coupling Facility (XCF) services to provide communication between certain PowerExchange CDC components. The couple data set should be sized to accommodate the additional PowerExchange XCF groups and members.

If you use the Post-Log Merge option of the PowerExchange Logger, you need to plan for capacity for four XCF groups for each PowerExchange Logger. Otherwise, a single XCF group is used for a PowerExchange Logger.

Consult your MVS systems programmer to determine the number of existing XCF groups and ensure that additional XCF groups are available. PowerExchange CDC uses at least one, and up to four, XCF groups for each running PowerExchange Logger.

Recall of Archived Log Data Sets

The PowerExchange Logger for MVS uses the DFSMShsm ARCGIVER module to explicitly recall any archived log data sets that are identified as "migrated" but needed for data set allocation requests.

If ARCGIVER is not available, an allocation request for a migrated data will fail. The ARCHRCAL macro that attempts to invoke ARCGIVER issues an error code, such as 0x806, which is used as a DYNALLOC Info Code (S99INFO).

Configuring the PowerExchange Logger for MVS

Review the information in the following sections before using the PowerExchange Logger. Review this information when performing maintenance tasks on PowerExchange log data sets.

Consider the following general operational information about the PowerExchange Logger:

- A PowerExchange Logger can log data from multiple ECCRs that operate on the same MVS system. Using Post-Log Merge, changes from multiple MVS systems can be accessed as if stored in a single Logger environment.
- If you use multiple PowerExchange Loggers, you need to have a copy of the default options module (EDMSDIR) for each instance of the PowerExchange Logger. Because you cannot rename EDMSDIR, you
must allocate a separate user library (YOUR.USERLIB) for each copy of EDMSDIR. You can reduce the chance of data loss by establishing dual active log data sets and dual archive log data sets.

- If you reinitialize the PowerExchange Logger after you start capturing changes, the RBA is reset to 0 and you lose all the changes that have been captured but not yet applied.

If you have to reinitialize the PowerExchange Logger, you also need to reinitialize all PowerExchange processes, which use data that is read from the PowerExchange Logger. The normal restart of these processes uses the last-read PowerExchange Logger RBA to generate the correct restart point. Reinitializing the PowerExchange Logger invalidates the last-read RBA value.

**Related Topics:**

- “Managing Log and Restart Data Sets” on page 52

### Configuring the EDMUPARM Module Options

Define the PowerExchange Logger options in the EDMUPARM module, which resides in the USERLIB library. The EDMUPARM module is created during the PowerExchange installation process by job SETUPCC2 in the RUNLIB library. This job runs the PowerExchange Logger in batch mode to create the EDMUPARM module.

To redefine an existing PowerExchange Logger EDMUPARM module options, you must first stop the PowerExchange Logger.

Consider the following issues when you configure the PowerExchange Logger EDMUPARM module options:

- With dual logging and dual emergency restart data sets, allocate the primary and secondary data sets to different volumes to ensure recovery in the case of a disk failure.

- To create an effective logging configuration, balance the following key factors:
  - Size of the input and output buffers, depending on the volume of captured data
  - Number of active log data sets defined (minimum 3, maximum 31), depending on the volume of captured data and how rapidly the data can be archived.
  - Size of active log data set, depending on the volume of data and the size requirements of the archive media.
  - Size of archive log data set, which should be based on the active log data set size, the block size used for archive data sets, and the device type to which you are archiving.

**Related Topics:**

- “Size and Number of Active Log Data Sets” on page 53

### DEFINE Statement Syntax Overview

Specify the PowerExchange Logger options in the EDMUPARM module using the DEFINE statement, which has the following syntax:

```plaintext
DEFINE
  [LOGGER_TITLE=name]
  [SYSTEM_OPTIONS
    [LOGGER_NAME=id,] [CHKPT_FREQUENCY=nnnn,] [START_TRACE=Y|N] [SUFFIX=n,]
    [TIMECHKPT_FREQ=nn,] [TIMER_INTERVAL=nnnn] ]
  [ARCHIVE_OPTIONS
    [PREFIX_COPY1=name,] [PREFIX_COPY2=name,] [ARCHIVE_BLKSIZE=n,]
    [ARCHIVE_DACL=] [ARCHIVE_DACL2=] ]
```
[ARCHIVE_MOC2=name,]
[ARCHIVE_MOC22=name,]
[ARCHIVE_RTPD=n,]
[ARCHIVE_RTPD2=n,]
[ARCHIVE_STCL=name,]
[ARCHIVE_STCL2=name,]
[ARCHIVE_UNIT=name,]
[ARCHIVE_UNIT2=name,]
[ARCUNIT_CNT=n,]
[PRIM_SPACE=n,]
[SEC_SPACE=n,]
[SPACEALLOC=unit,]
[LOGGING_OPTIONS
[LOGINSUFF=n,]
[LOGOUTSUFF=n,]
[ACTIVELOGMODE=mode,]
[ARCHIVERLOGMODE=mode,]
[ERGSLOGMODE=mode]]
END

**DEFINE Statement**

Use the DEFINE statement to configure the PowerExchange Logger options.

The DEFINE statement has the following general syntax:

```
DEFINE
  LOGGER_TITLE=name
  [SYSTEM_OPTIONS options]
  [ARCHIVE_OPTIONS options]
  [LOGGING_OPTIONS options]
END
```

Enter all of the parameters using a single DEFINE statement. If you omit a parameter, the PowerExchange Logger uses its default value.

The only required statements are DEFINE and END.

The system, archive, and logging option groups have additional parameters. You must specify at least one options group keyword with at least one parameter. Each option group has a different set of options associated with it.

<table>
<thead>
<tr>
<th>Parameter or Statement</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGGER_TITLE</td>
<td>Optional</td>
<td>Specifies a PowerExchange Logger name. This name can be up to 16 characters long.</td>
</tr>
<tr>
<td>SYSTEM_OPTIONS</td>
<td>Optional</td>
<td>Specifies configuration options for the PowerExchange Logger system.</td>
</tr>
<tr>
<td>ARCHIVE_OPTIONS</td>
<td>Optional</td>
<td>Specifies configuration options for the archive log data sets.</td>
</tr>
<tr>
<td>LOGGING_OPTIONS</td>
<td>Optional</td>
<td>Specifies configuration options for the active and archive log data sets.</td>
</tr>
<tr>
<td>END</td>
<td>Required</td>
<td>Indicates that the input for the DEFINE statement is complete.</td>
</tr>
</tbody>
</table>

**SYSTEM_OPTIONS Statement**

Use the following syntax for the SYSTEMS_OPTIONS statement:

```
SYSTEM_OPTIONS
  [LOGGER_NAME=id,]
  [CHKPT_FREQUENCY=n,]
  [START_TRACE=Y|N,]
  [SUFFIX=n,]
  [TIMER_INTERVAL=n,]
  [TIMECHKPT_FREQ=n,]
```
The parameters are optional, but you must specify at least one of them.

If you specify multiple parameters, use a comma (,) to separate them. Do not put a comma at the end of the last parameter.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGGER_NAME</td>
<td>Specifies the PowerExchange Logger ID.</td>
<td>A string from one to four characters in length. The following rules apply:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- The value can begin with and contain alphanumeric characters and the characters #, @, and $.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Because other PowerExchange CDC components use this value to refer to the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>PowerExchange Logger, the value must match the LOGGER parameter in the Power-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Exchange Agent EDMSDIR options module and the LOG parameter on LR API</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CAPI_CONNECTION statement in the DBMOVER configuration member.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- In a Post-Log Merge environment, all member Loggers must use the same</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LOGGER_NAME value.</td>
</tr>
<tr>
<td>CHKPT_FREQUENCY</td>
<td>Specifies the number of log records to process before taking a checkpoint.</td>
<td>A number from 1 to $2^{31}$-1. Default is 10,000.</td>
</tr>
<tr>
<td>START_TRACE</td>
<td>Specifies whether the Logger trace is active.</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Y for yes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- N for no.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is N.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Warning:</strong> The value Y causes additional overhead in the Logger. Enter Y</td>
</tr>
<tr>
<td></td>
<td></td>
<td>only at the request of Informatica Global Customer Support.</td>
</tr>
<tr>
<td>SUFFIX</td>
<td>Specifies the unique suffix for a member in a Post-Log Merge group.</td>
<td>A unique number from 1 through 9.</td>
</tr>
<tr>
<td>TIMER_INTERVAL</td>
<td>Specifies how frequently the Logger performs its internal management</td>
<td>An interval in hundredths of seconds in the following range:</td>
</tr>
<tr>
<td></td>
<td>operations, such as freeing unused virtual storage or detecting inactive</td>
<td>- Minimum is 50 (0.5 seconds).</td>
</tr>
<tr>
<td></td>
<td>tasks that need to be POSTed.</td>
<td>- Maximum is 6000 (1 minute).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is 100.</td>
</tr>
<tr>
<td>TIME_CHKPT_FREQ</td>
<td>Specifies how frequently time-based checkpoint records are created in a</td>
<td>The checkpoint frequency expressed in number of elapsed TIMER_INTERVAL</td>
</tr>
<tr>
<td></td>
<td>Post-Log Merge environment.</td>
<td>periods. This number must be in the following range:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum is 5.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Maximum is 60.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is 30.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you use the default TIMER_INTERVAL value of 100 hundredths of a second</td>
</tr>
<tr>
<td></td>
<td></td>
<td>with the default of 30 for this parameter, a time-based checkpoint record is</td>
</tr>
<tr>
<td></td>
<td></td>
<td>written every 30 seconds ($100 * 1/100 * 30$).</td>
</tr>
</tbody>
</table>

**Related Topics:**

- “Using Post-Log Merge” on page 69
ARCHIVE_OPTIONS Statement

The ARCHIVE_OPTIONS statement has the following syntax:

```
ARCHIVE_OPTIONS
  [PREFIX_COPY1-name,]
  [PREFIX_COPY2-name,]
  [ARCHIVE_BLKSIZE=n,]
  [ARCHIVE_DACL-name,]
  [ARCHIVE_MIGL-name,]
  [ARCHIVE_MIGL2-name,]
  [ARCHIVE_RTPD-n,]
  [ARCHIVE_RTPD2-n,]
  [ARCHIVE_STCL-name,]
  [ARCHIVE_STCIL2-name,]
  [ARCHIVE_UNIT-name,]
  [ARCHIVE_UNIT2-name,]
  [ARC_UNIT_CNT-n,]
  [PRIM_SPACE=n,]
  [SEC_SPACE-n,]
  [SPACE_ALLOC-unit]`
```

If you use the ARCHIVE_OPTIONS statement, you must specify at least one parameter.

Use a comma (,) as a separation character for the parameters. The last parameter in an options group must not end in a comma.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFIX_COPY1</td>
<td>Specifies the prefix for the first archive log data set name.</td>
<td>If you use multiple qualifiers, enclose the prefix in quotation marks. The value can be up to 17 alphanumeric characters long and must follow MVS data set name rules. Note: With Post-Log Merge, all member Loggers must have a unique value for this parameter.</td>
</tr>
<tr>
<td>PREFIX_COPY2</td>
<td>Specifies the prefix for the second archive log data set name.</td>
<td>If you use multiple qualifiers, enclose the prefix in quotation marks. The value can be up to 17 alphanumeric characters long and must follow MVS data set name rules. If you use this keyword, the value cannot be blank, even if you specified ARCHIVE_LOG_MODE=SINGLE. Note: With Post-Log Merge, all member Loggers must have a unique value for this parameter.</td>
</tr>
<tr>
<td>ARCHIVE_BLKSIZE</td>
<td>Specifies the block size of the archive log data set.</td>
<td>The block size must be compatible with the device type you specify in the ARCHIVE_UNIT parameter. The value must be a multiple of 4096 and must be in the range 4096 through 28672. Default is 24576.</td>
</tr>
<tr>
<td>ARCHIVE_DACL</td>
<td>Specifies the SMS data class name of the archive log data set.</td>
<td>If this value is omitted, no SMS data class is specified when allocating the primary archive log data set. One might be assigned by your SMS ACS routines.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARCHIVE_DACL2</td>
<td>Specifies the SMS data class name of the second archive log data set.</td>
<td>If this value is omitted, the second archive log takes the data class of the first archive log data set, if specified. Specify ARCHIVE_DACL2=, to prevent a data class name specified for the first archive log data set being used as a default for the second.</td>
</tr>
<tr>
<td>ARCHIVE_MGCL</td>
<td>Specifies the SMS management class name of the archive log data set.</td>
<td>If this value is omitted, no SMS management class is specified when allocating the primary archive log data set. One might be assigned by your SMS ACS routines.</td>
</tr>
<tr>
<td>ARCHIVE_MGCL2</td>
<td>Specifies the SMS management class name of the second archive log data set.</td>
<td>If this value is omitted, the second archive log takes the management class of the first archive log data set, if one is specified. Specify ARCHIVE_MGCL2=, to prevent a management class name specified for the first archive log data set being used as a default for the second.</td>
</tr>
<tr>
<td>ARCHIVE_RTPD</td>
<td>Specifies the number of days to retain the archive log data set.</td>
<td>0 through 9999. Default is 9999.</td>
</tr>
<tr>
<td>ARCHIVE_RTPD2</td>
<td>Specifies the number of days to retain the second archive log data set.</td>
<td>0 through 9999. The default is 9999.</td>
</tr>
<tr>
<td>ARCHIVE_STCL</td>
<td>Specifies the SMS storage class name of the archive log data set.</td>
<td>If this value is omitted, no SMS storage class is specified when allocating the primary archive log data set. One might be assigned by your SMS ACS routines.</td>
</tr>
<tr>
<td>ARCHIVE_STCL2</td>
<td>Specifies the SMS storage class name of the second archive log data set.</td>
<td>If this value is omitted, the second archive log takes the storage class of the first archive log data set, if specified. Specify ARCHIVE_STCL2=, to prevent a storage class name specified for the first archive log data set being used as a default for the second.</td>
</tr>
<tr>
<td>ARCHIVE_UNIT</td>
<td>Specifies the device type or unit name of the device used to store the archive log data set.</td>
<td>Specify a device type or unit name up to 8 alphanumeric characters long. Note: Informatica Corporation recommends that you write the primary archive log data set to DASD.</td>
</tr>
<tr>
<td>ARCHIVE_UNIT2</td>
<td>Specifies the device type or unit name of the device used to store the second archive log data set.</td>
<td>Specify a device type or unit name up to 8 alphanumeric characters long. If this value is omitted, the second archive log takes the UNIT value of the first archive log data set. Specify ARCHIVE_UNIT2=, to prevent a unit type specified for the first archive log data set being used as a default for the second.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ARC_UNIT_CNT</td>
<td>Specifies the number of DASD units to use for archiving.</td>
<td>Use this parameter in the same way you use the count option of the MVS UNIT parameter. If using SMS, the SMS data class specifies the volume count for SMS-managed data sets. Default is 2 units.</td>
</tr>
<tr>
<td>PRIM_SPACE</td>
<td>Specifies the primary space allocation for DASD data sets in the unit type specified by SPACE_ALLOC.</td>
<td>Must be greater than 0. Default is 4320 blocks.</td>
</tr>
<tr>
<td>SEC_SPACE</td>
<td>Specifies the secondary space allocation for DASD data sets in the unit type that you specify in SPACE_ALLOC.</td>
<td>Must be greater than 0. Default is 540 blocks.</td>
</tr>
<tr>
<td>SPACE_ALLOC</td>
<td>Specifies the unit in which primary and secondary space allocations are made.</td>
<td>- BLK allocates space in blocks. BLK is the default. - CYL allocates space in cylinders. - TRK allocates space in tracks.</td>
</tr>
</tbody>
</table>

LOGGING_OPTIONS Statement

The LOGGING_OPTIONS statement has the following syntax:

```
LOGGING_OPTIONS
[LOG_INBUFF=nn,]
[LOG_OUTBUFF=nn,]
[ACTIVE_LOG_MODE=mode,]
[ARCHIVE_LOG_MODE=mode,]
[ERDS_LOG_MODE=mode]
```

If you use the LOGGING_OPTIONS statement, you must specify at least one parameter.

Use a comma (,) as a separation character for the parameters. The last parameter in an options group must not end in a comma.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOG_INBUFF</td>
<td>Defines the number of 4-KB buffers used for reading the active and archive logs.</td>
<td>1 through 60 (decimal). Default is 28.</td>
</tr>
<tr>
<td>LOG_OUTBUFF</td>
<td>Specifies the size, in 4 KB buffers, of the output buffer that the PowerExchange Logger uses for writing the active and archive log data sets.</td>
<td>1 through 50 (decimal).</td>
</tr>
<tr>
<td>ACTIVE_LOG_MODE</td>
<td>Specifies whether the PowerExchange Logger writes to one or two active log data sets at a time.</td>
<td>- SINGLE. The PowerExchange Logger uses one active log at a time. - DUAL. The PowerExchange Logger writes to a primary log and a secondary backup log simultaneously. Note: Informatica strongly recommends that you use dual logging. Default is DUAL.</td>
</tr>
</tbody>
</table>
Parameter | Description | Valid Values
---|---|---
ARCHIVE_LOG_MODE | Specifies whether the PowerExchange Logger writes to one or two archive log data sets at a time. The PowerExchange Logger generates archive logs when the active log is off-loaded. | - SINGLE. The PowerExchange Logger writes to one archive log at a time. - DUAL. The PowerExchange Logger writes to a primary log and a secondary backup log simultaneously. **Note:** Informatica strongly recommends that you use dual logging. Default is DUAL.

ERDS_LOG_MODE | Specifies whether the PowerExchange Logger writes to one or two PowerExchange restart data sets (ERDS) at a time. | - SINGLE. The PowerExchange Logger uses one restart data set at a time. - DUAL. The PowerExchange Logger writes to a primary restart data set and a secondary backup restart data set simultaneously. **Note:** Informatica strongly recommends that you use dual logging. Default is DUAL.

Creating Active Log and Emergency Restart Data Sets

The PowerExchange installation process creates the PowerExchange Logger active logs and emergency restart (ERDS) data sets when you run job XICDC500 in the RUNLIB library. The active logs are VSAM linear data sets whereas the emergency restart data sets are VSAM KSDS data sets.

PowerExchange defines the PowerExchange Logger active logs using IDCAMS. Consider the following before creating the active log data sets:

- Do not specify secondary allocation.
- Specify a single VOLSER in the VOLUME parameter.
- You cannot use VSAM record-level sharing (RLS) with linear data sets (LDS). If you use SMS, do not associate any RLS attributes with these data sets.
- If you use SMS, use an SMS STORCLAS that does not specify GUARANTEED SPACE=YES.

Defining Log Data Sets to the PowerExchange Logger

The PowerExchange Logger requires an entry in the ERDS to access active and archive log data sets. Use the DEFINE_LOG command to define the active and archive logs to the emergency restart data sets (ERDS). During the archival processing, the PowerExchange Logger automatically defines archive logs to the ERDS.

The PowerExchange installation process defines the active logs when you run job SETUPCC2 in the RUNLIB library. This job runs the PowerExchange Logger in batch mode to create the EDMUPARM module and define the active logs to the ERDS data set.

**RELATED TOPICS:**
- “Defining Log Data Sets to the ERDS” on page 63
Customizing the PowerExchange Logger JCL

The PowerExchange Logger can run as a started task or a batch job. Informatica recommends that you run the PowerExchange Logger as a started task because it is a long-running task that is integral part of the change data capture environment.

After you have customized the JCL for your installation, you must copy the JCL to a system procedure library (PROCLIB) for started tasks. PowerExchange provides sample JCL for the PowerExchange Logger. Cleanup job XIZZZ998 in the RUNLIB library, which runs as a part of the PowerExchange installation process, moves the PowerExchange Logger JCL to the PowerExchange PROCLIB library.

The member name in the PROCLIB library is the value you specify for the PowerExchange Agent / Logger Prefix property in the MVS Installation Assistant followed by the letter L. For example, by default the PowerExchange Agent / Logger Prefix value is "PWX." Therefore, the default member name for the PowerExchange Logger JCL in the PROCLIB library is PWXL.

The PowerExchange Logger has the following JCL statements and parameters:

EXEC PGM=EDMLC000,PARM='logger_id[,BATCH][,,smf_id]'

Invokes the PowerExchange Logger.

Enter the following required and optional positional parameters for the PARM parameter:

* logger_id
  PowerExchange Logger ID that you specified in the LOGGER_NAME parameter in the EDMUPARM module options.

  PowerExchange uses this value to locate the PowerExchange Logger options in the EDMUPARM module.

* BATCH
  Optional. Run the PowerExchange Logger in batch mode to perform maintenance activities.

  Normally, use this option only when you update the EDMUPARM module options or define or delete logs from the ERDS.

* smf_id
  Optional. For Post-Log Merge configurations, this value overrides the system SMF ID value that PowerExchange appends to the PowerExchange Logger ID to form the XCF group name.

  Each PowerExchange Logger XCF group name must be unique within the sysplex.

  By default, the PowerExchange Logger uses the SMF ID for the MVS system on which it runs. If the SMF ID value for the MVS system on which a member Logger runs is not unique within the Post-Log Merge group, use this option to provide a unique value.

  The following example shows an EXEC card that uses a symbolic parameter, &SMFID, to override the system SMF ID:

  //LOGGER EXEC PGM=EDMLC000,REGION=ON,TIME=NOLIMIT,
  //      PARM='&LOGNAME, , , , &SMFID',ACCT=XXXX

  Valid values are 1 through 4 alphanumeric characters.

* JOBLIB or STEPLIB DD
  Defines the PowerExchange LOAD library, which contains the PowerExchange Logger load modules.

  This library must be APF-authorized.
Managing the PowerExchange Logger for MVS

You can control certain aspects of PowerExchange Logger processing using commands.

Starting the PowerExchange Logger for MVS

To start the PowerExchange Logger, issue the MVS START command with the name of the started task. For example:

```
START FWXL
```
Start the PowerExchange Logger after you start the PowerExchange Agent but prior to starting any other PowerExchange CDC component address spaces.

**Stopping the PowerExchange Logger for MVS**

To stop the PowerExchange Logger, issue the MVS STOP command with the name of the started task. For example:

```mvs
STOP PWXL
```

The PowerExchange Logger does not stop until all reader and writer connections have terminated.

**Controlling the PowerExchange Logger for MVS**

You can use commands to control and display information about PowerExchange Logger processing. Enter PowerExchange Logger command by using the MVS MODIFY command. For example:

```mvs
F logger,command
```

The following table briefly describes the PowerExchange Logger commands:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>DISPLAY OBJECT=LOG displays information about the active or archive log data sets.</td>
</tr>
<tr>
<td></td>
<td>DISPLAY OBJECT=CONNECTION displays information about tasks connected to the PowerExchange Logger.</td>
</tr>
<tr>
<td>PRINT</td>
<td>Prints log records to a dynamically allocated SYSOUT data set.</td>
</tr>
<tr>
<td>STOP</td>
<td>Stops the PowerExchange Logger. The MVS STOP command can also be used.</td>
</tr>
<tr>
<td>RESOLVE_INDOUBT</td>
<td>Forces the PowerExchange Logger to either commit the log records as valid changes or to discard them.</td>
</tr>
</tbody>
</table>

**Overriding Log-Read API Timed Defaults**

The Log-Read API (LRAPI) waits a fixed amount of time for a response after it sends commands to the PowerExchange Logger. In some customer environments, the default wait times for Log-Read API command may be too short. You can override the amount of time that the LRAPI waits for any request type by specifying parameters using the EDMLRPRM DD name.

**EDMLRPRM Parameters**

Specify the parameters by including the EDMLRPRM DD name in the JCL of the job issuing the Log-Read API calls. The parameters can be specified using an in-stream data set in the JCL or in a sequential data set.

Use the following DCB attributes when using a sequential data set to specify the parameter values:

- RECFM=FB or RECFM=VB
- LRECL less than or equal to 255
- Any valid block size

Specify one parameter statement per record or line. An asterisk (*) or a hash (#) in the first character indicates a comment. The syntax of the parameter statement is:

```plaintext
parameter=parm
```
The `parm` variable represents the wait time in hundredths of a second. The available parameters are:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Values (in hundredths of second)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTLST</td>
<td>6000 (60 seconds)</td>
<td>Specifies the time spent waiting for the PowerExchange Logger to respond to the Resource Interest List command. This time starts after the PowerExchange Logger issues the PWXEDM172791I message.</td>
</tr>
<tr>
<td>REQTRN</td>
<td>24000 (240 seconds)</td>
<td>Specifies the time spent waiting for the PowerExchange Logger to start sending data. This time starts after the PowerExchange Logger issues the PWXEDM263011I message.</td>
</tr>
<tr>
<td>SIGNON</td>
<td>6000 (60 seconds)</td>
<td>Specifies the time spent trying to connect to the PowerExchange Logger. This time starts after the PowerExchange Logger issues the PWXEDM263010I message.</td>
</tr>
<tr>
<td>STPTRN</td>
<td>12000 (120 seconds)</td>
<td>Specifies the time spent waiting for the PowerExchange Logger to stop sending more data. This time starts after the PowerExchange Logger issues the PWXEDM263014I message.</td>
</tr>
<tr>
<td>TERM</td>
<td>4500 (45 seconds)</td>
<td>Specifies the time when the Log Read API is disconnecting from the PowerExchange Logger. This time starts after the PowerExchange Logger issues the PWXEDM263012I message.</td>
</tr>
</tbody>
</table>

The Request Data Transfer (REQTRN) command is the most likely PowerExchange Logger command to require additional time. In processing a Request Data Transfer (REQTRN) command, the PowerExchange Logger may have to wait for archive log data sets to be recalled or for a tape mount. If the PowerExchange Logger cannot access the required log data sets in four minutes and provide the data to the LRAPI, the LRAPI request times out and returns reason code 0x0A0E0062 (LoggerDidNotRespondToCommand) and terminates the extraction request. In some environments, the LRAPI might frequently encounter this situation due to operational issues and so extending the wait time for the REQTRN command is necessary.

**Note:** Setting these parameter values in the PowerExchange Listener affects each instance of the Log-Read API. Therefore, all extractions use the same values.

The following examples specifies a value of 3 minutes for the REQTRN parameter using an in-stream data set:

```plaintext
//EDMLRPFM DD *
/*
/* Set REQTRN timeout value to 3 minutes (i.e. 3*60*100 )
/*
REQTRN=18000
/*
```

### Resolving In-Doubt Units of Work

Use this procedure to resolve in-doubt units of work (UOWs). UOWs that have not been committed may be left in an in-doubt state (for example when a CICS/VSAM or IMS region ABENDs). When the ECCR for that region reconnects to the PowerExchange Logger, the PowerExchange Logger exchanges information with the CICS, IMS, or DB2 regions and attempts to resolve in-doubt UOWs. The PowerExchange Logger generates a message that reports how many in-doubt UOWs were detected and if any UOWs are not resolved by this process. Use the following procedure to resolve the status of the in-doubt UOWs.
To resolve in-doubt units of work:

1. Run the DISPLAY command to the PowerExchange Logger to determine the data set names and RBAs of the UOWs that are in doubt.
2. Access the capture source environment and determine which UOWs you want to commit to the target database and which you want to abort.
3. In the PowerExchange Logger environment, run the RESOLVE_INDOUBT command for each in-doubt UOW:
   - Run the command with ACTION=COMMIT for UOWs that you want to commit to the source.
   - Run the command with ACTION=ABORT for UOWs that you want to abort.

---

### Monitoring the PowerExchange Logger for MVS

The PowerExchange Logger archives active logs when they become full. You must monitor the PowerExchange Logger to ensure that the archiving process keeps pace with the data flow. If the PowerExchange Logger uses all available active log space, PowerExchange change data capture and extraction will be impacted until the PowerExchange Logger archival process makes active log space available. Specifically, PowerExchange ECCRs will be unable to record new change data and extraction operations may be unable to read captured change data.

The PowerExchange Logger issues the following messages to allow you to monitor the status of the active log data sets:

- **PWXEDM1726721 EDM Logger last active log data set is nn percent full**

The PowerExchange Logger issues this message when the last available active log data set is 75 percent full, and reissues this message after each additional five percent of the remaining data set space is filled. The PowerExchange Logger retries the archive process each time it issues this message.

You should also monitor the PowerExchange Logger for other operational issues that may be unrelated to the active logs and archive log process. For example, if the PowerExchange Logger runs with a lower dispatching priority or class of service than a highly-active ECCR, it may be delay the ECCR because it cannot write change data to the active log data sets fast enough. PowerExchange issues the following Write-To-Operator (WTO) messages to allow you to monitor the status of change data recording:

- **PWXEDM172824W EDM Change Capture waiting on [the Logger’s queue | the ECCR-to-CIC queue] since date time. Using EDM Logger loggerid.**

A PowerExchange ECCR issues this message if it cannot send change data to the PowerExchange Logger because the circular queue used to do this is full.

For synchronous ECCRs, the transaction or VSAM batch job that encounters the full queue waits until it can log the change data to the circular queue. For asynchronous ECCRs, the ECCR address space waits until it can log the change data to the circular queue.

- **PWXEDM172825W UOWs are waiting on EDM syncpoint; see EDM log**

If the PowerExchange Logger does not respond to a PowerExchange ECCR within approximately one minute of the ECCR sending an end-UOW, the ECCR issues this message. In addition, PowerExchange writes message PWXEDM172826W with the UOW ID to the EDMMSG data set in the ECCR.

The PWXEDM172825W message may indicate that the PowerExchange Logger cannot keep pace with the ECCR. Alternatively, this message may indicate a transitory slowdown in the PowerExchange Logger due to other system issues, such as an SVC dump.

For synchronous ECCRs, the transaction or VSAM batch job waits until the PowerExchange Logger indicates that the end-UOW has been logged to the active log data set. For asynchronous ECCRs, the ECCR address space waits until this indication is received.
Performance Rules and Guidelines

To achieve the best performance for the PowerExchange Logger, consider the following rules and guidelines:

- The PowerExchange Logger is a high-performance started task. Informatica Corporation recommends that you define the Logger with the same dispatching priority as other high-performance started tasks on your system.
- If you anticipate a large volume of captured data, allocate buffers and data sets that are larger than those allocated in the sample startup procedures.
- Consider defining more active log data sets than the number specified in the sample startup procedures.
- Allocate the Logger active logs, emergency restart data sets, and the Archive Log Copy1 on high-performance DASD.
- The PowerExchange Logger is a long-running MVS started task. Therefore, ensure that your existing MVS system parameters or JCL does not cancel the PowerExchange Logger after a specified amount of CPU time or time.

To prevent cancellation of the PowerExchange Logger after a specified amount of CPU time or time, you need to specify `TIME=1440` or `TIME=NOLIMIT` in the EXEC statement of the PowerExchange Logger startup procedure.

**RELATED TOPICS:**

- “Size and Number of Active Log Data Sets” on page 53

Managing Log and Restart Data Sets

You can manage log data sets, including the archive log and active log data sets. You can also allocate and manage restart data sets.

**RELATED TOPICS:**

- “Archive Log Rules and Guidelines” on page 53
- “Size and Number of Active Log Data Sets” on page 53
- “Data Set Size Determination” on page 54
- “Number of Data Sets” on page 56
- “Defining Log Data Sets to the ERDS” on page 63
- “Deleting Log Data Sets from the ERDS” on page 64
- “Allocating Restart Data Sets” on page 56
- “Adding Active Log Data Set Definitions to the Restart Data Set” on page 57
- “Changing the Size of Active Log Data Sets” on page 58
- “Formatting Log Data Sets” on page 62
- “Recovering Damaged Restart Data Sets” on page 66
- “Moving Log Data Sets to Other Devices” on page 67
Archive Log Rules and Guidelines

Use the following rules and guidelines when you manage archive logs:

- Archive log data sets are dynamically allocated. When you install or reconfigure the PowerExchange Logger, you specify the data set name prefix, block size, unit name, and DASD sizes needed for allocation.
- The emergency restart data sets (ERDS) contains approximately 1,000 entries for the archive log data sets. When the PowerExchange Logger reaches the last entry, it wraps to the beginning, overwriting the oldest entry.
- Define dual archive logs to prevent potential data loss if one copy is corrupted or accidentally deleted.
- Define dual archive logs to prevent potential data loss if one copy is corrupted or accidentally deleted.
- Configure the Logger parameters so at least the first archive log copy is created on DASD. The second archive log copy can be placed on tape.
- You can archive DASD archive logs to tape provided that the storage management system automatically restores them to DASD when they are dynamically allocated.
- You can specify that your secondary archive log data sets be stored on a different device and device type from that used to store your primary archive log data sets. You can also specify different SMS classes for your primary and secondary archive logs.
- If you archive data to tape, adjust the size of the log data sets so that each set contains the amount of space that can be stored on a tape volume. Doing so minimizes tape handling and volume mounts and maximizes the use of tape resources.
- Because archive log data sets written to DASD cannot extend to another volume, make the primary space allocation (both quantity and block size) large enough to contain all of the data coming from the active log data sets. Allocate primary space with the PRIM_SPACE option of the DEFINE statement.
- As each active log becomes full, the PowerExchange Logger off loads the log data to an active archive log. If the rate of changes flowing into the Logger fills all the active logs before the Logger finishes off loading to an archive, the Logger stops accepting changes for two minutes. During the pause, the Logger attempts to finish its current archive log. The PowerExchange Logger continues in this mode until it completes off loading data to an archive, or until you stop the PowerExchange Logger manually.
- When the PowerExchange Logger abends due to data set out-of-space conditions, the PowerExchange Logger action depends on the abend code:
  - If the abend code is B37, the PowerExchange Logger increments the primary and secondary allocations by 25 using the units you specified in your definition and attempts to continue archiving.
  - If the abend code is D37 or E37, examine your system configuration (particularly the volumes that your PowerExchange active logs use) and determine the reason for the lack of space. If you fix the problem, the PowerExchange Logger continues attempting to archive until it is successful. If you do not fix the problem, you must use the MVS CANCEL command to cancel the PowerExchange Logger.

Warning: Do not place both archive log copies on tape. This limits the number of log readers to a single reader per archive log and allows only two concurrent extractions.

Related Topics:

- "ARCHIVE_OPTIONS Statement" on page 43

Size and Number of Active Log Data Sets

The PowerExchange installation process allocates three active log data sets with minimum size requirements. Use the information in this section to determine whether you need to increase the size of the data sets, and whether you should allocate additional log data sets. When you define your active log data sets, consider the system capacity and your change data requirements, including archiving and performance issues.

After the PowerExchange Logger is active, you can change the log data set configuration as necessary.
You must balance the following variables:

- Data set size
- Number of data sets
- Amount of archiving

Configure the log data set based on the following factors:

- Resource availability requirements
- Performance requirements
- Type of PowerExchange installation: whether you are running near-real-time or batch replication
- Data recovery requirements

The Logger format utility (EDMLUTL0) formats only the primary space allocation. This means that the Logger does not use secondary allocation. This includes Candidate Volumes and Space, such as that allocated by SMS when using a STORCLAS with the Guaranteed Space attribute.

**Related Topics:**

- “Formatting Log Data Sets” on page 62

---

**Data Set Size Determination**

This section provides criteria for determining the size of the active log data sets.

The maximum size of an active log data set is 2,912 cylinders on 3390 DASD and 3,495 cylinders on a 3380 DASD. The maximum size of an active log data set is limited by the maximum size of the associated data space. The maximum size of data space is approximately 2 GB.

**Factors Affecting Data Set Size**

When determining the size of active log data sets, consider the following factors:

- Informatica recommends that you use the same size for all log data sets. If the PRILOG and SECLOG data sets in the selected active log pair are not the same size, the amount of data that the PowerExchange Logger writes is limited to the size of the smallest data set in the log pair.

- An inverse relationship exists between the size of the log data sets and the archiving frequency. A large data set needs to be archived less often than a small data set. However, the archiving of a small data set takes less time.

- The PowerExchange header adds to the size of change records. For the header size in each record, use approximately 300 bytes plus the key length.

- You should include an overhead rate of 5-10 percent to log data set size. This overhead rate provides space for control information and recovery-related information such as system checkpoints. You can control the frequency of system checkpoints by setting the PowerExchange Logger CHKPT_FREQUENCY parameter.

- The type of change transaction affects if PowerExchange CDC captures a before-image, after-image, or both:
  - For a DELETE, PowerExchange captures the before-image.
  - For an INSERT, PowerExchange captures the after-image.
  - For an UPDATE, PowerExchange captures both the before- and after-images.

- For some data sources such as IMS and VSAM, PowerExchange CDC captures the entire object that contains a change. For example, if a field in an IMS segment changes, PowerExchange captures the entire segment.

**Related Topics:**

- “SYSTEM_OPTIONS Statement” on page 41
Calculating the Data Set Size

Use the following formulas to estimate the size of each active log data set in bytes and then convert that value to tracks and cylinders for space allocation.

- **Formula 1.** To estimate the active log data set size in bytes:

\[
\text{active log data set size in bytes} = \frac{\text{average change record size in bytes}}{\text{number of changes captured per hour}} \times \frac{\text{hours between archiving}}{\text{number of usable bytes per track}} \times (1 + \text{overhead rate})
\]

For the overhead rate, use 5-10 percent.

- **Formula 2.** To convert the active log data set size from bytes to tracks:

\[
\text{active log data set size in tracks} = \frac{\text{active log data set size in bytes}}{\text{number of tracks per cylinder}}
\]

- **Formula 3.** To convert the active log data set size from tracks to cylinders:

\[
\text{active log data set size in cylinders} = \frac{\text{active log data set size in tracks}}{\text{number of usable bytes per track}}
\]

The number of tracks per cylinder and the number of usable bytes per track depend on the type of DASD you use. The following table provides these values for 3390 and 3380 DASD devices. This table applies only to the PowerExchange Logger and is based on the fact that the PowerExchange Logger writes 4-KB blocks.

<table>
<thead>
<tr>
<th>Space Information</th>
<th>Model 3390</th>
<th>Model 3380</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tracks per cylinder</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Usable bytes per track</td>
<td>49,152</td>
<td>40,960</td>
</tr>
</tbody>
</table>

Calculating the Total Space for Each Active Log Data Set - Example

This example uses 3390 DASD and the following assumptions:

- Average change record size including the PowerExchange header = 600 bytes
- Number of changes captured per hour = 40,000
- Hours between archiving = 12
- Overhead rate = 5%
- Number of tracks per cylinder = 15

To calculate the total space for each active log data set:

1. **Use Formula 1 to calculate the size of each active log data set in bytes:**

\[
600 \times 40,000 \times 12 \times (1 + .05) = 302,400,000 \text{ bytes}
\]

2. **Use Formula 2 and Formula 3 to calculate the number of tracks and cylinders to allocate:**

\[
302,400,000 \div 49,152 = 6,152 \text{ tracks}
\]

\[
6,152 \div 15 = 410 \text{ cylinders}
\]
Number of Data Sets

You must specify between two and 31 active log data sets. Consider the following:

- Each active log is held on a single dataspace. After an active log is opened, it remains open as long as the PowerExchange Logger is active. Therefore, the more active logs you allocate, the more dataspaces you have open while the PowerExchange Logger is active.
- If you are running near-real-time replication, consider using a small number of data sets. In near-real-time replication mode, PowerExchange is available continuously, providing continuous replication.
- If you are not concerned about controlling the amount of archiving, specify a greater number of data sets. Although archiving occurs more frequently, it takes less time.

Allocating Restart Data Sets

The installation process creates at least one PowerExchange restart data set (ERDS). Use this procedure to expand the data sets.

You should define dual restart data sets and allocate them to different DASD volumes to ensure recovery in case of a disk failure. The restart data set names must match the data set names that you specify in the ERDS01 and ERDS02 DD statements in the PowerExchange Logger EDMUPARMS module options. To help distinguish restart data sets for different PowerExchange Logger subsystems, include the Logger ID as part of these data sets.

To allocate restart data sets:

1. Make a working copy of the #DEFRDS sample JCL procedure from the SAMPLIB library, and edit the copy as required.

The following table lists the required JCL statements:

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specifies the IDCAMS program.</td>
</tr>
<tr>
<td>SYSPRINT DD</td>
<td>Specifies the output data set for MVS system messages.</td>
</tr>
<tr>
<td>SYSIN DD</td>
<td>Specifies the IDCAMS commands DELETE, SET MAXCC, and DEFINE. For more information about these utility commands, see your IBM documentation.</td>
</tr>
</tbody>
</table>

2. Run the JCL procedure to create and configure the restart data sets.

The following sample JCL (#DEFRDS) shows how to define the restart data set in dual mode:

```
// JOB
// *--------------------------------------------------------------------------*
// * PowerExchange Change Data Capture - ALLOCATE LOGGER RESTART DATASETS
// *--------------------------------------------------------------------------*
// * REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES
// * 1. JCL DATA SET NAMES
// * 2. IDCAMS COMMAND SPECIFICATIONS
// * 3. REPLACE ???? WITH YOUR LOGGER NAME. USING THE LOGGER NAME AS A
// *    DATA SET NAME QUALIFIER PROVIDES A STANDARD TO INDICATE WHICH
// *    DATA SET BELONGS TO WHICH LOGGER.
// *--------------------------------------------------------------------------*
// ALLOCDS EXEC PGM=IDCAMS,REGION=4M
// SYSPRINT DD SYSOUT=* 
// SYSPRINT DD SYSOUT=* 
// SYSIN DD *
DELE (YOUR.??.?.ERDS01) ERASE
DELE (YOUR.??.?.ERDS02) ERASE
SET MAXCC = 0
DEFINE CLUSTER
```
Adding Active Log Data Set Definitions to the Restart Data Set

To add active log data set definitions to the restart data set:

- Determine the size and number of active log data sets required for your organization.
- Adjust the CYL parameters for the active log data sets according to the expected volume of logging.
- Use the IDCAMS parameters to define the active log data sets.
- To help distinguish log data sets from different PowerExchange Logger subsystems, include the subsystem name in the high-level qualifiers of these data sets. You can have a maximum of 31 active logs.
- To help manage the data set definitions, dynamically allocate the active log data sets named in the restart data sets. Use this procedure to create two active sets and one set always available for selection.

The installation process creates definitions for at least three active log data sets. With three data sets allocated, two are active and one is always available for selection. The startup procedure for the PowerExchange Logger dynamically allocates the active log data sets named in the restart data sets. Use this procedure to create additional data set definitions as required for your site. You can have a maximum of 31 active logs.

To help distinguish log data sets from different PowerExchange Logger subsystems, include the subsystem name in the high-level qualifiers of these data sets. Use the IDCAMS parameters to define the active log data sets. Adjust the CYL parameters for the active log data sets according to the expected volume of logging.

Determine the size and number of active log data sets required for your organization.

To add active log data set definitions to the restart data set:

1. Make a working copy of the #ADDLOGS sample JCL procedure from the HLQ.SAMPLIB library, and edit the copy as required.

   In the following JCL example, HLQ and YOUR represent high-level qualifiers that you specified during installation. (The question marks represent the PowerExchange Logger ID associated with the log data sets.) Using the subsystem name in the data set name helps you distinguish between the different log data sets.

   ```
   //*---------------------------------------------------------*
   DEFINE CLUSTER
   (NAME(Your????.ERDS01)  =
   VOLUMES(VVYYYY)  =
   SHAREOPTIONS(2,3)  =
   DATA  =
   (NAME(Your????.ERDS01.DATA)  =
   RECORDS(200)  =
   RECORDSIZE(4089 4089)  =
   CONTROLINTERVALSIZE(4096)  =
   FREESPACE(0 20)  =
   KEYS(0 0 )  =
   INDEX  =
   (NAME(Your????.ERDS01.INDEX)  =
   RECORDS(5 5)  =
   CONTROLINTERVALSIZE(1024)  =
   )
   //*---------------------------------------------------------*
   
   EXEC
   PARM
   
   Specify the EDMLC000 program.
   Include the Logger name, followed by BATCH.
   ```

   Related Topics:

   - “Data Set Size Determination” on page 54
Chapter 4: PowerExchange Logger for MVS

Changing the Size of Active Log Data Sets

2. Stop the PowerExchange Logger.
3. Run the JCL procedure to define the active log data sets.
4. Restart the PowerExchange Logger.

The following sample JCL (#ADDLGOS) shows how to add active log data sets:

```
// JOB
///*---------------------------------------------------------------*
///* PowerExchange CDC - DEFINE ACTIVE LOG DATA SETS TO LOGGER
///*---------------------------------------------------------------*
///* REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES
///* 1. JCL DATA SET NAMES
///* 2. REPLACE ???? WITH YOUR LOGGER NAME. USING THE LOGGER NAME AS A
///* DATA SET NAME QUALIFIER PROVIDES A STANDARD TO INDICATE WHICH
///* DATA SET BELONGS TO WHICH LOGGER.
///*---------------------------------------------------------------*
//DEFLOG EXEC PGM=EMLCOO00,PARM='????,BATCH'
//STEPLIB DD DISP=SHR,DSN=HLQ.LOAD,<<<=PWX LOAD
//EDMPARMS DD DISP=SHR,DSN=YOUR.USERLIB,<<<=EDMSDIR,EDMPARMS
//ERDS01 DD DISP=SHR,DSN=YOUR.????,ERDS01,<<<=PRI RESTART DSN
//ERDS02 DD DISP=SHR,DSN=YOUR.????,ERDS02,<<<=SEC RESTART DSN
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
//DEFINE_LOG
DSNAME=YOUR.????,PRLOG,DS03,  
COPY=PRLOG
END
//DEFINE_LOG
DSNAME=YOUR.????,SECLOG,DS03,  
COPY=SECLOG
END
/*

Related Topics:
- "Configuring the PowerExchange Logger for MVS" on page 39
- "Sample JCL Procedure for the PowerExchange Logger" on page 48
- "Size and Number of Active Log Data Sets" on page 53

Changing the Size of Active Log Data Sets

Use this procedure to change the size of existing active log data sets.
Before you begin, estimate the average active log data set size and the space to allocate for each of these data sets.

To resize the data sets, use the JCL in the #SIZELOG member of the hlq.SAMPLIB member, where hlq the high-level qualifier that you specified during installation. This member contains IDCAMS DEFINE statements for allocating space for the resized active log data sets, such as:

```
DEFINE CLUSTER -
    (NAME (hlq.EDML.PRILOG.DS01)  =
     LINEAR =
     VOLUMES (volser) =
     SHAREOPTIONS (2,3) =
     CYL (nnn) )
     DATA =
     (NAME (hlq.EDML.PRILOG.DS01.DATA) )
```

**Note:** To resize the active log data sets, you must shut down the PowerExchange Logger and stop all capture and extraction tasks.

To change the size of active log data sets:

1. Make a copy of the sample #SIZELOG member in the hlq.SAMPLIB library. This member contains JCL for changing the size of log data sets.
2. Edit the JCL statements in the copy of the #SIZELOG member, as needed.

The following table describes the JCL statements for the IBM IDCAMS program:

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the IDCAMS program so that you can run the IDCAMS ALTER, DEFINE, and REPRO commands, which are specified in the SYSIN DD.</td>
</tr>
<tr>
<td>SYSPRINT DD</td>
<td>Specify the output data set for MVS system messages.</td>
</tr>
<tr>
<td>SYSIN DD</td>
<td>Specify the IDCAMS commands ALTER, DEFINE, and REPRO. For more information about these commands, see your IBM documentation.</td>
</tr>
</tbody>
</table>

The following table describes the JCL statements for the PowerExchange EDMUTIL0 program:

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the EDMLUTL0 program. This program formats the expanded portions of the active log data sets for the PowerExchange Logger.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>Add the PowerExchange CDC load library to the STEPLIB DD concatenation unless you added it to the system LNKLST concatenation.</td>
</tr>
<tr>
<td>PRILOG DD</td>
<td>Specify the active log data set name that you used to create the log data set.</td>
</tr>
</tbody>
</table>

3. Stop all PowerExchange jobs and tasks for which the PowerExchange Logger writes data to or reads data from the active log data sets. These jobs and tasks include the PowerExchange Listener, all ECCRs associated with the PowerExchange Logger, PowerExchange Condense tasks, and PowerExchange netport jobs.
4. After all log reader and writer threads stop, stop the PowerExchange Logger.
5. Customize and run the JCL in the #DISPLOG member of the hlq.SAMPLIB sample library. This JCL uses the PowerExchange Logger batch interface to display the “in-use” active log data sets.
If you want to display only the active log data sets, without the archive data sets, include the following TYPE parameter in the DISPLAY OBJECT=LOG command:

DISPLAY OBJECT=LOG,TYPE=ACTIVE,DSNAME=* END

When you run the batch job, the following output is written to the EDMMSG data set:

LOG START
PXEDMX1725021 EDM Logger BATCH initialization in-progress product level V2.4.04 10/15/2003
PXEDMX1726381 EDM Logger system timestamp for ERDS = 2006.241 16:08:25.95
DISPLAY OBJECT=LOG,TYPE=ACTIVE,DSNAME=* END
PXEDMX1725721 EDM Logger input commands accepted execution started
PXEDMX1726791 EDM Logger LOG ACTIVE report follows:

*Start RBA   End RBA   Log Dsname   Status
000001PA4000 000002A2FFFP EDMUSR.PWX.PRILOG.DS01 REUS
000002A3000 00000034BFPP EDMUSR.PWX.PRLOG.DS02 REUS,IN-USE
000001518000 000001FA3FFP EDMUSR.PWX.PRLOG.DS03 REUS
000001FA4000 0000002A2FFP EDMUSR.PWX.SECLOG.DS01 REUS
000002A3000 00000034BFPP EDMUSR.PWX.SECLOG.DS02 REUS,IN-USE
000001518000 000001FA3FFP EDMUSR.PWX.SECLOG.DS03 REUS

PXEDMX1725061 EDM Logger BATCH Shutdown in progress
PXEDMX1725081 EDM Logger #### TASK EDMLPC0 COMPLETE RC-00
PXEDMX1725081 EDM Logger #### TASK EDMLCP0 COMPLETE RC-00
PXEDMX1725081 EDM Logger #### TASK EDMLRPC COMPLETE RC-00
PXEDMX1725081 EDM Logger #### TASK EDMLRUC COMPLETE RC-00
PXEDMX1725091 EDM Logger BATCH shutdown complete

Note: The PRILOG and SECLOG data sets that have the status of REUS,IN-USE are the in-use active log data sets.

6. To change the size of the active log data sets, run the customized #SIZELOG job.

7. Review the specifications for ARCHIVE_OPTIONS in the SETUPCC2 member of the hlq.RUNLIB library.
   Make any necessary adjustment to accommodate the new size of the active log data sets.

   An archive log data set requires the same amount of space as the active log from which it was created. If you increase the size of the active log data sets and you archive these logs to disk, you might also need to increase the space for the archive log data sets. You specify the amounts of primary and secondary space for archive log data sets in the ARCHIVE_OPTIONS parameter of the EDMUPARM options module. If you change these space amounts, update the corresponding values in the SETUPCC2 member.

   Tip: To change the archive log data set size, run only the first step of the job in the SETUPCC2 member. You do not need to run the second step, which defines the active log data sets to the PowerExchange Logger.

8. Restart the PowerExchange Logger.

9. Restart all of the PowerExchange jobs and tasks that you stopped in step 3.

   Note: If you issue the PowerExchange Logger DISPLAY OBJECT=LOG command immediately after this procedure, the RBA range that is displayed for the active log data sets might not reflect the increased data set size. The PowerExchange Logger does not adjust the RBA ranges to account for additional space until it nears the end of the in-use active log data sets.

RELATED TOPICS:

♦ “Data Set Size Determination” on page 54

Example #SIZELOG Member

The following example #SIZELOG member contains JCL that resizes two PRILOG and SECLOG pairs of active log data sets:

```c
//PXEDMXJG JOB (MYJOB), 'EXPAND LOGS',CLASS=A,MSGCLASS=X,
// MSGLEVEL=(1,1),NOTIFY=SYSUID
//*****************************************************************************
//RENAME EXEC PGM=IDCMS,REGION=0M
//SYSIN DD SYSOUT=*
//SYSIN DD *
ALTER PWX.PRILOG.DS01 -
   NEWNAME(PWX.TEMPLOGI.DS01)
```
ALTER PWX.FRILOG.DS01.DATA - NEWNAME(PWX.TEMPLOG1.DS01.DATA) ALTER PWX.SECLOG.DS01 - NEWNAME(PWX.TEMPLOG2.DS01.DATA) ALTER PWX.FRILOG.DS02 - NEWNAME(PWX.TEMPLOG1.DS02.DATA) ALTER PWX.FRILOG.DS02.DATA - NEWNAME(PWX.TEMPLOG1.DS02.DATA) ALTER PWX.SECLOG.DS02 - NEWNAME(PWX.TEMPLOG2.DS02.DATA) ALTER PWX.SECLOG.DS02.DATA - NEWNAME(PWX.TEMPLOG2.DS02.DATA) /* --------------------------------------------------------------- */ //ALLOCLOG EXEC PGM=IDCAMS,REGION=0M,COND=(0,LT) //SYSPRT DD SYSOUT=* //SYSIN DD * DEFINE CLUSTER - (NAME(PWX.FRILOG.DS01) - LINEAR - STORCLAS(SMSPool) - CYL(300)) - DATA - (NAME(PWX.FRILOG.DS01.DATA)) DEFINE CLUSTER - (NAME(PWX.SECLOG.DS01) - LINEAR - STORCLAS(SMSPool) - CYL(300)) - DATA - (NAME(PWX.SECLOG.DS01.DATA)) DEFINE CLUSTER - (NAME(PWX.FRILOG.DS02) - LINEAR - STORCLAS(SMSPool) - CYL(300)) - DATA - (NAME(PWX.FRILOG.DS02.DATA)) DEFINE CLUSTER - (NAME(PWX.SECLOG.DS02) - LINEAR - STORCLAS(SMSPool) - CYL(300)) - DATA - (NAME(PWX.SECLOG.DS02.DATA)) /* --------------------------------------------------------------- */ //FRILOG EXEC PGM=IDCAMS,REGION=0M,COND=(0,LT) //SYSPRT DD SYSOUT=* //SYSIN DD * REFRO INDDATAster(PWX.TEMPLOG1.DS01) - OUTDDATAster(PWX.FRILOG.DS01) REFRO INDDATAster(PWX.TEMPLOG2.DS01) - OUTDDATAster(PWX.SECLOG.DS01) REFRO INDDATAster(PWX.TEMPLOG1.DS02) - OUTDDATAster(PWX.FRILOG.DS02) REFRO INDDATAster(PWX.TEMPLOG2.DS02) - OUTDDATAster(PWX.SECLOG.DS02) /* --------------------------------------------------------------- */ /* NOTE: */ /* THE FOLLOWING STEPS WILL *NOT* DESTROY THE DATA THAT WAS JUST */ /* COPIED INTO THE LOG DATASETS. INSTEAD, THE UTILITY DETECTS */ /* WHETHER ANY PART OF THE DATASETS HAVE BEEN ALLOCATED BUT NOT */ /* YET FORMATTED, AND ONLY FORMATS *THOSE* PARTS OF THE DATASETS. */ /* --------------------------------------------------------------- */ //FORMATP EXEC PGM=EDMUTL0,REGION=0M,COND=(0,LT) //STEPLIB DD DISP=SHR,DSN=PWX.LOAD //FRILOG DD DISP=OLD,DSN=PWX.FRILOG.DS01 // --------------------------------------------------------------- */ //FORMATS EXEC PGM=EDMUTL0,REGION=0M,COND=(0,LT)
You must format the log data sets as you create them. PowerExchange CDC provides a utility, EDMLUTL0, that you can use to format the log data sets.

To format log data sets:

1. Make a working copy of the #EDMLFMT sample JCL procedure in the HLQ.SAMPLIB library, and edit the copy as required.

   For each log data set, you must include the statements shown in the following table. For example, if your system uses dual logging and two active logs, include four job steps in the utility JCL, one for each primary log and one for each secondary log. See the following sample JCL.

   In the sample JCL, HLQ and YOUR represent high-level qualifiers that you specified during installation. (The question marks represent the PowerExchange Logger ID associated with the log data sets.) Using the subsystem name in the data set name helps you distinguish between the different log data sets.

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the utility EDMLUTL0. This utility processes the log data sets so that they are formatted for change capture.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>Include the PowerExchange CDC load library. If you added the load library to your system's LNKLST concatenation, you do not need to add it to the STEPLIB statement.</td>
</tr>
<tr>
<td>PRILOG</td>
<td>Specify one of the log data set names that you used when you created the log data sets.</td>
</tr>
</tbody>
</table>

2. Repeat Step 1 until you have defined all of the log data sets that you want to format.

   See the following sample JCL for a site with four PowerExchange log data sets, two primary data sets, and two secondary data sets.

3. Run the job.

   The utility processes each data set, formatting it for change capture. The utility formats the data sets according to the following conditions:

   - If the data set is empty when the format utility processes it, the utility formats the entire data set, from the beginning of the data set to the highest-allocated RBA for the log.
   - If the data set contains data when the format utility processes it, the utility formats the data set from the highest used log RBA to the highest allocated log RBA. The utility does not format the existing data in the log data set. This is useful if you want to format a data set when you move or copy it to a different physical location.

   The following sample JCL (#EDMLFMT) shows how to format log data sets:

   //------------------------------------------------------------------------
   // PowerExchange CDC - FORMAT ACTIVE LOG DATA SETS FOR LOGGER
   //------------------------------------------------------------------------
Defining Log Data Sets to the ERDS

The PowerExchange Logger requires an entry in the ERDS to access active and archive log data sets. Use the DEFINE_LOG command to define the active and archive logs to the emergency restart data sets (ERDS).

The PowerExchange installation process defines the active logs when you run job SETUPCC2 in the RUNLIB library. This job runs the PowerExchange Logger in batch mode to create the EDMUPARM module and define the active logs to the ERDS data set.

DEFINE_LOG Command

The DEFINE_LOG command adds log definitions to the emergency restart data set. Use the DEFINE_LOG command to perform the following tasks:

- Add a definition for a new active log to the restart data set.
- Add a definition for a replacement active log to the restart data set.
- Add a definition for a replacement archive log to the restart data set.

The DEFINE_LOG command has the following syntax for active logs:

```
DEFINE_LOG
  DSNName=data_set_name,
  COPY={PRILOG|SECLOG},
  [STARTRBA=X'start_rba',ENDRBA=X'end_rba']
END
```

The DEFINE_LOG command has the following syntax for archive logs:

```
DEFINE_LOG
  DSNName=data_set_name,
  [STARTRBA=X'start_rba',ENDRBA=X'end_rba']
END
```

The following table shows the parameters of the DEFINE_LOG command:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNAMEN</td>
<td>Specifies a log data set name.</td>
<td>The data set name can be up to 44 characters long.</td>
</tr>
<tr>
<td>COPY</td>
<td>Specifies which copy of the active log you are defining. This parameter is valid only when you are specifying active log options.</td>
<td>PRILOG indicates that you are defining a primary log data set for the PowerExchange Logger to use. SECELOG indicates that you are defining a secondary log (backup copy).</td>
</tr>
</tbody>
</table>
### Parameter | Definition | Valid Values
--- | --- | ---
**STARTRBA** | Specifies the log RBA of the beginning of either the replacement active log data set or the replacement archive log data set volume specified by data_set_name. You can obtain the start RBA from messages or by using the PowerExchange Logger DISPLAY command. You must enter this parameter for archive log definitions. It is optional for active log definitions. | Enter up to 12 hexadecimal digits for the start_rba value preceding them with the character X and enclosing them in single quotation marks. If you enter fewer than 12 digits, the PowerExchange Logger adds leading zeros. Use this parameter only for replacement log data sets. 

**ENDRBA** | Specifies the log RBA of the end of either the replacement active log data set or the replacement archive log data set volume specified by data_set_name. You can obtain the end RBA from messages or by using the PowerExchange Logger DISPLAY command. You must enter this parameter for archive log definitions. For active log definitions, this parameter is required if you specified STARTRBA. | Enter up to 12 hexadecimal digits for the end_rba value preceding them with the character X and enclosing them in single quotation marks. If you enter fewer than 12 digits, the PowerExchange Logger adds leading zeros. Use this parameter only for replacement log data sets. 

**END** | Indicates that the input for this command is complete. | This parameter is required.

### RELATED TOPICS:
- "Adding Active Log Data Set Definitions to the Restart Data Set" on page 57

### Deleting Log Data Sets from the ERDS

The DELETE_LOG command deletes all information about a specified log data set from the emergency restart data set (ERDS). For example, you can use this command to delete outdated archive log data sets.

Run the DELETE_LOG command as part of a JCL procedure anytime you need to delete a log.

The DELETE_LOG has the following syntax when issued as a MVS command:

```
F jobname,DELETE_LOG DSNNAME=archive_log_dataset_name
```

The DELETE_LOG has the following syntax when used in batch mode:

```
DELETE_LOG DSNNAME=archive_log_dataset_name END
```

The following table shows the parameters of the DELETE_LOG command:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DSNAME</strong></td>
<td>Specifies a log data set name.</td>
<td>The data_set_name can be up to 44 characters long.</td>
</tr>
<tr>
<td><strong>END</strong></td>
<td>Indicates that the input for this command is complete.</td>
<td>This parameter is required.</td>
</tr>
</tbody>
</table>
The following sample JCL shows how to run the PowerExchange Logger in batch mode to delete an archive log data set. Specify the PowerExchange Logger ID in place of the question marks (????) in the PARM parameter on the EXEC card:

```
//jobname JOB
//DEFLOG EXEC PGM=EDMLC000,PARM='????,BATCH'
//STEP1 DD DISP=SHR,DSN=HLQ_USERLIB
//EDMPARMS DD DISP=SHR,DSN=HLQ_USERLIB
//ERDS01 DD DISP=SHR,DSN=HLQ_USERLIB
//SISPRINT DD SYSOUT=''
//SYSPRINT DD SYSOUT=''
DELETE_LOG
DSNAME=archive_log_dataset_name
END
```

### Recovering Damaged Active Log Data Sets

To recover damaged active log data sets you can use the following procedure. You can delete the damaged set and replace it with a copy of the corresponding backup log data set. You can use this procedure only if you defined dual logging to create a backup.

Before you run the procedure to recover the damaged data sets, you must stop the PowerExchange Logger. Following recovery, restart the PowerExchange Logger.

To recover damaged active log data sets:

1. Make a working copy of the #RCOVADS sample JCL from the HLQ.SAMPLIB sample library (where HLQ is the high-level qualifier you specified at installation), and edit the copy as required.
   
The following table describes the JCL statements in this member.

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>For the allocation step and the REPRO command, specify the IDCAMS program. To format the active log data sets for the PowerExchange Logger, specify the EDMLUTL0 program.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>Include the PowerExchange CDC load library. If you added the load library to your system's LNKLST concatenation, you do not need to add it to the STEPLIB concatenation.</td>
</tr>
<tr>
<td>SYSIN DD</td>
<td>Specify the IDCAMS commands DELETE, SET, DEFINE, and REPRO. For more information about these IDCAMS utility commands, see your IBM documentation.</td>
</tr>
<tr>
<td>PRILOG DD</td>
<td>Specify the log data set names you used when you created the log data sets.</td>
</tr>
</tbody>
</table>

2. Stop the PowerExchange Logger.

3. Run the job to recover the damaged data sets. The following JCL example shows statements that recover damaged active log data sets.

   The following sample JCL (#RCOVADS) shows how to recover damaged active log data sets:

   ```
   // JOB
   // *-------------------------------------------------------------------
   // PowerExchange Change Data Capture - RECOVER PRIMARY LOG FROM SECONDARY LOG
   ```
4. Restart the PowerExchange Logger.

Recovering Damaged Restart Data Sets

Use this procedure to recover a damaged restart data set. You delete the damaged data set and rename a copy of the corresponding backup restart data set to take the place of the damaged data set.

To recover damaged restart data sets:

1. Make a working copy of the #RCOVRDS sample JCL from the HLQ.SAMPLIB sample library (where HLQ is the high-level qualifier you specified at installation), and edit the copy as required.

   The following table describes the JCL statements in this member.

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the IDCAMS program.</td>
</tr>
<tr>
<td>SYSPRINT DD</td>
<td>Specify the output data set for MVS system messages.</td>
</tr>
<tr>
<td>SYSIN DD</td>
<td>Specify the IDCAMS commands DELETE, SET, DEFINE, and REPRO. For more information about these IDCAMS utility commands, see your IBM documentation.</td>
</tr>
<tr>
<td>PRILOG DD</td>
<td>Specify the log data set names you used when you created the log data sets. You created these data sets during installation.</td>
</tr>
</tbody>
</table>

2. Stop the PowerExchange Logger.
3. Run the edited #RCOVRDS job to recover the damaged data sets.

4. Restart the PowerExchange Logger.

The following sample JCL (#RCOVRDS) shows how to recover damaged restart data sets:

```java
//
// JOB
// *---------------------------------------------------------------*
// * PowerExchange Change Data Capture = RECOVERING A RESTART DATA SET*
// *----------------------------------------------------------------------*
// * REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES*
// * 1. JCL DATA SET NAMES*
// * 2. IDCAMS COMMAND SPECIFICATIONS*
// * 3. REPLACE ??? WITH YOUR LOGGER NAME. USING THE LOGGER NAME AS A*
// * DATA SET NAME QUALIFIER PROVIDES A STANDARD TO INDICATE WHICH*
// * DATA SET BELONGS TO WHICH LOGGER.*
// *----------------------------------------------------------------------*
//ALLOCRDS EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE (YOUR.???? .ERDS01) ERASE
SET MAXCC = 0
DEFINE CLUSTER
   (NAME(YOUR.???? .ERDS01) -
   VOLUMES(volser) -
   SHAREOPTIONS(2 3 ) -
DATA
   (NAME(YOUR.???? .ERDS01.DATA) -
   RECORDS(100) -
   RECORDSIZE(4089 4089)
   CONTROLINTERVALSIZE(4096) -
   FREESPACE(0 20) -
   KEYS(4 0 ) -
INDEX
   (NAME(YOUR.???? .ERDS01.INDEX) -
   RECORDS(5 5 ) -
   CONTROLINTERVALSIZE(1024) )
/*
//----------------------------------------------------------------------*
//REPRODS EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
REPRO INDATASET(YOUR.???? .ERDS02) -
   OUTDATASET(YOUR.???? .ERDS01)
/*
```

**Moving Log Data Sets to Other Devices**

You can move log data sets to another device if this is required in your installation. You must stop the PowerExchange Logger and run a procedure that allocates space on the other device and moves the data sets. You can restart the PowerExchange Logger.

To move log data sets to other devices:

1. Make a working copy of the #MOVELOG sample JCL from the HLQ.SAMPLIB sample library (where HLQ is the high-level qualifier specified at installation), and edit the copy as required.
The following table describes the JCL statements in this member.

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>For the ALTER, DEFINE, and REPRO commands, specify the IDCAMS program. To format the active log data sets for the PowerExchange Logger, specify the EDMLUTL0 program.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>Include the PowerExchange Change Data Capture load library. If you added the load library to your system's LNKLIST concatenation, you do not need to add it to the STEPLIB concatenation.</td>
</tr>
<tr>
<td>SYSPRINT DD</td>
<td>Specify the output data set for MVS system messages.</td>
</tr>
<tr>
<td>SYSIN DD</td>
<td>Specify the IDCAMS commands ALTER, DEFINE, and REPRO. For more information about these IDCAMS utility commands, see your IBM documentation.</td>
</tr>
<tr>
<td>PRILOG DD</td>
<td>Specify the log data set names you used when you created the log data sets.</td>
</tr>
</tbody>
</table>

2. Stop the PowerExchange Logger.
3. Run the job to move the data sets.
4. Restart the PowerExchange Logger.

The following sample JCL (#MOVELOG) shows how to move log data sets to other devices:

```plaintext
// JOB
//*----------------------------------------------------------------------*
//* PowerExchange Change Data Capture - MOVING A LOG DATA SET            *
//* REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES          *
//* 1. JCL DATA SET NAMES                                               *
//* 2. IDCAMS COMMAND SPECIFICATIONS                                   *
//* 3. REPLACE ???? WITH YOUR LOGGER NAME. USING THE LOGGER NAME AS A    *
//*     DATA SET NAME QUALIFIER PROVIDES A STANDARD TO INDICATE WHICH    *
//*     DATA SET BELONGS TO WHICH LOGGER.                                 *
//*----------------------------------------------------------------------*
//ALTERLOG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
ALTER YOUR.????..PRILOG.DS01 = NEWNAME(YOUR.????..TEMPLOG.DS01)
ALTER YOUR.????..PRILOG.DS01.DATA = NEWNAME(YOUR.????..TEMPLOG.DS01.DATA)
//----------------------------------------------------------------------
//ALLOCLOG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
DEFINE CLUSTER -  
   (NAME(YOUR.????..PRILOG.DS01) -  
    LINEAR -  
    VOLUMES(VVVVVV) -  
    CYL(CCC) ) -  
    DATA -  
       (NAME(YOUR.????..PRILOG.DS01.DATA) )
//----------------------------------------------------------------------
//REPROLOG EXEC PGM=IDCAMS,REGION=0M
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *
REPRO INDASSET(YOUR.????..TEMPLOG.DS01) = OUTDATASET(YOUR.????..PRILOG.DS01)
/*
//----------------------------------------------------------------------
```
Using Post-Log Merge

In a multi-system MVS environment with shared DASD, it is possible to change a database or VSAM data set on any MVS system. To use PowerExchange CDC in these types of environments, changes must be captured from all MVS systems. This in itself is not sufficient. Changes from multiple MVS systems for the same database or data set must also be merged to reserve the chronological context of the change.

For example, the online CICS system runs on one MVS system but the overnight batch workload, which updates the same VSAM data sets, runs on a different MVS system. In this example, the VSAM data sets are being changed on multiple MVS systems but in a serial fashion (either through CICS or batch). It is also possible to change the same database or data set at the same time (or nearly) on multiple MVS systems. An example of this is an IMS system where it is possible to have changes being made to IMS databases from multiple MVS systems at the same time.

Post-Log Merge is a configuration option of the PowerExchange Logger that allows the change data that has been captured and logged (into multiple Loggers) on multiple MVS systems to be merged and extracted as if it has been captured on a single system.

The multi-Logger merge process is performed by the Post-Log Merge job, also referred to as the Post-Log Merge task. It extracts logged data from each of the PowerExchange Loggers, referred to as member Loggers, and merges this data in the proper chronological order for use by the PowerExchange extraction process. This results in a single merged change stream, which is provided to the extraction process.

Post-Log Merge System Requirements

The collection of all of the member Loggers is referred to as the Post-Log Merge group. To use Post-Log Merge to merge logged data from multiple Loggers running on multiple MVS systems, the following criteria must be met:

- All of the MVS systems running member Loggers in the Post-Log Merge group must be a part of the same base sysplex (parallel sysplex is not required).
- There must be sufficient available XCF groups to support the Post-Log Merge environment. Each member Logger creates an XCF group. The Post-Log Merge job creates an XCF group, which is named by using the PowerExchange Logger ID value. All member Loggers join the Post-Log Merge XCF group.

Therefore, the total number of XCF groups that PowerExchange requires is the sum of all of the member Loggers plus one for the Post-Log Merge XCF group. For example, if you have three member Loggers on three MVS systems, there are four XCF groups created.

- Each PowerExchange Logger XCF group name must be unique within the sysplex. PowerExchange creates the name for the member Logger XCF group by appending the SMF ID of the MVS system to PowerExchange Logger ID value from the LOGGER_NAME parameter in the EDMUPARM module options.

If the SMF ID value for the MVS system on which a member Logger runs is not unique within the Post-Log Merge group, you can specify a unique value to override the SMF ID in the PARM parameter of the EXEC JCL card for the member Logger.
The Logger emergency restart data sets (ERDSnn) and the active log data sets for all member Loggers in the Post-Log Merge group must be on shared DASD.

If the archive logs are on DASD, they must also be on shared DASD. If the archive logs are on TAPE, the tapes must be accessible to the system on which the Post-Log Merge job runs. This applies to all member Loggers in the Post-Log Merge group.

All PowerExchange MVS capture sources that support multi-system access and update can utilize Post-Log Merge. You must run the appropriate capture source ECCR (along with the Agent and the Logger) on each MVS system for which you want the Post-Log Merge job to merged changes.

Note: DB2 data sharing does not require Post-Log Merge. The DB2 IFI 306 interface calls utilized by the DB2 ECCR result in all changes being captured from a database on any system in the data sharing group. Running multiple DB2 ECCRs in a DB2 data sharing group results in changes being captured numerous times.

Post-Log Merge Restrictions

The following restrictions apply to capture and extraction environments that use Post-Log Merge:

- Capture for any synchronous data sources must run on the MVS system where the changes are being made. Synchronous capture data sources include IMS, Batch VSAM, CICS/VSAM, and Datacom Synchronous. You must run an ECCR for the synchronous data sources on every MVS system in the sysplex where changes are made. You must also run a PowerExchange Agent on each system that you run an ECCR, and a Post-Log Merge member Logger on one MVS system. Therefore, the minimum capture environment on any one system includes a PowerExchange Agent, PowerExchange Logger, and an ECCR.

- All log readers must run on the same MVS system as the Post-Log Merge job. Log readers include the PowerExchange Listener, netport jobs, Condense jobs, and the DTLUAPPL utility.

- The DTLUAPPL utility and Condense jobs require that a Post-Log Merge member Logger run on the same system as the Post-Log Merge job.

Post-Log Merge Configuration

There are differences in the Logger setup when using Post-Log Merge. To set up Post-Log Merge on your system, you must make changes to the default PowerExchange installation. In addition to changes to the Logger installation, you must also configure a Post-Log Merge job. The Post-Log Merge job provides the merged view of all of the data captured in the member Loggers.

You should configure your system to use the Post-Log Merge configuration during the initial installation of the Loggers on each system. The Logger id for all member Loggers must be the same.

Note: You cannot change an existing Logger environment that isn't configured for Post-Log Merge to the Post-Log Merge configuration without losing data captured in your Logger.
The following figure shows an example of Post-Log Merge configuration:

![Post-Log Merge Diagram]

**Configuring Post-Log Merge**

The following steps detail the necessary changes required to the installation to create a Post-Log Merge environment.

To configure post-log merge:

1. Define unique Logger data sets for each system.
   
   Member XICDC500 in RUNLIB defines the Logger data sets. Ensure that the Logger active logs and ERDS data sets defined are unique for each Logger that are a part of the Post-Log Merge group.

2. Ensure a unique USERLIB data set is created.
   
   Member SETUPCC1 in RUNLIB creates the USERLIB data set. The default data sets name created is &HLQ..&LOGGER..USERLIB. This pattern might not create a unique USERLIB data set for each Logger. If necessary, change this name to ensure it is unique.

   **Note:** RUNLIB contains many members that refer to this USERLIB and they also need to be changed as well.

3. Create an EDMSDIR module for each USERLIB data set.
   
   Member XICDC600 in RUNLIB creates the EDMSDIR member in USERLIB. This member contains specifications that should be reviewed and changed where required:

   - LOGRGRP= must be changed from N (no Post-Log Merge) to Y (Post-Log Merge)
   - LOGGER= must be the Logger name. This Logger name must be the same for all member Loggers in the Post-Log Merge group.

4. Create a unique EDMUPARM for each USERLIB data set.
   
   Member SETUPCC2 in RUNLIB creates the EDMUPARM member in USERLIB. This member contains specifications that should be reviewed and changed where required:

   - SUFFIX= in SYSTEM_OPTIONS must be a unique number for each member Logger of the Post-Log Merge group
   - LOGGER_NAME= in SYSTEM_OPTIONS must be the Logger name. This Logger name must be the same for all member Loggers in the Post-Log Merge group.
   - PREFIX_COPY1= and PREFIX_COPY2= in ARCHIVE_OPTIONS must be specify unique high-level qualifiers (HLQ) for the archive logs of each member Logger of the Post-Log Merge group.
The variable For the DDNAME of the ERDS, you must use the following format:

```
#POSTLOG.
```

This JCL needs to be customized for your environment. The following example shows sample JCL for Sample JCL for the Post-Log Merge job can be found in the PowerExchange SAMPLIB data set in member connect to member Loggers rather than the Post-Log Merge job.

Note:

Creating the Post-Log Merge Job

Each Post-Log Merge group requires a single Post-Log Merge job. This is a long-running job (just like the Logger) and is generally best setup as a Started Task. This job (or started task) can run on any MVS system within the sysplex.

Note: All log readers (PowerExchange Listener, netport, and Condense jobs) connect to the Post-Log Merge job, which means that they must run on the same MVS system as the Post-Log Merge job. Log writers like ECCRs connect to member Loggers rather than the Post-Log Merge job.

Sample JCL for the Post-Log Merge job can be found in the PowerExchange SAMPLIB data set in member #POSTLOG. This JCL needs to be customized for your environment. The following example shows sample JCL for this job where the Post-Log Merge group is comprised of three member Loggers.

Sample JCL Statements for a Post-Log Merge Job:

```
// JOB
// *-------------------------------------------------------------*
// * POST LOG MERGE
// *-------------------------------------------------------------*
// WARNING: DO NOT PLACE THE SECONDARY ERDS IN THE JOB OR INCORRECT
// RESULTS WILL OCCUR.
// *-------------------------------------------------------------*
// REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES
// 1. JCL DATA SET NAMES
// 2. PRIMARY ERDS FROM EACH LOGGER
// *-------------------------------------------------------------*
//READER EXEC PGM=EDMLCTRD,FARM='TRACEE'
//STEPLIB DD DISP=SHR,DSN=<HLQ>.LOAD  ==> LOAD modules
//EDMPARMS DD DISP=SHR,DSN=<HLQ>.USERLIB  ==> EDMSDIR,EDMGPARM
//ERDS001 DD DISP=SHR,DSN=YOUR.SYSTEM1.ERDS  ==> ERDS OF SYSTEM 1
//ERDS002 DD DISP=SHR,DSN=YOUR.SYSTEM2.ERDS  ==> ERDS OF SYSTEM 2
//ERDS003 DD DISP=SHR,DSN=YOUR.SYSTEM3.ERDS  ==> ERDS OF SYSTEM 3
//SYSPRINT DD SYSOUT=*  <== SYSTEM_OUTPUT
//SYSDUMP DD SYSOUT=*  <== SYSTEM_DUMP
// *-------------------------------------------------------------*
```

Use the USERLIB that has been created for the MVS system on which the Post-Log Merge job runs.

For the DDNAME of the ERDS, you must use the following format:

```
//ERDSnn
```

The variable `nn` represents a two-digit value from 01 to 99. When you set up the Post-Log Merge job, specify only one ERDSnn DD statement, usually the primary one, for each PowerExchange Logger.
Performance Considerations

Post-Log Merge does not impact the performance of the change capture process. All change capture ECCRs connect to the member Logger on their MVS system to write their captured changes.

During change extraction process, if one MVS system or member Logger is running slowly, the log-merge process performed by the Post-Log Merge task for the log readers are impacted. The change extraction process must wait for the data from the slow MVS system/member Logger as the change data from all members must be merged and presented in the proper chronological order.

The Post-Log Merge task reads records from each member Logger’s active log data set as they are written. To ensure the required responsiveness for the extraction process, there are two key performance characteristics of the Post-Log Merge environment to consider:

- Time-based checkpoint frequency in inactive member loggers.
- Dispatching priority of the Post-Log Merge job.

Timed Checkpoint Considerations for Dormant Member Loggers

There are configurations where you should consider increasing the frequency of time-based checkpoint records by reducing the values of TIME_CHKPT_FREQ= and TIMER_INTERVAL= parameters set in EDMUPARM. These parameters control the frequency with which a member Logger produces time-based checkpoint records that are written to the Logger’s active log data set.

To understand why this is necessary and determine appropriate values for these parameters, you must first understand the concept of dormant and quiesced member Loggers.

A member Logger is quiesced if no ECCRs are connected to it because either no ECCR was started or all ECCRs have shut down. In this situation, the member Logger notifies the Post-Log Merge task that it is being quiesced. PowerExchange writes message PWXEDM172552I in the EDMMSG of the member Logger when the Logger enters a quiesced state and writes message PWXEDM172544I when logging is resumed.

A member Logger is dormant if ECCRs are connected to the Logger but they are not supplying any change data to be logged. For example, if the member Logger is running on a system that has only one active CICS/VSAM ECCR but no transactions are running, the member Logger is dormant.

The Post-Log Merge task does not wait for data from quiesced member Loggers. It does, however, wait for data from dormant member Loggers. The active ECCRs that are connected to the dormant member Loggers can send data at any time. The only records written to the active log are time-based checkpoint records.

Time-based checkpoint records are not produced if there are active ECCRs that are writing change data to the member Logger. The record-based checkpoints, referred to as extended checkpoints, are still written to the active log when change data is being recorded. Time-based checkpoint records are produced only in dormant Loggers.

Reducing the TIME_CHKPT_FREQ and, if necessary, TIMER_INTERVAL values can reduce the latency of data that is being extracted from active member Loggers in the Post-Log Merge environment. The default values are TIME_CHKPT_FREQ=30 and TIMER_INTERVAL=100 hundredths of a second, or 1 second. This means that the member Logger produces time-based checkpoint records every 30 seconds if the Logger is dormant.

If you have member Loggers that are occasionally dormant, you should consider at least reducing the TIME_CHKPT_FREQ to a value less than 30. The minimum value for TIME_CHKPT_FREQ is 5, and the minimum value for TIMER_INTERVAL is 50 hundredths of a second. This results in a time-based checkpoint frequency of 2.5 seconds. This lower value reduces the latency of extractions in this type of Post-Log Merge environment.

Note: All checkpoints (time-based or record-based) cause records to be generated in the Logger’s active log data set. In the case of frequently dormant Loggers, you need to balance the space consumed by frequent time-based checkpoints with the desired extraction latency.
Dispatching Priorities

It is recommended that the member Loggers have a dispatching priority (or service class) at least equal to the ECCRs which are writing data to them. This is especially important with transaction-oriented synchronous capture sources (such as IMS, CICS/VSAM) as the change logging process is a part of the transactions path length so delays in logging delay the transaction. The dispatching priority of Post-Log Merge does not impact the capture process. However, the extraction process is dependent upon the responsiveness of the Post-Log Merge task. Based on your extraction needs, the Post-Log Merge job may need to have a higher dispatching priority than standard batch jobs or general started tasks. For example, if you require the best-possible extraction response from the Post-Log Merge task, its dispatching priority (or service class) should be equal to or higher that the PowerExchange Listener (or whatever job is performing the extraction).

Recovery Scenarios

When you run Post-Log Merge, you need to consider the recovery options for the Post-Log Merge job as well as the other PowerExchange CDC components. Consider the following types of recovery scenarios:

- PowerExchange CDC component failures
- MVS system failures

PowerExchange CDC component failures might interrupt capture or extraction processing.

The following table lists the capture and extraction components in the Post-Log Merge configuration and the result if that component fails:

<table>
<thead>
<tr>
<th>Component</th>
<th>Result If the Component Fails</th>
<th>Recovery</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECCR</td>
<td>Capture for that ECCR is interrupted.</td>
<td>Restart the ECCR.</td>
</tr>
<tr>
<td>PowerExchange Agent</td>
<td>Capture registrations cannot be verified.</td>
<td>Restart the PowerExchange Agent.</td>
</tr>
<tr>
<td>PowerExchange Logger</td>
<td>The ECCRs that reside on the same system as the failed PowerExchange Logger also fail.</td>
<td>Restart the PowerExchange Logger and then the ECCRs.</td>
</tr>
<tr>
<td>PowerExchange Listener</td>
<td>The member Loggers and the Post-Log Merge job continue to run. Real-time extraction CDC sessions fail.</td>
<td>Restart the PowerExchange Listener and then restart the failed CDC sessions.</td>
</tr>
<tr>
<td>Post-Log Merge job</td>
<td>The member Loggers continue to run but real-time extraction CDC sessions fail.</td>
<td>Restart the Post-Log Merge job and then restart the failed CDC sessions.</td>
</tr>
</tbody>
</table>

Recovery from MVS System Failures

If an MVS system fails, all PowerExchange components on that system are unavailable. After you IPL the MVS system, normal operation usually resumes. In certain circumstances, you might want to move the PowerExchange CDC components from the failed MVS system to another MVS system, called the destination MVS system.

To quickly reestablish the ability to perform change data extractions, you can move the PowerExchange components that relate to extraction to another MVS system in the sysplex. If you also want to capture new change data, then you must move all of PowerExchange CDC components and, in most cases, the source database system or region. For example, to move the PowerExchange CICS/VSAM capture environment to another system, you must also move CICS region in which the CICS/VSAM ECCR runs.
The following table lists the extraction components in a Post-Log Merge configuration and the considerations for moving these components to another MVS system in the sysplex:

<table>
<thead>
<tr>
<th>Component</th>
<th>Considerations</th>
</tr>
</thead>
</table>
| PowerExchange Listener   | - If a PowerExchange Listener runs on the destination MVS system and uses the same PowerExchange CDC environment, then you can change the NODE statement that points to the failed MVS system in the dbmover.cfg file on the Integration Service machine to point to the PowerExchange Listener on the destination system.  
- If you move the PowerExchange Listener from the failed system, you must either redirect network traffic for the failed MVS system to the destination MVS system or change the NODE statement for the failed MVS system in the dbmover.cfg file on the Integration Service machine to point to the destination MVS system.  
- To restart extraction CDC sessions, you must also move the Post-Log Merge job. |
| Post-Log Merge Job       | - The Post-Log Merge job can be restarted on any MVS system in the sysplex, including systems that do not currently run a member Logger.  
- Move the PowerExchange Agent if there is not one running on the destination MVS system.  
- To restart extraction CDC sessions, you must either move the PowerExchange Listener and redirect network traffic for that PowerExchange Listener or change the NODE statement in the dbmover.cfg file on the Integration Service machine to point to a PowerExchange Listener that runs on the destination MVS system. |

The following table lists the capture components in a Post-Log Merge configuration and the considerations for moving these components to another MVS system in the sysplex:

<table>
<thead>
<tr>
<th>Component</th>
<th>Considerations</th>
</tr>
</thead>
</table>
| ECCR                    | - Only move a synchronous ECCR to another MVS system if the source database region or workload moves. In this case, a PowerExchange Agent and a member Logger must be available on the destination MVS system. If a member Logger of the same Post-Log Merge group runs on the destination MVS system, do not move the PowerExchange Agent and PowerExchange Logger from the failed system.  
- For the Adabas, Datacom table-based, IDMS log-based, and IMS log-based ECCRs, the PowerExchange Agent and PowerExchange Logger from the failed MVS system must be moved to the destination MVS system. The destination system cannot run another PowerExchange Logger that has the same Logger name or is part of the same Post-Log Merge group. The destination MVS system must also run the Post-Log Merge job and the PowerExchange Listener used for change data extraction.  
- For a DB2 ECCR that attaches to a data sharing group, you can only move the ECCR to a destination MVS system that does not have a member Logger that is a part of the same Post-Log Merge group. If so, then you must move the member Logger from the failed system. The destination system must also have a DB2 subsystem that is a member of the same data sharing group. This DB2 subsystem can be one moved from the failed system or one that normally runs on the destination system. If there is a member Logger on the destination system, you cannot move the DB2 ECCR to that system.  
- For a DB2 subsystem that attaches to a non-data sharing DB2 subsystem, the related PowerExchange Agent and PowerExchange Logger must be available on the destination MVS system. The destination MVS system cannot run another PowerExchange Logger that has the same Logger name or is part of the same Post-Log Merge group. You must also move the DB2 subsystem to the destination system. |
| PowerExchange Agent      | None                                                                                                                                              |
| PowerExchange Condense   | - A PowerExchange Logger that is part of the Post-Log Merge group must run on the destination MVS system.  
- The destination MVS system must also run the Post-Log Merge job. |
### Considerations

<table>
<thead>
<tr>
<th>Component</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerExchange Logger</td>
<td>- If no PowerExchange Logger runs on the destination MVS system, then you must also move the related PowerExchange Agent from the failed MVS system.</td>
</tr>
<tr>
<td></td>
<td>- If a member Logger in the same Post-Log Merge group runs on the destination MVS system, do not move another member Logger to that system.</td>
</tr>
<tr>
<td>PowerExchange Listener</td>
<td>If you use the PowerExchange Listener on the failed MVS system to extract change data, then also move the Post-Log Merge job to the destination MVS system.</td>
</tr>
</tbody>
</table>

### Post-Log Merge Job Commands

You can issue commands against the Post-Log Merge job to interrogate the status of the Log Reader process, stop the Post-Log Merge job, or instigate traces for problem determination.

The standard format of these commands uses the MVS operator command `MODIFY` (which can be abbreviated as `F`) as follows:

```plaintext
MODIFY job_name,DISPLAY
```

The `job_name` is the Post-Log Merge JOB name.

Also, you can use the MVS `STOP` command (`STOP job_name`). It has the same effect as the following `MODIFY` command:

```plaintext
MODIFY job_name,QUIT
```

The following table shows the full set of `MODIFY` commands that you can use:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY or DIS</td>
<td>Displays information about Log Reader processes that are connected to the Post-Log Merge task, including what Loggers are being merged, and what the current read location is within each Logger’s data. Information is displayed in the Log.</td>
</tr>
<tr>
<td>STATUS or STAT</td>
<td>Same as for DISPLAY.</td>
</tr>
<tr>
<td>QUIT</td>
<td>Causes Post-Log Merge to terminate. Any active Log Reader processes end abnormally.</td>
</tr>
<tr>
<td>STOP</td>
<td>Same as for QUIT.</td>
</tr>
<tr>
<td>TRACEE</td>
<td>Disables tracing for the Post-Log Merge task.</td>
</tr>
<tr>
<td>TRACES</td>
<td>Activates short-form tracing. No more than 32 bytes of data for each trace entry are produced.</td>
</tr>
<tr>
<td>TRACEL</td>
<td>Activates long-form tracing, which causes the entire trace entry to be produced.</td>
</tr>
</tbody>
</table>
Chapter 5

PowerExchange Condense

This chapter includes the following topics:

- PowerExchange Condense Overview, 77
- Configuring PowerExchange Condense, 77
- Configuring PowerExchange Condense Parameters, 84
- Starting and Stopping PowerExchange Condense, 96
- Controlling PowerExchange Condense, 102
- Backing Up PowerExchange Condense Output Files, 102

PowerExchange Condense Overview

PowerExchange Condense provides the ability to capture change data from the PowerExchange Logger log files to condense files, and make these files available at user-determined intervals.

Changes can be presented in a complete condensed form, providing just the net change over the specified interval. This is called full condense processing. With full condense processing, PowerExchange writes the change data to VSAM condense files, which allows the data set to be keyed so that changes can be condensed. Changes written to full condense files no longer have transactional integrity.

Alternatively, changes can be partially condensed into chronological order without eliminating any of the individual changes made to a record. This is called partial condense processing. With partial condense processing, PowerExchange writes the change data to sequential condense files. Because PowerExchange does not condense the change data with partial condense processing, transactional integrity is maintained.

To extract change data from PowerExchange Condense files, run CDC sessions in batch extraction mode by using PWX CDC Batch application connections.

Configuring PowerExchange Condense

To configure the PowerExchange Condense, you must define a CAPTPARM configuration file for each source type and instance, as defined in a registration group. Also, verify that the **Condense** option is set to **Part** or **Full** in the capture registrations for all source tables.

If you select **Full** for the **Condense** option, the source table or data map must identify at least one column as a key column. Otherwise, the PowerExchange Navigator does not allow you to select the **Full** option.
Restriction: PowerExchange does not support full condense processing for Adabas or IDMS log-based CDC.

If you want the PowerExchange Condense to create separate condense files for one or more groups of tables, create a PowerExchange group definition file that defines groups of capture registrations for the tables.

Related Topics:
- “Configuring Condense Group Definitions” on page 94

Enabling Capture Registrations for PowerExchange Condense Use

PowerExchange Condense captures and logs change data only for capture registrations that have a Status of Active and a Condense value of Part or Full.

To enable capture registrations for PowerExchange Condense use:
1. In the PowerExchange Navigator, open the capture registration.
2. In the Resource Inspector, select Active from the Status list.
3. From the Condense list, select Part or Full.
4. Click Apply.
5. Click File > Save to save the changed capture registration. Alternatively, press CTRL+S.

You must refresh or recycle the ECCR that captures changes for the data source, and recycle the PowerExchange Condense job if it is running.

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

PowerExchange Condense Job

The PowerExchange Condense job can run as a batch job or started task. Usually, a batch job is used to run PowerExchange Condense in batch mode, and a started task is used to run PowerExchange Condense in continuous mode. A batch job can be submitted by a job scheduling system at certain intervals.

A PowerExchange Condense job is comprised of the following unique tasks:
- Controller. This is the job step task and controls the address space and starts the subtasks.
- Condense subtask. This subtask is specifically responsible for condensing data.
- Command handler subtask. This subtask provides the command interface to the Condense job.
- Dump subtask. This subtask provides dump services to the Condense job.

The PowerExchange log contains messages indicating when the various tasks start and end and, generally, from which task a message is being issued.

Condense Operational Modes

You can run a PowerExchange Condense job in batch mode or continuous mode. You specify the mode in the COLL_END_LOG parameter of the CAPTPARM member.

Related Topics:
- “Configuring PowerExchange Condense Parameters” on page 84
Batch Mode

In batch mode, a single condense operation runs and then the Condense job shuts down. Running the Condense job in this manner is well suited for batch applications.

For example, single condense runs might be inserted at appropriate points in an application’s automated schedule following update jobs.

The following sample messages display for batch mode:

```
PWX-06455 Command Handler: received CAPTURE_STARTUP_COMPLETE event.
PWX-06417 Condense: Start to Condense because initialization complete
PWX-09957 CAPI 1/f: Read times out after 10 seconds
PWX-09967 CAPI 1/f: End of log for time 06/08/16 16:59:43 reached
PWX-06415 Condense: Condense completed. Total Records=584, Data=289, UOMs =6
PWX-06416 Condense: Shutting down because Single Condense run completed
PWX-06418 Condense: Closed file EDMUSR.D811.CND.CP060816.T1659144
PWX-06136 Checkpoint taken to file=EDMUSR.D811.CNKFTV0 time=06/08/16 17:00:05
PWX-06420 Condense: Checkpoint done. Sequence=0000035CA140000000000035CA1400000000
PowerExchange Logger=C5C4D4D34D040000035C5EEB00000000
```

In this example, the Condense job ran in batch mode and so it shut down after the first condense operation as indicated by message PWX-06416.

Continuous Mode

In continuous mode, the Condense job runs for a long period, perhaps on a 24-hour basis. In this mode, the condense subtask “sleeps” after each condense operation.

One of the following events triggers the next condense operation:

- The number of minutes defined by the NO_DATA_WAIT parameter elapses.
- You issue a CONDENSE command from the command line. Alternatively, you issue a pwxcmd condense command from a Linux, UNIX, or Windows system to the PowerExchange Condense process running on the z/OS system.
- You issue a FILESWITCH command from the command line. Alternatively, you issue a pwxcmd fileswitch command from a Linux, UNIX, or Windows system to the PowerExchange Condense process running on the z/OS system.
- You issue a SHUTCOND command from the command line. Alternatively, you issue a pwxcmd shutcond command from a Linux, UNIX, or Windows system to the PowerExchange Condense process running on the z/OS system.

In continuous mode, the Condense job does not automatically shut down. Instead, you must shut the Condense job down by using either the SHUTDOWN or SHUTCOND commands.

Condense files become available for reading by the change extraction process after a file switch. File switch processing closes open condense 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.

PowerExchange performs the file switch when the file switch criteria defined by tags FILE_SWITCH_CRIT and FILE_SWITCH_VAL are met or when you issue a FILESWITCH command.

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

The following are sample messages for continuous mode:

```
PWX-06455 Command Handler: received CAPTURE_STARTUP_COMPLETE event.
PWX-06417 Condense: Start to Condense because initialization complete
PWX-09957 CAPI 1/f: Read times out after 10 seconds
PWX-09967 CAPI 1/f: End of log for time 06/08/16 17:14:56 reached
PWX-06421 Condense: 06/08/16 17:15:09 Starting wait on commands for 5 minute
PWX-06417 Condense: Start to Condense because no commands received after 5 minute
PWX-06419 Condense: Doing file switch. Records=1017 Reason=Records criteria met
```
In this example, the Condense job is running in continuous mode so the condense operation runs periodically and waits for the next interval as specified in NO_DATA_WAIT. Message PWX-06421 indicates that the condense operation waits the NO_DATA_WAIT interval, which is five minutes in this example. Message PWX-06417 indicates that the condense operation starts after the NO_DATA_WAIT interval expires. If PowerExchange condenses data in a condense operation, PowerExchange issues message PWX-06420.

**Related Topics:**
- “Configuring PowerExchange Condense Parameters” on page 84

### Configuring PowerExchange Condense JCL

The PowerExchange installation provides two sample Condense jobs in the RUNLIB data set:

- **CONDDDB2.** This JCL is setup for running the Condense job as a batch job.
- **PCNDDDB2.** This JCL is setup for running the Condense job as a started task.

The following example JCL is from the PCNDDDB2 member in RUNLIB:

```plaintext
//PCNDD2  PROC SCERUN=CEE.SCEERUN,
// HLQ=YOUR_HLQ,
// LOGGER=PWXL,
// HLQVS=YOUR_HLQVS,
// RUNLIB=YOUR_HLQ.RUNLIB
//RUN EXEC FGM=DTLCACON,PARM=('')
//*/
//STEPLIB DD DISP=SHR,DSN=4HLQ..LOADLIB
// DD DISP=SHR,DSN=4HLQ..LOAD
// DD DISP=SHR,DSN=4SCERUN
//*/
//EDMPARMS DD DISP=SHR,DSN=4HLQ..4LOGGER..USERLIB
// SYSTCPD EXPLICITLY IDENTIFIES WHICH DATA SET IS TO BE USED TO
// OBTAIN THE PARAMETERS DEFINED BY TCP1P.DATA. THIS DD STATEMENT
// MIGHT BE NECESSARY IF YOUR CONFIGURATION CANNOT BE FOUND USING
// USING THE STANDARD IP SEARCH. CONSULT YOUR NETWORKING SYSTEMS
// PROGRAMMER FOR FURTHER INFORMATION.
//SYSTCPD DD DSN=YOUR.TCP1P.DATA,DISP=SHR
//*/
// CDC DATASETS FOLLOW - WITH SPECIFIC PARMS
//*/
//DTLCACFR DD DISP=SHR,DSN=4HLQVS..CCT
//DTLCACDE DD DISP=SHR,DSN=4HLQVS..CDEP
//DTLCACDC DD DISP=SHR,DSN=4HLQVS..CDCT
//DTLCAMAP DD DISP=SHR,DSN=4HLQVS..DTLCAMAP
//DTLCACFG DD DISP=SHR,DSN=4RUNLIB(CAPTD2)
//*/
//DTLMIGG DD DISP=SHR,DSN=4HLQ..DTLMIGG
//DTLCFG DD DISP=SHR,DSN=4RUNLIB(DMover)
//DTLKHU DD DISP=SHR,DSN=4RUNLIB(LICENSE)
//DTLSIGN DD DISP=SHR,DSN=4RUNLIB(SIGNON)
//DATAMAP DD DISP=SHR,DSN=4HLQVS..DATAMAP
//DTLSLOG DD SYSDUMP=*.
//DTLSLOG0 DD SYSDUMP=*
//SYSDUMP DD SYSDUMP=*
//SYSDUMP DD SYSDUMP=*
//SYSPRINT DD SYSDUMP=*
//CCEEDUMP DD SYSDUMP=*
//EDMNCAP DD DUMMY
```
Condense Input Files

The following topics discuss the DD statements in the PowerExchange Condense JCL that point to input files used by the Condense job to read capture registrations and change data.

**DTLAMCPR**

This DD statement points at the hlqvs.CCT, which is a VSAM KSDS data set containing the capture registrations defined using the Navigator. When the Condense job is started, it processes all active registrations in the CCT requesting condense processing, which match the CAPTPARM parameters DB_TYPE and DBID. For example, if the CAPTPARM specifies DB_TYPE=DB2 and DBID=DSN1, the Condense uses all active DB2 registrations with condense of either Part or Full with an instance name of DSN1.

**Note:** The value for DBID is the value specified when the Registration Group is created. The name of the field in the Registration Group varies based on DB_TYPE. In the case of DB2, the field is called Database Instance. When opening an existing Registration Group in the Navigator, this value is contained in the Instance field in the Registration Group tab in Resource Inspector.

The CCT pointed to by the Condense DTLAMCPR DD statement must be the same CCT pointed to by the PowerExchange Listener, which was used when the capture registration was created.

The CCT must also be the same CCT that is read on behalf of or by the PowerExchange Agent. The recommended Agent setup is to process registrations through the PowerExchange Listener but it is possible for the Agent to read the CCT directly. In either case, this must be the same CCT as used by the Condense job.

**EDMPARMS**

This DD statement points to the hlq.logger.USERLIB data set, which is created during the installation of PowerExchange. This data set contains the EDMSDIR module, which defines the default Agent ID and Logger name and is used to initialize services required by the Log Read API. PowerExchange uses the Log Read API (LRAPI) to access the change data captured by the DB2 ECCR and recorded by the PowerExchange Logger.

**DTLCFG**

This DD statement points at the DBMOVER member of the hlq.RUNLIB data set, which is created during the installation of PowerExchange. The DBMOVER member contains the PowerExchange configuration parameters.

The DBMOVER member includes the CAPI_CONNECTION statements used by the Log Read API (TYPE=LRAP) and the UOW Cleanser (TYPE=UOWC).

The Log Read API (LRAPI) CAPI_CONNECTION statement defines the Agent ID and Logger name to which it connects. PowerExchange uses the UOW Cleanser in conjunction with the LRAPI to reconstruct the UOWs read from the Logger into complete UOWs in the proper chronological order.

The Logger specified in the CAPI_CONNECTION for the LRAPI must be the same that the DB2 ECCR uses (in the EDMSDIR pointed to by the EDMPARMS DD statement) to capture the change data.

Condense Output Files

The following topics discuss the DD statements in the PowerExchange Condense JCL that point to output files used by the Condense job to write condense data and control information about the condense files and messages.

**DTLCACDC (CDCT)**

The hlqvs.CDCT, a VSAM KSDS data set, is written to by the condense process and read by the PowerExchange Listener when extracting condensed change data. This VSAM data set is created during the PowerExchange
installation and is initially primed with a high values (9s) record. After each file switch, tracking records are written with information about each Condense file.

These keyed records contain information about the data that has been condensed such as the condense file name, whether it is a partial or full condense file, start and end times, whether before images are included, the number of records in the file, and other control information.

After each file switch, the Condensed files are closed, tracking records are inserted into the CDCT, and a new checkpoint is taken to the Checkpoint data set. The CDCT tracking records are also written to the checkpoint file. Each time the Condense job is warm started, these records are checked and adjustments are made to the CDCT file if necessary, by either inserting or deleting records.

**Condense Files**

Condense files are created as a part of the condense process in the Condense job. They contain the change data for the active registrations found by the Condense job during initialization.

The names of these data sets are determined by various parameters specified in the CAPTPARM parameters; specifically, EXT_CAPT_MASK and CONDF_FULL_FILE_CTL.

The data set type of the condense files varies based on whether they are partial or full condense files:

- **Partial condense creates variable-blocked sequential data sets.** The data set name has the following format:

  \( hlq.CND.CPyyymmdd.Thhmmssn \)

  Where:
  - \( hlq \) is an EXT_CAPT_MASK value.
  - \( yymmdd \) is a date.
  - \( hhmmss \) is a time.
  - \( n \) is a sequence number, starting at 1, for establishing uniqueness.

- **Full condense creates VSAM KSDS data sets.** The cluster data set name has the following format:

  \( hlq.CND.CFyyymmdd.Thhmmssn \)

  Where:
  - \( hlq \) is an EXT_CAPT_MASK value.
  - \( yymmdd \) is a date.
  - \( hhmmss \) is a time.
  - \( n \) is a sequence number, starting at 1, for establishing uniqueness.

Condense files are read by the PowerExchange Listener or a netport job using the CAPX access method. You can use the PowerExchange Navigator to view the data contained within closed condense files. Open the extraction map that is in the appropriate Extraction Group and select Database Row Test.

The Condense change data can be extracted and processed by a variety of methods, including PowerCenter sessions and workflows.

**RELATED TOPICS:**

- "Configuring PowerExchange Condense Parameters" on page 84

**Checkpoint Files**

The checkpoint files are VSAM KSDS data sets. Their names are determined based on the prefix specified in the CHKPT_BASENAME parameter in CAPTPARM and, if specified, the suffix specified in the template pointed to by CAPTPARM parameter CHKPT_FILE_CTL.
It is possible to run with a single checkpoint data set. This is not advisable as future restart could be compromised. It is recommended that at least 2 checkpoint files are specified for the Condense job in the CAPTPARM parameters.

During initialization of the Condense job, a new checkpoint is taken. The following message, which includes the checkpoint file name and a timestamp, indicates that a checkpoint has been taken:

```
PWX-06136 Checkpoint taken to file=hlq.CHKPTVn time=yy/mm/dd hh:mm:ss
```

This checkpoint reflects the results of merging the current registrations from the CCT file with the information from the last checkpoint of the previous run (if this is a warm start). For a cold start, no data is merged because previous checkpoint files do not exist.

After each FILESWITCH or SHUTDOWN command is issued, a new checkpoint is taken.

In addition to the PWX-06136 message, the following PWX-06420 message displays the contents of the restart tokens at checkpoint:

```
PWX-06420 Condense: Checkpoint done. Sequence=sequence_restart_token Logger=logger_restart_token
```

The following table describes the information that is stored in the checkpoint files:

<table>
<thead>
<tr>
<th>Checkpoint Record Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERT records</td>
<td>Registration tags and restart tokens, which indicate the point to start receiving records from the PowerExchange Logger.</td>
</tr>
<tr>
<td>DCT records</td>
<td>Information which is also held in the CDCT file, describing completed Condensed files. The purpose of this record type is to be able to restore the CDCT file to a consistent point during either cold start or warm start. This information is purged using the number of days defined in CAPTPARM parameter COND_CDCT_RET_P.</td>
</tr>
<tr>
<td>SRT record</td>
<td>A single record defining system-wide information.</td>
</tr>
</tbody>
</table>

**PowerExchange Message Data Sets**

The Condense job prints important information to the following message data sets, which are defined by DD statements in the JCL:

- DTLLOG
- DTLLOGnn (if alternative logging is used)
- DTLOUT
- EDMMSG

The following information assumes that alternative logging, which is the default during the installation of PowerExchange, is being used.

**DTLLOG**

With alternative logging, DTLLOG only contains messages up until the point that the alternative logging subtask is successfully initialized. For the Condense job, this means that it generally only contains the print of the DTLCFG DD statement parameters (DBMOVER).

**DTLLOGnn (PowerExchange Alternative Logging)**

With alternative logging, the standard PowerExchange run-time message information is sent to the alternative log data sets.
These can be DD statements in the JCL of the form DTLOGnn (where nn are numbers from 01 through 99) or dynamically allocated data sets (if no DD statements are provided).

**DTLOUT**
When alternative logging is used, the DTLOUT DD statement only contains messages if there are errors allocating condense files. Without alternative logging, it contains a subset of the messages written to the DTLOG DD statement.

**EDMMSG**
The EDMMSG DD statement is dynamically allocated if it is not included in the JCL. It contains messages from the Log Read API, which connects to the PowerExchange Logger to read the captured change data.
These messages indicate to which PowerExchange Logger and PowerExchange Agent the Condense job attaches as well as the starting point at which to begin, which is passed to the Logger.

Configuring PowerExchange Condense Parameters

The CAPTPARM parameters for PowerExchange Condense are pointed to by the DTLCACFG DD statement in the PowerExchange Condense job or started task.
PowerExchange provides sample parameters for PowerExchange Condense for each data source in the following members of the RUNLIB library:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adabas</td>
<td>CAPTADA1</td>
</tr>
<tr>
<td>DB2</td>
<td>CAPTDB2</td>
</tr>
<tr>
<td>Datacom</td>
<td>CAPTDCOM</td>
</tr>
<tr>
<td>IDMS log-based</td>
<td>CAPTIDML</td>
</tr>
<tr>
<td>IMS</td>
<td>CAPTIMSS</td>
</tr>
<tr>
<td>VSAM</td>
<td>CAPTVSM</td>
</tr>
</tbody>
</table>

If you plan to run multiple PowerExchange Condense jobs, each job must use a unique CAPTPARM member. Each PowerExchange Condense job must have unique checkpoint and condense file data set names.

Parameter Descriptions
This topic describes the PowerExchange Condense parameters that you can specify in the CAPTPARM member.
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. 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. If you enter <strong>AI</strong> for this parameter, the following limitations apply: - You cannot extract before images to the target. - You cannot use DTL_BI columns in extraction maps. - If you add DTL_CI columns to extraction maps, any Insert or Delete operations result in Null values in these columns. Informatica recommends that you specify <strong>BA</strong> so that you have the flexibility to use either <strong>AI</strong> or <strong>BA</strong> for the PowerCenter Image Type connection attribute for extraction processing.</td>
<td>- <strong>AI</strong>. After images only. - <strong>BA</strong>. Before and after images. Default is <strong>AI</strong>.</td>
</tr>
<tr>
<td>CHKPT_BASENAME</td>
<td>Specifies the high-level data set name qualifiers for generating the checkpoint data sets. For example: <strong>INFA.D.CHKPT</strong> Checkpoint data sets are VSAM KSDS clusters. To create the full checkpoint VSAM KSDS cluster name, PowerExchange appends Vn to the last qualifier, where n is a number from 0 to the value of CHKPT_NUM-1. For example: <strong>INFA.D.CHKPTV0</strong> By default, the names of the index and data components of the checkpoint VSAM KSDS clusters are the full cluster names with the suffix .D or .I. The following example is for a data component: <strong>INFA.D.CHKPTV0.D</strong></td>
<td>Maximum length is calculated as 41 - (CHKPT_NUM-1).</td>
</tr>
<tr>
<td>CHKPT_FILE_CTL</td>
<td>Specifies the template file that contains the IDCAMS DEFINE CLUSTER control statements for the checkpoint files. A sample template, TMLCHKPT, is supplied in the RUNLIB library. This parameter is mutually exclusive with: - CHKPT_VOLSERS - CHKPT_PRIM_ALLOC - CHKPT_SCND_ALLOC</td>
<td>A fully qualified sequential data set name or PDS member name. For a PDS member, enclose the entire name string in quotation marks (&quot;).</td>
</tr>
<tr>
<td>CHKPT_NUM</td>
<td>Specifies the number of checkpoint data sets. <strong>Warning</strong>: Decreasing CHKPT_NUM on a warm start can result in a restart from an incorrect location. Decrease CHKPT_NUM only if doing a cold start.</td>
<td>A number from 1 to 999999. Default is 3.</td>
</tr>
<tr>
<td>CHKPT_PRIM_ALLOC</td>
<td>Specifies the primary space allocation for checkpoint files. This parameter is mutually exclusive with CHKPT_FILE_CTL.</td>
<td>Any number greater than 0.</td>
</tr>
<tr>
<td>CHKPT_SCND_ALLOC</td>
<td>Specifies the secondary space allocation for checkpoint files. This parameter is mutually exclusive with CHKPT_FILE_CTL.</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>
| CHKPT_VOLSERS        | Specifies the DASD volume serial numbers where checkpoint data sets are allocated.  
If specified, three VOLSERS must be specified, even if they are all the same.  
This parameter is mutually exclusive with CHKPT_FILE_CTL.          | Valid MVS VOLSERS, such as DSK100, DSK101, DSK102.                           |
| COLL_END_LOG         | Specifies the operational mode of the Condense job.                          | - 0. Continuous mode. After each Condense run, the system waits for the number of minutes defined in the NO_DATA_WAIT parameter and then performs another Condense.  
- 1. Batch mode. The system shuts down after a single Condense run. For example, a single condense run might be scheduled following a particular batch update job.  
Default is 0.                                                         |
| COND_CDCT_RET_P      | Specifies the retention period, in days, for CDCT records and condense files.  
Files older than this period and their corresponding CDCT records are automatically deleted during cleanup processing.  
Cleanup processing occurs during startup, fileswitch, or shutdown processing.  
Tip: Enter enough days for change data to be extracted from the condense files before the files are deleted. | Any number greater than 0.  
Default is 60.                                               |
| CONDENSENAME         | Optional. A name for the command-handling service for a PowerExchange Condense process to which pwxcmd commands are 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 configuration file. | Maximum length is 64 characters.  
No default.                                               |
| CONDF_FULL_FILE_CTL  | Specifies a template file that contains the IDCAMS DEFINE CLUSTER control statements for the full condense files.  
A sample template, TMLCONF, is supplied in RUNLIB. | Any fully qualified sequential data set name or PDS member name.  
If specifying a member name, enclose the entire name string in quotation marks ("). |
| CONDF_PART_BLSKZ      | Specifies the block size for partial condense files.                        | A number from 0 to 32760.  
Default is 0.                                               |
| CONDF_PART_DATACLAS   | Specifies the SMS DATACLAS value for partial condense files.                | Any valid SMS DATACLAS.                                              |
| CONDF_PART_LRECL      | Specifies the logical record length (LRECL) value for partial condense files.  | A number from 4044 to 32756.  
Default is (block size - 4).                                       |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONDF_PART_MGMTCLAS</td>
<td>Specifies SMS MGMTCLAS value for partial condense files.</td>
<td>Any valid SMS MGMTCLAS.</td>
</tr>
<tr>
<td>CONDF_PART_STORCLAS</td>
<td>Specifies the SMS STORCLAS value for partial condense files.</td>
<td>Any valid SMS STORCLAS.</td>
</tr>
<tr>
<td>CONDF_PRIM_ALLOC</td>
<td>Specifies the primary space allocation for condense files, used in conjunction with CONDF_TYPE. This parameter is ignored for full condense files if CONDF_FULL_FILE_CTL is specified.</td>
<td>Any number greater than 0.</td>
</tr>
<tr>
<td>CONDF_SCND_ALLOC</td>
<td>Specifies the secondary space allocation for condense files, used in conjunction with CONDF_TYPE. This parameter is ignored for full condense files if CONDF_FULL_FILE_CTL is specified.</td>
<td>Any number greater than 0.</td>
</tr>
</tbody>
</table>
| CONDF_TYPE              | Specifies the space unit type for condense files, used in conjunction with CONDF_PRIM_ALLOC and CONDF_SCND_ALLOC. This parameter is ignored for full condense files if CONDF_FULL_FILE_CTL is specified.                                      | - CYL. Cylinders.  
  - TRK. Tracks.  
  Default is CYL.                                                                                                                                                                                          |
| CONDF_UNIT              | Specifies the unit for condense files. This parameter is ignored for full condense files if CONDF_FULL_FILE_CTL is specified.                                                                               | Valid MVS generic or esoteric unit name. For example, 3390 or SYSDA.                                                                                                                                       |
| CONDF_VOL               | Specifies the VOLSER for condense files. This parameter is ignored for full condense files if CONDF_FULL_FILE_CTL is specified.                                                                               | Valid MVS VOLSER.                                                                                                                                                                                             |
| CONN_OVR                | Specifies the CAPI_CONNECTION name to use when running PowerExchange Condense.                                                                                                                                 | Valid source CAPI connection name. If you do not specify this name, PowerExchange Condense uses the default connection. adam |                                        |
| DB_TYPE                 | Specifies the data source type.                                                                                                                                                                               | - ADA for Adabas  
  - DB2 for DB2 for z/OS  
  - DCM for Datacom  
  - IDM for IDMS synchronous  
  - IDL for IDMS log-based  
  - IMS for IMS  
  - VSM for VSAM                                                                                                                                                                             |
| DBID                    | Specifies the instance name. When used with DB_TYPE, it defines selection criteria for capture registrations in the CCT file. This value must match the instance name that is displayed in the PowerExchange Navigator for the Registration Group that contains the capture registrations.  
  For DB2, this value is either a DB2 subsystem ID (SSID) or the name of a data-sharing group.                                                                 | Instance name of captured registrations.                                                                                                                                                                   |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EXT_CAPT_MASK</strong></td>
<td>Specifies a unique high-level qualifier (HLQ) that PowerExchange Condense uses to allocate condense data sets. For example: <strong>INFA.D</strong> To create condense data sets, PowerExchange appends the following information for sequential partial condense data sets: <strong>.CND.CFyymmd.CNdmmmm</strong> For VSAM full condense data sets: <strong>.CND.CFyymmd.Tthmmnn</strong> Where: - <strong>yymmd</strong> is year (yy), month (mm), and day (dd). - <strong>hh</strong> is hour. - <strong>mm</strong> is minutes. - <strong>nnn</strong> is a sequence number starting from 001. For example: <strong>INFA.D.CND.CF080718.T11545001</strong> <strong>Warning:</strong> Do not use the same EXT_CAPT_MASK value for multiple Condense tasks. Otherwise, a Condense task might corrupt condense data sets that another Condense task is using. Also, do not reuse an EXT_CAPT_MASK value until the Condense task has finished processing all condense data sets that match the mask.</td>
<td>Verify that this HLQ does not match data sets other than condense data sets on the system. PowerExchange considers any data sets that match this HLQ to be condense data sets, even if they are unrelated to condense processing. Maximum length is 21.</td>
</tr>
<tr>
<td><strong>FILE_SWITCH_CRIT</strong></td>
<td>Specifies whether to use minutes or records for determining when to do an automatic file switch. Used in conjunction with FILE_SWITCH_VAL.</td>
<td>- <strong>M.</strong> Minutes. - <strong>R.</strong> Records. Default is <strong>M.</strong></td>
</tr>
<tr>
<td><strong>FILE_SWITCH_VAL</strong></td>
<td>Specifies the number of FILE_SWITCH_CRIT units at which to do a file switch. For example, if this value is 30, the Condense task performs a file switch every 30 records if FILE_SWITCH_CRIT=R, or every 30 minutes if FILE_SWITCH_CRIT=M. If a condense file contains no data when the FILE_SWITCH_VAL limit is reached, the file switch does not occur.</td>
<td>Any number greater than 0. Default is 30.</td>
</tr>
<tr>
<td><strong>GROUPDEFS</strong></td>
<td>Specifies the fully qualified data set name for the Condense Group Definitions file that defines condense groups. This parameter is required to use condense definition groups.</td>
<td>Any fully qualified data set name or PDS member name. If you specify a member name, enclose the entire name string in quotation marks (*). For example: <strong>GROUPDEFS=&quot;DTLSR.V810.RUN LIB(CONDGRP)&quot;</strong></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>KEY_CHANGE_ALW</td>
<td>Specifies whether changes to the source key columns are allowed for full condense. With DB2, it is possible to do an UPDATE and change any or all key columns in a row. This parameter is not applicable to partial condense.</td>
<td>- N. If a key is changed for a source using full condense, the condense job fails when this is detected during condense processing. - Y. If a key is changed for a source using full condense, the condense job, during condense processing, ignores the change to the key and continue processing. Default is N.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>Specifies the wait period, in minutes, between condense operations when running in continuous mode. If file switching is done on minutes criteria and FILE_SWITCH_VAL is smaller than NO_DATA_WAIT, the wait period is reduced to the smaller of the two values. This parameter is not used if running in batch mode (COLL_END_LOG = 1).</td>
<td>Any number greater than 0. Default is 60.</td>
</tr>
<tr>
<td>NO_DATA_WAIT2</td>
<td>Specifies the wait period, in seconds, for additional data to be received after the end-of-log is reached, indicated by the PWX-09967 message. This parameter sets the Consumer API (CAPI) interface timeout value, which is shown in message PWX-09957. The completion of a condense operation occurs when this number of seconds expires without data being provided by PowerExchange Logger.</td>
<td>Any number greater than 0. 2 seconds is recommended. Default is 600 seconds. The optimal value for the parameter varies according to change data activity on the system: If the parameter is set too low, the Condense operation might end prematurely causing a delay in capturing all available changes to a condense file so they can be extracted. - If the parameter is set too low and a large unit of work for a source not being condensed is encountered by the PowerExchange Logger, the condense operation might also end prematurely because no data is being returned. - If the parameter is set too high, an individual condense operation might never end.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| OPER_WTO                 | Specifies whether condense file close WTO messages are issued. **Note:** File switch processing does not occur for empty condense files.                                                               | - N. PWX-06418 messages are written to the PowerExchange log when a condense file is closed.  
- Y. PWX06418 WTOs are issued when a condense file is closed. These messages can be used by an automation product if desired. The PWX-06418 messages are also written to the PowerExchange log.  
Default is N. |
| RESTART_TOKEN and        | Parameters that define a restart point for starting change data processing when PowerExchange Condense is cold started.  
A restart point is defined by both a restart token and a sequence token.  
Depending on how you set these parameters, PowerExchange Condense processing starts from one of the following restart points during a cold start:  
- If you do not specify these parameters, processing starts from the current end-of-log position.  
- If you enter 0 for both parameters, processing starts from the earliest possible start location in the PowerExchange Logger.  
- If you enter restart token and sequence token values other than 0, processing resumes from the specific restart point defined by these token values. | - Specific restart and sequence token values.  
- 0  
- Not specified. |
| SEQUENCE_TOKEN           |                                                                                                                                                                                                             |                                                                                                                                                                                                           |
| SIGNALLING               | Specifies whether PowerExchange Condense should attempt to handle certain abnormal end conditions, such as ABEND 0C4, SIGSEGV, SIGABEND.                                                                     | - Y. Condense takes automatic action in the event of certain abnormal ends, such as memory corruption (S0C4 ABENDs), and attempt to close down in an orderly manner.  
- N. No automatic trapping of errors is done by Condense. Instead, the operating system uses its default error handling, which is usually to report the offending program line and dump memory.  
Default is N. |
| CONDENSE_SHUTDOWN_TIME   | Specifies the maximum amount of time, in seconds, that PowerExchange Condense waits after receiving a SHUTDOWN command before stopping.                                                                     | A number from 0 to 2147483647. Set this value based on your environment, such as the number of tables being condensed. Default is 600 seconds. |
| OUT                      |                                                                                                                                                                                                             |                                                                                                                                                                                                           |
| VERBOSE                  | Specifies whether PowerExchange Condense issues verbose or terse messages for frequent condense activities such as cleanup, checkpoint, condense, and file switch processing.                                      | - Y. Verbose messaging  
- N. Terse messaging  
Default is Y. |
Controlling Allocation Attributes of Condense Data Sets

There are a number of parameters in the CAPTPARM parameters that can be used to control the allocation of the various data sets created by the Condense job. Specifically, you can control the allocation attributes for the checkpoint files, the partial condense output files, and the full condense output files.

Checkpoint Files

The allocation attributes of the checkpoint files can be controlled in two ways in the CAPTPARM parameters:

- Specifying the data set prefix, space allocation, and volumes using the following parameters:
  - CHKPT_BASENAME
  - CHKPT_VOLSERS
  - CHKPT_PRIM_ALLOC
  - CHKPT_SCND_ALLOC

- Specifying the IDCAMS DEFINE CLUSTER control statements using the CHKPT_FILE_CTL parameter.

Note: The CHKPT_BASENAME parameter is still used to provide the data set prefix for the checkpoint files.

With the exception of CHKPT_BASENAME, the various parameters of the two options are mutually exclusive. This means that you cannot specify the parameters noted in #1 if you specify CHKPT_FILE_CTL. The reverse is also true.

Using CHKPT_FILE_CTL

If you use the CHKPT_FILE_CTL parameter to specify DEFINE CLUSTER control statements, you have some additional flexibility in controlling the allocation attributes of the checkpoint files.

For example, you can:

- Specify SMS DATACLAS, STORCLAS, and MGMTCLAS parameters.
- Change the default suffix for the DATA and INDEX components from D and I, respectively, to some other desired value.
- Specify different CONTROLINTERVALSIZE values to override the default of 32768.

The sample template that follows is provided in RUNLIB member TMLCHKPT:

```plaintext
/* template for PowerExchange chkpt definition */
/* max 35 lines cols 2-80 only, Lines of comments do not count */
/* NAME<<name>> should occur three times */
/* must otherwise be valid define of cluster */
/* KEYS(40 0) is required for smooth running */

DEFINE CLUSTER -

(NAME<<name>>) -

KEYS(40 0) -

RECORDSIZE(4096 32756) -

DATACLASS(dataclas) -

STORAGECLASS(storclas) -

MANAGEMENTCLASS(mgmtclas) -

TRACKS (5 5) -

VOLUMES(volser) -

REUSE -

FREESPACE (20 20) -

SHAREOPTIONS (2 3) -
```

Related Topics:

- “Condense Operational Modes” on page 78
- “Controlling Allocation Attributes of Condense Data Sets” on page 91
- “Configuring Condense Group Definitions” on page 94
The MVS Installation Assistant customizes the values for \textit{DATACLAS}, \textit{STORCLASS}, \textit{MANAGEMENTCLASS}, and \textit{VOLUMES} based on information specified on the install dialog boxes.

When using the template to allocate the checkpoint files, the following restrictions apply:

- The control statements of the \texttt{DEFINE CLUSTER} must be valid IDCAMS control statements as they are passed to IDCAMS as-is (with the exception of the \texttt{NAME} statements).
- The control statements of the \texttt{DEFINE CLUSTER} must be in uppercase.
- The control statements cannot start in column 1 and only 35 lines of input is allowed.
- The PowerExchange variable \texttt{<<name>>} must appear in the \texttt{NAME} parameter of the \texttt{DEFINE CLUSTER}, \texttt{DATA}, and \texttt{INDEX} statements. The variable is populated with the value specified in the \texttt{CHKPT_BASENAME} parameter of \texttt{CAPTPARM}. Ensure that the \texttt{CHKPT_BASENAME} prefix combined with any changes made to the suffix for the \texttt{DATA} and \texttt{INDEX} statements do not exceed 44 characters.
- The \texttt{KEYS} parameter must be specified as shown in the template.
- Comments must start with /* and should only be placed before or after all of the IDCAMS control statements.

\section*{Partial Condense Files}

The allocation attributes of the partial condense files, which are variable-blocked (VB) sequential data sets, are controlled by the following parameters:

- \texttt{EXT\_CAPT\_MASK}
- \texttt{CONDF\_PART\_DATACLAS}
- \texttt{CONDF\_PART\_STORCLAS}
- \texttt{CONDF\_PART\_LRECL}
- \texttt{CONDF\_PART\_BLKSZ}
- \texttt{CONDF\_PRIM\_ALLOC}
- \texttt{CONDF\_SCND\_ALLOC}
- \texttt{CONDF\_VOL}
- \texttt{CONDF\_UNIT}
- \texttt{CONDF\_TYPE}

The only required parameter is \texttt{EXT\_CAPT\_MASK}. Any combination of the remaining parameters is allowed. The following parameters have default values provided by PowerExchange:

- \texttt{CONDF\_PART\_LRECL}. Default is (blocksize - 4).
- \texttt{CONDF\_PART\_BLKSZ}. Default is 0.
- \texttt{CONDF\_PRIM\_ALLOC}. Default is from \texttt{DBMOVER SPACE=} parameter, if specified.
- \texttt{CONDF\_SCND\_ALLOC}. Default is from \texttt{DBMOVER SPACE=} parameter, if specified.
- \texttt{CONDF\_TYPE}. Default is CYL.

If some or all volume and space allocation parameters are omitted, the partial condense file allocations may still succeed, depending upon the MVS/SMS configuration on the system.

It is also possible for the data set allocation to succeed but for the data set to be unusable. For example, if no space allocation parameters are provided in \texttt{CAPTPARM} or \texttt{DBMOVER}, none is passed on the dynamic allocation request. If the MVS system on which this occurs does not have space allocation defaults defined, the data set is created with a primary and secondary space allocation value of 0. The data set is successfully created but when the Condense job attempts to write to this data set, it fails.
**Full Condense Files**

The allocation attributes of the full condense files, which are VSAM KSDS data sets, can be controlled in two ways in the CAPTPARM parameters:

- Specifying the data set prefix, space allocation, and volumes using the following parameters:
  - EXT_CAPT_MASK
  - CONDF_PRIM_ALLOC
  - CONDF_SCND_ALLOC
  - CONDF_VOL
  - CONDF_TYPE

- Specifying the IDCAMS DEFINE CLUSTER control statements using the CONDF_FULL_FILE_CTL parameter.

  **Note:** The EXT_CAPT_MASK parameter is still used to provide the data set prefix for the full condense files.

The only required parameter is EXT_CAPT_MASK. Any combination of the remaining parameters is allowed. The following parameters have default values provided by PowerExchange:

- CONDF_PRIM_ALLOC. Default is 1.
- CONDF_SCND_ALLOC. Default is 1.
- CONDF_TYPE. Default is CYL.

If the CONDF_VOL parameter is omitted, the full condense file allocations may still succeed, depending upon the MVS/SMS configuration on the system.

**Using CONDF_FULL_FILE_CTL**

If you use the CONDF_FULL_FILE_CTL parameter to specify DEFINE CLUSTER control statements, you have some additional flexibility in controlling the allocation attributes of the full condense files.

For example, you can:

- Specify SMS DATACLAS, STORCLAS, and MGMTCLAS parameters.
- Change the default suffix for the DATA and INDEX components from D and I respectively to some other desired value.
- Specify different CONTROLINTERVALSIZE values to override the default of 32768.

The sample template that follows is provided in RUNLIB member TMLCONDF:

```
/* template for PowerExchange full condense data files */
/* max 35 lines cols 2-80 only, Lines of comments do not count */
/* do not put parameters after comments on any line */
/* NAME[<<name>>] should occur three times */
/* must otherwise be valid define of cluster */
/* KEYS(246 0) is required for smooth running */

DEFINE CLUSTER -
  (NAME[<<name>>]) -
  KEYS(246 0) -
  RECORDSIZE(400 32756) -
  DATACLASS(dataclas) -
  STORAGECLASSstorclas -
  MANAGEMENTCLASSmgmtclas) -
  TRACKS (5 5) -
  VOLUMES(1volser) -
  REUSE -
  FREESPACE (20 20) -
  SHAREOPTIONS (2 3)) -

DATA -
  (NAME[<<name>>].D) -

INDEX -
  (NAME[<<name>>].I)}
```
The MVS Installation Assistant customizes the values for DATACLAS, STORCLASS, MANAGEMENTCLASS, and VOLUMES based on information specified on the install dialog boxes.

When using the template to allocate the full condense files, the following restrictions apply:

- The control statements of the DEFINE CLUSTER must be valid IDCAMS control statements as they are passed to IDCAMS as-is (with the exception of the NAME statements).
- The control statements of the DEFINE CLUSTER must be in uppercase.
- The control statements cannot start in column 1 and only 35 lines of input is allowed.
- The PowerExchange variable <<name>> must appear in the NAME parameter of the DEFINE CLUSTER, DATA, and INDEX statements. The variable is populated with the value specified in the EXT_CAPT_MASK parameter of CAPTPARM. Ensure that the EXT_CAPT_MASK prefix combined with any changes made to the suffix for the DATA and INDEX statements do not exceed 44 characters.
- The KEYS parameter must be specified as shown in the template.
- Comments must start with /* and should only be placed before or after all of the IDCAMS control statements.

**Configuring Condense Group Definitions**

If you want PowerExchange Condense to create separate sets of condense files for groups of data, create a group definition file. The file groups data based on groups of capture registrations.

When you use a group definition file, CDC sessions can extract change data more efficiently by targeting a more specific set of condense files.

To use a group definition file with z/OS data sources, you must set the **Condense** option to **Part** in the capture registrations. You cannot use the **Full** condense option.

Also, you must specify the fully qualified data set name for the group definition file in the GROUPDEFS parameter in the CAPTPARM configuration member.

Without a group definition file, PowerExchange Condense processes data for all tables that are registered with the **Condense** option set to **Full** or **Part**. All changes are written to a single set of condense files, not taking into account file-switching. To extract change data from a table with low level of change activity, the extraction process might need to read through a lot of data before finding the changes of interest.

**Condense Group Definition File**

PowerExchange Condense group definitions are defined in a sequential file, called the *group definition file*.

For PowerExchange Condense to create separate sets of condense files for the groups you define, you must enter the path and file name of the group definition file in the GROUPDEFS parameter in the CAPTPARM configuration member.

A group definition file contains one or more GROUP statements, each with one or more REG statements. The following table describes the GROUP and REG statements:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Positional Parameter</th>
<th>Type (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GROUP</td>
<td>group_name</td>
<td>VARCHAR(255)</td>
<td>Identifier for the Condense group.</td>
</tr>
<tr>
<td></td>
<td>external_capture_mask</td>
<td>VARCHAR(21)</td>
<td>Fully-qualified prefix for the name of the data set to contain the condense files for the data group.</td>
</tr>
<tr>
<td>REG</td>
<td>registration_name</td>
<td>VARCHAR(8)</td>
<td>Full or wild-carded registration name (has to be the prefix). Registration names are case sensitive.</td>
</tr>
</tbody>
</table>
The following rules and guidelines apply:

- Each `group_name` value must be unique.
- Each `external_capture_mask` value must be unique.
- Each REG statement applies to the single preceding GROUP statement.
- If a REG statement without a preceding GROUP statement is found, a syntax error is generated.
- You must not specify the same REG statement for more than one GROUP statement.
- If a REG statement is not specified for a group, all of the registrations that belong to that group will be considered for condense processing, as if REG=* had been specified.

**Condense Group Definition Example**

Use this example to learn how you might use a group definition file.

The following capture registrations exist for the specified tables in the DBID named MVSINST:

<table>
<thead>
<tr>
<th>Registration</th>
<th>Table Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>regemp1</td>
<td>COMPANY.EMPLOYEES</td>
</tr>
<tr>
<td>regemp2</td>
<td>COMPANY.EXEMPLOYEES</td>
</tr>
<tr>
<td>regmgr</td>
<td>COMPANY.MANAGERS</td>
</tr>
<tr>
<td>regloc1</td>
<td>COMPANY.UK_LOCATIONS</td>
</tr>
<tr>
<td>regloc2</td>
<td>COMPANY.US_LOCATIONS</td>
</tr>
<tr>
<td>regloc3</td>
<td>COMPANY.JAPAN_LOCATIONS</td>
</tr>
<tr>
<td>regdept1</td>
<td>COMPANY.DEPTS</td>
</tr>
</tbody>
</table>

Based on these registrations, the following example group definition file creates separate sets of condense files for the groups called Personnel, Locations, and Departments:

```plaintext
GROUP=(Personnel, DTLUSR.PERSCOND)
REG=regemp*
REG=regmgr
GROUP=(Locations, DTLUSR.LOCCOND)
REG=regloc*
GROUP=(Departments, DTLUSR.DEPTCOND)
REG=regdept1
```

In this definition file, the asterisk (*) is used a wildcard character. Consequently, the REG=regemp* specification includes both the regemp1 and regemp2 registrations. The REG=regloc* specification includes the regloc1, regloc2, and regloc3 registrations.

**Output Files**

Condense files for data groups are written to data sets that have data set names with the prefix values that are specified by the `external_capture_mask` parameters of the GROUP statements.

Extraction processes can then extract the change data from the condense files in those data sets.
Starting and Stopping PowerExchange Condense

You can start and stop the Condense job several different ways.

Starting Condense

The Condense job can be run as a MVS batch job or as a started task. Generally, continuous mode condense is run as a started task as it is a long-running job whereas batch mode condense is run as a batch job.

If you are running the Condense job as a batch job, it is started by submitting the job to the MVS Job Scheduler using such products as TSO/E, a job scheduler, automation, or other mechanisms that submit batch jobs. Sample JCL for running Condense as a batch job is supplied in RUNLIB member CONDBB2.

If you are running the Condense job as a started task, the PROC needs to be placed into a system PROCLIB. After which, the MVS START command is used to start the Condense job as a started task. Sample JCL for running Condense as a started task is supplied in RUNLIB member PCNDBB2.

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

Before starting the Condense job, verify the following:

- Ensure that the PowerExchange Logger and Agent have already been started.
- Ensure that the checkpoint files are in the desired state:
  - If a cold start is required, no checkpoint files should exist for the mask defined by CAPTPARM parameter CHKPT_BASENAME.
  - If a warm start is required, ensure that all of the checkpoint files created in the last Condense job exist and are available.
- Ensure that the required registrations have been added through the PowerExchange Navigator to the CCT file for the DBTYPE and DBID being used in this run. If required, existing registrations can be disabled or deleted using the Navigator.

**RELATED TOPICS:**

- “Configuring PowerExchange Condense JCL” on page 80
- “Cold Start Processing” on page 96
- “Warm Start Processing” on page 97

Cold Start Processing

When the Condense job is started, it checks for existing checkpoint files using the prefix specified in CHKPT_BASENAME in the CAPTPARM member pointed to by the DTLCACFG DD statement.

If no checkpoint data sets are found, the Condense job cold starts. A Write-To-Operator-with-Reply (WTOR) is issued to confirm the cold start. The following is an example of the WTOR that is issued:

```
*nn PWX06101A No checkpoint files, cold start from specified restart point ? (Y/N)
```

To continue with the cold start, reply Y to the PWX06101A message. The Condense job issues the following WTO to indicate that the request to cold start has been accepted:

```
PWX061031 Cold Start accepted
```

If you reply N to this message, the cold start is canceled and the Condense job ends immediately. The following message is issued in this case:

```
PWX06104W Cold Start declined
```
For each possible checkpoint file (the CHKPT_NUM number), the following message is written to the PowerExchange log (DTLOG or DTLLOGnn if using alternative logging):

FWX-06365 Warning: Checkpoint file chkpt_basenameVn could not be read and was ignored: Checkpoint FILE chkpt_basenameVn Does not exist. OPEN retcodes 268/4/5996

These messages indicate that the Condense job was unable to allocate the checkpoint file because it does not exist.

The point at which the Condense job starts receiving change data from the PowerExchange Logger depends upon whether RESTART_TOKEN and the SEQUENCE_TOKEN are specified in the CAPTPARM and, if so, what values are specified:

♦ If the RESTART_TOKEN and SEQUENCE_TOKEN are not present in the CAPTPARM parameters then the condense starts from the current position in the Logger (the current end-of-log).

♦ If the RESTART_TOKEN and SEQUENCE_TOKEN are present but set to zero then the condense starts from the earliest available point in the PowerExchange Logger. The Logger goes back to the oldest available RBA (or timestamp in Post-Log Merge). Be aware that this could take some time depending upon the number and size of Logger archive logs available.

The following messages are issued in the PowerExchange log (DTLOG or DTLLOGnn if using alternative logging) to indicate that zeroes are provided for the restart tokens:

FWX-06100 Sequence_token 00000000000000000000000000000000000000
FWX-06100 Logger_token 00000000000000000000000000000000000000

♦ If the RESTART_TOKEN and SEQUENCE_TOKEN are set to a specific value, the Logger starts reading from this point, provided the values are a valid restart point. The following messages are issued in the PowerExchange log (DTLOG or DTLLOGnn if using alternative logging) to indicate the restart tokens provided:

FWX-06100 Sequence_token sequence_token_value
FWX-06100 Logger_token restart_token_value

These restart points can be values obtained from utilities DTLUAPPL or DTLUCDEP. They could also be values obtained from previous Condense job runs or provided by Informatica Support (for error recovery situations).

At this point in the initialization process, the other subtasks of the Condense job (dump task, command task, and condense task) are started by the controller task. The restart tokens that are to be used as the starting point for data extraction from the PowerExchange Logger are echoed in the PowerExchange log with the following message:

FWX-06413 Condense: Highest Restart Token. Sequence=sequence_token_value
PowerExchange Logger=Restart_token_value

After the restart point is established, PowerExchange Condense performs cleanup processing for condense files and CDCT entries that are being expired as a result of the cold start, and writes checkpoint information to the current checkpoint file. Then, the initialization is complete as indicated by the following messages in the PowerExchange log:

FWX-06111 Controller: All tasks initialization complete.
FWX-06455 Command Handler: received CAPTURE_STARTUP_COMPLETE event.

Then, the first condense operation is triggered.

**Warm Start Processing**

When the Condense job is started, it checks for existing checkpoint files using the prefix specified in CHKPT_BASENAME in the CAPTPARM member pointed to by the DTLCACFG DD statement. If at least one checkpoint file is found, the Condense job warm starts. The following message is written to the PowerExchange log (DTLOG or DTLLOGnn if using alternative logging) for each Checkpoint data set that is found:

FWX-06038 Checkpoint file chkpt_basenameVn has time yy/mm/dd hh:mm:ss.
This message indicates the latest checkpoint time in that checkpoint file. You may also see the following message if some of the data sets defined by the CHKPT_NUM do not exist:

PWX-06365 Warning: Checkpoint file chkpt basename Vn could not be read and was ignored: Checkpoint FILE chkpt basename Vn Does not exist. OPEN retnoces 268/4/5996

**Warning:** Do not change CHKPT_NUM to a lower value and warm start Condense. This action can cause incorrect warm start processing and duplicate data being condensed. The Condense job only verifies as many checkpoint files as specified in CHKPT_NUM. For example, if the latest checkpoint is in V3 and CHKPT_NUM is changed to 3, only checkpoint files V0, V1, and V2 are checked to determine the latest checkpoint.

After the existing checkpoint file have been read and the latest checkpoint has been determined, the following message indicates which checkpoint file is being used for Condense restart:

PWX-06040 Checkpoint restart using file chkpt basename Vn.

The capture registrations eligible for Condense are processed (as indicated by the PWX-06118 messages) and the warm start complete message is issued:

PWX-06048 Controller: Warm start complete. Tables restored from checkpoint file.

At this point in the initialization process, the other subtasks of the Condense job (dump task, command task, and condense task) are started by the controller task. The restart tokens that are to be used as the starting point for data extraction from the PowerExchange Logger are echoed in the PowerExchange log with the following message:

PWX-06113 Condense: Highest Restart Token. Sequence=sequence_token_value
PowerExchange Logger=restart_token_value

After the restart point is established, cleanup processing occurs for condense files and CDCT entries that are being expired as a result of the cold start, a checkpoint is taken to the current checkpoint file, and the initialization process is now complete. This is indicated by the following messages in the PowerExchange log:

PWX-06111 Controller: All tasks initialisation complete.
PWX-06455 Command Handler: received CAPTURE_STARTUP_COMPLETE event.

Then, the first condense operation is triggered.

**Note:** When a condense operation is in progress, you can shut down the Condense job by issuing the SHUTDOWN command from the command line. The SHUTDOWN command might cause an incomplete UOW being written to the final condense file. When the Condense job is restarted, this is detected and a file switch is done when an end UOW record is encountered. The following messages are issued to indicate this has occurred:

PWX-06414 Condense: Checkpoint ERT shows incomplete UOW on previous partial Condense
PWX-06419 Condense: Doing file switch. Records=nn Reason=1st EndUOW after previous file switch Cdcts=nn
CPU: TotMa=nnnnnnn Diff=nnnnnnn

### Shutting Down Condense

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

**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 report when shutdown is complete. Finally, the Controller shuts down, ending the Condense job.

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

**SHUTCMD**

The SHUTCMD 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.
Alternatively, on a Linux, UNIX, or Windows system, you can issue a pwxcmd shutcond command to a
PowerExchange Condense process running on a z/OS system.
Issue these commands by using the MVS MODIFY (F) command.

RELATED TOPICS:
¨ “Sample Condense Job Messages” on page 99

Using the MVS STOP Command
The Condense job, specifically the Command Handler subtask, does not accept or process the MVS STOP (P)
command.

Canceling the Condense Job
If the Condense job is canceled, it resumes at the most recent complete checkpoint, which have been taken either
at start up or at the last file switch. All processing since that checkpoint is rolled back.
The unwanted CDCT records are deleted and unwanted Condense files are deleted. Some processing time is lost,
but data integrity is preserved.

Sample Condense Job Messages
The following is an example of the messages issued for a continuous Condense job. This Condense job had been
run previous but this time was started with a cold start from the earliest point (zero restart tokens):
PWX-21605 Connection selected CHANGES found from covr< > tag< > type<DB2> int<FALSE> method<CONN_NAME>.
PWX-21605 Connection selected CHANGES found from covr< > tag< > type<DB2> int<FALSE> method<CONN_NAME>.
PWX-06365 Warning: Checkpoint file EDMUSR.D811.CHKPTV0 could not be read and was ignored: Checkpoint
FILE EDMUSR.D811.CHKPTV0 Does not exist. OPEN retcodes 268/4/5896
PWX-06365 Warning: Checkpoint file EDMUSR.D811.CHKPTV1 could not be read and was ignored: Checkpoint
FILE EDMUSR.D811.CHKPTV1 Does not exist. OPEN retcodes 268/4/5896
PWX-06365 Warning: Checkpoint file EDMUSR.D811.CHKPTV2 could not be read and was ignored: Checkpoint
FILE EDMUSR.D811.CHKPTV2 Does not exist. OPEN retcodes 268/4/5896
PWX-06100 Sequence token 0000000000000000000000000000000000000000
PWX-06100 Logger
token 00000000000000000000000000000000
PWX-06103 Cold Start accepted
PWX-06118 Registration loaded: DBName: DSN8 RegName: db2demo1.1 Creator: EDMUSR Table: DB2DEMO1
PWX-06118 Registration loaded: DBName: DSN8 RegName: db2demo2.1 Creator: EDMUSR Table: DB2DEMO2
PWX-06118 Registration loaded: DBName: DSN8 RegName: db2demo3.1 Creator: EDMUSR Table: DB2DEMO3
PWX-06118 Registration loaded: DBName: DSN8 RegName: docuser1.1 Creator: EDMUSR Table: OPB_USERS
PWX-06119 Controller: added new registration tag DB2DSN8db2demo11
PWX-06119 Controller: added new registration tag DB2DSN8db2demo21
PWX-06119 Controller: added new registration tag DB2DSN8db2demo31
PWX-06119 Controller: added new registration tag DB2DSN8docuser11
PWX-06049 Controller: Cold start complete.
PWX-06112 Controller: Starting the capture subtasks.
PWX-06076 Starting Subtask program DTLCDUMP.
PWX-06490 Dump: starting.
PWX-06494 Dump: task is waiting for an event.
PWX-06076 Starting Subtask program DTLCCMD0.
PWX-06450 Command Handler: Starting.
PWX-06076 Starting Subtask program DTLCCND2.
PWX-06404 Condense: Deleting file EDMUSR.D811.CND.CP060816.T1309001.
PWX-06404 Condense: Deleting file EDMUSR.D811.CND.CP060816.T1322002.
PWX-06404 Condense: Deleting file EDMUSR.D811.CND.CP060816.T1604003.
PWX-06412 Condense: Registration Tag=DB2DSN8db2demo11
PWX-06412 Condense: Registration Tag=DB2DSN8db2demo21
PWX-06412 Condense: Registration Tag=DB2DSN8db2demo31
PWX-06412 Condense: Registration Tag=DB2DSN8docuser11
PWX-06413 Condense: Highest Restart Token. Sequence=0000000000000000000000000000000000000000
PowerExchange Logger=00000000000000000000000000000000
Changed 4 sources to earliest sequence token
PWX-09959 CAPI i/f: Earliest UOW restart tokens: Sequence=0000000000000000000000000000000000000000
PowerExchange Logger=C5C4D4D34040EFFFFFFFFFFFFFFFFFFF

Starting and Stopping PowerExchange Condense

99


The following table describes useful messages to look for in the output.

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWX-21605</td>
<td>Indicates the CAPI_CONNECTION that is used (in this case from DTLCFG because covr is blank).</td>
</tr>
<tr>
<td>PWX-06365</td>
<td>Indicates that none of the Checkpoint data sets are found.</td>
</tr>
<tr>
<td>PWX-06100</td>
<td>Shows the restart tokens used for restart.</td>
</tr>
<tr>
<td>PWX-06103</td>
<td>Indicates that the operator responded Y to the PWX06101A WTOR message.</td>
</tr>
<tr>
<td>Message</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>PWX-06118, PWX-06119, PWX-6412</td>
<td>Lists information about each capture registration. The PWX-06119 and PWX-06412 messages list the registration tag. The PWX-06118 message includes: - DBID / instance (DBName:) - Registration name and version - Creator - Table</td>
</tr>
<tr>
<td>PWX-06049</td>
<td>Indicates that the cold start completed successfully.</td>
</tr>
<tr>
<td>PWX-06112</td>
<td>Reports that the Controller is starting the three sub tasks: the Command Handler, the Condense and the Dump sub tasks.</td>
</tr>
<tr>
<td>PWX-06404, PWX-06405</td>
<td>Indicates that old condense files and their CDCT entries are being removed because the restart point is prior to the restart points at which these files were created.</td>
</tr>
<tr>
<td>PWX-06413</td>
<td>Lists the highest restart tokens across all registration tags. Restart tokens contain two components: - Sequence (20 bytes) containing UOW and sub UOW sequencers. - Logger (16 bytes) containing the Logger started task name and the RBA of the last successfully processed UOW.</td>
</tr>
<tr>
<td>PWX-06136</td>
<td>Reports the initial checkpoint which is done before any processing starts. This file is a merge of any checkpoint data brought forward from the last run (if warm start) and any new data being added or deleted from the CCT registrations file.</td>
</tr>
<tr>
<td>PWX-06111</td>
<td>Reports that all sub tasks have successfully completed their initialization.</td>
</tr>
<tr>
<td>PWX-09950</td>
<td>Reports that a successful connection has been made to the Consumer API (CAPI) and the number of registration tags used.</td>
</tr>
<tr>
<td>PWX-06417</td>
<td>Reports that a Condensing has begun and the reason why it started, which can be: - Initialization is complete. - Timeout waiting for commands. - A CONDENSE command was issued to end the wait period: F jobname,CONDENSE - On a Linux, UNIX, or Windows system, a pwxcmd condense command was issued to the PowerExchange Condense process running on the z/OS system.</td>
</tr>
<tr>
<td>PWX-09957</td>
<td>Is issued on the first read from the Consumer API. It reports some parameters used by the interface to the Consumer API. Here it indicates that Condensing stops if no data is received for a maximum of 10 seconds. This parameter is set from CAPTPARM parameter NO_DATA_WAIT2.</td>
</tr>
<tr>
<td>PWX-06419</td>
<td>Indicates that a file switch has occurred and why, which can be: - Number of records reached if FILE_SWITCH_CRIT=R. - Number of minutes reached if FILE_SWITCH_CRIT=M. - A FILES Witch command was received (F jobname,FILESWITCH). - A pwxcmd fileswitch command was received.</td>
</tr>
<tr>
<td>PWX-06418</td>
<td>Indicates the data set name of the Condense file(s) closed by the file switch.</td>
</tr>
<tr>
<td>PWX-09967</td>
<td>Indicates that what was the end-of-log (EOL) at the point the read started has been reached. The NO_DATA_WAIT2 time is now waited to see if there is more data. If not, this condense operation stops.</td>
</tr>
</tbody>
</table>
### Controlling PowerExchange Condense

Use the following commands to control PowerExchange Condense processing or display the status of PowerExchange Condense tasks:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONDENSE</td>
<td>Starts a condense operation instead of waiting for the sleep time to elapse.</td>
</tr>
<tr>
<td>DISPLAY STATUS</td>
<td>Displays the status of the PowerExchange Condense tasks, including the Controller task.</td>
</tr>
<tr>
<td>FILESWITCH</td>
<td>Closes the current log file or files and starts new ones.</td>
</tr>
<tr>
<td>SHUTCOND</td>
<td>Stops a PowerExchange Condense task running in continuous mode without first performing a final condense operation.</td>
</tr>
<tr>
<td>SHUTDOWN</td>
<td>Shuts down a Condense job after a PowerExchange performs a final condense operation.</td>
</tr>
</tbody>
</table>

You can issue these commands by using the MODIFY (F) command on the z/OS system.

Alternatively, use the pwxcmd program to issue condense, displaystatus, fileswitch, shutdown, or shutcond commands from a Linux, UNIX, or Windows system to a PowerExchange Condense process on a z/OS system.

### Backing Up PowerExchange Condense Output Files

Periodically, back up PowerExchange Condense CDCT data set, checkpoint files, 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 \(((2 \times 8) - 1) \times 20 = 300\) 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.
Part III: CDC Sources
Configuration and Management

This part contains the following chapters:

- Adabas Change Data Capture, 105
- Batch VSAM Change Data Capture, 114
- CICS/VSAM Change Data Capture, 122
- Datacom Synchronous Change Data Capture, 130
- Datacom Table-Based Change Data Capture, 139
- DB2 Change Data Capture, 148
- IDMS Log-Based Change Data Capture, 183
- IMS Log-Based Change Data Capture, 196
- IMS Synchronous Change Data Capture, 208
CHAPTER 6

Adabas Change Data Capture

This chapter includes the following topics:
- Introduction to Adabas Change Data Capture, 105
- Adabas Planning Considerations, 105
- Configuring Adabas for Change Data Capture, 107
- Configuring the Adabas ECCR, 108
- Managing Adabas Change Data Capture, 112

Introduction to Adabas Change Data Capture

After configuring and starting the PowerExchange Listener, PowerExchange Agent and PowerExchange Logger and registering Adabas data sources, you can configure the Adabas ECCR.

The following figure displays a configuration for a single Adabas database installation:

Adabas Planning Considerations

Before designing the change data capture environment it is important to analyze requirements. When planning for a production environment, consider the following questions:
- How often do PLOG switches occur?
- How often are changes required for extraction?
What is the volume of changes?
How will the initial load of the data be performed?

Operational Considerations

The following CDC operational considerations apply to PowerExchange Adabas sources:

- PowerExchange imports Long Alpha (LA) fields with a default length of 1,024 bytes. You can override this default length from the PowerExchange Navigator by editing the data map. Open the Record view of an Adabas file and then open the Field Properties dialog box for the LA field. In the Length field, you can enter an override value of up to 16,381.
- The PowerExchange PCAT program, DTLCCADW, can read archived Adabas PLOG records from tape data sets, including data sets that have a block size value greater than 32,760. The Adabas ECCR can then capture change data from those PLOG records.

Accessing Multiple Databases

To capture changes for multiple Adabas databases, you must configure an Adabas ECCR for each Adabas database. The JCL for each Adabas ECCR must reference unique versions of the following:

- PowerExchange Adabas configuration file, which is specified in the DTLCACFG DD statement.
- PowerExchange PLOG Catalog (PCAT) file, which is specified in the DTLADKSD DD statement.
- The Adabas database data sets, which are specified in the DDASSOR1, DDDATAR1, and DDWORKR1 DD statements.

PowerExchange CDC Component Relationships

The Adabas ECCR uses other PowerExchange components, such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- An Adabas ECCR must log all changes to a single PowerExchange Logger running on the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must run on the same MVS system as the Adabas ECCR.
- Operational issues in the PowerExchange Logger can cause the Adabas ECCR to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the Adabas ECCR continues the capture and recording of change data without any loss of data.

You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

Related Topics:

- “Monitoring the PowerExchange Logger for MVS” on page 51
Configuring Adabas for Change Data Capture

PowerExchange provides sample RUNLIB library members to populate the PowerExchange PCAT with information on the latest archived PLOG data sets.

The following alternative sample members are provided in PowerExchange DTLEXPL library.

- **SAMPUEX2** provides PLOG archiving JCL that is submitted from within the Adabas UEX2 exit.
- **SAMPEXTU** provides PLOG archiving JCL that can be submitted as a job from outside the Adabas UEX2 exit.

Depending on the requirements of your site processes, the Adabas DBA must modify, assemble, link, stop, and start the Adabas nucleus, only if the JCL is submitted directly from within UEX2.

The DTLEXPL library contains template members, which may be used to copy and paste the changes into the desired site members. **SAMPEXTU** provides an example of a JCL program called outside of UEX2.

To configure Adabas for CDC:

1. Verify that the ADARUN parameters of the SAMPUEX2 reflect the correct settings for the installation environment. For example:
   
   ```
   ADARUN DB=200,DE=3390,SVC=249,PROG=ADASEL
   ```

2. Replace your current UEX2 with the contents of SAMPUEX2.
   
   Alternatively, you can replace your current PLOG archive JCL with SAMPEXTU.
   
   Verify that the ADARUN parameters of the user exit JCL reflect the correct settings for the installation environment. For example:
   
   ```
   ADARUN DB=DBID,DE=3390,SVC=249,PROG=ADASEL
   ```

3. Perform an Adabas PLOG file switch. The PLOG file switch has two functions:
   
   - It confirms the successful change to the PLOG archive JCL.
   - It verifies that, after the Adabas ECCR is brought up, the PCAT contains an initial archived PLOG data set name entry for subsequent change processing.

   **Note:** PowerExchange creates the PCAT VSAM data set during the installation process if Adabas change capture is selected.

Customizing the PowerExchange Adabas Exit 2 Sample

**SAMPUEX2** contains sample code that you can customize.

- Amend the following code immediately before the comment block for Protection Log flips only:

  ```
  * CLOSE THE INTERNAL READER
  ```

Add the following lines of JCL:

```plaintext
CLI 0(4),EOJ
BNE SUBMIT1
*
   *STR-01*
* End of cards spotted - if this copy is for Command Log, finish -
* but if it's a Protection Log, continue to submit further cards to
* register PLOG into the plog control file...
*
   *STR-01*
CLI CASE,C'P'
BNE CLOSE i.e. it's a CLOG *
   *STR-01*
LA 4,1(4) Skip over first EOJ mark *
   *STR-01*
SUBMIT2 DS 0H *
   *STR-01*
MVC CARD[50],0(4) *
   *STR-01*
PUT INTRDR2,CARD *
   *STR-01*
LA 4,50(4) *
   *STR-01*
CLI 0(4),EOJ LAST CARD PROCESSED ? *
   *STR-01*
BNE SUBMIT2 *
   *STR-01*
*
   *CLOSE THE INTERNAL READER
```
Configuring the Adabas ECCR

Prior to starting the Adabas ECCR, you must configure the ECCR parameter file and the ECCR JCL.

Configuring the Adabas ECCR Parameters

Configure the Adabas ECCR parameters in the RUNLIB library member ADAECRP1. The Adabas ECCR points to these parameters using the DTLACFG DD statement.

Add the following cards to the JCL cards that are immediately before the comment block * READER DCB:

* BELOW ARE PXW ADDITIONAL CARDS
DC CL50'//FLOGCNTL EXEC PGM=DTLCCADW,COND=(4,LT),'
DC CL50'// PARM=(A)' DC CL50'//STEPLIB DD DSN=sceerun,DISP=SHR'
DC CL50'// DD DSN=hlq.LOADLIB,DISP=SHR'
DC CL50'//DTLCCFLG DD DSN=*.COPY.DSDIAUS1,DISP=SHR'
DC CL50'//DTLCCADA DD DSN=hlq.DDbid.PCAT,'
DC CL50'// DISP=SHR'
DC CL50'//DTLcfg DD DSN=hlq.runlib(DMOVER),'
DC CL50'// DISP=SHR'
DC CL50'//DD DSN=hlq.DTLMG,'
DC CL50'// DISP=SHR'
DC CL50'//DTLKEY DD DSN=hlq.runlib(LICENSE),'
DC CL50'// DISP=SHR'
DC CL50'//DTLSGN DD DSN=hlq.runlib(SIGNON),'
DC CL50'// DISP=SHR'
DC CL50'//DD SYSOUT=*
DC CL50'//SYSUOMP DD DUMMY'
DC CL50'//SYSPRINT DD SYSOUT=*
ENDALL DC ALL('EOJ')
* END OF PXW ADDITIONAL CARDS
The following table describes the Adabas ECCR parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADASEL_DSN</td>
<td>Name of a dataset that is set up by the user to contain ADASEL parameters.</td>
<td>String representing the name of a dataset.</td>
</tr>
<tr>
<td></td>
<td>These are propounded to those that are generated by DTLCCADA.</td>
<td></td>
</tr>
<tr>
<td>COLDSTART</td>
<td>Indicates whether to perform a cold or warm start.</td>
<td>Y. Directs the ECCR to perform a cold start, which means it starts processing</td>
</tr>
<tr>
<td></td>
<td>Use any of the following methods to invoke a cold start of the Adabas ECCR:</td>
<td>from the first (oldest) log in the PCAT.</td>
</tr>
<tr>
<td></td>
<td>- Code COLDSTART=Y in the RUNLIB(ADAECRP1) member. This should be pointed to</td>
<td>N. Directs the ECCR to perform a warm start, which means it continues</td>
</tr>
<tr>
<td></td>
<td>by the DTLCCFG DD in the JCL.</td>
<td>processing where it left off. Default is N.</td>
</tr>
<tr>
<td></td>
<td>- Start the ECCR using a new PowerExchange Logger that the Adabas ECCR has</td>
<td></td>
</tr>
<tr>
<td></td>
<td>never connected to.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Change the value of ECCRNAME in the HLQ.RUNLIB(ADAECRP1) member. This</td>
<td></td>
</tr>
<tr>
<td></td>
<td>should be pointed to by the DTLCCFG DD in the JCL.</td>
<td></td>
</tr>
<tr>
<td>COLL_END_LOG</td>
<td>The ECCR execution can be controlled by a combination of parameters</td>
<td>0. The number of PLOGs processed has no influence on whether the collector</td>
</tr>
<tr>
<td></td>
<td>COLL_END_LOG, NO_DATA_WAIT, and NO_DATA_WAIT2.</td>
<td>shuts down.</td>
</tr>
<tr>
<td></td>
<td>These parameters can be combined to ensure the ECCR runs continuously or</td>
<td>- Any number (n) greater than 0. If greater than 0 this specifies the number</td>
</tr>
<tr>
<td></td>
<td>closes down after a specified number of PLOGs are processed.</td>
<td>of PLOGS to be processed before closing down. Default is 0.</td>
</tr>
<tr>
<td>DBID</td>
<td>Used in conjunction with DB_TYPE, it defines selection criteria for which</td>
<td>Collection Identifier used on registrations.</td>
</tr>
<tr>
<td></td>
<td>registrations in the CCT file to be processed.Adabas Database ID.</td>
<td></td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Database type</td>
<td>ADA</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ECCRNAME               | Required. The ECCR name for the Adabas ECCR. The ECCR name value must be unique within a PowerExchange Logger group. *Warning:* If you change the ECCRNAME value, the ECCR cannot warm start from the last stopped position. The Adabas ECCR uses the value specified for the following purposes:  
  - The ECCR name that connects to the PowerExchange Logger to write change data  
  - The member name that joins the XCF group of the PowerExchange Logger  
  - As part of the ECCR-UOW field in the control information for each change record written to PowerExchange Logger log files  
  *Tip:* Informatica recommends that you use the same value for the ECCRNAME parameter and the Adabas ECCR started task or job name. This practice allows you to easily identify the Adabas ECCR when reviewing messages and data from the PowerExchange Logger. | 1 through 8 alphanumeric characters. Default is PWXAD1EC.                                     |
| IGNORENOCHANGEUPDATES | Controls whether the Adabas ECCR ignores records for which update operations did not change the data. You can use this parameter to have the Adabas ECCR ignore the many unchanged records that are typically produced by the ADAORD utility for online reorder operations. When you reorder Adabas files, Adabas logs the before and after images of unchanged records to PLOG files. The ECCR captures the unchanged records from the PLOG files unless you instruct the ECCR to ignore these records. | *Y.* The Adabas ECCR checks the before image and after image of the source data to determine if the data changed and then passes only the changed records to the PowerExchange Logger. The ECCR ignores records for which data did not change. This setting can reduce the number of records that are sent to the PowerExchange Logger.  
 *N.* The Adabas ECCR passes all records to the PowerExchange Logger, including the records with unchanged data. Default is N. |
| NO_DATA_WAIT           | The ECCR execution is controlled by a combination of parameters COLL_END_LOG, NO_DATA_WAIT, and NO_DATA_WAIT2. You can combine these parameters to ensure that the ECCR runs continuously or closes down after a specified number of PLOGs are processed. | *0.* Shut down the ECCR as soon as all PLOG entries in the PCAT are processed.  
 *Any number (n) greater than 0.* Wait n minutes before checking for new PCAT entries. After the initial wait, NO_DATA_WAIT2 controls subsequent waits. |
| NO_DATA_WAIT2          | Specifies the number of seconds for the ECCR to wait for new PLOGs to be entered into the PCAT after processing all existing entries. The ECCR continues to retry every NO_DATA_WAIT2 period until the ECCR is stopped if COLL_END_LOG is 0 and NO_DATA_WAIT is greater than 0. | *Any number greater than 0.* Default is 600. |
Configuring the Adabas ECCR JCL

PowerExchange provides a sample Adabas ECCR PROC called ECCRADA in the RUNLIB library.

Installation job XIZZZ998 copies the ECCRADA member to the PowerExchange PROCLIB library as xxxAD1EC. The variable xxx is the *PowerExchange Agent / Logger Prefix* value specified in the MVS Installation Assistant.

Customize the following statements in the Adabas ECCR JCL:

```
//DTLCACFG DD DISP=SHR,DSN=4RUNLIB(ADACRP1)
//DTLADKSD DD DISP=SHR,DSN=4HLQVS..DDbid.PCAT
//DDASSGR1 DD DISP=SHR,DSN=adanas.ASSOR
//DDDATA1 DD DISP=SHR,DSN=adanas.DATA
//DDWORK1 DD DISP=SHR,DSN=adanas.WORK
```

Run a separate Adabas ECCR with unique Adabas ECCR parameters for each Adabas database from which change data is captured. The member that holds the parameters is specified in the DTLCACFG DD statement.

To configure the Adabas ECCR JCL:

1. Verify that the ADARUN parameters in the ADACARD1 member of the RUNLIB library reflect the correct settings for the installation environment. For example:
   
   ADARUN DB=dbid,DE=3390,SVC=249,PROG=ADASEL

2. Run the Adabas ECCR as a started task or batch job.
   
   A sample Adabas ECCR batch job, ECCRADA, is delivered in RUNLIB. The job XIZZZ998 copies the ECCRADA member to the PowerExchange PROCLIB library as xxxAD1EC, where xxx is the *PowerExchange Agent / Logger Prefix* value that was specified in the MVS Installation Assistant.

3. Verify that the Adabas ECCR DBID parameter is correct in the RUNLIB library member ADACRP1.

   The DBID parameter must be the same as the collection identifier used for the registration group containing the capture registrations in the PowerExchange Navigator.

4. If you use PowerExchange Condense, verify that the DBID parameter in the RUNLIB(CAPTADA1) member is correct.

   The DBID parameter must be the same as the collection identifier used for the registration group containing the capture registrations in the PowerExchange Navigator.

Testing the Installation

Use the following procedure to test the installation.

To test the installation:

1. Perform an update to the Adabas file that was registered in the PowerExchange Navigator.

2. Perform a PLOG switch.

3. Look at the PLOG switch job output to confirm condition code 0 on both the PLOG Copy and PCAT population steps. Note the name of the newly created archived PLOG data set name.

4. Look at the Adabas ECCR job output to note the collected changes. To verify that collection occurred, look in DD name EDMMSG for the message that begins with:

   FWXEDM1728081 Change Capture active for PowerExchange:

   **Note:** The ECCR determines whether it is time to collect archived PLOG data and move the data to the PowerExchange Logger (based on the existence of new PCAT entries) when the following events occur:

   - The ECCR is first started.
   - The criteria in the NO_DATA_WAIT and NO_DATA_WAIT2 parameters are met.
In summary, the first parameter specifies the number of minutes the ECCR will wait before doing another read after the ECCR received an end of PCAT file condition (which means there are no more archived PLOGs to process at the time).

On receiving another end-of-file condition on the first read following the previous end-of-file, the ECCR will wait NO_DATA_WAIT2 seconds before retrying another read (over and over again). These parameters are located in the RUNITLIB(ADAECRP1) member.

5. Look at the PowerExchange Logger output to verify that the archived PLOG was read.

Look in DDNAME EDMMSG for the message that begins with:

```
PWXEDM1727741 Event Mark generated by ECCR xxxADIEC for:
Finished with Plog copy ADABAS.DB00199.PLOG.G0022V00
```

Example archived PLOG. This can also be verified by reviewing the PCAT file and locating the archived PLOG data set name.

6. If you do not use PowerExchange Condense, perform a database row test in the PowerExchange Navigator, as follows:
   a. Open the extraction map.
   b. Click File > Database Row Test.
   c. Specify CAPXRT in DB_Type, and click Go.

7. If you use PowerExchange Condense, issue the fileswitch command to make the condense file available for extraction processing.

Look at the PowerExchange Condense job output to determine the records added to the condense file. Review the PowerExchange log file to find the PWX-06415 message that contains information about a completed condense. Then, perform a database row test in the PowerExchange Navigator. Specify CAPX in DB_Type, and click Go.

Managing Adabas Change Data Capture

You manage Adabas CDC processing by using the Adabas ECCR.

Starting the Adabas ECCR

To start the Adabas ECCR, issue the MVS START command with the name of the started task, such as:

```
START PWXADIEC
```

The Adabas ECCR can also be run as a batch job.

Start the Adabas ECCR after starting the PowerExchange, Listener, PowerExchange Agent, and PowerExchange Logger. The Adabas ECCR terminates with a return code 8 if there are no active Adabas capture registrations. PowerExchange issues messages about active registrations to the PowerExchange log file.

The Adabas ECCR issues message DTL07901 as a WTOR to the MVS operator console, requesting confirmation for cold start processing in the following cases:

- The ECCR is being started for the first time
- The ECCRNAME statement in the Adabas ECCR parameters specifies a new name for the Adabas ECCR
- COLDSTART=Y is specified in the Adabas ECCR parameters
Stopping the Adabas ECCR

To stop the Adabas ECCR, issue the MVS STOP command with the name of the started task or batch job, such as:

STOP PWXAD1EC

Using the DTLCCADW Utility

The PCAT utility program, DTLCCADW, is used by the Adabas ECCR process to manipulate the contents of the PCAT file. The PCAT utility is controlled by settings of the parameters passed through the PARM= on the EXEC statement. There are examples of the JCL required for each function in the PowerExchange DTLEXPL library with names DTLCCADx, where x corresponds to the parameter value.

Typically, these functions are used only internally by PowerExchange. However, there may be times when manual overrides are desired. When in doubt about usage, contact Informatica Global Customer Support.
Batch VSAM Change Data Capture

This chapter includes the following topics:

- Introduction to Batch VSAM Change Data Capture, 114
- Configuring Batch VSAM Jobs for Change Data Capture, 116
- Managing Batch VSAM Change Data Capture, 118
- Managing VSAM Schema Changes, 121

Introduction to Batch VSAM Change Data Capture

PowerExchange for VSAM CDC synchronously captures changes made in batch jobs to VSAM data sets registered for capture. PowerExchange captures changes made to registered VSAM data sets when the batch job is configured to run the batch VSAM ECCR. The batch VSAM ECCR captures changes from GET, PUT, and ERASE requests for registered VSAM data sets.

The batch VSAM ECCR runs in the same address spaces as the batch job that makes changes to VSAM data sets. It captures changes as they occur using a VSAM JRNAD exit and passes the changes to the PowerExchange Logger for recording. After the batch program opens the VSAM data set, PowerExchange records a single unit of work (UOW) in the PowerExchange Logger for all changes the batch program makes to that VSAM data set. PowerExchange commits the UOW containing the changes for the VSAM data set when the batch program closes the VSAM data set.

Relationships with Other PowerExchange Components

The batch VSAM ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- The batch VSAM ECCR must log all changes to a single PowerExchange Logger running in the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must both run on the same MVS system as the ECCR.
If you use Post-Log Merge option of the PowerExchange Logger, PowerExchange allows you to capture and propagate changes even if the changes originate from different MVS systems. You must run a PowerExchange Logger on each MVS system that makes changes to the source VSAM data sets.

Operational issues in the PowerExchange Logger can cause the batch job to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, PowerExchange continues the capture and recording of change data without any loss of data.

You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

**RELATED TOPICS:**
- "Monitoring the PowerExchange Logger for MVS" on page 51
- "Using Post-Log Merge" on page 69

**Batch VSAM ECCR Restrictions**

The batch VSAM ECCR does not support the following items:

- Multiple task control block (TCB) environments
- Natively updated alternate indexes
- Records larger than 32,660 bytes
- Spanned ESDSs
- Paths defined over ESDSs
- Control interval (CI) mode updates
- VSAM data sets opened with record-level sharing (RLS) protocols
- Applications that use request parameter lists (RPLs) that are coded with OPTCD=ASY for asynchronous processing for VSAM files

If you use these applications, unpredictable results can occur.

The batch VSAM ECCR uses an internal exclude table to exclude VSAM data sets from change data capture processing. This internal exclude table contains the following types of entries:

- Complete load module names
- Prefixes for load module names
- Prefixes for data set names

The batch VSAM ECCR does not capture changes for the following data sets:

- VSAM data sets that begin with any data set prefix in this table
- VSAM data sets that are opened by any load modules that match specific load module names or begin with any load module prefix in this table

The following table lists the load module names and prefixes included in the batch VSAM ECCR internal exclude table:

<table>
<thead>
<tr>
<th>Load Module Name or Prefix</th>
<th>Generic or Specific</th>
<th>Excludes Product, Component, or Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CRLFSM</td>
<td>Specific</td>
<td>ASG Software Solutions ASG-TMON</td>
</tr>
<tr>
<td>$TMONTMP</td>
<td>Specific</td>
<td>ASG Software Solutions ASG-TMON</td>
</tr>
</tbody>
</table>
### Configuring Batch VSAM Jobs for Change Data Capture

To configure batch jobs to use the batch VSAM ECCR, you must update the batch job JCL to add the PowerExchange libraries and activate the batch VSAM ECCR interface.

<table>
<thead>
<tr>
<th>Load Module Name or Prefix</th>
<th>Generic or Specific</th>
<th>Excludes Product, Component, or Data Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACF2</td>
<td>Generic</td>
<td>Data sets prefixed with ACF2</td>
</tr>
<tr>
<td>ARC</td>
<td>Generic</td>
<td>IBM DFSMShsm</td>
</tr>
<tr>
<td>DFH</td>
<td>Generic</td>
<td>IBM CICS Transaction Server</td>
</tr>
<tr>
<td>DFSMVRC0</td>
<td>Specific</td>
<td>IBM IMS - Online control region</td>
</tr>
<tr>
<td>DSI</td>
<td>Generic</td>
<td>IBM Tivoli Netview for z/OS</td>
</tr>
<tr>
<td>DSN</td>
<td>Generic</td>
<td>IBM DB2 for z/OS</td>
</tr>
<tr>
<td>EDML</td>
<td>Generic</td>
<td>PowerExchange Logger</td>
</tr>
<tr>
<td>EDMSTART</td>
<td>Specific</td>
<td>PowerExchange Agent</td>
</tr>
<tr>
<td>ERB</td>
<td>Generic</td>
<td>IBM Resource Measurement Facility (RMF)</td>
</tr>
<tr>
<td>FDR</td>
<td>Generic</td>
<td>Innovation Data Processing FDR</td>
</tr>
<tr>
<td>GIM</td>
<td>Generic</td>
<td>IBM SMP/E for z/OS</td>
</tr>
<tr>
<td>IEF1IC</td>
<td>Specific</td>
<td>IBM z/OS - MVS Initiator</td>
</tr>
<tr>
<td>JMPMAINT</td>
<td>Specific</td>
<td>BMC Software JOURNAL MANAGER PLUS</td>
</tr>
<tr>
<td>LANDMARK</td>
<td>Specific</td>
<td>ASG Software Solutions ASG-TMON</td>
</tr>
<tr>
<td>RPCMAINT</td>
<td>Specific</td>
<td>BMC Software RECOVERY PLUS for CICS/VSAM</td>
</tr>
<tr>
<td>SYS1</td>
<td>Generic</td>
<td>Data sets prefixed with SYS1</td>
</tr>
<tr>
<td>TMVSMSTR</td>
<td>Specific</td>
<td>IBM TMON for MVS</td>
</tr>
<tr>
<td>UCC1</td>
<td>Generic</td>
<td>Data sets prefixed with UCC1</td>
</tr>
</tbody>
</table>
Making the Batch VSAM ECCR Available to Batch Jobs

To make the batch VSAM ECCR available to batch jobs, make the following updates to the batch job JCL:

- Add the PowerExchange LOAD library to the STEPLIB concatenation in every step of any batch jobs that update VSAM data sets registered for capture. Alternatively, you can add the LOAD library to the JOBLIB DD of the batch job.
- Add the EDMPARMS DD statement in every step of any batch jobs that update VSAM data sets registered for capture. The EDMPARMS DD statement references the PowerExchange USERLIB library that contains the EDMSDIR module options. For example:

  //EDMPARMS DD DISP=SHR,DSN=hlq.logger_name.USERLIB

  If the EDMSDIR module is included in the LOAD library or if the USERLIB library is include in the JOBLIB or STEPLIB concatenation, you do not need to add the EDMPARMS DD statement.

MVS LNKLST Concatenation

Informatica strongly recommends against including the PowerExchange libraries in the MVS LNKLST concatenation as unexpected abends can occur. When PowerExchange software is included in the LNKLST concatenation, PowerExchange gets control during OPEN processing for all VSAM data sets. PowerExchange does a registration check to determine if the VSAM data set is registered for capture. The registration check process requires that the PowerExchange Agent be active.

If site standards require that the PowerExchange libraries are included in the LNKLST concatenation, the following rules apply:

- The library containing the EDMSDIR module must also be included in the LNKLST concatenation.
- EDMSDIR should specify the option CCERR=CONT as OPEN processing for any VSAM data set causes PowerExchange to get control. If CCERR=ABEND is coded, VSAM OPEN requests fail if the PowerExchange Agent is not active.

  Source for EDMSDIR is supplied in member XICDC600 in the RUNLIB library. Change and rerun this job if changing the CCERR parameter is necessary.
- To override the EDMSDIR included in the LNKLST concatenation and use CCERR=ABEND for VSAM batch jobs, add the EDMPARMS DD statement to the VSAM batch jobs updating VSAM data sets registered for capture. Specify a different data set name in the EDMPARMS DD statement than is specified in the LNKLST concatenation, and include an EDMSDIR module that specifies CCERR=ABEND.
- If you add the PowerExchange LOAD library to the LNKLST concatenation, you can stop an ECCR from capturing changes for a specific job by including the following DD statement:

  //EDMNOCAP DD DUMMY

Activating and Loading the Batch VSAM ECCR Interface

To use the Batch VSAM ECCR, you must first activate the batch VSAM ECCR interface using the PowerExchange Agent. You can activate the Batch VSAM ECCR interface automatically when the PowerExchange Agent starts. Alternatively, you can manually activate Batch VSAM ECCR by using a command after the PowerExchange Agent starts.

**Note:** Activating the Batch VSAM ECCR interface in one PowerExchange Agent makes it active globally on the MVS system. If you are running multiple PowerExchange Agents on a single MVS image, only one PowerExchange Agent needs to activate the batch VSAM ECCR interface.

Activate the Batch VSAM ECCR InterfaceAutomatically

To activate the batch VSAM ECCR interface whenever the PowerExchange Agent starts, set the PowerExchange Agent AGENTCTL parameter CCVACTIVE to YES before you start the PowerExchange Agent.
Activate the Batch VSAM ECCR Interface Manually

Enter the following command to manually activate the batch VSAM ECCR interface:

```
cmd_prefix START VSAMECCR
```

For `cmd_prefix`, use the MVS command prefix specified in the CmdPrefix parameter in the PowerExchange Agent AGENTCTL parameters. The EDMSCTL DD statement in the PowerExchange Agent JCL points to the AGENTCTL parameters.

Restoring VSAM Data Sets When Using the Batch VSAM ECCR

The batch VSAM ECCR captures changes from VSAM batch jobs and passes the changes to the PowerExchange Logger to be recorded. If the VSAM batch job step terminates abnormally, PowerExchange aborts any open units of work in the PowerExchange Logger for that job step. When you extract change data, PowerExchange provides only successfully committed units of work and skips aborted units of work.

**Note:** If the batch job closes the VSAM data set registered for capture before it terminates abnormally, the PowerExchange Logger unit of work containing the changes for that VSAM data set is successfully committed. When you extract changes for this VSAM data set, PowerExchange provides the changes from the failed batch job because the UOW was successful even though the batch job ultimately failed.

If you restart batch VSAM processing from the point of failure rather than restoring the data set and restarting the batch job from the beginning, you must change the default PowerExchange operation to capture change data properly. To change the default PowerExchange processing, add the following DD statement in each batch VSAM job where you restart processing from the point of failure:

```
//EDCMUOW DD DUMMY
```

When you use the EDMCMUOW DD statement and the batch VSAM job step terminates abnormally, PowerExchange commits all open units of work (UOWs) generated by the batch VSAM job. Consider the following points before using the EDMCMUOW DD statement:

- Depending upon the failure circumstances, the batch VSAM ECCR may not get control to commit the open units of work. If so, any uncommitted units of work from the failed VSAM batch job are left in IN-DOUBT status. You must use the PowerExchange Logger RESOLVE_INDOUBT command to commit these uncommitted units of work.
- Do not use EDMCMUOW if you have specified full condense in the capture registration for a VSAM data set.

Managing Batch VSAM Change Data Capture

Controlling the Batch VSAM ECCR

You can control the batch VSAM ECCR interface using PowerExchange Agent commands that use the following syntax:

```
cmd_prefix keyword VSAMECCR
```

**Where:**

- The `cmd_prefix` variable is the command prefix for the PowerExchange Agent. You specify this prefix in the CmdPrefix statement in the PowerExchange Agent AGENTCTL parameters.
- The `keyword` variable is one of the valid controlling keywords.
The following table describes these keywords:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Displays the number of active and inactive batch VSAM ECCR interface modules that have been loaded on this MVS system.</td>
</tr>
<tr>
<td>START</td>
<td>Activates the Batch VSAM ECCR interface regardless of the value specified in the CCVActive statement in the PowerExchange Agent control parameters (AGENTCTL). Use VSAMECCR/RELOAD to a new batch VSAM batch ECCR interface module into Extended Common Storage Area (ECSA). The module is placed at the beginning of the LPA queue in an active state. Warning: This command affects all Batch VSAM ECCRs on the same MVS system.</td>
</tr>
<tr>
<td>STOP</td>
<td>Deactivates the Batch VSAM ECCR interface regardless of the value specified in the CCVActive statement in the PowerExchange Agent control parameters (AGENTCTL). To stop capture for a particular VSAM data set, inactivate the capture registration using the PowerExchange Navigator. Warning: This command affects all Batch VSAM ECCRs on the same MVS system.</td>
</tr>
</tbody>
</table>

**Related Topics:**
- “Configuring AGENTCTL Parameters” on page 25

### Output from the Batch VSAM ECCR

When you start the batch VSAM ECCR by opening a VSAM data set, PowerExchange generates a report that shows the default options that are in effect for the ECCR. After the batch VSAM ECCR ends, the report indicates the number of captured changes. You can find this report in the EDMMSG SYSOUT data set.

The following is a sample report:

```
PWXEDM1728521 Options in effect:
  Load Library containing EDMSDIR. . . . . : EDM.AUSL.USERLIB
  EDMSDIR assembly date/time . . . . . : 20070406 18.19
  Product distribution date. . . . . . . : 20060831
  Product distribution level . . . . . . . : 2.4.05
  Agent Id . . . . . . . . . . . . . . . . : AUSA
  Logger Id. . . . . . . . . . . . . . . . : AUSL
  SYSOUT class . . . . . . . . . . . . . : "
  Action if ECCR error encountered . . . : Continue

PWXEDM1728181 Joined XCF group 'AUSL' as member 'AUSVSUPD'
PWXEDM1728411 EDM ECCR AUSVSUPD connected to EDM Logger AUSL, Log RBA=x'0000560078040000'
PWXEDM1728081 Change Capture active for VSAM file AUSQA.VSAM.VSMDMO1
  Edition=C4E3D30000000001, EDMMNAME=AUSMAUSQA.VSAM.VSMDMO1
PWXEDM1728091 Change Capture counts for AUSQA.VSAM.VSMDMO1: Insert=0, Update=5, Delete=0
PWXEDM1728441 EDM ECCR AUSVSUPD disconnected from EDM Logger AUSL, Log RBA=x'00005600840D0000'
PWXEDM1728181 Left XCF group 'AUSL' as member 'AUSVSUPD'
PWXEDM1728291 EDM ECCR sent 5 records to Logger AUSL (5 change records)
```

**Note:** This report also includes message PWXEDM172886I, which indicates any load module replacements that have been applied.
Stopping Change Data Capture for VSAM Sources

You can use the following methods to stop change data capture for VSAM data sets:

<table>
<thead>
<tr>
<th>To Stop Capturing Changes for...</th>
<th>Use This Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>All VSAM data sets</td>
<td>Stop the batch VSAM ECCR interface.</td>
</tr>
<tr>
<td>A specific registered VSAM data set</td>
<td>Deactivate or delete the capture registration and close the data set.</td>
</tr>
</tbody>
</table>

**Warning:** When you stop the change data capture process without stopping updates to the source, you lose change data. To avoid losing change data and rematerializing the target tables, stop updates to the source instead of stopping the batch VSAM ECCR interface.

Stopping the Batch VSAM ECCR

**Closing a VSAM Data Set**

When you close a source data set, the batch VSAM ECCR no longer captures changes associated with that source. Closing data sets with the batch VSAM ECCR generally means stopping the batch job, which also stops the batch VSAM ECCR.

**Stopping a Batch VSAM ECCR Job**

When you stop a batch VSAM ECCR job, PowerExchange no longer captures change data for any VSAM data sets in that batch job. The batch VSAM ECCR running in the batch job disconnects from the PowerExchange Logger and displays a set of messages, including the number and type of changes captured since the last time the VSAM data sets were opened. For example:

```
PWXEDM1728181 Joined XCF group 'AUSL' as member 'AUSVSUPD'
PWXEDM1728411 EDM ECCR AUSVSUPD connected to EDM Logger AUSL, Log RBA=X'0000560078040000'
PWXEDM1728081 Change Capture active for VSAM file AUSQA.VSAM.VSMDEMO1 Edition=C4E3D70000000000, EDMNAME=AUSMACQ.AVSAM.VSMDEMO1
PWXEDM1728091 Change Capture counts for AUSQA.VSAM.VSMDEMO1: Insert=0, Update=5, Delete=0
PWXEDM1728411 EDM ECCR AUSVSUPD disconnected from EDM Logger AUSL, Log RBA=X'0000560084DD0000'
PWXEDM1728181 Left XCF group 'AUSL' as member 'AUSVSUPD'
PWXEDM1728291 EDM ECCR sent 5 records to Logger AUSL (5 change records)
```

**Stopping the Batch VSAM ECCR Interface**

Stop the batch VSAM ECCR interface by using the PowerExchange Agent STOP command. This command disables the batch VSAM ECCR interface for the entire z/OS system. After the batch VSAM ECCR interface stops, PowerExchange does not capture changes for any VSAM data set that is subsequently opened. Change data capture activity that is in progress continues until the data sets are closed.

To stop the VSAM batch ECCR, enter the following command:

```
cmd_prefix STOP VSAECR
```

The `cmd_prefix` variable is the command prefix for the PowerExchange Agent. You specify this prefix in the CmdPrefix statement in the PowerExchange Agent AGENTCTL parameters.

For more information about batch VSAM ECCR interface commands, see the **PowerExchange Command Reference**.

**Related Topics:**

* “Configuring AGENTCTL Parameters” on page 25
Refreshing the Batch VSAM ECCR

The batch VSAM ECCR does not refresh capture registrations once it starts. You must rerun the batch job to activate new or changed capture registrations for VSAM data sets in that batch job.

Note: If the capture registrations specify condense processing, you must also recycle PowerExchange Condense.

Application Recovery Considerations

The following section describes batch execution and recovery issues that you must consider when using PowerExchange CDC. You may have to change some of your existing operational recovery procedures to accommodate changed-data propagation.

Point-in-Time Recovery

Point-in-time recovery invalidates those changes on the PowerExchange Logger that were recorded by the jobs which were recovered. Standard point-in-time recovery does not indicate to processors of PowerExchange Logger data that this data is invalid.

What the processor of PowerExchange log data must do when point-in-time recovery is necessary is as follows:
- Recover the source to the correct point-in-time.
- Recover the output of the PowerExchange Condense to the state that it was in at the time of recovery.
- Reset the change processor to restart processing when the recovery is complete.

DFSMSdfp Checkpoint/Rarent

PowerExchange for VSAM CDC does not support DFSMSdfp Checkpoint/Rarent.

Managing VSAM Schema Changes

If the record layout of the VSAM source data set changes, use the following procedures to ensure that data previously captured remains available for use.

To manage VSAM schema changes:
1. Stop updates to the VSAM source file.
2. If you are using PowerExchange Condense, ensure that PowerExchange Condense has extracted all captured change data from the PowerExchange Logger.
3. Extract all captured changes using the existing extraction map.
4. In the VSAM capture registration, set the **Status** option to **History**.
5. Change the VSAM file structure as needed.
6. Delete the extraction map.
7. Create a data map for the new VSAM data structure.
8. Create capture registration using the new data map.
9. Allow changes to the altered VSAM file.
C H A P T E R 8

CICS/VSAM Change Data Capture

This chapter includes the following topics:
- Introduction to CICS/VSAM Change Data Capture, 122
- Planning for CICS/VSAM Change Data Capture, 122
- Configuring CICS for Change Data Capture, 124
- Activating the CICS/VSAM ECCR, 126
- Managing CICS/VSAM Change Data Capture, 126
- Managing VSAM Schema Changes, 129

Introduction to CICS/VSAM Change Data Capture

PowerExchange for VSAM CDC synchronously captures change data for VSAM data sets from CICS regions. The CICS/VSAM ECCR allows you to capture changes to VSAM data sets that were generated by CICS transactions. After the CICS/VSAM ECCR captures changes, the change data is available for propagation. After you activate the capture registrations and the CICS/VSAM ECCR, PowerExchange automatically captures changes made to all registered VSAM data sets.

The CICS/VSAM ECCR runs in the CICS region. It captures changes as they occur using CICS global user exits (GLUE) and task-related user exits (TRUE) and passes the changes to the PowerExchange Logger for recording.

Planning for CICS/VSAM Change Data Capture

Before you configure CICS/VSAM CDC, verify that the following prerequisites are met. Also, review the requirements and restrictions so that you can properly configure CDC.

Prerequisites for CICS/VSAM CDC

CICS/VSAM CDC has the following prerequisites:
- To capture change data, you must define VSAM data sets in CICS with either RECOVERY(BACKOUTONLY) or RECOVERY(ALL). The CICS/VSAM ECCR only supports capture for recoverable data sets.
**Warning:** If you register a non-recoverable VSAM data set for capture and the CICS/VSAM ECCR is active, PowerExchange closes the data set and prevent it from being reopened.

- You must define and open the CSMT queue in the CICS region. CSMT is a file used for sending messages within CICS. The MVS equivalent is SYSLOG.

### Requirements and Restrictions for CICS/VSAM CDC

The following restrictions apply to CICS/VSAM CDC processing:

- The CICS/VSAM ECCR supports only VSAM KSDS, RRDS, or VRRDS data sets. You cannot capture changes to VSAM ESDS or LINEAR data sets.
- If you specify CCERR=ABEND in the EDMSDIR module options and the CICS/VSAM ECCR encounters a serious error or abnormally ends (abends) during initialization, PowerExchange terminates the CICS region to prevent data loss and ensure change data replication integrity. The termination process aborts current tasks and backs out in-flight transactions similar to if you had issued the CICS command CEMT PERFORM SHUTDOWN IMMEDIATE.
- If you activate the CICS/VSAM ECCR and open a file before you activate the PowerExchange Agent, you must close and reopen the file to start capturing changes.
- In a CICS/MRO environment, you must load the CICS/VSAM ECCR into each CICS region that owns VSAM files from which you want to capture changes.
- Exclude PowerExchange from XPEDITER/CICS. XPEDITER/CICS erroneously believes that it is detecting storage violations.
- PowerExchange requires unique data source-specific ECCRs to capture changes for that data source. A single ECCR cannot capture changes for multiple data source types, such as CICS/VSAM and DB2. Therefore, PowerExchange cannot maintain transactional integrity for transactions that change CICS/VSAM data sets and DB2 tables or IMS databases in the same unit of work.

If you need to apply the changes made by a CICS transaction that changes multiple data source types in the same order, you could use a staging table. First, extract the changes for each unique source type and insert them into the staging table, which includes the DTL__CAPXTIMESTAMP value as a column. Then, you can extract these changes from the staging table, ordering them by the DTL__CAPXTIMESTAMP value, to apply the changes to the target tables in the appropriate order.

### CICS XFCFRIN and XFCFROUT Global Exits

The CICS/VSAM ECCR uses CICS XFCFRIN and XFCFROUT global exits to capture changes to VSAM data sets defined to CICS. These exits perform the following processing:

- The XFCFROUT exit writes the change data to the PowerExchange Logger.
- If the XFCFRIN exit detects DELETE operations that use the RIDFLD operand, it reads the record being deleted with UPDATE and issues a new DELETE operation without the RIDFLD operand. The new DELETE operation causes the XFCFRIN and XFCFROUT exits to get control again, which allows the XFCFROUT exit to capture and log all of the required information for this type of DELETE operation.

If your CICS region has other, active programs that use the XFCFRIN or XFCFROUT exit point, verify that those exits do not impact the processing of the PowerExchange-supplied exits. For example, if your exit receives control...
prior to the PowerExchange exit and changes the data record, PowerExchange might be unable to properly capture change data. Also, consider the following:

- The PowerExchange XFCFROUT exit must receive uncompressed data records.
- If you have other XFCFRIN or XFCFROUT global user exits and use the EDMC INIT command, you might impact the processing of either your application or the CICS/VSAM ECCR. This command initializes the CICS/VSAM ECCR and dynamically installs the XFCFRIN and XFCFROUT exits, which might cause the PowerExchange exits to get control in an improper order.

  Note: CICS gives control to multiple exits at the same exit point based on the order in which the exits are activated in CICS.

To determine whether a CICS region has other exits programs installed at the XFCFRIN and XFCFROUT exit points, you can use the CECI transaction with the following commands to browse the exit list:

```
INQUIRE EXITPROGRAM EXIT(XFCFRIN) START
INQUIRE EXITPROGRAM EXIT(XFCFROUT) START
```

For more information about the CICS-supplied CECI transaction and the INQUIRE EXITPROGRAM command, see the IBM CICS Transaction Server documentation.

**Relationships with Other PowerExchange Components**

The CICS/VSAM ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- The CICS/VSAM ECCR must log all changes to a single PowerExchange Logger running on the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must both run on the same MVS system as the ECCR.
- If you use the Post-Log Merge option of the PowerExchange Logger, PowerExchange allows you to capture and propagate changes even if the changes originate from different MVS systems. You must run a PowerExchange Logger on each MVS system that makes changes to the source VSAM data sets.
- Operational issues in the PowerExchange Logger can cause the CICS transactions to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the waiting transactions continue and PowerExchange captures and records the change data without any loss of data.

  You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

**Related Topics:**

- "Monitoring the PowerExchange Logger for MVS" on page 51
- "Using Post-Log Merge" on page 69

**Configuring CICS for Change Data Capture**

To use the CICS/VSAM ECCR, you must modify the CICS regions that make changes to the VSAM data sets for which you want to capture changes.
To configure CICS for capture change data:

1. Modify the CICS JCL.
   - Add the PowerExchange LOAD library to the STEPLIB and DFHRPL DD statements.
     **Note**: If you added the PowerExchange LOAD library to the MVS LNKLST concatenation, you do not need to add it to the STEPLIB statement but you do need to add it to the DFHRPL DD statement.
   - Add the EDMPARMS DD statement. The EDMPARMS DD statement references the PowerExchange USERLIB library. For example:
     ```
     //EDMPARMS DD DISP=SHR, DSN=hlq.logger_name.USERLIB
     ```

2. Modify the CICS startup procedures.
   Add the module name EDMKOPER to the second phase of the PLT initialization list (PLTPI) so that the CICS/VSAM ECCR is initialized during the third stage of CICS initialization. Alternatively, you can manually activate the CICS/VSAM ECCR by entering the CICS command:
   ```
   EDMC INIT
   ```

   **Tip**: Informatica Corporation recommends that you add the module name EDMKOPER to the initialization list, rather than manually activating the CICS/VSAM ECCR with EDMC INIT. When you include EDMKOPER in the initialization list, the CICS/VSAM ECCR activates during CICS startup, which ensures that PowerExchange captures change data.

3. Verify that each CICS region that connects to PowerExchange uses a unique ECCR name.
   The ECCR name value must be unique within a PowerExchange Logger group. You must cold-start CICS to change the ECCR name. The CICS/VSAM ECCR uses the value specified for the following purposes:
   - The ECCR name that connects to the PowerExchange Logger to write change data
   - The member name that joins the XCF group of the PowerExchange Logger
   - As part of the ECCR-UOW field in the control information for each change record written to PowerExchange Logger log files
   
   The default name is the CICS SYSID specified in the SIT SYSIDNT parameter. To override the default name, code the following statement in the SIT or SIT override file:
   ```
   INITPARM=(EDMKOPER='option')
   ```

   Replace *option* with one of the following options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>SYSID</em></td>
<td>Uses the CICS SYSID</td>
</tr>
<tr>
<td><em>JOBNAME</em></td>
<td>Uses the JOB or STC name</td>
</tr>
<tr>
<td><em>APPLID</em></td>
<td>Uses the VTAM ACB name</td>
</tr>
<tr>
<td>1 through 8 alphanumeric characters</td>
<td>Uses this value</td>
</tr>
</tbody>
</table>
**Tip:** Informatica recommends that you use the same value for the ECCR name and the CICS started task or job name. This practice allows you to easily identify the CICS/VSAM ECCR when reviewing messages and data from the PowerExchange Logger.

4. Add the CICS/VSAM ECCR programs and transaction to CICS. The PowerExchange SAMPLIB library contains sample members for each level of CICS:

<table>
<thead>
<tr>
<th>CICS TS Version</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2, 1.3, and 2.1</td>
<td>#CICSV52</td>
</tr>
<tr>
<td>2.2 and later</td>
<td>#CICSV62</td>
</tr>
</tbody>
</table>

Edit the copy as required for your site. Use DFHCSDUP or RDO to add the CICS/VSAM ECCR programs and transaction to the processing program table (PPT) and program control table (PCT).

### Activating the CICS/VSAM ECCR

Use the following procedure to activate the CICS/VSAM ECCR.

To activate the CICS/VSAM ECCR:

1. Verify that you are using a supported version of CICS and that any prerequisite CICS maintenance has been applied.
2. Verify that the CICS FCT for the file is defined as RECOVERY(BACKOUTONLY) or RECOVERY(ALL).
3. Start the CICS region. If you did not add the module name EDMKOPER to the initialization list, you must manually activate the CICS/VSAM ECCR by using the command EDMC INIT.

Each time the CICS/VSAM ECCR starts, PowerExchange generates a report with message number PWXEDM 172852 I. The report contains the default options that are in effect. For example:

```
LOG START
PWXEDM172852I Options in effect:
Load Library containing EDMSDIR . . . . : EDM.EDMUSERLIB
EDMSDIR assembly date/time . . . . . . : 20070406 18.19
Product distribution date . . . . . . . : 20060831
Product distribution level . . . . . . : 2.4.05
Agent Id . . . . . . . . . . . . . . . : EDMA
Logger Id . . . . . . . . . . . . . . : EDML
SYSOUT class . . . . . . . . . . . . : *
Action if ECCR error encountered . . . : Continue
```

### Managing CICS/VSAM Change Data Capture

To control the CICS/VSAM ECCR, use PowerExchange-supplied CICS transactions. You also might need to temporarily stop CDC or change the capture registrations for data sets.
Controlling CICS/VSAM ECCR Processing

You can control the CICS/VSAM ECCR by issuing the EDMC command from a CICS terminal. With this command, you can perform the following tasks:

- Initialize the ECCR.
- Terminate change-capture processing.
- Display the files that are participating in the change-capture process.
- Display a Help panel for the ECCR.

The syntax for the EDMC command is as follows:

```
EDMC keyword
```

The `keyword` variable can be any valid EDMC keyword. The following table describes these keywords:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Displays the names of the files that are registered for capture and have been opened since the CICS/VSAM ECCR initialized. The file names are displayed at the CICS terminal. Both DISP and DISPLAY are valid keywords.</td>
</tr>
<tr>
<td>HELP</td>
<td>Displays a Help panel for the CICS/VSAM ECCR that lists the available commands and their functions.</td>
</tr>
</tbody>
</table>
| INIT    | Initializes CICS/VSAM ECCR in the CICS region. Use this keyword following a TERM command to restart the CICS/VSAM ECCR. You can also set up the ECCR to start automatically as part of the PLTPI startup procedure.  
  **Note:** To capture changes for a particular VSAM file, you must activate the corresponding capture registration and close and reopen the VSAM file to CICS. |
| TERM    | Terminates the CICS/VSAM ECCR in the CICS region. When you run this command, the CICS/VSAM ECCR stops immediately. 
  **Note:** To terminate the change data capture for a particular VSAM file, you must deactivate or delete the corresponding capture registration and close and reopen the VSAM file to CICS. |

**Warning:** If you have initialization programs that enable either the XFCFRIN or XFCFROUT exit, do not use the INIT or TERM commands.

**RELATED TOPICS:**
- “CICS XFCFRIN and XFCFROUT Global Exits” on page 123

Sample Results from the EDMC DISP Command

The following sample output is from a EDMS DISP command:

```
EDMC DISP  Connect  Date : 12/22/00
DTECDM CICS Display Processor  Time : 13:27:27

File Name  Dataset Name  Using ID : EDC1
EDMTFL01  DTL.EDM.TESTFL01
EDMTFL02  DTL.EDM.TESTFL02
```
Output from the CICS/VSAM ECCR

When the CICS/VSAM ECCR starts, PowerExchange issues a report that shows the default options that are in effect for this run. The CICS/VSAM ECCR reports the number of captured changes at termination. You can find this report in the EDMMSG SYOUT data set. The following information is a sample of this report:

FWXEDM172852I Options in effect:
  Load Library containing EDMSDIR. . . . . : EDM.AUSL.USERLIB
  EDMSDIR assembly date/time . . . . . . . : 20070406 18.19
  Product distribution date. . . . . . . . . : 20060831
  Product distribution level . . . . . . . . : 2.4.05
  Agent Id . . . . . . . . . . . . . . . . . . . . : EDMA
  Logger Id. . . . . . . . . . . . . . . . . . . : EDML
  SYSOUT class . . . . . . . . . . . . . . . . . : "
  Action if ECCR error encountered . . . : Continue

FWXEDM172830I CICGLB loaded at 0F2871A8
FWXEDM172811I XCF is in local mode only
FWXEDM172818I Joined XCF group 'EDML' as member 'VSM3'
FWXEDM172841I EDM ECCR VSM3 connected to DETAIL Logger EDML, Log RBA=X'000000001DFE'
FWXEDM172808I Change Capture active for VSAM file CCV.EDM.VCC1
FWXEDM172811I Change Capture active for VSAM file CCV.EDM.VCC1
FWXEDM172841I EDM ECCR VSM3 disconnected from DETAIL logger EDML,
  Log RBA=X'00000000AED19
FWXEDM172818I Left XCF group 'EDML' as member 'VSM3'
FWXEDM172829I EDM ECCR sent 11 records to logger EDML (5 change records)
FWXEDM172809I Change Capture counts for CCV.EDM.VCC1: Insert=5, Update=0, Delete=0

Note: This report also includes message PWXEDM172886I, which indicates any load module replacements that have been applied.

Stopping Change Data Capture for VSAM Sources

You can use the following methods to stop change data capture for VSAM data sets:

<table>
<thead>
<tr>
<th>To Stop Capturing Changes for...</th>
<th>Use This Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>All VSAM data sets</td>
<td>Stop the CICS/VSAM ECCR.</td>
</tr>
<tr>
<td>A specific registered VSAM data set</td>
<td>Deactivate or delete the capture registration and close the data set.</td>
</tr>
</tbody>
</table>

Warning: When you stop the change data capture process without stopping updates to the source, you lose changed data. You can avoid losing changed data and avoid rematerialization by stopping updates to the source instead of stopping the change data capture process.

Closing a VSAM Data Set

When you close a source data set, the CICS/VSAM ECCR no longer captures changes associated with that source. VSAM data sets accessed through CICS can be closed dynamically using CICS commands.

Stopping the CICS/VSAM ECCR

When you stop the CICS/VSAM ECCR, PowerExchange no longer captures change data for any VSAM data set in that environment. The CICS/VSAM ECCR disconnects from the PowerExchange Logger and displays a set of messages, which include the number and type of changes captured since the last time the data set was opened.

To stop the CICS/VSAM ECCR, issue the EDMC transaction TERM command from CICS terminal, as follows:

EDMC TERM

EDMC is the default CICS transaction code for the CICS/VSAM ECCR. The CICS/VSAM ECCR stops immediately.

Warning: If you have initialization programs that enable either the XFCFRIN or XFCFROUT exit, do not use the INIT or TERM command.
**RELATED TOPICS:**
- “CICS XFCFRIN and XFCFROUT Global Exits” on page 123

### Refreshing Capture Registrations in the CICS/VSAM ECCR

Refresh the capture registrations in the CICS/VSAM ECCR to activate the new or changed capture registrations.

To refresh capture registrations in the CICS/VSAM ECCR:
- Close and reopen the VSAM data set. If the file definition is set up to open the file when referenced, the CICS/VSAM ECCR will be refreshed the next time the file is opened.
- If the capture registrations specify condense processing, you must recycle the PowerExchange Condense task.

### Managing VSAM Schema Changes

If the record layout of the VSAM source data set changes, use the following procedure to ensure that data previously captured remains available for use.

To make VSAM schema changes:
1. Stop updates to the VSAM source file.
2. Close the VSAM file in CICS region.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has extracted all data from the PowerExchange Logger.
4. Extract all captured changes using the existing extraction map.
5. In the VSAM capture registration, set the **Status** option to **History**.
6. Change the VSAM file structure as needed.
7. Delete the extraction map.
8. Create a data map for the altered VSAM data set.
9. Create a capture registration using the new data map.
10. Open the VSAM file in the CICS region.
11. Allow updates to the altered VSAM data set.
PowerExchange for Datacom synchronous CDC captures changes while the changes are occurring in the Datacom Multi-User Facility (MUF) address space.

**Note:** PowerExchange for Datacom provides both synchronous and table-based CDC. For Datacom Version 12 or later, you must use table-based CDC.

You can configure the Datacom synchronous ECCR to use the direct-log-write method. This method uses the following components:

**Datacom Change Collector**
- Runs in the Datacom MUF address space. It captures changes as they occur and passes the changes to the PowerExchange Logger for recording.

**Datacom Change Controller**
- Runs in a separate address space. It manages the capture registrations for the Datacom Change Collector.

The direct-log-write method is the recommended configuration because it has the following advantages:
- It reduces the latency between the time when the changes occur and the time when changes can be extracted.
- It reduces the operational complexity and system resource usage to capture change data.

For compatibility with older configurations of Datacom CDC, you can configure the Datacom synchronous ECCR to store the changes in a data space before they are passed to the PowerExchange Logger. This method uses the following components:
### Datacom Change Collector

Runs in the Datacom MUF address space. It captures changes as they occur and passes the changes to the Datacom Change Controller.

### Datacom Change Controller

Runs in a separate address space. It manages the capture registrations for the Datacom Change Collector and stores the captured changes in a data space.

### Datacom Log Feeder

Runs in a separate address space. It reads the captured change data from the data space and passes this data to the PowerExchange Logger for recording.

### Relationships with Other PowerExchange Components

The Datacom ECCR uses other PowerExchange components, such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- The Datacom Change Collector must log all changes to a single PowerExchange Logger running on the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must both run on the same MVS system as the Datacom ECCR.
- Operational issues in the PowerExchange Logger can cause the Datacom Change Collector to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the Datacom Change Collector continues the capture and recording of change data without any loss of data.

You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

**Related Topics:**

- "Monitoring the PowerExchange Logger for MVS" on page 51

### Configuring Datacom for CDC

Before you can capture changes to Datacom tables, you must configure the Datacom MUF for PowerExchange for CA Datacom CDC. Configuration requires changes to the MUF JCL to provide access to load modules and parameters.

**Note:** Configuration of the MUF JCL varies depending on whether you use the direct-log-write method or the original method of logging change data.

### Adding the Datacom Change Collector Subtask

The Datacom Change Collector captures changes to tables as they occur inside the Datacom Multi-User Facility (MUF) address space. The Change Collector uses the subtask facility of CA Datacom to establish its intercepts. You must change the Multi-User Facility JCL to include the Datacom Change Collector.

Add one of the following statements to the MUF SYSIN stream:
SUBTASK DTLXDPDX

Use this statement when using the direct-log-write method for logging changes. The Change Collector, when using the direct-log-write method, passes changes directly to the PowerExchange Logger for recording. Informatica recommends using the direct-log-write method because it provides the lowest latency between the time the change occurs and the time it is recorded.

SUBTASK DTLXDPDT

Use this statement when using the original method for logging changes. The Change Controller passes changes to the Change Controller so they can be stored in a data space. The Log Feeder reads the changes from the data space and passes them to the PowerExchange Logger for recording.

Configuring the Datacom MUF JCL When Using the Direct-Log-Write Method

Use the following procedure to configure the Datacom MUF JCL when you want to use the direct-log-write method of logging change data. These JCL changes are required to provide access to the load modules and parameters that are needed for CDC.

1. Add the following PowerExchange load libraries to the STEPLIB concatenation in the MUF JCL:
   - hlq.LOADLIB
   - hlq.LOAD
   
   The hlq variable is the high-level qualifier for the PowerExchange data sets chosen during installation.

2. Verify that all libraries in the STEPLIB DD statement are APF-authorized.

   The Datacom MUF must run under this authorization if you use Datacom CDC.

3. Add the following EDMPARMS DD statement:
   
   //EDMPARMS DD DISP=SHR,DSN=hlq.logger.USERLIB

   This statement allocates the USERLIB library that contains the EDMSDIR module options.

4. Add the following DTLINPUT DD statement:
   
   //DTLINPUT DD DISP=SHR,DSN=data_set(DTLINPUT)

   This statement allocates the data set that contains the Datacom Change Collector parameters. The variable data_set is the name of this data set.

5. Add the following DTLPRINT DD statement:
   
   //DTLPRINT DD SYSOUT=* 

   This statement allocates the SYSOUT data set PowerExchange uses for messages.

Configuring the Datacom MUF JCL When Using the Original Logging Method

Use the following procedure to configure the Datacom MUF JCL when you want to use the original method of logging changes in a data space. These JCL changes are required to provide access to the load modules and parameters that are needed for CDC.

1. Add the hlq.LOADLIB to the STEPLIB concatenation.

2. Verify that all libraries in the STEPLIB DD statement are APF-authorized.

   The Datacom MUF must run under this authorization if you use Datacom CDC.

3. Add the following DTLINPUT DD statement:
   
   //DTLINPUT DD DISP=SHR,DSN=data_set(DTLINPUT)
This statement allocates the data set that contains the Datacom Change Collector parameters. The variable `data_set` is the name of this data set.

4. Add the following DTLPRINT DD statement:
   
   ```
   //DTLPRINT DD SYSOUT=*
   ```
   
   This statement allocates the SYSOUT data set that PowerExchange uses for messages.

**RELATED TOPICS:**

- "Configuring the Datacom Change Collector Parameters" on page 133

**Configuring the Datacom ECCR Components**

The Datacom ECCR is composed of a Change Collector, a Change Controller, and if you use the original logging method, a Log Feeder. The Change Collector runs in the Datacom MUF address space, whereas the Change Controller and Log Feeder each run in their own address space.

Configuration depends on whether you use the direct-log-write method or original logging method to log change data.

If you use the direct-log-write method, configure the following parameters:

- Datacom Change Collector parameters
- Datacom Change Controller JCL and parameter

If you use the original logging method, configure the same Change Collector and Change Collector items and also configure the Log Feeder and its parameters.

**Configuring the Datacom Change Collector Parameters**

Specify parameters to control the Datacom Change Collector using the DTLINPUT DD statement in the Datacom MUF JCL.

The Change Collector parameters are:

- **DSPACE_ID `nnnnnnnn`**
  
  The DSPACE_ID parameter overrides the dataspace ID that is used. If a DSPACE_ID is specified, the MUF parameter is ignored.

- **END**
  
  The END parameter terminates the SYSIN datastream. This statement is optional. Synonyms for END are EXIT and QUIT. If these statements are not present, EOF is assumed to be a valid END statement.

- **MUF `nnnnnnnn`**
  
  The `nnnnnnnn` is the MUF name. It should match the MUF name as determined in the previous sentence. Multiple MUF statements are valid in a single job.

  The MUF parameter specifies the Multi-User Facility on the system to which all other Change Collector parameters will be applied. If the MUF parameter is specified in the standard input stream for the MUF, Datacom takes this value as the MUF name. Otherwise, this value is a started task or job name.
ON_ERROR (ABEND|DISABLE)

The ON_ERROR parameter specifies how the MUF handles change capture fatal errors:

- ON_ERROR ABEND causes the MUF to shut down.
- ON_ERROR DISABLE allows the MUF to remain functioning, which is the default behavior if ON_ERROR is not specified.

Configuring the Datacom Change Controller

Each Multi-User Facility (MUF) needs to have tables defined for replication within its environment.

While a production system can have many tables selected for replication, there can only be a few selected on a test system. The Change Controller reads the PowerExchange registrations, which define the data that is to be captured.

The Change Controller can act as a root module to control all the dataspaces for a set of Multi-User Facilities on a single system. It can also be run to modify the current settings for each of those.

Configuring the Datacom Change Controller JCL

The Change Controller runs continuous when Datacom change data capture is active to anchor common control blocks in the data space and to perform capture registration checks. If you use the original method of logging change data, the Change Controller also stores captured changes in the data space.

PowerExchange provides sample JCL in member DCOMCTRL in the RUNLIB library.

To configure the Datacom Change Controller JCL:

1. Edit the DCOMCTRL member in the RUNLIB library.
2. If using local access to capture registrations, include all of the DTLxxxxx DD statements that are marked as comments in the step called STEP1.
3. Configure the Change Controller parameters in the step called COPYCMD.
4. To run as a started task, change the JCL in the DCOMCTRL member to a started task procedure, and copy the configured DCOMCTRL member to a system PROCLIB library for started tasks.
   To run as a batch job, add the JOBCARD member from the RUNLIB library, changing the jobname and other JOB card parameters as appropriate.

Configuring the Datacom Change Controller Parameters

The mode of execution of the Change Controller is determined by a series of statements read from the SYSIN file. The syntax of each statement is listed. Anything input on the same line after the required portion is considered a comment and will be ignored. The input is echoed to the SYSPRINT file along with any processing or error messages.

DSPACE_ID nnnnnnnnn

Overrides the dataspace ID that is used. If a DSPACE_ID is specified, the MUF nnnnnnnnn definition is ignored.

END

This terminates the SYSIN datastream. This statement is optional. Synonyms for END are EXIT and QUIT. If these statements are not present, EOF is assumed to be a valid END statement.
Running Datacom as Part of a SHADOW MUF

You can run Datacom as part of a SHADOW MUF on two LPARs, with PowerExchange running on each LPAR. When the Primary LPAR is disabled and the Secondary LPAR assumes the work load, change propagation stops because the same MUF name is normally used by PowerExchange.

To overcome this, PowerExchange provides the capability to override the Dataspace ID that is used and to disassociate it from the predefined Datacom MUF name. For example, a different dataspace could be used for each LPAR.

To run Datacom as part of a SHADOW MUF:

1. Specify the 'DSPACING _nnn' control statement and value in the input stream for the Controller, DTLXCTRL.
2. Define the DTLINPUT member.
Configuring the Datacom Log Feeder

The Datacom Log Feeder is only used when using the original method of logging change data. If you use the recommended direct-log-write method for writing change data to the PowerExchange Logger, you do not need to configure the Datacom Log Feeder.

The Datacom Log Feeder writes the changes stored in the data space by the Change Controller to the alternate target environments. By separating the duties of identification and the actual propagation function, Datacom performance should be minimally impacted. All delays due to the replication process are processed asynchronously.

The Log Feeder will normally run continuously as a started task to allow communication with the PowerExchange functions. It may be started and stopped as necessary depending upon the needs of an individual site. Once started, the Log Feeder will wait for notification of any information to propagate.

Configuring the Datacom Log Feeder Parameters

Log Feeder execution is controlled through control statements read from the SYSIN file.

You can specify the following control statements using the syntax given. Any input on the same line after the required portion is considered a comment and is ignored. The input is echoed to the SYSPRINT file along with any processing or error messages.

**MUF nnnnnnnn**

The Log Feeder can process changes for one, and only one, MUF. A unique copy of the Log Feeder must be run for each MUF on a system. Additional copies of the Log Feeder may be run for each additional MUF on a system. If the MUF parameter is specified in the standard input stream for the MUF then Datacom takes this as the MUF name, otherwise this is taken to be started task/jobname. The MUF statement selects the specific MUF to be processed by this copy of the Log Feeder. The nnnnnnnn value is the name of the MUF to be associated with this copy of the Log Feeder. The Datacom MUF name, PowerExchange Controller MUF name, and Log Feeder name must be the same. This statement is required.
**BYPASS**

The BYPASS statement allows testing of the Log Feeder without invoking the PowerExchange Change Capture software code. All other processes are available. This statement is optional.

**FAKEIT**

The FAKEIT statement allows testing of the Log Feeder without invoking the actual PowerExchange Change Capture software code. Different from the bypass command, this invokes a simulator for the PowerExchange Change Capture software that prints messages describing the processes that would be sent to the PowerExchange Change Capture software. This allows testing and verification without the overhead and difficulties associated with running dual testing and production facilities. All other processes are available. This statement is optional.

**END**

The END statement terminates the SYSIN datastream. This statement is optional. Synonyms for END are EXIT and QUIT. If these statements are not present, EOF is assumed to be a valid END statement.

---

**Managing Datacom Change Data Capture**

PowerExchange provides commands to start, stop, and control the Datacom ECCR components.

**Datacom Change Controller Commands**

Use the following commands to control the Datacom Change Controller:

- To start the Change Controller address space, use the MVS START command.
- To stop the Change Controller address space, use the MVS STOP command.
- To refresh the Change Controller with new or changed registrations, use the following command:

  ```
  MODIFY jobname,REFRESH
  ```

  The `jobname` variable is the name of the Change Controller started task.

**Datacom Log Feeder Commands**

Use the following commands to control the Datacom Log Feeder:

- To start the Log Feeder address space, use the MVS START command if running as a started task procedure or submit the JCL if running as a batch job.

  When starting up the Log Feeder ensure the Log Feeder has connected to the PowerExchange Logger. The following message in the Log Feeder confirms the connection has been successfully made.

  ```
  FWXEDM172841I EDM ECCR DOCLF1 connected to EDM Logger DOCLR, Log RBA=X'000000056AD40000'
  ```

- To stop the Log Feeder address space, use the MVS STOP command.
- To display Logger Feeder statistics or get diagnostic information, use the following MVS MODIFY commands:

  - `F jobname,STATUS` to get current propagation statistics.
  - `F jobname,DEBUG {ON|OFF}` to turn debugging on or off during processing.
  - `F jobname,TRACE {ON|OFF}` to turn tracing on or off during processing.
Stopping Change Data Capture for Datacom Sources

The following table lists the methods for stopping the change capture process, based on the level at which you want to stop capturing changes:

<table>
<thead>
<tr>
<th>To Stop Capturing Changes for</th>
<th>Use This Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Datacom environment</td>
<td>Stop access with the ACCESS command. Ensures that the change data capture integrity is retained by preventing updates to the source data.</td>
</tr>
<tr>
<td>Any registered data object</td>
<td>Set the registration to history.</td>
</tr>
<tr>
<td>Any registered data object</td>
<td>Deactivate or delete the corresponding data-resource registration.</td>
</tr>
</tbody>
</table>

**Note:** If change data capture is unavailable, changes applied to the database might be missed. Restricting database access to read-only with the ACCESS command can be an option if change data capture is unavailable.

Managing Datacom Schema Changes

This procedure describes how to manage schema changes for the source and target tables that are involved in change propagation.

To manage Datacom schema changes:

1. Stop all update activity against the relevant Datacom data.
2. Ensure that all changes that occurred under the old schema are processed by PowerExchange.
3. Make the change to the Datacom data structures.
4. Reflect the schema changes in the PowerExchange Registration.
5. Refresh the PowerExchange Controller.
7. Allow update activity to the Datacom data.
Datacom Table-Based Change Data Capture

This chapter includes the following topics:

- Introduction to Datacom Table-Based CDC, 139
- Architectural Overview, 140
- Configuring Datacom for CDC, 142
- Configuring the Datacom Table-Based ECCR, 142
- Managing Datacom Change Data Capture, 146
- Managing Datacom Schema Changes, 147

Introduction to Datacom Table-Based CDC

PowerExchange for Datacom table-based change data capture (CDC) captures changes asynchronously from Datacom CDC tables. PowerExchange works with the Datacom Change Data Capture feature that was introduced in Datacom Release 11 SP4. When Change Data Capture is enabled in Datacom, Datacom records changes in its CDC tables, TSN and MNT. The table-based ECCR listens for changes to the CDC tables and writes the changed data to the PowerExchange Logger.

PowerExchange also provides a Datacom synchronous ECCR, which does not require the Datacom Change Data Capture feature.

Related Topics:

- “Datacom Synchronous Change Data Capture ” on page 130
Relationships with Other PowerExchange Components

The Datacom table-based ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following requirements:

- The Datacom table-based ECCR logs all changes to a single PowerExchange Logger. The PowerExchange Logger and PowerExchange Agent must run on the same MVS system as the Datacom table-based ECCR.
- The PowerExchange Logger stores the changes in its log files. The PowerExchange Logger archives active logs when they become full. You must monitor the PowerExchange Logger to ensure that the archiving process keeps pace with the data flow.
  
  If the PowerExchange Logger uses all available active log space, the Datacom table-based ECCR enters a wait state until the PowerExchange Logger archival process makes active log space available.

**RELATED TOPICS:**

- "Managing the PowerExchange Logger for MVS" on page 48

Tasks for Implementing Datacom Table-Based CDC

Complete the following tasks to implement Datacom table-based CDC.

1. “Configuring Datacom for CDC” on page 142.
2. “Configuring the Datacom Table-Based ECCR” on page 142.
3. “Starting the Datacom Table-Based ECCR” on page 146.

**RELATED TOPICS:**

- “CDC Implementation Summary” on page 9

Architectural Overview

This overview describes the Datacom and PowerExchange components that are involved in Datacom table-based CDC.

Datacom CDC Components

The following Datacom components are involved in CDC:

- Source MUF in which the transactions occur.
- Target MUF that contains the CDC tables, if different from the source MUF.
- CDC tables with the change data.
- Programs that capture change data and monitor CDC execution.

For more information about these components, see the *CA Datacom/DB Database and System Administrator Guide*. 
Source MUF

The source Multi-User Facility (MUF) is the Datacom MUF in which the inserts, updates, and deletes occur and are written to the Log Area (LXX) file. For CDC purposes, any MUF configuration that shares a single LXX file is considered a source MUF, including:

- A single MUF
- A MUFPLEX consisting of multiple MUFs that share a single LXX file
- A MUF with a shadow MUF

Target MUF

The target MUF contains the CDC tables. A program supplied with Datacom captures the changes in the LXX file in the source MUF and records the changes in the CDC tables in the target MUF.

The target MUF can match, or differ from, the source MUF.

Datacom CDC Tables

Datacom provides the following tables for CDC:

- TSN (transaction sequence number). Each row of the TSN table defines the boundaries of a unit of work.
- MNT (maintenance records). The rows of the MNT table contain the change data.

Datacom CDC Programs

Datacom provides or defines the following programs for CDC:

- CDC listener program (CDCL). This program monitors the LXX in the source MUF and writes the change data to the CDC tables in the target MUF. The program runs within the target MUF address space. This program is provided with Datacom.
- CDC user listener program (CDCU). This program detects, processes, and deletes committed records in the TSN and MNT tables. PowerExchange uses this program interface to capture change data.
- CDC monitor program (CDCM). This program monitors the CDCL and the CDCU. The task runs within the source MUF address space. This program is provided with Datacom.

Related Topics:

- "Datacom Table-Based ECCR" on page 141

Datacom Table-Based ECCR

The Datacom table-based ECCR is a PowerExchange component that functions as the Datacom CDCU program. The Datacom table-based ECCR performs the following functions:

- Reads change data from the TSN and MNT tables.
- Writes change data to the PowerExchange Logger.
- Removes records from the CDC tables that have been committed to the PowerExchange Logger.

The PowerExchange CDC components run in a separate address space from the target MUF.
Configuring Datacom for CDC

Before PowerExchange can capture changes to Datacom tables, you must configure the following Datacom MUF startup options:

**CDC**

Enables the Datacom Change Data Capture feature and defines this MUF as a source MUF. By default, this option also starts the CDCM subtask in the MUF. You can specify this option during MUF startup only. You cannot specify CDC through the console.

**CDC_BASE**

Enables the specified database or databases for CDC. You can specify CDC_BASE during MUF startup or through the console.

**CDC_TABLE**

Enables the specified database or databases for CDC. You can specify CDC_TABLE during MUF startup or through the console.

**CDCL**

Enables the CDCL task. Specify the following parameters:

- *name* specifies the MUF in which CDCL is enabled, the CDC target MUF.
- *control_ID* specifies the version identifier of the Datacom CDC tables. If you specify a value other than A, specify the same value for the CDC_ID ECCR parameter.

You can specify this option during MUF startup only. You cannot specify CDCL through the console.

**CDCL_DBID**

Specifies the database ID where the CDCL runs. If you specify a value other than 2009, be sure to specify the same value for the CDC_BASE ECCR parameter. You can specify CDCL_DBID during MUF startup or through the console.

For more information about MUF startup options, console commands, and Datacom CDC operation, see the CA Datacom/DB Database and System Administrator Guide.

**Note:** Before starting CDC, ensure that the CDC tables are adequately sized for your environment. For more information, see your CA Datacom documentation.

Configuring the Datacom Table-Based ECCR

Before starting the Datacom table-based ECCR, you must configure the ECCR parameters and JCL.

**ECCR Parameters**

Specify input parameters for the Datacom table-based ECCR inline in the ECCR JCL or in the data set designated by the //DTLCACFG DD statement in the ECCR JCL.

The ECCRDRCMP member in the PowerExchange RUNLIB data set includes the following sample ECCR parameters:

```
MUF=muf_name
REG MUF=registered_muf_name
NO_DATA_WAIT=60
NO_DATA_WAIT2=600
```
The variables shown for the ECCR parameters in the sample member will have the values that you specified with the MVS Installation Assistant.

MUF=muf_name

Required. Specifies the name of the Datacom MUF for which change data is captured. This name must match the internal MUF name that is recorded as part of the key data within the CDC TSN table. Unless REG_MUF specifies a different value, the MUF value is also the same as the MUF name that you specified when you defined the PowerExchange Datacom registration group.

This parameter has no default.

REG_MUF=registered_muf_name

Optional. Specifies the MUF name that you specified when you defined the PowerExchange Datacom registration group. This parameter allows capture registrations defined for one MUF to be used to collect changes for a different MUF. For example, test and production MUFs that have capture active for the same tables can use the same set of registrations.

The ECCR uses the REG_MUF parameter value to read PowerExchange registrations and the MUF parameter value to read the change data from the Datacom CDC tables.

Default is the value specified with the MUF parameter.

NO_DATA_WAIT=seconds

Optional. Specifies the number of seconds that the ECCR waits before doing another read after reading the CDC tables and determining that no new change records have been written. If subsequent reads also return no new records, the ECCR waits NO_DATA_WAIT2 before retrying another read.

The ECCR always waits simultaneously for console input.

Default is 60 seconds.

NO_DATA_WAIT2=seconds

Optional. Specifies the number of seconds that the ECCR waits before performing another read after the second and subsequent consecutive times that the ECCR reads the CDC tables and determines that no new records have been written.

If data is subsequently received, the ECCR reverts to the value for NO_DATA_WAIT.

The ECCR always waits simultaneously for console input.

Default is 600 seconds.

ECCRNAME=eccr_name

Required. The ECCR name for the Datacom ECCR. The ECCR name value must be unique within a PowerExchange Logger group.

Warning: If you change the ECCRNAME value, the ECCR cannot warm start from the last stopped position.
The Datacom ECCR uses the value specified for the following purposes:

- The ECCR name that connects to the PowerExchange Logger to write change data
- The member name that joins the XCF group of the PowerExchange Logger
- As part of the ECCR-UOW field in the control information for each change record written to PowerExchange Logger log files

Valid values are 1 through 8 alphanumeric characters. Default is PWXDCMEC.

Tip: Informatica recommends that you use the same value for the ECCRNAME parameter and the Datacom ECCR started task or job name. This practice allows you to easily identify the Datacom ECCR when reviewing messages and data from the PowerExchange Logger.

**DB_TYPE=db_type**

Required. Specifies the type of database and ECCR. For the Datacom table-based ECCR, this value must be DCM.

This parameter has no default.

**COLDSTART={Y|N}**

Optional. Controls the method by which the ECCR is started. Specify Y to perform a cold start or N to perform a warm start, which restarts the change capture process from its previous stopping point, without loss of data.

Default is N.

**CLEANUP={Y|N}**

Optional. Specifies whether PowerExchange invokes the cleanup subtask at a specific interval to remove changes from the Datacom CDC tables that have been committed to the PowerExchange Logger. By specifying Y, you can ensure that the CDC tables do not fill up. Specify N to prevent the cleanup subtask from being invoked.

Default is Y.

**CLEANUP_INTERVAL=seconds**

Optional. Specifies the number of seconds for the cleanup subtask to wait before removing change data that has been committed from the Datacom CDC tables.

If CLEANUP=Y, the cleanup subtask connects to the Datacom MUF and removes any data that is no longer needed. The subtask then waits again for the specified wait interval.

Default is 300.

**CDC_BASE=dbid**

Optional. Specifies the database identifier for the database to contain the change data. By convention, Datacom uses a database ID of 2009. If this ID is already in use at your site, you can assign a different ID to the CDC database with the Datacom MUF CDCL_DBID startup option. The value that you specify with CDC_BASE must match the value specified with CDCL_DBID.

Default is 2009.

**CDC_ID=A**

Optional. Specifies the version identifier of the Datacom CDC tables. This value must match the value specified with the Datacom MUF CDCL startup option.

If the format of the Datacom CDC tables changes in a later Datacom release, a new version identifier will be assigned.

Default is A.
JCL for the Datacom Log-Based ECCR

To configure the JCL for the Datacom log-based ECCR, edit the ECCRDACM member in the PowerExchange RUNLIB data set.

The JCL in the ECCRDACM member contains the following statements:

```
// ******************************************************************************
// *                                                                           *
// *       RUN DETAIL DATA COM TABLE BASED ECCR                                *
// *                                                                           *
// ******************************************************************************
//ECCRDACM EXEC PGM=DTLCCDCR,REGION=50M
//STEPLIB DD DISP=SHR, DSN=4HLQ..LOADLIB
// DD DSN=4COMCXX, // DISP=(SHR)
// DD DSN=4COMCUST, // DISP=(SHR)

//DD DSN=4COMSPL, // DISP=(SHR)
//DD DSN=4COMMAI, // DISP=(SHR)
//DD DSN=4COMIN, // DISP=(SHR)
//DD DSN=4COMIC, // DISP=(SHR)
//DD DSN=4COMIP, // DISP=(SHR)
//DD DSN=4COMIPC, // DISP=(SHR)
//EDMPARMS DD DISP=SHR, DSN=4HLQEDM..&LOGGER&SUFX..USERLIB
//DTLCFG DD DISP=SHR, DSN=4RUNLIB(EDMSDIR)
//DTLCACFG DD DISP=SHR, DSN=4RUNLIB(EDMDSF)
//DTLMACF DD DISP=SHR, DSN=4HLQVW..CCT
//DTLMGR DD DISP=SHR, DSN=4HLQ..DTLMGR
// EXEC                                                                
// Specifies the ECCR program name (DTLCCDCR).
//STEPLIB DD                                                                
// Includes the PowerExchange load libraries (LOADLIB and LOAD). If you added the load libraries to the system LNKLST concatenation, you do not need to add it to the STEPLIB statement.
//EDMPARMS                                                                
// Specifies the name of the PowerExchange USERLIB library that contains the default options module (EDMSDIR) associated with the PowerExchange Agent and PowerExchange Logger that you are using.
// If you do not include an EDMPARMS statement, or if the library that you specify does not contain the options modules, PowerExchange CDC uses the STEPLIB concatenation to obtain the configuration options.
//DTLCFG                                                                
// Specifies the DBMOVER configuration file for PowerExchange. Some of the parameters are applicable to the Datacom table-based ECCR.
```
Managing Datacom Change Data Capture

PowerExchange provides commands to start and stop the Datacom ECCR.

Starting the Datacom Table-Based ECCR

Before starting the Datacom table-based ECCR, perform the following tasks:

- Start the PowerExchange Listener, PowerExchange Agent, and PowerExchange Logger.
- Configure the Datacom table-based ECCR.
- If you want to run the ECCR as a started task, convert the ECCRDCM JCL to a PROC and copy it to the system PROCLIB library for started tasks.

To start the ECCR:

- If you want to run the ECCR as a started task, issue the MVS START command.
- If you want to run the ECCR as a batch job, submit the configured JCL.

Tip: Because the ECCR is a long-running, it usually is run as a started task.

Related Topics:

- "Configuring the Datacom Table-Based ECCR" on page 142

Stopping the Datacom Table-Based ECCR

To stop the Datacom table-based ECCR, use the MVS STOP command. Enter the STOP command followed by the name of the started task or batch job, for example:

```
STOP DTLCCDCR
```
Managing Datacom Schema Changes

This procedure describes how to manage schema changes for the source or target tables that are involved in change propagation.

To manage Datacom schema changes:

1. Stop all update activity against the relevant Datacom data.
2. Ensure that all changes that occurred under the old schema are processed by PowerExchange.
3. Change the Datacom data structures, and recycle Datacom.
4. Reflect the schema changes in the PowerExchange capture registrations.
5. Recycle the Datacom table-based ECCR.
6. Allow update activity to the Datacom data.
CHAPTER 11

DB2 Change Data Capture

This chapter includes the following topics:
- Introduction to DB2 Change Data Capture, 148
- Planning for DB2 Change Data Capture, 148
- Configuring DB2 for Change Data Capture, 156
- Configuring the DB2 ECCR, 158
- Managing DB2 Change Data Capture, 166
- Managing DB2 Schema Changes, 179

Introduction to DB2 Change Data Capture

The DB2 ECCR allows you to capture changes that are made to a DB2 table. The DB2 ECCR sends the change data to the PowerExchange Logger for recording.

You can have multiple DB2 ECCRs in a single DB2 subsystem and multiple DB2 ECCRs on a single MVS image connecting to multiple DB2 subsystems. However, a single DB2 ECCR must connect to a single DB2 subsystem and communicate with a single PowerExchange Logger.

In a DB2 data-sharing environment, a single DB2 ECCR can capture changes for all members of the data-sharing group.

To capture change data, 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, merge the data map with an extraction map.

Planning for DB2 Change Data Capture

Review the considerations in this section when planning for DB2 CDC.
## DB2 Datatypes

The following table indicates the DB2 datatypes for which PowerExchange supports change data capture:

<table>
<thead>
<tr>
<th>DB2 datatype</th>
<th>Supported for CDC?</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIGINT</td>
<td>Yes</td>
</tr>
<tr>
<td>BINARY</td>
<td>Yes</td>
</tr>
<tr>
<td>BLOB</td>
<td>No</td>
</tr>
<tr>
<td>CHAR</td>
<td>Yes</td>
</tr>
<tr>
<td>CLOB</td>
<td>No</td>
</tr>
<tr>
<td>DATE</td>
<td>Yes</td>
</tr>
<tr>
<td>DBCLOB</td>
<td>No</td>
</tr>
<tr>
<td>DECFLOAT</td>
<td>No</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>Yes</td>
</tr>
<tr>
<td>DOUBLE</td>
<td>Yes</td>
</tr>
<tr>
<td>FLOAT</td>
<td>Yes</td>
</tr>
<tr>
<td>GRAPHIC</td>
<td>Yes</td>
</tr>
<tr>
<td>LONG VARCHAR</td>
<td>Yes</td>
</tr>
<tr>
<td>LONG VARCHARGRAPHIC</td>
<td>Yes</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Yes</td>
</tr>
<tr>
<td>REAL</td>
<td>Yes</td>
</tr>
<tr>
<td>ROWID</td>
<td>No</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>Yes</td>
</tr>
<tr>
<td>TIME</td>
<td>Yes</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>Yes</td>
</tr>
<tr>
<td>VARBINARY</td>
<td>Yes</td>
</tr>
<tr>
<td>VARCHAR</td>
<td>Yes</td>
</tr>
<tr>
<td>VARGRAPHIC</td>
<td>Yes</td>
</tr>
<tr>
<td>XML</td>
<td>No</td>
</tr>
<tr>
<td>User-defined (distinct)</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Must map to a supported datatype.*
DB2 CDC Rules and Guidelines

The following rules and guidelines apply for DB2 change data capture:

- If you did not select Select All Columns or Select all and notify changes when you created the capture registration for a DB2 table, the DB2 ECCR captures changes only for columns that are selected.
- The DB2 ECCR does not support change data capture for the TRUNCATE SQL statement with the IMMEDIATE option.
- DB2 views and aliases are not eligible for change data capture.
- The DB2 ECCR only captures changes that are recorded in the DB2 log as structured query language (SQL) inserts, deletes, updates, or truncates to the source table.
- The DB2 ECCR does not capture the following types of changes:
  - Deletions that result from the DROP TABLE SQL statement or from the DB2 REORG utility with the DISCARD option.
  - Changes from all DB2 utilities, even if you specify the LOG=YES option, with the exception of the DB2 LOAD utility with the RESUME YES and SHRLEVEL CHANGE options. With these options, the DB2 ECCR does capture changes from the DB2 LOAD utility.
- The DB2 ECCR does not support a single unit-of-work (UOW) that contains both DML and DDL for the same table, such as:
  - SQL inserts, deletes, updates, or truncates
  - CREATE or ALTER statements

Limitations for Compressed DB2 Table Spaces

The following limitations apply when using change data capture for tables, which reside in compressed table spaces:

- The compression dictionary must be available at the time that the DB2 ECCR requests the DB2 log data from DB2. You should not stop databases or table spaces that contain the tables for which you wish to capture changes, unless you are certain that the DB2 ECCR has processed all pending DB2 log data for that table.
- In a DB2 data sharing environment, the DB2 subsystem to which the DB2 ECCR connects needs access to compression dictionaries. You must make the table spaces and the buffer pools for any tables registered for capture accessible to the DB2 subsystem to which the DB2 ECCR connects. If the DB2 subsystem to which the DB2 ECCR connects cannot access table spaces or buffer pools for registered compressed tables, DB2 passes decompressed change records to the DB2 ECCR. When the DB2 ECCR receives compressed change records, it abends with a user abend code 3680 and reason code 02710009 after issuing the following message to the EDMMSG log:

```
PWXEDM177462E ROW NOT DECOMPRESSED, TABLE=table_name, DB2 LOG LOCATION=irsn
```

If this occurs, you must remove the table from capture by inactivating or deleting its registration. After the table is removed from capture, you can warm start the DB2 ECCR.
- Because the compression dictionary must match the DB2 log records, the following compression restrictions apply if you did not specify KEEPDICTIONARY:
  - When compression is enabled with the COMPRESS YES table space option and you use one of the following utilities, the compression dictionary is rebuilt or recovered:
    - DB2 REORG TABLESPACE utility
    - DB2 LOAD utility with the REPLACE or RESUME NO options
  - If the DB2 log records that you want to capture were written prior to these utilities being executed, DB2 may no longer be able to decompress those rows. In this situation, DB2 passes the compressed rows to the DB2 ECCR and the DB2 ECCR abends.
When you start the DB2 ECCR, it uses the current compression dictionary. For this reason, be careful when using either the START WARM statement or the START STARTLOC= statements to start from a specified point in the DB2 log. If the DB2 log records that you want to capture need an earlier compression dictionary, the DB2 subsystem may not be able to decompress the change records for the DB2 ECCR, which will abend.

Considerations for FIELDPROC and EDITPROC Exit Routines

The following considerations apply to exit routines for DB2 source tables:

- Libraries that contain FIELDPROC or EDITPROC exit routines that processes updated rows must be concatenated in the STEPLIB statement of the DB2 ECCR startup procedure.
- If you update a FIELDPROC or EDITPROC exit routine, you must do the following:
  - Refresh or restart the DB2 ECCR to initiate the new routine.
  - Ensure that the DB2 ECCR uses a version of the exit routine that matches the DB2 log records that you want to capture.

Relationships with Other PowerExchange Components

The DB2 ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- A DB2 ECCR must log all changes to a single PowerExchange Logger that runs on the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must run on the same MVS system as the DB2 ECCR.
- A single DB2 ECCR that attaches to a single member of the DB2 data sharing group can process changes for all members in the data sharing group. You do not need to use the Post-Log Merge configuration of the PowerExchange Logger to capture DB2 change data when you use DB2 data sharing.
- If you use the Post-Log Merge configuration of the PowerExchange Logger for another reason, a single DB2 ECCR can attach to a single member Logger of the Post-Log Merge group.
- Operational issues in the PowerExchange Logger can cause the DB2 ECCR to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the DB2 ECCR continues the capture and recording of change data without any loss of data.

You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

Related Topics:

- “Monitoring the PowerExchange Logger for MVS” on page 51

DB2 ECCR Capture Directory Tables

The DB2 ECCR uses a set of DB2 tables, called the capture directory tables, to track information about the DB2 tables that are registered for change data capture. These tables are created during PowerExchange installation.

These capture directory tables must reside on the DB2 subsystem to which the DB2 ECCR connects to capture change data from registered tables. The capture directory tables reside in their own database and table space and have their own unique index.
The following table describes the purpose of each DB2 ECCR capture directory table:

<table>
<thead>
<tr>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCAPCOLUMNS</td>
<td>Stores catalog and status information about all of the columns in the tables registered for change data capture.</td>
</tr>
<tr>
<td>TCAPFIELDS</td>
<td>Stores information about the columns that use a field procedure exit routine (FIELDPROC) and that are in tables registered for change data capture.</td>
</tr>
<tr>
<td>TCAPSTATUS</td>
<td>Stores status information about all of the tables registered for change data capture.</td>
</tr>
<tr>
<td>TCAPTABLEPART</td>
<td>Stores information about table space parts for all table spaces that contain tables registered for change data capture.</td>
</tr>
<tr>
<td>TCAPTABLES</td>
<td>Stores catalog and status information about the tables registered for change data capture.</td>
</tr>
<tr>
<td>TCAPTABLESPACE</td>
<td>Stores catalog and status information about all table spaces in the DB2 catalog, including those table spaces containing tables that are not registered for change data capture.</td>
</tr>
<tr>
<td>TCAPUPDATE</td>
<td>Stores information that the DB2 ECCR uses to coordinate handling of the DB2 log read process.</td>
</tr>
<tr>
<td>TCAPWORK</td>
<td>Stores changes to the DB2 system catalog tables until the UOW that contains the catalog table change is committed.</td>
</tr>
</tbody>
</table>

In the MVS Installation Assistant, you specify a DB2 creator name for the DB2 capture directory tables and a DB2 owner for the DB2 ECCR plans and packages. You also specify the following information for customizing the jobs that create these tables and related DB2 objects:

- DB2 subsystem identifier (SSID)
- Database name
- STOGROUP
- TCAPWORK buffer pool name

The XIDDB220 member of the RUNLIB library creates the DB2 tables spaces, tables, and indexes for the DB2 ECCR capture directory tables. The SETUPDB2 job submits the XIDDB220 job. The DDL for the capture directory tables are in the following RUNLIB members:

**DB2TGEN**

Creates the database and the table space for each table.

**DB2SGEN**

Creates the tables, except for DB2 Version 8 new-function mode and later.

**DB2SGEN8**

Creates the tables for DB2 Version 8 new-function mode and later.

**DB2IGEN**

Creates the unique index for each of the tables.

**Capture Table Buffer Pool Requirements**

With the exception of the TCAPWORK table, the minimum buffer pool size required is 4 KB.
If you are using DB2 Version 8 new-function mode or DB2 Version 9.1, assign a buffer pool of at least 16 KB to the TCAPWORK table. Prior to DB2 Version 8 new-function mode, the TCAPWORK table required only an 8 KB buffer pool.

You can assign larger buffer pool sizes than the minimum requirements for the DB2 ECCR.

Capture Table Sizing

The capture directory tables are each created in their own unique DB2 table space. The following table describes the default space specifications from the PowerExchange installation process and the contents of the DB2 ECCR capture directory tables:

<table>
<thead>
<tr>
<th>Default Tablespace / Table Name</th>
<th>Install PRIQTY</th>
<th>Install SECQTY</th>
<th>Table Sizing Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWXPCOLS / TCAPCOLUMNS</td>
<td>180 KB</td>
<td>20 KB</td>
<td>Up to three rows per column across all the tables being captured</td>
</tr>
<tr>
<td>PWXPFLDS / TCAPFIELDS</td>
<td>3 KB</td>
<td>1</td>
<td>One row per column that has a FIELDPROC across all tables being captured</td>
</tr>
<tr>
<td>PWXPSTAT / TCAPSTATUS</td>
<td>3 KB</td>
<td>1 KB</td>
<td>One row per table being captured</td>
</tr>
<tr>
<td>PWXPTBPPT / TCAPTABLEPART</td>
<td>180 KB</td>
<td>20 KB</td>
<td>One row for each non-partitioned table space and a row for each partition in a partitioned table space</td>
</tr>
<tr>
<td>PWXPTABL / TCAPTABLES</td>
<td>180 KB</td>
<td>20 KB</td>
<td>Up to three rows per table being captured</td>
</tr>
<tr>
<td>PWXPTBSPT / TCAPTABLESPACE</td>
<td>180 KB</td>
<td>20 KB</td>
<td>Up to three rows per tablespace in the DB2 catalog, including those table spaces containing tables that are not registered for capture</td>
</tr>
<tr>
<td>PWXPUPDT / TCAPUPDATE</td>
<td>3 KB</td>
<td>1 KB</td>
<td>One row per DB2 ECCR</td>
</tr>
<tr>
<td>PWXPWORK / TCAPWORK</td>
<td>720 KB</td>
<td>48 KB</td>
<td>One row per in-flight catalog change</td>
</tr>
</tbody>
</table>

The default values provided in the PowerExchange installation process are usually sufficient for most DB2 subsystems, although some of the table spaces may create secondary extents. If you have more than 5,000 tables in the DB2 subsystem or a large number of tables or columns in tables registered for capture, the PowerExchange install values may need adjustment. You should monitor these table spaces to ensure they are able to extend as needed. The DB2 ECCR abends if it cannot extend a capture directory table.

Running Multiple DB2 ECCRs

There are some typical scenarios where you might want to run multiple DB2 ECCRs on the same MVS image:

- You have multiple DB2 subsystems on that MVS image and they are either not data sharing DB2 subsystems or not apart of the same data sharing group. For example, you have test and production DB2 subsystems on the same MVS image and you need to capture change data from both subsystems.

- You have a single DB2 subsystem and you want a separate capture environment for certain tables. For example, if the DB2 subsystem contains both test and production tables, you might want to have a capture environment for the test tables and another capture environment for the production tables.
Multiple DB2 Subsystems, Single MVS Image

You might need to capture changes from multiple DB2 subsystems that all run on a single MVS image. The following considerations apply to capturing changes on a single MVS image that contains multiple, non-data-sharing DB2 subsystems:

- A unique instance of the DB2 ECCR is required for each unique subsystem. The DB2 ECCR connects only to a single DB2 subsystem and, with the exception of DB2 data sharing environments, can only capture changes for that specific DB2 subsystem.
- The capture name that you specify in the CA statement in the REPL2CTL file must be unique for each DB2 ECCR and also within a single MVS image and a sysplex.
- Each DB2 ECCR can, but is not required to, have its own unique set of PowerExchange Listener, Agent, and Logger tasks. For instance, if you are capturing data from test and production DB2 subsystems, you might want to keep the capture environments separate. If you are capturing data from two separate test systems, using the same capture environment might be acceptable.

Single DB2 Subsystem with Multiple DB2 ECCRs

The following considerations apply to multiple DB2 ECCRs in the same DB2 subsystem:

- Each execution must have its own unique parameter files which are specified in the REPL2CTL and REPL2OPT DD statements.
- Each DB2 ECCR must have its own set of DB2 Capture Directory tables.
- Each DB2 ECCR must have its own unique qualifier and plan name in the BIND for the packages and plans.
- The capture name you specify in the CA statement in the REPL2CTL file must be unique for each DB2 ECCR, and also within both a single MVS image and a sysplex.
- DB2 registrations contain either the DB2 subsystem ID (SSID) or group attachment name. Therefore, each DB2 ECCR must have its own unique set of PowerExchange Listener, Agent, and Logger tasks. This allows the registrations to be split up as desired between the various capture environments.

DB2 Data-Sharing Considerations

DB2 data sharing is a collection of DB2 subsystems, called a data sharing group, that operate from the same DB2 catalog. In general, any table in the DB2 catalog can be read or updated from any DB2 subsystem in the data sharing group. Access control that is provided by DB2 through grants, plans, and so on still applies.

When initially implementing the DB2 ECCR in a data-sharing environment, the following considerations apply:

- In the DB2 bind JCL for the DB2 ECCR in member XIDDB225 in the RUNLIB library, you can use either the DB2 group attachment name or the SSID when specifying the SYSTEM operand of the DSN command.
- The DB2 ECCR captures changes for tables registered under the name specified in the RN parameter of the REPL2OPT control statement. The RN parameter can specify an SSID of a member of the data sharing group or the group attachment name. Use of the group attachment name is recommended.
  
  All tables must be registered under either a single DB2 SSID or group attachment name.
- The DB2 ECCR uses the CN parameter to attach to DB2. You can use either the SSID or group attachment name to attach. Specification of the CN parameter is optional unless you want to attach to a specific DB2 subsystem. If it is not specified, the DB2 ECCR uses the RN value to do the attach. Of course, should the ECCR be moved, it still must have access to its proper Agent and Logger.
If you want to have the flexibility to move the DB2 ECCR to other MVS systems running members in the same DB2 data-sharing group without making parameter changes, either use the DB2 group attachment name in the CN parameter or allow it to default from the RN parameter.

- If you run DB2 for z/OS Version 9.1 in new-function mode in a data-sharing environment, multiple DB2 log records in a single data-sharing member can have the same LRSN. In this case, the DB2 ECCR generates unique, ascending sequence tokens for these records. Also, if two of the records are begin-UR records with the same LRSN, the PowerExchange Logger generates corresponding begin-UOW records with unique UOWIDs.

**Considerations If You Migrate to DB2 Version 8 New-Function Mode or DB2 Version 9.1**

To capture change data when running DB2 Version 8 new-function mode or DB2 Version 9.1, the DB2 ECCR requires changes to its capture directory tables. PowerExchange provides SQL and procedures to upgrade existing capture directory tables to support DB2 Version 8 new-function mode and DB2 Version 9.1.

Even if you altered the capture directory tables to support DB2 Version 8 new-function mode while running a previous release of PowerExchange, you must upgrade the capture directory tables as a part of the upgrade to PowerExchange 8.6 or later. PowerExchange 8.6 includes improvements in the DB2 ECCR for DB2 Version 8 new-function mode, and also supports DB2 Version 9.1.

**Note:** With PowerExchange 8.6 or later, you only need to upgrade the capture directory tables once to support both DB2 Version 8 new-function mode and DB2 Version 9.1.

**New Users of the DB2 ECCR**

If you selected DB2 CDC when you ran the MVS Installation Assistant, the SETUPDB2 job created the DB2 ECCR capture directory tables. Use the XJOBS member of the RUNLIB library to verify whether DB2 CDC was selected. If DB2 CDC was not selected when you installed PowerExchange, upgrade your PowerExchange installation to support DB2 CDC.

The SETUPDB2 job submits the XIIDDB210 job, which creates the capture directory tables using one of the following members from the RUNLIB library:

- DB2SGEN, if you did not select **DB2 V8+ New-Function Mode** in the MVS Installation Assistant
- DB2SGEN8, if you selected **DB2 V8+ New-Function Mode** in the MVS Installation Assistant

If XIIDDB210 used the DB2SGEN8 member, you do not need to perform any further migration tasks for the DB2 ECCR. Otherwise, you must upgrade the format of the capture directory tables. Alternatively, you can do the following if you are not yet using the DB2 ECCR to capture changes:

- Set the value of the DB28NFM variable to 1 in the GENBULK member in the RUNLIB library so that the DB2SGEN8 member is used.
- Delete the capture directory tables.
- Rerun the XIIDDB210 job, which can be found in the RUNLIB library.

**Related Topics:**

- “DB2 ECCR Capture Directory Table Upgrades” on page 172

**Existing Users of the DB2 ECCR**

If you are an existing user of the DB2 ECCR, you must alter the DB2 ECCR capture directory tables to support DB2 Version 8 new-function mode and DB2 Version 9.1. You must also alter these tables if you did not select the **DB2 V8+ New-Function Mode** option when you installed PowerExchange and if you have run the install jobs.
PowerExchange provides SQL to upgrade the capture directory tables to allow the DB2 ECCR to support DB2 Version 8 new-function mode and DB2 Version 9.1. The following rules apply to altering the capture directory tables:

- You must run DB2 Version 8 new-function mode or DB2 Version 9.1 to be able to change the capture directory tables for support of the DB2 version you are running. If you run DB2 Version 8 compatibility mode or earlier, you cannot change the capture directory tables.
- Change the capture directory tables to support DB2 Version 9.1 before making any schema change to the tables that are registered for change capture and before starting the DB2 ECCR with DB2 Version 9.1.

**RELATED TOPICS:**
- “DB2 ECCR Capture Directory Table Upgrades” on page 172

**Consideration If You Migrate to DB2 Version 9.1 New-Function Mode**

After you upgrade to DB2 Version 9.1 new-function mode, install the fix for APAR PK41156 if you plan to reload or reorganize compressed table spaces that contain tables registered for change capture. You might also need to enable a DSNZPARM option that is provided in the fix.

By default, DB2 Version 9.1 ignores the KEEPDICTIONARY specification the first time a table space is processed by any of the following utilities:

- REORG
- LOAD REPLACE
- LOAD PART REPLACE

**Note:** DB2 does honor the KEEPDICTIONARY specification if a table in the table space contains an EDITPROC or VALIDPROC. For more information, see the description of APAR PK41156.

DB2 Version 9.1, with APAR PK41156, provides a new DSNZPARM option called HONOR_KEEPDICTIONARY. You can enable this option to cause DB2 to honor the KEEPDICTIONARY specification during the first reorganization or reload of a table space in new-function mode.

For table spaces containing tables for which the DB2 ECCR is capturing changes, do one of the following:

- When you install the fix for APAR PK41156, enable the HONOR_KEEPDICTIONARY option in DSNZPARM.
- When you perform the first reload or reorganization, verify that the DB2 ECCR has captured all of the changes for the tables in the table space.

Otherwise, the DB2 ECCR may be unable to process compressed change records and fail.

**Configuring DB2 for Change Data Capture**

Before you can capture changes for DB2 tables, you must configure the DB2 catalog tables and the DB2 logs for change data capture. Verify the following system requirements are met:

- Your system is running a supported version of DB2 and the recommended IBM maintenance has been applied.
- The DB2 subsystem is running on the MVS subsystem where you plan to start the DB2 ECCR.
Activating Change Data Capture for DB2 Catalog Tables

PowerExchange requires that DATA CAPTURE CHANGES be enabled for the following IBM DB2 catalog tables when using DB2 Version 8 or later:

- SYSTABLES
- SYSCOLUMNS
- SYSTABLESPACE
- SYSFIELDS
- SYSCOPI

**Warning:** The DB2 ECCR fails if DATA CAPTURE CHANGES is not enabled for these catalog tables, when running DB2 Version 8 or later.

Prior to DB2 Version 8, the DB2 catalog tables only required DATA CAPTURE CHANGES to be enabled when using the DB2 ECCR IFI306OPT statement. The DB2 ECCR issues the following messages when running DB2 Version 7 without DATA CAPTURE CHANGES enabled for the aforementioned DB2 catalog tables:

- PWXEDM177540W Some DB2 catalog tables not defined with Data Capture Changes
- PWXEDM177541W Migration to DB2 V8 will not be allowed

In DB2 Version 8 and later, the DB2 ECCR issues the following message when one or more of the DB2 catalog tables do not have DATA CAPTURE CHANGES enabled:

- PWXEDM177543E Capture program of DB2 Replication ending - DB2 Catalog tables not Data Capture Changes

When this message is issued, the DB2 ECCR terminates without processing any data.

**RELATED TOPICS:**

- "Altering DB2 System Tables for DATA CAPTURE CHANGES" on page 171

Managing DB2 Logs

The DB2 ECCR relies on the DB2 logs for change information. If the logs are lost, changes that should be captured are also lost. To avoid losing the log data sets, use dual logging including dual archive logs. Archive logs are required if it ever becomes necessary to retrieve data that has become inactive due to DB2 log switch processing.

**Note:** If you lose DB2 log data that the DB2 ECCR has not already processed, you must rematerialize the target tables before restarting the DB2 ECCR. Because your source and target tables are synchronized, you should begin capturing from the current DB2 log location. To do so, be sure that you use the START COLD statement in your REPDB2OP parameter file when you restart the DB2 ECCR.

DB2 Logging in a Data Sharing Environment

The Post-Log Merge option of the PowerExchange Logger allows you to capture changes to multiple PowerExchange Loggers on multiple MVS systems, and extract those merged changes from a single Logger. With DB2 data sharing, the Post-Log Merge option of the PowerExchange Logger is not required. The DB2 ECCR uses DB2 IFI 306 calls that return change information from all members in the data sharing group.

**Note:** Post-Log Merge is not required for DB2 data sharing. However, if Post-Log Merge is being used for another reason, the DB2 ECCR can also attach to a member Logger of the Post-Log Merge group, even when running in data sharing mode. A single DB2 ECCR is still all that is required, even when connecting to a PowerExchange Logger in a Post-Log Merge configuration.
Configuring the DB2 ECCR

To successfully capture DB2 change data using the DB2 ECCR, there are various operational considerations that must be understood and requirements that must be met.

DB2 ECCR Rules and Guidelines

The DB2 ECCR has the following rules and guidelines:

- You must define DB2 source tables with the DATA CAPTURE CHANGES option. For more information about this option, see the IBM DB2 documentation.
- Tables with a long table name, owner, or column name can only be registered for change data capture with PowerExchange 8.0 and later.
- The first start of the DB2 ECCR should be performed with START COLD parameter. Thereafter it should normally be run with START WARM unless a cold or special start is required for recovery purposes.
- The DB2 ECCR requires at least one active registration to start successfully. If there are no active registrations, the DB2 ECCR abends with U3680 and message PWXEDM177509E indicating there are no active registrations.
- The DB2 ECCR issues IFCID 306 READS requests to read the DB2 log data. To issue the READS request, MONITOR TRACE 1 needs to be started. Therefore, the user ID under which the DB2 ECCR runs must have the following authorities:
  - TRACE authority to issue the START TRACE command
  - DISPLAY authority to issue a DISPLAY TRACE to determine if the MONITOR TRACE is already active
  - MONITOR2 authority to issue the READS request to get the log data containing the changes it needs to capture

If the user ID for the DB2 ECCR has SYSOPR, SYSCTL, or SYSADM authority, you do not need to grant additional authority.

If the DB2 ECCR starts the trace during initialization, it issues message:

```
PWXEDM177008I -START TRACE(MONITOR) PLAN(plan) LOCATION(caname) CLASS(1) HAS BEEN EXECUTED
```

If MONITOR TRACE is started, the DB2 ECCR does not issue the START TRACE command. If MONITOR TRACE is stopped or has never started, the DB2 ECCR starts it.

- The first time that the DB2 ECCR receives a change record for a particular table, it compares the registered schema for that table to the schema for the table in the DB2 catalog. If the schemas do not match, the DB2 ECCR issues a report and terminates.

The DB2 ECCR also performs schema verification the first time that the ECCR receives a change record for a table following a schema change on that table. To prevent the ECCR from terminating when the table schemas do not match, you must update the corresponding capture registration any time that the source schema changes.

DB2 ECCR REPL2CTL Statement

The REPL2CTL statement control the name of the DB2 ECCR and where it stops processing. PowerExchange provides a sample REPL2CTL file in the REPDB2CT member in the RUNLIB library. Use the following syntax for the REPL2CTL statement:

```
CA NAME=eccr_name [STOPAFT |LOGLOC=rba|LOGTS=timestamp]
```

The REPL2CTL statement has the following parameters:
CA NAME=ecr_name

Required. The ECCR name for the DB2 ECCR. The ECCR name value must be unique within a sysplex.

Warning: If you change the CA NAME value, the ECCR cannot warm start from the last stopped position.

The DB2 ECCR uses the value specified for the following purposes:

♦ The ECCR name that connects to the PowerExchange Logger to write change data
♦ The member name that joins the XCF group of the PowerExchange Logger
♦ The minor name of the DB2CAPT ENQ

During initialization, the DB2 ECCR issues the DB2CAPT ENQ as an exclusive ENQ with SCOPE=SYSTEMS.
♦ As part of the ECCR-UOW field in the control information for each change record written to PowerExchange Logger log files

Valid values are 1 through 8 alphanumeric characters. Default is PWXDB201. The default value can be modified in the MVS Installation Assistant during the installation of PowerExchange.

Tip: Informatica recommends that you use the same value for the CA NAME parameter and the DB2 ECCR started task or job name. This practice allows you to easily identify the DB2 ECCR when reviewing messages and data from the PowerExchange Logger.

STOPAFT [LOGLOC=rba][LOGTS=timestamp]

Optional. Specifies when the DB2 ECCR should stop processing. STOPAFT can be used with any type of start for the DB2 ECCR. Only one STOPAFT statement can be used in the REPL2CTL file.

LOGLOC=rba

The value of rba is the RBA, or LRSN if the DB2 ECCR is connected to a member of a DB2 data sharing group, at which the DB2 ECCR is to terminate.

The value must be larger than the RBA or LRSN location at which the DB2 ECCR starts. If not, the DB2 ECCR stops as soon as the first record is obtained from the DB2 log data.

LOGTS=timestamp

The value of timestamp is the time at which the DB2 ECCR is to terminate. The timestamp value has the following format: yyyy-mm-dd-hh.mm.ss.nnnnnn. The variables in the timestamp value are:

♦ yyyy is the year, such as 2005
♦ mm is the numeric month
♦ dd is the numeric day of the month
♦ hh is the hour of the day
♦ mm is the minute of the hour
♦ ss is the second of the minute
♦ nnnnnn is the fraction of the second

The timestamp must be a valid date. For example, 2005-02-31-17.15.59.000000 is invalid because February 31st does not exist.

The timestamp value must be later than the time at which the DB2 ECCR starts. If not, the ECCR stops as soon as the first record is obtained from the DB2 log data.
Sample REPL2CTL Parameters

The following is an example of the REPL2CTL parameters from the REPDB2CT member in the RUNLIB library:

```
CA NAME=FWXDB201
```

DB2 ECCR REPL2OPT Statements

Specify DB2 ECCR processing options using the REPL2OPT statements. PowerExchange provides a sample REPL2OPT file in member REPDB2OP in the RUNLIB library.

Configure the REPL2OPT file prior to starting the DB2 ECCR for the first-time. You can change the REPL2OPT file after you have started the DB2 ECCR. Some REPL2OPT statements can be changed by refreshing the DB2 ECCR. Others requires that you restart the DB2 ECCR.

Use the following syntax for the REPL2OPT statements:

```
[CHKSCHEM [NO|YES|WARN]]
DB2 PLAN=plan_name {RN=rn_ssid|CN=cn_ssid|RN=nn_ssid CN=cn_ssid}
[EC ERRNR=err_num]
[IPII3OPT]
[MODE={RB|CM}]
START {COLD|WARN|STARTLOC=rba [USEDIR],[USESTAT]}
[STAT LEV=ST [SEC=secs]}
[TRACE [option]]
```

All of the REPL2OPT statements must begin in column 1.

REPL2OPT has following statements:

**CHKSCHEM (NO|YES|WARN)**

Optional. Specifies whether the DB2 ECCR is to verify the schema registrations at ECCR initialization and, if so, how to handle errors. This schema verification is in addition to the verification performed when the ECCR receives the first change record for a registered schema.

**NO**

Default. Does not verify your registered schema at initialization. The DB2 ECCR continues to verify each registered schema against the information in the DB2 catalog when the ECCR receives the first change record for that schema.

**YES**

Checks all registered schema information against the information in the DB2 catalog at initialization and when you refresh the ECCR. This option terminates the ECCR startup process if the verification encounters errors.

**WARN**

Checks all registered schema information against the information in the DB2 catalog at initialization and when you refresh the ECCR. This option issues a warning message if the verification encounters errors, but the DB2 ECCR continues.

Refresh or restart the DB2 ECCR to activate a new value for this statement.

**DB2 PLAN=plan_name {RN=nn_ssid|CN=nn_ssid|RN=nn_ssid CN=nn_ssid}**

Required. Specifies the plan and subsystem name or group name for the DB2 system to which the DB2 ECCR attaches.

You can code both RN, CN, or RN and CN. Only one of these keywords is required. The specified keyword substitutes for the non-specified keyword, if only one of RN and CN is coded.
Tip: When implementing the DB2 ECCR in a data sharing environment, Informatica recommends using the group attachment name for the RN keyword and in the registration group in the PowerExchange Navigator. The PowerExchange Logger uses the registration tag name to capture changes. The registration tag name contains the value specified in the Database Instance field in the registration group. Using the group attachment name makes registration tag names and captured change data independent of a specific data sharing group member SSID.

**PLAN=plan_name**

Identifies the DB2 plan name that the ECCR uses.

The following rules and guidelines apply:

- The PLAN keyword must be in uppercase and begin in column 5.
- Plan names must be in uppercase.
- Plan names can be between 1 and 8 characters long.
- Plan names less than eight characters must be padded with spaces to make eight characters.

For example, if your plan name is MYPLAN, you must add three spaces between the plan name and the RN keyword.

**RN=rn_name**

Identifies the DB2 subsystem name used in the capture registrations.

The value specified for this keyword must match the value specified in the Database Instance field in the registration group defined in the PowerExchange Navigator. The value for this keyword defaults to the CN value if not specified.

The following rules and guidelines apply:

- The RN keyword must be in uppercase and begin in column 19.
- The rn_name value must be in uppercase.
- The rn_name value can be 1 to 4 characters long and is either the DB2 subsystem ID (SSID) or the DB2 group attachment name.

**CN=cn_ssid**

Identifies the DB2 subsystem to which the DB2 ECCR should connect. This value for this keyword defaults to the RN value if not specified.

The following rules and guidelines apply:

- CN must be in uppercase and begin in column 27.
- The cn_name value must be in uppercase.
- The cn_name value can be 1 to 4 characters long and is either the DB2 subsystem ID (SSID) or the DB2 group attachment name.

The following examples show combinations of RN and CN keywords:

- If you have a non-data-sharing environment with a DB2 subsystem SS01, code the DB2 statement as follows:
  
  DB2 PLAN=plan_name RN=SS01

- If you migrate SS01 to a data-sharing environment called GRP1, code the DB2 statement as follows:

  DB2 PLAN=plan_name RN=SS01 CN=GRP1
If you add another DB2 subsystem, SS02, to the data-sharing group GRP, continue to use the previous statement to run one instance of the ECCR on either SS01 or SS02. You must continue to register new tables under the name SS01.

If you have a data sharing environment with the previous configuration and do not have existing capture registrations, code the DB2 statement as follows:

```
DB2 PLAN=plan_name RN=GRP1
```

Create all capture registrations under the GRP1 name.

Restart the DB2 ECCR to activate a new value for this statement.

**EC PERMIL=err_num**

Optional. Specifies the maximum number of acceptable errors per thousand updates.

The default value is zero.

Refresh or restart the DB2 ECCR to activate a new value for this statement.

**IFI306OPT**

Optional. IFI306OPT changes the manner in which the DB2 ECCR interacts with the DB2 catalog tables.

For this option to be effective, the DB2 catalog tables must have the DATA CAPTURE CHANGES option. When the DB2 catalog tables have DATA CAPTURE CHANGES enabled and you specify IFI306OPT, DB2 passes a reduced volume of change information to the DB2 ECCR.

Restart the DB2 ECCR to activate a new value for this statement.

**MODE={RB|CM}**

Optional. Specifies the DB2 ECCR mode of operation.

**RB**

Default. Designates rollback mode. This option does not send aborted UOW records to the PowerExchange Logger.

**CM**

Designates compensation mode. This option sends compensation and SQL records to the PowerExchange Logger.

Restart the DB2 ECCR to activate a new value for this statement.

**START {COLD|WARM|STARTLOC=rba [USEDIR],[USESTAT]}**

Required. Controls the method by which the DB2 ECCR is started.

**COLD**

Starts the DB2 ECCR for the first time or restarts the ECCR after a major system failure.

**WARM**

Restarts change-capture process from its previous stopping point, without loss of data.

Use this option to restart the DB2 ECCR after a successful shutdown using the STOP command or the MODIFY QUIESCE command. Typically, you should use the WARM keyword when starting the ECCR.

**STARTLOC=rba [USEDIR],[USESTAT]**

Restarts change-capture process from a specific point in the DB2 log.

The `rba` value specifies the 12-digit hexadecimal DB2 log RBA or log record sequence number (LRSN) at which the DB2 ECCR should start.
If you use the STARTLOC parameter, include the following options:

- **USEDIR.** Specifies that the DB2 ECCR uses the source table information from the data-resource information that was registered in the PowerExchange repository when the STARTLOC option was specified.

- **USESTAT.** Specifies that the DB2 ECCR uses the status active (C) or inactive (N) for the table registration that existed when the STARTLOC option was specified.

Restart the DB2 ECCR to activate a new value for this statement. Ignored when you refresh the DB2 ECCR.

**STAT LEV={ST|SQ} [SEC=secs]**

Optional. Specifies the level at which the DB2 ECCR displays capture statistics.

The DB2 ECCR displays statistics before terminating, when you issue a DISPLAY command, or when you issue a REFRESH command. You can find these statistics in the EDMMSG file in DB2 ECCR JCL.

**LEV={ST|SQ}**

Identifies the level of table statistics printed. This value can ST for table summary statistics or SQ for table SQL operation statistics.

**Note:** The SQ option prints two lines of output per table registered for capture. To minimize the size of the EDMMSG output, use LEV=ST. You can issue the DISPLAY command with the SQ option to write a table SQL operation statistics report.

**SEC=secs**

Specifies the number of seconds in the reporting period. The default is 3600 (1 hour).

Refresh or restart the DB2 ECCR to activate a new value for this statement.

**TRACE [option]**

Optional. The TRACE statement can help to troubleshoot the behavior and performance of the DB2 ECCR.

**Note:** Use the TRACE statement and its keywords only under the advice of Informatica Global Customer Support.

To activate more than one trace, you must specify the TRACE statement multiple times. If you specify TRACE without an keyword, a minimal trace is activated, which is the same level of tracing specified by the MINI keyword.

The TRACE statement must start in column 1 and the trace option, if specified, must start in column 7.

You can specify the following TRACE options:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINI</td>
<td>Default. Activates a minimal trace.</td>
</tr>
<tr>
<td>ALL</td>
<td>Activates all tracing within the DB2 ECCR.</td>
</tr>
<tr>
<td>DB2CAT</td>
<td>Traces DB2 catalog access.</td>
</tr>
<tr>
<td>CAPDIR</td>
<td>Traces DB2 ECCR capture directory access.</td>
</tr>
<tr>
<td>COMMIT</td>
<td>Traces DB2 ECCR commit and rollback activity.</td>
</tr>
<tr>
<td>ROLLBACK</td>
<td>Has the same function as COMMIT.</td>
</tr>
</tbody>
</table>
The following is an example of the REPL2OPT file which is supplied in RUNLIB member REPDB2OP:

```
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SERVICES</td>
<td>Traces DB2 ECCR services.</td>
</tr>
<tr>
<td>RECCDC</td>
<td>Traces log record processing for captured DB2 change data.</td>
</tr>
<tr>
<td>RECDLL</td>
<td>Traces DB2 DDL log record processing.</td>
</tr>
<tr>
<td>RECURCTL</td>
<td>Traces DB2 log UR Control record processing.</td>
</tr>
<tr>
<td>LOGREC</td>
<td>Traces reading a DB2 log record.</td>
</tr>
<tr>
<td>LOGSEG</td>
<td>Traces reading a DB2 log record segment.</td>
</tr>
<tr>
<td>LOGIFI</td>
<td>Traces reading the DB2 log through IFI.</td>
</tr>
<tr>
<td>LOGDSNJ</td>
<td>Traces reading the DB2 log through DSNJ.</td>
</tr>
<tr>
<td>CB</td>
<td>Traces DB2 ECCR internal control block activity.</td>
</tr>
<tr>
<td>DECOMPRESSION</td>
<td>Traces record decompression for captured records.</td>
</tr>
<tr>
<td>EDITPROC</td>
<td>Traces EDITPROC processing for captured records.</td>
</tr>
<tr>
<td>FIELDPROC</td>
<td>Traces FIELDPROC processing for captured records.</td>
</tr>
<tr>
<td>FMSG</td>
<td>Traces message formatting for captured records.</td>
</tr>
</tbody>
</table>
```

**Note:** When you use the TRACE statement and its keywords, the REPL2TRA DD statement must be present in the JCL.

Refresh or restart the DB2 ECCR to activate a new value for this statement.

**RELATED TOPICS:**
- “Sample REPL2CTL Parameters” on page 160
- “DB2 Catalog Tables” on page 174

**Sample REPL2OPT Statements**

The following is an example of the REPL2OPT file which is supplied in RUNLIB member REPDB2OP:

```
* Use only one START statement for an execution of the DB2 ECCR.  *
* Use only one DB2    statement for an execution of the DB2 ECCR.  *
* Other statements contain default values.                       *
* All the parameters below are column specific, beginning in column 1 *
**************************************************************************
START WARM
 * START COLD
 * START STARTLOC=00000000000000 USEDIR,USESTAT
 * DB2 PLAN=DTLCFV80 RN=DSN1
 DB2 PLAN=<plan_name> RN=<ssid>
 * DB2 PLAN=DTLCFV52 CN=DSN1
 EC_PERMIL=000
 STATLEV=ST SEC=3600
 CHRSCHM NO
 MODE RB
```
Configuring the DB2 ECCR JCL

The following sample JCL for the DB2 ECCR is provided in the RUNLIB library member ECCRDB2:

```plaintext
//PX0DB2EC PROC HLQ=<libname>,LOGGER=<zlogger>,
// RUNLIB=<runlib>
/*
*/ PROC OR JOB
/*
*/ CHANGEDATAMOVE = DB2 CHANGE CAPTURE (ECCR) JCL
/*
*/ NOTE: THIS PROCEDURE CAN BE RUN AS AN MVS STARTED TASK OR AS A JOB
/*
*/ REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES
/*
*/ 1. JCL DATA SET NAMES
/*
*/ 2. REPDB2CT MEMBER OF YOUR RUNLIB
/*
*/ 3. REPDB2OP MEMBER OF YOUR RUNLIB
/*
*/ ECCRDB2 EXEC PGM=PX029200,TIME=NOLIMIT
//STEPLIB DD DISP=SHR,DSN=4HLQ..LOADLIB
// DD DISP=SHR,DSN=4HLQ..LOAD
// DD DISP=SHR,DSN=<db2exit>
// DD DISP=SHR,DSN=<db2load>
//EDMPARMS DD DISP=SHR,DSN=4HLQ..&LOGGER..USERLIB
//REPL2CTL DD DISP=SHR,DSN=4RUNLIB(REPDB2CT)
//REPL2OPT DD DISP=SHR,DSN=4RUNLIB(REPDB2OP)
//REPL2TRA DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
/*
*/ REFER TO SAMPJCL MEMBER TRACJCL FOR FURTHER
/*
*/ INFORMATION ON TRACE AND DIAGNOSTIC DD STATEMENTS
/*
*/ EDMTRACE DD SYSOUT=* (COMMON SERVICES TRACE)
/*
*/ DB2 ECCR DIAGNOSTIC DD'S
/*
*/ (REQUIRE REPL2OPT TRACE STATEMENT(S),
/*
*/ WHICH ARE CURRENTLY NOT DOCUMENTED)
/*
*/ REPL2TRU DD SYSOUT=*
/*
*/ (UNFORMATTED TRACE(TRACE TYPE U))
/*
*/ BMCTRACE DD DSN=??,UNIT=SYSDA, (DUMP OF DATA RECORDS CAPTURED
/*
*/ DISP=(NEW,CATLG,CATLG), FROM DB2 LOG)
/*
*/ DCB=(RECFM=VB,
/*
*/ LRECL=32756,BLKSIZE=32760)
```

The following table describes the JCL statements for the DB2 ECCR procedure:

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the PX029200 program.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>Include the PowerExchange load libraries (LOADLIB and LOAD) and the DB2 load library (DSNLOAD). If your DB2 subsystem uses EDITPROC or FIELDPROC exit routines, include the library that contains them as well. All libraries included in this STEPLIB concatenation must be APF-authorized. If any of the libraries are included in your system's LNKLST concatenation, you do not need to include them in the STEPLIB.</td>
</tr>
<tr>
<td>EDMPARMS DD</td>
<td>Specify the name of the PowerExchange USERLIB library that contains the EDMSDIR modules options module associated with the PowerExchange Logger you are using. If you do not include an EDMPARMS DD statement, or if the library you specify does not contain the EDMSDIR options module, the DB2 ECCR searches the STEPLIB concatenation for those options.</td>
</tr>
<tr>
<td>REPL2CTL DD</td>
<td>Specify the REPL2CTL file (REPDB2CT in RUNLIB) associated with the ECCR.</td>
</tr>
</tbody>
</table>
### Managing DB2 Change Data Capture

This section describes how to start and stop the DB2 ECCR. It also describes how to control DB2 ECCR statistics and output.

#### Starting the DB2 ECCR

The DB2 ECCR runs as an MVS started task or as an MVS batch job. For the DB2 ECCR to start successfully, DB2 must be running.

Use this procedure to start the DB2 ECCR for the first time, or to restart after a system shutdown.

To start the DB2 ECCR:

1. Configure the DB2 ECCR options:
   - Edit the REPDB2CT member in the PowerExchange RUNLIB data set as required.
   - Edit the REPDB2OP member in the PowerExchange RUNLIB data set as required.
   - **Important:** The default member that PowerExchange supplies specifies WARM for the start type. The first time you start the DB2 ECCR, temporarily change the start type to COLD to allow the DB2 ECCR to start. After the initial start, warm start the DB2 ECCR.

2. Edit the ECCRDB2 sample JCL in the PowerExchange RUNLIB data set as required.

3. Execute the procedure in a batch job. Alternatively, start it as a started task by using the MVS START command. Generally, the DB2 ECCR is run as a started task because it is a long-running job.

The process described previously details the requirements for starting a single DB2 ECCR in a simple environment.

**Related Topics:**
- “Running Multiple DB2 ECCRs” on page 153
- “DB2 Data-Sharing Considerations” on page 154
- “DB2 ECCR REPL2CTL Statement” on page 158
- “DB2 ECCR REPL2OPT Statements” on page 160
Stopping the DB2 ECCR

When you stop a DB2 ECCR, PowerExchange no longer captures changes in the capture environment. The DB2 ECCR disconnects from the PowerExchange Logger and displays a set of messages.

To stop the DB2 ECCR, issue the QUIESCE or the STOP command. If you use the QUIESCE command, the DB2 ECCR does not stop until the units of work (UOWs) associated with the ECCR complete their processing. For this reason, the QUIESCE command is preferable to the STOP command.

The messages that result from stopping the ECCR include the number and type of changes captured since the last time the data set was opened. The following are example messages:

```
FWXEDM172841I EDM ECCR FWXDB2CC disconnected from EDM Logger FWXL,Log RBA=X'0000000000000000'
FWXEDM1728411 EDM ECCR FWXDB2CC disconnected from EDM Logger FWXL,Log RBA=X'0000000000000000'
FWXEDM1728181 Left XCF group 'FWXL' as member 'FWXDB2CC'
FWXEDM172829I EDM ECCR sent 0 records to Logger FWXL (0 change records)
FWXEDM177012I DB2 ECCR STATUS : LAST READ RBA=00089D5F2D0/0000 OLDEST OPEN URID=NONE
```

Using the QUIESCE Command

To stop the DB2 ECCR, you can use the MVS MODIFY command to issue the QUIESCE command. This method stops the DB2 ECCR after all open UOWs for that ECCR finish processing. The DB2 ECCR sends the change records to the PowerExchange Logger and then stops.

Use the following syntax to issue the QUIESCE command:

```
F job_name,QUIESCE
```

The `job_name` variable is the name of the DB2 ECCR job or of the started task.

The QUIESCE command results in the following message output:

```
FWXEDM177048I CAPTURE PROGRAM ACKNOWLEDGES A QUIESCE COMMAND
FWXEDM177280I CAPTURE PROGRAM OF DB2 REPLICATION ENDING BECAUSE OF CAPTURE QUIESCE COMMAND
FWXEDM177282I BEGIN DB2 CAPTURE TERMINATION
FWXEDM177008I -STOP TRACE(MONITOR) PLAN(CCCDGPK0) LOCATION(DEBB0001) CLASS(1) HAS BEEN EXECUTED
FWXEDM1770001 DSN9022I DSN9022I 'DEBB STOP TRACE SUCCESSFUL FOR TRACE NUMBER(S) 04
FWXEDM1770001 DSN90221I DSN9022I 'DEBB DSNVCM1 '-STOP TRACE' NORMAL COMPLETION
FWXEDM177268I LAST READ DB2-LOG LOCATION=000017066F37
FWXEDM177265I PROCESSING IS COMPLETE
FWXEDM177260I Change Capture counts for DEBB/RADAGK.SOURCE: Insert=0, Update=0, Delete=0
FWXEDM177260I Change Capture counts for DEBB/RADAGK.DGKSRC01: Insert=0, Update=0, Delete=1
FWXEDM177261I DSN9022I DSN9022I disconnected from EDM Logger FWXL, Log RBA=X'00000014AA0540000'
FWXEDM177261I Last XCF group 'FWXL' as member 'FWXDB2CC'
FWXEDM177261I EDM ECCR sent 1 records to Logger FWXL (1 change records)
FWXEDM177012I DB2 ECCR STATUS : LAST READ RBA=000017066F37/0000 OLDEST OPEN URID=NONE
```

Using the MVS STOP Command

To stop the DB2 ECCR immediately, issue the MVS STOP command. If open UOWs are still processing when you issue this command, the DB2 ECCR cannot send the changed records to the PowerExchange Logger until you restart the ECCR. Informatica Corporation recommends that you use the QUIESCE command instead of the STOP command.

The syntax for the STOP command is as follows:

```
STOP!P job_name
```

For the `job_name` variable, enter the name of the DB2 ECCR job or the started task.

The following are examples of messages that result when you run STOP to stop the DB2 ECCR:

```
FWXEDM177046I CAPTURE PROGRAM ACKNOWLEDGES A MVS STOP COMMAND
FWXEDM177279I CAPTURE PROGRAM OF DB2 REPLICATION ENDING BECAUSE OF MVS STOP COMMAND
FWXEDM177282I BEGIN DB2 CAPTURE TERMINATION
FWXEDM177008I -STOP TRACE(MONITOR) PLAN(CCCDGPK0) LOCATION(DEBB0001) CLASS(1) HAS BEEN EXECUTED
FWXEDM1770001 DSN90221I DSN9022I 'DEBB STOP TRACE SUCCESSFUL FOR TRACE NUMBER(S) 04
FWXEDM1770001 DSN90222I DSN9022I 'DEBB DSNVCM1 '-STOP TRACE' NORMAL COMPLETION
FWXEDM177268I LAST READ DB2-LOG LOCATION=000017066F37
```
Controlling DB2 ECCR Processing

You can control the DB2 ECCR either by changing control statements in the REPL2OPT options file and restarting or refreshing, or by issuing MVS MODIFY, START, and STOP commands.

Modifying a Running DB2 ECCR

Use the MVS MODIFY command for a DB2 ECCR to perform the following operations:

- Display statistics about the change-capture process.
- Stop the DB2 ECCR.
- Refresh the DB2 ECCR.
- Display the open UOWs.

The following table briefly describes the DB2 ECCR commands you can use with the MVS MODIFY command to control DB2 ECCR processing:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Displays current ECCR-processing statistics.</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>Stops the DB2 ECCR after all open UOWs for that ECCR finish processing.</td>
</tr>
<tr>
<td>STOP</td>
<td>Terminates the DB2 ECCR immediately.</td>
</tr>
<tr>
<td>REFRESH</td>
<td>Activates the updated options file and validates the new table registration information. Note: The REFRESH keyword ignores any changes that you make to the CA, IFI306OPT, and START statements in the REPL2CTL file. The REFRESH command is equivalent to stopping the DB2 ECCR and then restarting it with the START WARM statement.</td>
</tr>
<tr>
<td>URID</td>
<td>Displays any open UOWs for the DB2 subsystem or data-sharing group to which the DB2 ECCR is connected.</td>
</tr>
<tr>
<td>TERM</td>
<td>Commits an outstanding UOW.</td>
</tr>
</tbody>
</table>

MVS START Command

The START command starts the DB2 ECCR if it has been stopped. Use the following syntax:

```
START|S job_name
```

The variable `job_name` refers to the MVS batch job or started task name that you need to run the DB2 ECCR.

MVS STOP Command

The STOP command terminates the DB2 ECCR immediately. Use the following syntax:

```
STOP|P job_name
```

The variable `job_name` refers to the MVS started task name that you need to run the DB2 ECCR.
Output from the DB2 ECCR

When the DB2 ECCR starts, PowerExchange generates a report that shows the default options that are in effect for the ECCR. After processing ends, the ECCR reports the number of captured changes. If you have applied any zaps or load module replacements to PowerExchange, this report also indicates which ones have been applied. The DB2 ECCR prints the report to the output queue, or to a location that the ECCR startup procedure specifies. The following example shows a sample report.

You can change the frequency of the interval statistics reported in messages PWXEDM177084 and PWXEDM177085 by changing the value of the STAT statement in the REPL2OPT control file.

The following section describes the interval statistics that these messages provide.

Sample DB2 ECCR Report

PWXEDM1728521 Options in effect:
Load Library containing EDMSDIR. . . . . : PWX_PWXL_USERLIB
EDMSDIR assembly date/time . . . . . . . : 20080306 22.53
Product distribution date. . . . . . . . . : 20060831
Product distribution level . . . . . . . . : 2.4.05
Agent Id. . . . . . . . . . . . . . . . . . . . : PWX
Logger Id. . . . . . . . . . . . . . . . . . . : PWXL
SYSGUT class . . . . . . . . . . . . . . . . : */
Action if ECCR error encountered . . . . : Continue
PWXEDM1728181 Joined XCF group 'PWXL' as member 'PWXDB2CC',
PWXEDM1728411 EDM ECCR PWXDB2CC connected to EDM Logger PWXL,
Log RBA=X'0000003A58E40000'
PWXEDM1728081 Change Capture active for DB2 table DBDT/CCD_SRC02
Edition=B4283F1D309C18C3, EDMNAME=DB2.DBDT.CCD_SRC02
PWXEDM1728201 Change Capture initialized for DB2 on V9.1.0 - DSN9(N)
PWXEDM1770081 -START TRACE (MONITOR) PLAN(PWXCPDB2) LOCATION(PWXDB2CC) CLASS(1) HAS BEEN EXECUTED
DSNW1501 -DSN9 MONITOR TRACE STARTED, ASSIGNED TRACE NUMBER 04
DSN90121 -DSN9 DSNW1CM1 "-START TRACE" NORMAL COMPLETION

RELATED TOPICS:
♦ “DB2 ECCR REPL2OPT Statements” on page 160

Statistics Provided by the DB2 ECCR

The DB2 ECCR writes statistics to the EDMMSG data set under the following circumstances:

♦ At the end of each statistical reporting period, as defined by the STAT statement in the REPL2OPT file

By default, the DB2 ECCR writes a table summary statistics report for each interval. If you specify LEV=SQ on the STAT statement, the DB2 ECCR writes a table SQL operation statistics report.

♦ When you issue a DISPLAY command to the DB2 ECCR

If you specify the SQ option, the DB2 ECCR writes a table SQL operation statistics interval report.

The DISPLAY command also writes the PWXEDM177084I message to the MVS hardcopy log and to the JES job log of the DB2 ECCR.

Sample Statistics Report for the DB2 ECCR

The following example shows a table summary statistics report of the DB2 ECCR:

LAST DELAY= 1.52 SEC AVERAGE DELAY= 1.82 SEC
NBR OF ERRORS= 0
DB2 LOG C1'S CI_TOT CI_INTV CI_FSEC 380,719 0 0
EDM MESSAGES MSG_TOT MSG_INTV MSG_FSEC 63 0 0
PWXEDM177085I DETAIL-LEVEL STATISTICS FOLLOW BELOW
The followed example shows a table SQL operation statistics report of the DB2 ECCR:

```
FPWEDM1770841 STATISTICS OF CAPTURE PGM FWDB2CC AT=2008-11-30 11.20.39
-----------------------------------------------------------------------
DB2-LOG LOCATION=001520A56E6/0000 DB2-LOG TIMESTAMP=2008-11-30 11.18.55
LAST DELAY= 2.20 SEC  AVERAGE DELAY= 2.20 SEC
NBR OF ERRORS= 0
DB2 LOG CI’S CI_TOT CI_INTV CI_PSEC
380,719 0 0
EDM MESSAGES MSG_TOT MSG_INTV MSG_PSEC
63 0 0
-----------------------------------------------------------------------
```

The following table describes each element of the interval statistics reports:

<table>
<thead>
<tr>
<th>Report Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2-LOG LOCATION</td>
<td>Displays the RBA of the current location of ECCR processing in the DB2 log.</td>
</tr>
<tr>
<td>DB2-LOG TIMESTAMP</td>
<td>Displays the time stamp of the last DB2 log record that the ECCR read. This time stamp reflects the date and time that the record was written to the DB2 log.</td>
</tr>
<tr>
<td>LAST DELAY</td>
<td>Displays the difference between the time when the ECCR read the last DB2 log record and the time when the record is written to the DB2 log.</td>
</tr>
<tr>
<td>AVERAGE DELAY</td>
<td>Displays the average delays for the statistical reporting period. The delay is defined as the difference between when ECCR read a DB2 log record and the time that record was written to the DB2 log.</td>
</tr>
<tr>
<td>NBR OF ERRORS</td>
<td>Displays the total number of errors since the DB2 ECCR was started.</td>
</tr>
<tr>
<td>CI_TOT</td>
<td>Displays the total estimated number of DB2 log control intervals read by the ECCR since the ECCR started.</td>
</tr>
<tr>
<td>CI_INTV</td>
<td>Displays the total estimated number of DB2 log control intervals read by the ECCR for the statistical reporting period.</td>
</tr>
<tr>
<td>CI_PSEC</td>
<td>Displays the estimated average number of DB2 log control intervals read per second by the ECCR for the statistical reporting period.</td>
</tr>
<tr>
<td>MSG_TOT</td>
<td>In the EDM MESSAGES section, displays the total number of changed records read by the DB2 ECCR for all tables since the ECCR started. This statistic also appears for each table in the table summary statistics report.</td>
</tr>
<tr>
<td>MSG_INTV</td>
<td>In the EDM MESSAGES section, displays the total number of changed records read by the DB2 ECCR for all tables in the statistical reporting period. This statistic also appears for each table in the table summary statistics report.</td>
</tr>
</tbody>
</table>
### Report Element Description

<table>
<thead>
<tr>
<th>Report Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSG_PSEC</td>
<td>In the EDM MESSAGES section, displays the average number of changed records read per second by the DB2 ECCR for the statistical reporting period. This statistic also appears for each table in the table summary statistics report.</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>In the table summary statistics report, displays the name of the table for which the MSG_TOT, MSG_INTV, and MSG_PSEC statistics are reported.</td>
</tr>
<tr>
<td>TABLE</td>
<td>In the table SQL operation statistics report, displays the name of the table for which the INSERTS, UPDATES, and DELETES statistics are reported.</td>
</tr>
<tr>
<td>INSERTS</td>
<td>In the table SQL operation statistics report, displays the total number of insert operations to the table since the ECCR started.</td>
</tr>
<tr>
<td>UPLOADS</td>
<td>In the table SQL operation statistics report, displays the total number of update operations to the table since the ECCR started.</td>
</tr>
<tr>
<td>DELETES</td>
<td>In the table SQL operation statistics report, displays the total number of delete operations to the table since the ECCR started.</td>
</tr>
</tbody>
</table>

### Recovering the DB2 ECCR

You must recover the DB2 ECCR:

- When the DB2 ECCR fails
- When the PowerExchange Logger stops or fails while attached to the DB2 ECCR

To recover the DB2 ECCR:

1. Determine the cause of the DB2 ECCR failure.
   - The DB2 ECCR allows you to specify a maximum number of errors to tolerate before the ECCR terminates. Use the EC PERMIL statement in the REPL2OPT file.
2. If the DB2 ECCR failed because the PowerExchange Logger stopped, restart the PowerExchange Logger.
   - If the PowerExchange Logger stops or ABENDs while attached to the DB2 ECCR, the DB2 ECCR also ABENDs when it receives the first change record following the PowerExchange Logger failure.
3. Restart the DB2 ECCR from the point at which it ABENDed.
   - Use the STARTUP WARM statement in the REPB2OP member. Be sure to use the same REPL2CTL file that you used prior to the abend.

When you restart the DB2 ECCR or the PowerExchange Logger after a failure, the PowerExchange Logger determines the point at which to begin capturing changes again for the corresponding CA name.

**RELATED TOPICS:**

- “DB2 ECCR REPL2OPT Statements” on page 160

### Altering DB2 System Tables for DATA CAPTURE CHANGES

To support DB2 Version 8 and later, the DB2 ECCR requires that certain DB2 system tables specify DATA CAPTURE changes. Use the following procedure to alter these tables.
To alter DB2 system tables for DATA CAPTURE CHANGES:

1. Perform a orderly shutdown of the DB2 ECCR by using the MVS command:
   
   ```
   MODIFY db2_eccr_name,QUIESCE
   ```
   
   Record the LAST READ RBA in the PWXEDM177012I message in the EDMMSG data set in DB2 ECCR output listing. You need this RBA value to restart the DB2 ECCR.
   
   For example:
   
   ```
   PWXEDM177012I DB2 ECCR STATUS : LAST READ RBA=000C9041C372/0000
   ```
   
   The LAST READ RBA value that is used to special start the DB2 ECCR in this example is 000C9041C372.

2. Alter any of the following catalog tables that are currently have DATA CAPTURE NONE to DATA CAPTURE CHANGES:
   
   - SYSTABLES
   - SYSCOLUMNS
   - SYSTABLESPACE
   - SYSFIELDS
   - SYSCOPY

   This can be done using the following SQL command:
   
   ```
   ALTER TABLE SYSIBM.<catalog table name> DATA CAPTURE CHANGES
   ```

3. Special start the DB2 ECCR using STARTLOC= parameter in the REPDB2OP member of RUNLIB library, pointed to by the REPL2OPT DD statement in the DB2 ECCR JCL.

   The value specified in the STARTLOC= parameter should be the LAST READ RBA value from the PWXEDM177012I message in the EDMMSG data set in the DB2 ECCR output listing from the quiesced DB2 ECCR.

   For example, using the example message from step 1, you would code the following STARTLOC parameter:
   
   ```
   START STARTLOC=000C9041C372 USEDIR,USESTAT
   ```

4. Verify that no PWXEDM177540W or PWXEDM177541W messages are issued when the DB2 ECCR special start completes.

5. Modify the REPDB2OP member of the RUNLIB library to specify START WARM so the next restart of the DB2 ECCR starts correctly.

**RELATED TOPICS:**

- “DB2 ECCR REPL2OPT Statements” on page 160

---

**DB2 ECCR Capture Directory Table Upgrades**

If you use the DB2 ECCR that PowerExchange 8.6 or later supplies, you must upgrade the ECCR capture directory tables to run DB2 for z/OS Version 8 in new-function mode or to upgrade to DB2 for z/OS Version 9.1. More specifically, you must upgrade the capture directory tables in the following circumstances:

- You migrate a source DB2 Version 8 subsystem to new-function mode.
- You use DB2 Version 8 in new-function mode and plan to upgrade to DB2 Version 9.1
- You use DB2 Version 8 in new-function mode and plan to upgrade PowerExchange to Version 8.6 or later.

If you previously used an earlier PowerExchange version and upgraded the capture directory tables when migrating to DB2 Version 8 new-function mode, you must upgrade the capture directory tables again as part of the upgrade to PowerExchange 8.6 or later. This step is necessary because PowerExchange 8.6 introduced DB2 ECCR enhancements for DB2 Version 8 new-function mode. After you upgrade the capture directory tables once for PowerExchange 8.6 or later, you do not need to update the tables again to support DB2 Version 9.1.
SAMPLIB Members for Upgrading the Tables

PowerExchange provides the following members in the SAMPLIB library to upgrade the capture directory tables:

**EXPNDCAP**

Alters the capture directory tables to add columns and change some existing columns. This member contains the following steps:

- Step 1 provides DDL that alters the capture directory tables and re-creates the TCAPWORK table.
- Step 2 identifies the capture directory tables that require reorganization.
- Step 3 provides DDL for reinstating DATA CAPTURE CHANGES for the altered capture directory tables.

**FIXTCAPC**

Updates the TCAPCOLUMNS capture directory table to correct some existing data that might have been set incorrectly by previous levels of the DB2 ECCR.

**FIXTCAPP**

Updates the TCAPTABLEPART capture directory table to populate new columns that are added to that table by the EXPNDCAP DDL, and to provide full support for DB2 Version 9.1 functionality such as clone tables and reordered row format.

**FIXTCAPS**

Updates the TCAPTABLESPACE capture directory table to populate the new columns that are added by the EXPNDCAP DDL. Also, provides full support for DB2 Version 9.1 functionality such as clone tables and reordered row format.

**FIXTCAPT**

Updates the TCAPTABLES capture directory table to populate new columns that are added by the EXPNDCAP DDL. Also, provides full support for DB2 Version 9.1 functionality such as clone tables and reordered row format.

Upgrading the DB2 ECCR Capture Directory Tables

Use the following procedure to upgrade the DB2 ECCR capture directory tables when migrating to DB2 Version 8 new-function mode or DB2 Version 9.1.

Important: While performing this procedure, ensure that no schema changes are made to any table that is registered for change data capture.

To upgrade the DB2 ECCR capture directory tables:

1. If DB2 ECCR is running, use the QUIESCE command to stop it.
2. Customize the DDL in the sample EXPNDCAP member. For instructions, see the comments in the member.
3. Run the customized DDL that is in step 1 of the EXPNDCAP member.
   
   Note: To run this DDL, the DB2 subsystem must be running DB2 Version 8 new-function mode or later.
4. Reorganize the capture directory tables. For instructions, see the comments in step 2 of the EXPNDCAP member.
5. Run the customized DDL in the step 3 of the EXPNDCAP member.
6. Customize and run the DDL in the FIXTCAPC, FIXTCAPP, FIXTCAPS, and FIXTCAPT members. For instructions, see the comments in these members.
   
   If any SELECT SQL statement in these members returns rows, uncomment and run the UPDATE SQL statement that is provided in the same member.
7. If necessary, update the GENBULK member in the RUNLIB library to change the DB28NFM variable from 0 to 1. A value of 1 indicates that you have DB2 Version 8 new-function mode or later.

8. Warm start the DB2 ECCR to resume change data capture with the upgraded catalog capture directory tables.

DB2 Catalog Tables

By default, DB2 sends the DB2 ECCR all log records. The DB2 ECCR must inspect the data for log records that pertains to tables registered for capture. This activity by the DB2 ECCR might cause high levels of CPU usage and I/O activity. If many or all of the tables in the DB2 subsystem specify DATA CAPTURE CHANGES, you cannot substantially reduce the amount of log data DB2 sends to the DB2 ECCR.

If a small proportion of the tables in the DB2 subsystem specify DATA CAPTURE CHANGES, you can use the IFI306OPT statement in the REPL2OPT file to reduce the amount of log data sent to the DB2 ECCR by DB2. Retrieving less DB2 log data decreases CPU and I/O usage of the DB2 ECCR. You must balance these savings against the operational restrictions that using IFI306OPT imposes.

Warning: Informatica recommends that you omit the IFI306OPT statement for the DB2 ECCR. Use of IFI306OPT imposes restrictions that, if ignored, can result in change data loss.

To use the IFI306OPT statement, you must enable DATA CAPTURE CHANGES for the following DB2 catalog tables:

- SYSTABLES
- SYSCOLUMNS
- SYSFIELDS
- SYSTABLESPACE
- SYSCOPY

With DB2 Version 8 and later, you must enable DATA CAPTURE CHANGES for these DB2 catalog tables whether you use IFI306OPT or not. The DB2 ECCR does not start up unless DATA CAPTURE CHANGES is specified for these catalog tables.

Optimizing Access to the DB2 Catalog Tables

Use the following procedure to implement the IFI306OPT statement in the DB2 ECCR.

To optimize access to the DB2 catalog tables:

1. Quiesce the DB2 ECCR using the MVS command:

   ```
   MODIFY db2_eCCR_name,QUIESCE
   ```

   Record the LAST READ RBA value in the PWXEDM177012I message in the EDMMSG data set in DB2 ECCR output listing. You need this RBA value to restart the DB2 ECCR.

   For example:

   ```
   PWXEDM177012I DB2 ECCR STATUS : LAST READ RBA=000C9041C372/0000
   ```

   The LAST READ RBA value that is used to special start the DB2 ECCR in this example is 000C9041C372.

2. Alter any of the following catalog tables that are currently have DATA CAPTURE NONE to DATA CAPTURE CHANGES:

   - SYSTABLES
   - SYSCOLUMNS
   - SYSTABLESPACE
3. Code statement IFI306OPT in the REPDB2OP member of hlq.RUNLIB, pointed to by the REPL2OPT DD statement in the DB2 ECCR JCL.

   **Note:** The IFI306OPT statement cannot be activated by a REFRESH of the ECCR.

4. Special start the DB2 ECCR using STARTLOC= parameter in the REPDB2OP member of hlq.RUNLIB, pointed to by the REPL2OPT DD statement in the DB2 ECCR JCL.

   The value specified in the STARTLOC= parameter should be the RBA value from the PWXEDM177012I message in the EDMMSG data set in the DB2 ECCR output listing from the quiesced DB2 ECCR.

   For example, using the example message from step 1, you would code the following STARTLOC parameter:

   ```sql
   START STARTLOC=000C9041C372 USEDIR,USESTAT
   ```

5. Verify that no PWXEDM177540W or PWXEDM177541W messages are issued when the DB2 ECCR special start completes.

6. Modify the REPDB2OP member of hlq.RUNLIB, pointed to by the REPL2OPT DD statement in the DB2 ECCR, to specify START WARM so the next restart of the DB2 ECCR starts correctly.

   If running DB2 Version 7 or less with IFI306OPT specified in the REPL2OPT statements, the following message appears if the relevant DB2 catalog tables have not been set up with DATA CAPTURE CHANGES:

   ```sql
   PWXEDM177542W Request for optimized IFI306 processing (IFI306OPT) ignored
   ```

   In this case the ECCR continues processing but uses the standard data capture processing.

**RELATED TOPICS:**

- “DB2 ECCR REPL2OPT Statements” on page 160
- “Restrictions Imposed by Optimized Access (IFI306OPT)” on page 175

**Restrictions Imposed by Optimized Access (IFI306OPT)**

Carefully consider the restrictions and potential for change data loss before using the IFI306OPT statement. The DB2 ECCR may see useful CPU and I/O improvement if the DB2 subsystem has only a small percentage of tables recording change data in the DB2 log. You do not see any substantial CPU and I/O reduction if many tables in the DB2 subsystem specify DATA CAPTURE CHANGES. You must balanced this benefit against more complex operational procedures.

When you specify the IFI306OPT statement, the DB2 ECCR does not detect the following operations:

- Table renames
- Column renames
- The DB2 QUIESCE utility for tables registered for capture

**Manually Renaming a Table or Columns in a Table**

Normally, the DB2 ECCR detects and handles table and column rename operations. You must handle these operations manually when using the IFI306OPT statement.
To manually rename a table or column in a table:
1. Make the table read-only.
2. Ensure the DB2 ECCR has captured all changes up until the point at which changes were stopped.
3. Rename the table or the column in the table.
4. Delete or inactivate the capture registration for the table.
5. If you renamed the table but still want to capture changes for the renamed table, create and activate a new capture registration for the table. Otherwise, skip this step.
6. Issue the DB2 ECCR REFRESH command.
7. Allow changes to the table.

Replacing a Table with Another Table with the Same Name

Use the following procedure to replace a table that is currently being captured with a table that is not registered for capture. The new table uses the same name as the old table.

To replace a table with another table with the same name:
1. Make both tables read-only.
2. Ensure the DB2 ECCR has captured all changes up until the point at which changes were stopped.
3. Rename the table being captured to a different name.
4. Delete or inactivate the capture registration for the table.
5. Issue the REFRESH command to drop the table from the DB2 ECCR.
6. Rename the new table to the name of the table previously registered for capture.
7. Create and activate a capture registration for the newly renamed table.
8. Issue the REFRESH command to add the newly renamed table to the DB2 ECCR.
9. Allow changes to both tables.

Handling the DB2 QUIESCE Utility Manually

Normally the DB2 ECCR detects a table space quiesce and creates an event marker containing restart information in the PowerExchange Logger. When you specify the IFI306OPT statement, the DB2 ECCR does not see the DB2 log records for the QUIESCE utility and does not create an event marker.

To manually generate an event marker when you use the QUIESCE utility for a table registered for capture, use either the PowerExchange EDMXLUTL Event Marker utility or the DTLUAPPL utility.

Migrating to a DB2 Data Sharing Environment

Keep the following considerations in mind when migrating from a DB2 non-data sharing environment to a DB2 data sharing environment:

- The DB2 ECCR connects to the name specified in the CN parameter (or the RN parameter if CN is not specified) of the DB2 control record in the REPL2OPT control file.
This single execution of the DB2 ECCR:
- Obtains the log records of all DB2 subsystems that are members of the DB2 data-sharing group from that 
  DB2 subsystem.

If the DB2 subsystem to which the DB2 ECCR normally attaches is unavailable, the DB2 ECCR does not run 
and does not capture table changes from the DB2 logs. The change data are not be lost as long as the DB2 
logs are still available, but access to the data might be delayed.
- Processes all updates performed by DB2 subsystems that are members of the DB2 data-sharing group

If you create a single data sharing group using existing DB2 subsystems and want to continue using their 
existing capture registrations, you must continue to run multiple DB2 ECCRs: one for each SSID for which you 
captured changes with the appropriate RN parameter to match that DB2 subsystem id.

After successfully migrating to a data sharing environment, you can consider minimizing the number of ECCRs 
by combining existing ones from the members of the data sharing group. This requires reregistration of all of 
the DB2 tables of the combined ECCRs under a common SSID (generally the group attachment name). Before 
attempting to combine ECCRs, ensure that you understand the ramifications as you may need to change your 
extraction mappings and processes in addition to re-registering your tables.

**Migrating from a DB2 Data-Sharing Environment**

If a migration from a data-sharing environment is required during change propagation, you must perform the 
following procedure so that the DB2 ECCR can cope with the migration.

**Note:** To migrate from the DB2 data-sharing mode, ideally you should wait until the DB2 ECCR has processed all 
of the updates created in data sharing mode. If you cannot wait, updates may be lost, causing processing 
inconsistencies and requiring that target tables be materialized again.

After you have migrated your environment from a DB2 data-sharing environment to a DB2 non-data-sharing 
environment, DB2 does not support the reading of any log records that were written in the data sharing mode.

To migrate from a DB2 data-sharing environment:
1. Verify (before you complete the migration out of a DB2 data-sharing environment) that the DB2 ECCR 
successfully captured all of the log records for any source table updates that were performed in data-sharing 
mode, as follows:
2. Set the table spaces that contain the source tables to read-only (RO) state by issuing the following 
   commands:
   - For a database:
     ```sql
     START DATABASE(database_name) ACCESS(RO)
     ```
   - For a table space:
     ```sql
     START DATABASE (database_name) SPACENM(table_space_name) ACCESS(RO)
     ```
3. To ensure that the DB2 ECCR processed all log records that were written before setting the table spaces to 
   RO state, issue the following command:
   ```sql
   MODIFY job_name,DISPLAY
   ```
   The result of this command includes the DB2 log timestamp that indicates when the last read log record was 
   created. Be sure that this timestamp is later than the recorded time at which the last table space (which 
   contains source tables) entered the read-only (RO) state.
4. Stop the DB2 ECCR by issuing the following command:
   ```sql
   STOP job_name
   ```
5. Complete the migration from the DB2 data sharing environment.
6. Start the DB2 subsystem in non-sharing mode and start the DB2 ECCR to propagate all of the updates that 
   are performed in non-sharing mode.
You must do one of the following:

- Cold start the DB2 ECCR before allowing updates of the source tables (that is, before setting the table spaces containing the source tables in RW mode). After completing the cold-start, allow the update of the source tables. PowerExchange CDC propagates these updates as usual.

- Special start the DB2 ECCR before completing any data definition language (DDL) operations on the source tables. You can perform the special start before or after you allow updates to the source tables.

For the special start, proceed as follows to determine the STARTLOC keyword value of the START control record in the file REPL2OPT:

- Run the DB2 DSNJU004 utility.
- From the DSNJU004 print output, obtain the value of MIN RBA FOR TORBA.
- Use the value of MIN RBA FOR TORBA as the keyword value for STARTLOC.
- If you specified the group attachment name in the CN parameter (or allowed it to default from the RN parameter) of the REPL2OPT control statement, you should now code a DB2 subsystem ID.

If you now run multiple ECCRs and had registered all resources under the group attachment name, you can continue to use the same repository and the same RN value as before. For each table registered that is not in the DB2 catalog, you receive the following message:

```
PWXEM17737W TABLE=creator.tablename DOES NOT EXIST IN DB2 CATALOG
```

This message is a warning message. It does not affect change capture for tables that are defined in the DB2 catalog for the DB2 subsystem under which the ECCR is running.

## Stopping Change Data Capture for DB2

The following table lists the methods for disabling the change data capture process, based on the level at which you want to stop capturing changes. Additionally, the table describes ways to stop the change data capture process:

<table>
<thead>
<tr>
<th>To Stop Capturing Changes for</th>
<th>Use This Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 tables</td>
<td>Alter the structure of your DB2 table to DATA CAPTURE NONE.</td>
</tr>
<tr>
<td>DB2 environment</td>
<td>Stop the ECCR.</td>
</tr>
<tr>
<td>Registered DB2 tables</td>
<td>Deactivate or delete the capture registration, and refresh the DB2 ECCR.</td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**

- “Stopping the DB2 ECCR” on page 167
- “Altering the DB2 Table Structure” on page 178
- “Deactivating or Deleting Registrations” on page 179

## Altering the DB2 Table Structure

Use the following DDL statement to change the structure of your DB2 table so that the table no longer allows data capture:

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

As a result, the DB2 ECCR can no longer capture changes associated with the specified table.

For more information about this command, see your IBM DB2 documentation.
Managing DB2 Schema Changes

To capture changes without interruption to DB2 tables registered for capture, you must manage changes to DB2 tables and table spaces.

Schema Verification

When the DB2 ECCR captures the first change record for a DB2 table, the ECCR verifies that the DB2 table schema in the DB2 catalog matches the corresponding schema registration in the PowerExchange repository.

The schema verification routine does not actually access the DB2 catalog. Instead, the routine uses the internal PowerExchange tables that were created from the DB2 catalog when you started the DB2 ECCR.

- If the DB2 table schema matches the activated schema registration, capture processing continues.
- If the DB2 table schema does not match the activated schema registration, the verification routine displays a report and the DB2 ECCR ABENDs.

You can request that the DB2 ECCR also run this schema verification routine at startup by using the CHKSCEM statement.

The following example shows a sample report and the messages that are displayed when schema verification fails. In this example, schema verification fails because the schema in the schema registration contains a column that is not defined in the DB2 catalog (suggesting that a column has been removed since the table was registered). The fields in the Schema Verification Report table describe the fields in the report.

Sample Schema Verification Report and ABEND Messages

The following example output shows a schema verification report with ABEND messages:

```
FWXEDM1775S21 The DB2 schema for table 'DTLUSR.DEFINFO' does not match the active profile schema. DB2 log time = 2004-06-21-17.10.13.296528.
-------------------- Db2 Catalog -------------------------------- Edm registration --------------------
Create timestamp = 2004-06-21-17.06.11.458379 | Create timestamp =
Alter timestamp = 2004-06-21-17.06.11.458379 | Alter timestamp =
# NL Column Name Datatype Len Pr Sc N | # NL Column Name Datatype Len Pr Sc N
1 9 EMPLOY_ID Char 6 0 0 N | 1 9 EMPLOY_ID Char 6 0 0 N
2 12 DEPENDENT_ID Char 3 0 0 N | 2 12 DEPENDENT_ID Char 3 0 0 N
| 3 14 DEPENDENT_NAME Char 20 0 0 Y
FWXEDM1775S11E Schema verification failed for table 'DTLUSR.DEFINFO'.
FWXEDM172807E ABEND issued by schema verification, Abend code=3680, Reason=10040001.
```
Fields in the Schema Verification Report

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create timestamp</td>
<td>Date and time when the DB2 table schema was created and registered.</td>
</tr>
<tr>
<td>Alter timestamp</td>
<td>Date and time when the DB2 table schema and schema registration were last altered.</td>
</tr>
<tr>
<td>#</td>
<td>Sequential number of the column in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>NL</td>
<td>Length of the column name in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>Column Name</td>
<td>Name of the column in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>Datatype</td>
<td>Datatype of the column in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>Len</td>
<td>Length of the column in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>Pr</td>
<td>Precision (total number of digits) of the column in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>Sc</td>
<td>Scale (number of digits to the right of the decimal point) of the column in the DB2 table and in the associated schema registration.</td>
</tr>
<tr>
<td>N</td>
<td>Whether the column in the DB2 table and in the associated schema registration can have null values.</td>
</tr>
</tbody>
</table>

Changing the Schema of DB2 Source Tables

Use the following procedure to change the schema for DB2 source tables registered for capture.

To change the schema of DB2 source tables:
1. Disable the ability to make changes to the source table.
2. Ensure all changes logged in DB2 for that table have been captured by the DB2 ECCR.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then, shut down PowerExchange Condense.
4. Extract all captured changes.
5. Make the schema change to the DB2 table. If necessary, reorganize the table space containing the table.
6. Delete the capture registration and extraction map.
7. Create a new capture registration using the new schema.
8. Refresh the DB2 ECCR using the REFRESH command to get the registration changes.
9. Allow changes to be made to the appropriate table.
10. Restart any extraction processes and, if applicable, PowerExchange Condense.

Recovering from Unplanned Schema Changes to DB2 Source Tables

To make schema changes for DB2 tables registered for capture, follow the procedure to make schema changes. If you do not follow this procedure when making schema changes, the DB2 ECCR abends when it reads the first
change record for the table after the schema change is made. The DB2 ECCR writes the following messages in the EDMMSG data set:

PWXEDM17751IE Schema verification failed for table 'owner.table_name'
PWXEDM172807E ABEND issued by schema verification, Abend code=3680, Reason=10040001.

To recover from an unplanned schema change to a DB2 source table:

1. If you use PowerExchange Condense, ensure that all of the captured changes have been condensed. After this has been done, shut down PowerExchange Condense.
2. Extract all captured changes.
3. Delete the capture registration and extraction map.
4. Create a new capture registration using the new schema.
5. Warm start the DB2 ECCR.
6. Restart any extraction processes and, if applicable, PowerExchange Condense.

**RELATED TOPICS:**

- “Changing the Schema of DB2 Source Tables” on page 180

### Altering Columns in DB2 Source Tables

You can make the following types of alterations to columns:

- Increase the length of a VARCHAR or VARGRAPHIC column.
- Alter the datatypd of a column.

**Important:** DB2 requires that you disable DATA CAPTURE CHANGES to alter the datatypd of a column. To enable DATA CAPTURE CHANGES after you alter the datatypd, you must reorganize the table space containing the table. Some releases and maintenance levels of DB2 also require the table space or partitions to be reorganized if you increase the length of variable columns. During this process, do not allow any data changes to the table because the DB2 ECCR does not see them. If changes occur while DATA CAPTURE CHANGES is disabled, you must rematerialize any target tables that use the source table.

- Rename a column

If you make these types of alterations to a column registered for capture with the DB2 ECCR, you must follow the procedure for schema changes.

If a column is not registered for capture with the DB2 ECCR, you do not need to change to the capture registration. When you create a capture registration in the PowerExchange Navigator, columns can be registered for capture in one of the following ways:

- Explicitly selected
- **Select All Columns** option
- **Select all and notify changes** option

**RELATED TOPICS:**

- “Changing the Schema of DB2 Source Tables” on page 180

### Changing the Qualifiers of DB2 Table Spaces

Usually, the DB2 ECCR continues to capture changes without interruption if you change the qualifier for a table space containing tables registered for capture.
However, you must take action to enable the DB2 ECCR to capture changes if the following conditions exist:

- The table space contains multiple tables.
- You have altered at least two tables containing a minimum of one variable-length column to add fixed-length columns.
- The altered tables are not registered for capture.

In this situation, take the following actions:

- Register one of the altered tables for capture, and refresh or warm start the DB2 ECCR.
- Change the qualifier of the table space that contains the tables by using ALTER TABLESPACE with the USING VCAT or USING STOGROUP clauses.
- Register one of the other altered tables for capture, and refresh or warm start the DB2 ECCR.

To allow the DB2 ECCR to capture changes for the second table, you must reorganize the table space before you make any changes to that table. Otherwise, the DB2 ECCR fails because it cannot process DB2 log records for the second table.

**Note:** The DB2 ECCR can capture changes for the altered tables if you change the qualifier for the table space containing the tables after you register both altered tables for capture and refresh or warm start the DB2 ECCR.
CHAPTER 12

IDMS Log-Based Change Data Capture

This chapter includes the following topics:

- Introduction to IDMS Log-Based Change Data Capture, 183
- Configuring IDMS Log Catalog Procedures, 185
- Configuring the IDMS Log-Based ECCR, 186
- Managing IDMS Log-Based CDC, 191
- Managing IDMS Schema Changes, 195

Introduction to IDMS Log-Based Change Data Capture

Change data capture is the process of capturing changes made to the source database and recording those changes in the PowerExchange Logger.

PowerExchange IDMS Log-Based CDC Components

The following figure depicts the PowerExchange IDMS log-based CDC architecture:
The components through which the data flows appear as shaded, rectangular shapes with numerical labels. The components that control the flow of data appear as elliptical shapes with alphabetic labels.

The following table describes the PowerExchange IDMS log-based CDC components. These components handle the change data as it progresses through the capture process.

**Note:** The user application and the source and target databases are not PowerExchange CDC components.

<table>
<thead>
<tr>
<th>Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Updates</td>
<td>Any software that updates the IDMS source database on an ongoing basis.</td>
</tr>
<tr>
<td>IDMS</td>
<td>The IDMS database where the source data resides. PowerExchange Change Capture can capture changes from more than one source database or data file.</td>
</tr>
<tr>
<td>IDMS Logs</td>
<td>PowerExchange IDMS log-based Data Capture uses data changes stored in the IDMS logs.</td>
</tr>
<tr>
<td>PWX LOGCAT</td>
<td>PowerExchange uses a catalog of IDMS logs. The PowerExchange Log Catalog is built and maintained using PowerExchange utilities to identify data available for capture.</td>
</tr>
<tr>
<td>ECCR</td>
<td>The IDMS ECCR is run as a batch job or started task to select data required for capture from the logs identified in the PowerExchange Log Catalog.</td>
</tr>
<tr>
<td>LOGGER</td>
<td>The PowerExchange Logger records the change data captured by the ECCRs in its log data set. Change consumers extract the changes from the PowerExchange Logger using the PowerExchange Listener.</td>
</tr>
<tr>
<td>CONDENSE</td>
<td>PowerExchange Condense extracts changes from the PowerExchange Logger, condenses them, and stores them in condense files.</td>
</tr>
<tr>
<td>AGENT</td>
<td>The PowerExchange Agent controls mainframe service routines and programs for data propagation in PowerExchange. The PowerExchange Agent obtains data from repositories, manages authorization, and facilitates communication between components.</td>
</tr>
</tbody>
</table>

To develop a data capture environment, you must perform the following tasks:

- Register a data source.
- Create the Capture catalog.
- Populate the Capture catalog.
- Configure and start the ECCR.
- Register a restart token.
- Enable data access.

The information here is enough for a basic working installation.

The following components must be configured for IDMS log-based change data capture:

- **PowerExchange Log Catalog.** This catalog contains information about all of the IDMS logs from which change data is captured.
- **SR2/SR3 journal record identifier.**
- **ECCR.** This routine captures changes from the IDMS logs and makes the data available to the PowerExchange Logger.

When you start the ECCR, PowerExchange begins capturing changes as the logs are included in the PowerExchange Log Catalog. PowerExchange sends the change data to the PowerExchange Logger. After the data has been sent, PowerExchange Condense, if used, can pull the changes from the PowerExchange Logger, or the data can be accessed directly from the PowerExchange Logger if you use the PowerExchange real-time option.
Warning: Multiple schemas can be registered within a single LOGSID. However, schemas, which include objects of the same name, cannot be differentiated. If you copy schemas under the same names, such as in system test and QA environments, configure the copies for their own environments. A separate PowerExchange Listener, PowerExchange Logger, and ECCR is required for each like-named schema.

**Related Topics:**
- “CDC Components Configuration and Management” on page 11
- “PowerExchange Agent” on page 20
- “PowerExchange Condense” on page 77
- “PowerExchange IDMS Log-Based CDC Components” on page 183
- “Configuring the IDMS Log-Based ECCR” on page 186
- “Creating the Log Catalog” on page 188
- “Providing SR2 and SR3 Information to the ECCR” on page 190

**Relationships with Other PowerExchange Components**

The IDMS log-based ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- The IDMS log-based ECCR must log all changes to a single PowerExchange Logger running on the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must all run on the same MVS system as the IDMS log-based ECCR.
- Operational issues in the PowerExchange Logger can cause the IDMS log-based ECCR to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the IDMS log-based ECCR continues the capture and recording of change data without any loss of data.

You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

**Related Topics:**
- “Monitoring the PowerExchange Logger for MVS” on page 51

**Configuring IDMS Log Catalog Procedures**

Care must be taken with the order in which the logs are added to the catalog. Operational procedures for the running of DTLULCAT and DTLULLOGC must be developed to ensure that logs are added in the correct sequence.

The preferred method of operation is to include DTLULCAT and DTLULLOGC in an archive log job. Run this job together with DTLULCAU JCL and submit it by a WTOEXIT, which intercepts a message written to the operator.

Include the following steps in the job:

1. Offload the active journal to an archive log.
2. Write the archive log to a file by using a utility program like IEBGENER, which can be kept as a unique data set name (a GDG is a good data set structure). This copy of the log must be retained until all changes have
been captured by PowerExchange. Add this file to the PowerExchange Log Catalog by running the following jobs:

- Run DTLULCAT to generate input statements for DTLULOGC.
- Run DTLULOGC to update the PowerExchange Log Catalog.

When Central Versions are varied offline to run in Local Mode, ensure Local Mode logs are added before any new Central Version logs. If a database, previously varied offline, is subsequently varied back online and the Local Mode log is not added immediately, if a later log is added to the catalog and a subsequent attempt made to add the Local Mode log, this will fail.

The rules for checking such log additions is:

- A local mode journal must not be added to the catalog if the last available timestamp within the journal is greater than the timestamp of the previously added CV mode journal.

- If logs are added in the incorrect sequence expect to see messages similar to the following:

```
51007 162240 MVS 1 PWX-19862 IDMS CATLG FILE: Add Entry Failure - Timestamp not greater than previous for key XYLOGSID0000000000000001AIDL15CDBAIIDMSE150DTLUSR.IDMS.D1SP0.OFF.J4
```

## Configuring the IDMS Log-Based ECCR

Perform the following procedures to configure the IDMS ECCR.

Before beginning these procedures, complete the following tasks:

- Activate the PowerExchange Agent, Listener, and Logger.
- Register and activate at least one source and one target as described in the previous sections.
- Create and populate the Log Catalog.

**RELATED TOPICS:**

- “CDC Components Configuration and Management” on page 11

## Starting the IDMS Log-Based ECCR

The IDMS log-based ECCR runs as a MVS started task or batch job. When running as a started task or as a long-running batch job, the IDMS log-based ECCR checks the access module on a regular basis to detect if new logs have been added to the log catalog for capture processing. As logs are added, the ECCR captures the data and directs it to the PowerExchange Logger.

To start the IDMS log-based ECCR for the first time, or to restart it after a system shutdown, complete the following steps:

1. Configure IDMS ECCR options. Edit the IDMS options ECCRIDLP member of RUNLIB as required.

   The following is a sample ECCRIDLP file:

   ```
   LOGSID=XYLOGSID
   NO_DATA_WAIT=60
   NO_DATA_WAIT2=999
   ```
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSID</td>
<td>Name of the LOGSID specified in the DBMOVER configuration file.</td>
<td></td>
</tr>
</tbody>
</table>
| NO_DATA_WAIT | Number of seconds for the Collector to wait before doing another read after the Collector received an end-of-log condition. On receiving another end-of-log condition on the first read following the previous end-of-log, the Collector will wait NO_DATA_WAIT2 before retrying another read. | - 0. Shut down the change capture routine as soon as there are no more logs to process.  
- n. Where n is the time in minutes to wait for more logs or changes before shutting down. |
| NO_DATA_WAIT2| Number of seconds for the Collector to wait before doing another read after the Collector received an end-of-log condition after waiting for NO_DATA_WAIT. The Collector waits for the number of seconds specified in NO_DATA_WAIT and retries another read over and over. The Log Catalog is read to check whether a new LOG data set has been registered. | 0 or greater (decimal).                                                        |
| ECCRNAME     | Required. The ECCR name for the IDMS log-based ECCR. The ECCR name value must be unique within a PowerExchange Logger group. **Warning:** If you change the ECCRNAME value, the ECCR cannot warm start from the last stopped position. The IDMS log-based ECCR uses the value specified for the following purposes:  
- The ECCR name that connects to the PowerExchange Logger to write change data  
- The member name that joins the XCF group of the PowerExchange Logger  
- As part of the ECCR-UOW field in the control information for each change record written to PowerExchange Logger log files  
**Tip:** Informatica recommends that you use the same value for the ECCRNAME parameter and the IDMS log-based ECCR started task or job name. This practice allows you to easily identify the IDMS log-based ECCR when reviewing messages and data from the PowerExchange Logger. | 1 through 8 alphanumeric characters. Default is PWXIDLEC.                       |
| DB_TYPE      | Specify IDL for MVS IDMS log-based.                                           | IDL                                                                          |

2. The ECCR is delivered in RUNLIB JCL member ECCRIDL.
The following table describes the JCL Statements for the IDMS ECCR Startup Procedure:

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the DTLCCIDL program.</td>
</tr>
<tr>
<td>STEPLIB DD</td>
<td>Specify the PowerExchange LOADLIB and LOAD.</td>
</tr>
<tr>
<td>EDMPARMS DD</td>
<td>Specify the name of the user library (YOUR.USERLIB) that contains the default options module (EDMSDIR) associated with the PowerExchange Logger you are using. If you do not include an EDMPARMS DD statement or if the library you specify does not contain the options modules, PowerExchange Change Capture uses the STEPLIB concatenation to obtain the configuration options.</td>
</tr>
<tr>
<td>DTLACFG DD</td>
<td>Points to the IDMS ECCR configuration file ECCRIDLP.</td>
</tr>
<tr>
<td>EDMMSG DD</td>
<td>Specify the output data set for IDMS log-based ECCR messages.</td>
</tr>
</tbody>
</table>

3. Run the procedure, or start it as a started task by using the MVS START command.

4. Ensure that all the IDMS logs that require processing have been added to the PowerExchange Log Catalog. Sample ECCRIDL JCL to activate the IDMS log-based ECCR (the place holders HLQ and LOGGER will be substituted with the appropriate information as added in the MVS Install Assistant when installing):

```c
        /**************************************************************************
        /*                        RUN DETAIL IDMS LOG BASED ECCR                     */
        /**************************************************************************
/ECCRAD1 EXEC PGM=DTLCCIDL,REGION=50M
/STEPLIB DD DISP=SHR,DSN=HLQ..LOADLIB
//EDMPARMS DD DISP=SHR,DSN=HLQ..LOGGER..USERLIB
//DTLCFG DD DISP=SHR,DSN=HLQ..RUNLIB(DM0V0R)
//DTLCFG DD DISP=SHR,DSN=HLQ..RUNLIB(ECCRIDLP)
//DTLACFG DD DISP=SHR,DSN=HLQ..CRT
//DTLMSG DD DISP=SHR,DSN=HLQ..DLMSG
//DTLOG DD SYSOUT=* 
//DDDPRINT DD SYSOUT=* 
//DDDPRNT DD SYSOUT=* 
//SYSOGMP DD SYSOUT=* 
//SYSOGDD DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//EDMMSG DD SYSOUT=* 
//CEEDUMP DD SYSOUT=* 
```

The process described previously details the requirements for starting an ECCR in a simple environment.

Creating the Log Catalog

The Log Catalog holds information about the IDMS logs, which are available for the use of PowerExchange log-based capture. During the initial installation of PowerExchange, a Log Catalog VSAM file will be created (default naming will be HLQ..LOGSCAT) and a dummy record will be added.

For IDMS log-based capture to work effectively, it is vital to ensure that the log catalog is updated in a timely fashion and that log information is both secure and available. If the logs are not in the catalog, the records they hold will be unknown to PowerExchange. The correct way to add information to the catalog is to use utility DTLULCAT to format the input, run DTLULOGC to amend the Log Catalog with that prepared input.
RUNLIB member DTLULCAU is supplied to run the two utilities one after the other. It is expected that this be scheduled to run as soon as the latest IDMS log had been spooled off. There may, however, be times when DTLULOGC is run in isolation, involving manual coding of the input file.

Correct scheduling of the addition logs to the Log Catalog is vital to obtaining timely data from the log-based IDMS capture environment.

**RELATED TOPICS:**

* "Configuring IDMS Log Catalog Procedures" on page 185

**Running DTLULCAT**

Use the DTLULCAT utility to take a supplied journal name and use it to prepare the input required for the catalog utility DTLULOGC. The DTLULCAT utility is delivered as an executable on Windows and as the DTLULCAT member in the RUNLIB on MVS.

Sample utility statements:

```
IDMS_VERSION=15
FILE_TYPE=C
MEDIA_TYPE=D
MEDIA_CONTENT=BI
SERVICE=IDMSE150
INSTANCE_IDENTIFIER=XYLOGSID.
```

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDMS_VERSION</td>
<td>Versions 15 and 16 are supported.</td>
</tr>
<tr>
<td>FILE_TYPE</td>
<td>One of the following file types:</td>
</tr>
<tr>
<td></td>
<td>- C. Central version.</td>
</tr>
<tr>
<td></td>
<td>- L. Local mode.</td>
</tr>
<tr>
<td>MEDIA_TYPE</td>
<td>One of the following media types:</td>
</tr>
<tr>
<td></td>
<td>- T. Tape.</td>
</tr>
<tr>
<td></td>
<td>- D. Disk.</td>
</tr>
<tr>
<td>MEDIA_CONTENT</td>
<td>One of the following options for the types of images of change records</td>
</tr>
<tr>
<td></td>
<td>delivered:</td>
</tr>
<tr>
<td></td>
<td>- BI. Before images.</td>
</tr>
<tr>
<td></td>
<td>- AI. After images.</td>
</tr>
<tr>
<td></td>
<td>- BA. Both before and after images.</td>
</tr>
<tr>
<td>SERVICE</td>
<td>IDMS CV name or Local Job name.</td>
</tr>
<tr>
<td>INSTANCE_IDENTIFIER</td>
<td>Chosen LOGSID identifier.</td>
</tr>
</tbody>
</table>

The DTLULCAT utility writes information to DDCARD SYSPUNCH. This file is the input to the DTLULOGC utility.

**Running DTLULOGC**

The DTLULOGC utility populates the log catalog with information about the logs to process.

The following example shows sample DTLULCAU JCL, which runs DTLULCAT followed by DTLULOGC. The DTLULCAU JCL is the recommended method of populating the Log Catalog.

The example adds the log DTLUSR.IDMS.E15SP0.OFF.LOADED.JOURNAL for an IDMS Version 15 environment with the CV Name IDMSE150. The log resides on disk storage and will be accessed using a LOGSID value of XYLOGSID. The SYSIN data is shown as instream data for clarity. However, the sample JCL points to
member DTLIDLC when running against a CV (DTLIDLL for Local Job mode) in which these statements would normally be placed.

Providing SR2 and SR3 Information to the ECCR

IDMS records may become split into SR2 and SR3 records where a record can no longer fit on its home page. This situation normally occurs because of database reorganization, resulting in an SR2 record of eight bytes held on the home page and the SR3 record held elsewhere. PowerExchange needs to understand the position of these SR3 records, and this must be facilitated by running utility DTLUCSR2. After DTLUCSR2 has been run, the ECCR can scan the internal tables for SR2 and matching SR3 records.

Running DTLUCSR2

You must run the DTLUCSR2 utility before running IDMS log-based capture the first time and after subsequent database reorganizations.
To run DTLUCSR2:

1. Edit RUNLIB member DTLICSRI.
   
   For each database for which records will be registered for capture, edit the sample statements with the relevant values as described in the following example and table:

   ```
   Read,
   DD_NAME=ddname
   PAGE_GROUP=n
   RADIX=x
   ```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD_NAME</td>
<td>Specify the DDNAME that must then be added to the DTLUCSR2 JCL. This value does not have to match a DD name from an IDMS region but must match exactly the DD name added to your DTLUCSR2 JCL. Format: DD_NAME=STUDENT</td>
</tr>
<tr>
<td>PAGE_GROUP</td>
<td>If the database file is normally accessed with a page group other than 0, the PAGE_GROUP number must be specified.</td>
</tr>
<tr>
<td>RADIX</td>
<td>RADIX must be specified if it is not the default value of 8. Valid range is 2 to 12.</td>
</tr>
</tbody>
</table>

   **Note:** DTLUCSR2 will write control information to DD SR2TOTAL, and SR2/SR3 link information to SR2OUT. These files are created with default information at installation time, but the file sizes may need to be reviewed and amended depending upon the number of SR3 records.

2. Add relevant DD cards to your DTLUCSR2 JCL, which match the DD names supplied in parameter file DTLICSRI.
   
   The DD cards added point to the relevant IDMS data set names.

3. Run RUNLIB member DTLUCSR2.

---

**Managing IDMS Log-Based CDC**

Occasionally, you might need to alter the Log Catalog or recover change capture processing after DB2 ECCR failures or IDMS restore operations.

**Manually Manipulating the Log Catalog**

During normal IDMS log processing, the Log Catalog is updated using the combination of DTLULCAT and DTLULOGC to add the next available log. There might be times when details about certain log entries need to be altered, or even for logs to be removed from the catalog. To do this, you can run the DTLULOGC utility standalone with hand-coded input.

The DTLULOGC utility is the DTLULOGC JCL in RUNLIB. The utility enables you to:

- Add an instance.
- Add a log.
- Update a log entry.
- Delete an entry.
- Export an entry to another data set for offloading.
The following table shows the parameters that you can code in an 80-byte file that is specified as input in the SYSIN DD card, as shown in the previous sample JCL:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDINSTANCE</td>
<td>Add a LOGSID instance to the catalog. Each LOGSID used will require an instance to be added to the log catalog.</td>
<td></td>
</tr>
<tr>
<td>INSTANCEIDENTIFIER</td>
<td>LOGSID value.</td>
<td></td>
</tr>
<tr>
<td>VERSION</td>
<td>Version number of the entry.</td>
<td></td>
</tr>
<tr>
<td>Keyword</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ADD_ENTRY</td>
<td>BLOCK_SIZE</td>
<td>Block size of the log. Required if the logs are to be shipped to another platform.</td>
</tr>
<tr>
<td></td>
<td>ENTRY_NUMBER</td>
<td>Sequential number, which should be incremented by 1 for each new log added to the log catalog.</td>
</tr>
<tr>
<td></td>
<td>FILE_TYPE</td>
<td>- C. Central or Shared Service Log or Journal.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- L. Local Mode or Unshared Service Log or Journal.</td>
</tr>
<tr>
<td></td>
<td>FIRST_RECORD_SEQUENCE_NUMBER</td>
<td>Sequence number of the first record in the block.</td>
</tr>
<tr>
<td></td>
<td>FIRST_RECORD_TIME_STAMP</td>
<td>Timestamp of the first record in the block.</td>
</tr>
<tr>
<td></td>
<td>IDMS_VERSION</td>
<td>Version number of IDMS. Specified as an integer.</td>
</tr>
<tr>
<td></td>
<td>INSTANCE_IDENTIFIER</td>
<td>LOGSID value</td>
</tr>
<tr>
<td></td>
<td>LAST_RECORD_IDENTIFIER</td>
<td>Record ID of the last record in the block or zeros if a non-data record.</td>
</tr>
<tr>
<td></td>
<td>LAST_RECORD_OFFSET</td>
<td>Offset of last valid offset in the block.</td>
</tr>
<tr>
<td></td>
<td>LOG_DATA_TYPE</td>
<td>IDL for MVS IDMS log data.</td>
</tr>
<tr>
<td></td>
<td>LOG_FILE_NAME</td>
<td>Name of IDMS log file.</td>
</tr>
<tr>
<td></td>
<td>MEDIA_CONTENT</td>
<td>- AI. Only contains After images.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- BI. Only contains Before images.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- BA. Contains both Before and After images.</td>
</tr>
<tr>
<td></td>
<td>MEDIA_TYPE</td>
<td>- D. Disk.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- T. Tape.</td>
</tr>
<tr>
<td></td>
<td>NUMBER_OF_BLOCKS</td>
<td>Number of blocks in the log.</td>
</tr>
<tr>
<td></td>
<td>SERVICE</td>
<td>CV name or Local Mode job name.</td>
</tr>
<tr>
<td></td>
<td>STATUS</td>
<td>- A. Active.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- S. Skip.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- T. Terminate.</td>
</tr>
<tr>
<td></td>
<td>ENTRY_TYPE</td>
<td>- 1. File entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2. Reserved for future use.</td>
</tr>
<tr>
<td></td>
<td>VERSION</td>
<td>Version number of the entry.</td>
</tr>
<tr>
<td>UPDATE_ENTRY</td>
<td>Valid parameters are those listed for ADD_ENTRY</td>
<td>Updates a log entry. The entry is identified by the value of INSTANCE_IDENTIFIER and ENTRY_NUMBER.</td>
</tr>
<tr>
<td>DELETE_ENTRY</td>
<td>INSTANCE_IDENTIFIER</td>
<td>Deletes the last log for the specified INSTANCE_IDENTIFIER.</td>
</tr>
</tbody>
</table>
### Keyword

**Keyword** | **Parameter** | **Description**
---|---|---
REPORT_INSTANCE | INSTANCE_IDENTIFIER | Lists catalog entries for the specified INSTANCE_IDENTIFIER.
EXPORT_INSTANCE | INSTANCE_IDENTIFIER | Used to export all information for a specified INSTANCE_IDENTIFIER to a file.

**Note:** Keyword commands are separated by a semicolon (;), parameters by a comma (,).

The following sample input adds two instances (LOGSIDs), adds entries (log files), deletes an entry, reports instance LOGSida, exports instance LOGSida to a file (dtdlgce.txt), and finally deletes instance LOGSIda:

```sql
ADD_INSTANCE INSTANCE_IDENTIFIER=LOGSIDA, VERSION=224;
ADD_ENTRY INSTANCE_IDENTIFIER=LOGSIDA, ENTRY_NUMBER=777, VERSION=0, ENTRY_TYPE=1, STATUS=A,
LOG_DATA_TYPE=IDL, IDMS_VERSION=15, FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE-IDMSE150,
LOG_FILE_NAME=xxxxxxxxxxxxxxxxxxxxxxx, BLOCK_SIZE=29000, NUMBER_OF_BLOCKS=445,
LAST_RECORD_OFFSET=1119, LAST_RECORD_IDENTIFIER=3, FIRST_RECORD_SEQUENCE_NUMBER=4,
FIRST_RECORD_TIME_STAMP="05/03/03 10:55:01";
ADD_ENTRY INSTANCE_IDENTIFIER=LOGSIDA, ENTRY_NUMBER=778, VERSION=0, ENTRY_TYPE=1, STATUS=A,
LOG_DATA_TYPE=IDL, IDMS_VERSION=15, FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE-IDMSE150,
LOG_FILE_NAME=xxxxxxxxxxxxxxxxxxxxxxx, BLOCK_SIZE=29000, NUMBER_OF_BLOCKS=445,
LAST_RECORD_OFFSET=1119, LAST_RECORD_IDENTIFIER=3, FIRST_RECORD_SEQUENCE_NUMBER=4,
FIRST_RECORD_TIME_STAMP="05/03/03 12:55:01";
ADD_ENTRY INSTANCE_IDENTIFIER=LOGSIDA, ENTRY_NUMBER=779, VERSION=0, ENTRY_TYPE=1, STATUS=A,
LOG_DATA_TYPE=IDL, IDMS_VERSION=15, FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE-IDMSE150,
LOG_FILE_NAME=xxxxxxxxxxxxxxxxxxxxxxx, BLOCK_SIZE=29000, NUMBER_OF_BLOCKS=333,
LAST_RECORD_OFFSET=1119, LAST_RECORD_IDENTIFIER=3, FIRST_RECORD_SEQUENCE_NUMBER=4,
FIRST_RECORD_TIME_STAMP="05/03/03 14:55:01";
ADD_INSTANCE INSTANCE_IDENTIFIER=ABCD, VERSION=0;
ADD_ENTRY INSTANCE_IDENTIFIER=ABCD, ENTRY_NUMBER=1, VERSION=0, ENTRY_TYPE=1, STATUS=A,
LOG_DATA_TYPE=IDL, IDMS_VERSION=15, FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE-IDMSE15P,
LOG_FILE_NAME=xxxxxxxxxxxxxxxxxxxxxxx, BLOCK_SIZE=29000, NUMBER_OF_BLOCKS=444,
LAST_RECORD_OFFSET=1112, LAST_RECORD_IDENTIFIER=2, FIRST_RECORD_SEQUENCE_NUMBER=3,
FIRST_RECORD_TIME_STAMP="05/04/03 02:55:01";
ADD_ENTRY INSTANCE_IDENTIFIER=ABCD, ENTRY_NUMBER=2, VERSION=0, ENTRY_TYPE=1, STATUS=A,
LOG_DATA_TYPE=IDL, IDMS_VERSION=15, FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE-IDMSE15P,
LOG_FILE_NAME=xxxxxxxxxxxxxxxxxxxxxxx, BLOCK_SIZE=29000, NUMBER_OF_BLOCKS=444,
LAST_RECORD_OFFSET=1119, LAST_RECORD_IDENTIFIER=3, FIRST_RECORD_SEQUENCE_NUMBER=4,
FIRST_RECORD_TIME_STAMP="05/03/03 10:55:01";
UPDATE_ENTRY INSTANCE_IDENTIFIER=LOGSIDA, ENTRY_NUMBER=779, VERSION=0, ENTRY_TYPE=1, STATUS=A,
LOG_DATA_TYPE=IDL, IDMS_VERSION=15, FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE-DTLXXXXX,
LOG_FILE_NAME=xxxxxxxxxxxxxxxxxxxxxxx, BLOCK_SIZE=29000, NUMBER_OF_BLOCKS=111,
LAST_RECORD_OFFSET=1119, LAST_RECORD_IDENTIFIER=3, FIRST_RECORD_SEQUENCE_NUMBER=4,
FIRST_RECORD_TIME_STAMP="05/04/03 12:55:01";
DELETE_ENTRY INSTANCE_IDENTIFIER=LOGSIDA;
REPORT_INSTANCE INSTANCE_IDENTIFIER=LOGSIDA;
EXPORT_INSTANCE INSTANCE_IDENTIFIER=LOGSIDA;
DELETE_INSTANCE INSTANCE_IDENTIFIER=LOGSIDA;
```

## Recovering from Failures

This section contains procedures to assist you in resuming change propagation after the propagation stops for any reason.

The information in this chapter includes:
- A description of the conditions that require you to perform recovery and restart procedures to resume a change propagation.
- Procedures to restart change propagations for every target.
Recovering the IDMS Log-Based ECCR

You need to recover the IDMS log-based ECCR in the following cases:

- When the IDMS log-based ECCR fails.
- When the PowerExchange Logger stops or fails while attached to the IDMS log-based ECCR.

If the PowerExchange Logger stops or abends while attached to the IDMS log-based ECCR, the ECCR also abends when it receives the first change record following the PowerExchange Logger failure. When you restart the IDMS log-based ECCR or the Logger after a failure, the Logger determines the point at which to begin capturing changes again.

To recover the IDMS log-based ECCR:

1. Determine the cause of the ECCR failure and correct it.
2. If the ECCR failed because the PowerExchange Logger stopped, restart the Logger.
3. Restart the IDMS log-based ECCR from the point at which it abended.

The ECCR undergoes a warm start if there is warm start data available from the Agent or Logger. It automatically restarts at the correct point. If there is no warm start data available, the ECCR issues a prompt for a cold start. Be sure that you use the same ECCRNAME in your ECCRIDLP parameter file that you used for the ECCR that abended.

Recovering after IDMS Restores or Reruns

The PowerExchange Logger should be used as a log of all PowerExchange activity and is not normally restored. When the source database is restored because of application failures, you typically reset the application extraction start points to the relevant point.

To help identify the correct point to start, use the Event Marker Utility, EDMXLUTL, to put markers into the Logger on a regular basis. When these markers are added, they appear in the PowerExchange log. These markers can also be read from the log using the EMR access method through the PowerExchange Navigator.

Managing IDMS Schema Changes

Use this procedure to manage schema changes for the source and target tables that are involved in change propagation.

1. Stop all update activity against the relevant IDMS data.
2. Ensure that PowerExchange has processed all changes that occurred under the old schema.
3. Change the IDMS schema.
4. Reflect the schema changes in the PowerExchange registration.
5. Ensure that any updated schemas are in place in the PowerExchange copies of the IDMS libraries.
6. Allow update activity to the IDMS data to resume.
Introduction to IMS Change Data Capture

PowerExchange for IMS change data capture (CDC) captures changes made to IMS databases and logs those changes to the PowerExchange Logger. You can use PowerCenter to extract the captured change data from the PowerExchange Logger or from condense files and apply that data to one or more target databases.

PowerExchange for IMS provides the following CDC options:

- Synchronous IMS CDC captures the changes as they occur and logs them to the PowerExchange Logger. The IMS synchronous ECCR runs as separate subtasks in IMS regions such as the control region, DBCTL, DL/1, and DBB batch jobs.

- Log-based IMS CDC reads the changes from the IMS archive logs and logs them to the PowerExchange Logger. The IMS log-based ECCR runs in a separate address space and can be either a started task or a batch job.

The following table describes the functional differences between the IMS synchronous ECCR and the IMS log-based ECCR implementations of CDC:
IMS Log-Based Change Data Capture

The IMS log-based ECCR is an asynchronous collection method that reads change data from closed IMS archive logs. The IMS log-based ECCR passes the captured changes to the PowerExchange Logger. After the changes are logged, they are available for extraction.

The IMS log-based ECCR reads the changed data for registered IMS databases from the IMS archive logs. Based on specified parameters, the ECCR periodically inspects the IMS RECON data sets for new archive logs to process. Because IMS log-based capture reads closed IMS archive logs, there is a time delay between the change being made and the capture of the changed data. The length of the time delay between the change being made and the change being captured is dependent upon the following factors:

- How quickly IMS archives the active logs after the change is made.
- How frequently the IMS log-based ECCR checks for new archive logs.

The IMS log-based ECCR is a separate address space and runs either continuously or in batch mode. During initialization, the ECCR reads the registration data set to determine which IMS databases are registered for capture. You must make the following changes to the IMS environment for all databases for which capture is required:

- Change the DBD to include the EXIT statement.
- Register the databases in DBRC, if not already registered.

IMS log-based change data capture is built around an architecture, which allows functions to capture changes, process them according to user-specified rules, and provide the input for business processes. The core architecture is a multi-tasking environment, which has the capability to perform many functions in parallel.

The IMS log-based ECCR task collects changes for registered data sources. Once the changes have been collected, they will be transformed into a PowerExchange internal format, which is essentially the same regardless of the source of the changes. Further processing, condensing and extraction is the same regardless of the source.
The IMS log-based ECCR captures changes to IMS databases from IMS logs and logs the changes in the PowerExchange Logger. On start-up, the IMS log-based ECCR goes through several steps:

- Initialization
- Processing blocks of data
- Waiting for data

**Initialization**

The initialization process performs the following tasks:

- Checks and loads the registrations.
- Determines which RECON data set in the list provided in the input parameters is the current data set.
- Uses the RECON to determine which log data sets to process and in which order.
- Opens a connection to the PowerExchange Logger and retrieves restart information.
- Sets up searchable structures and allocates work buffers.

**Processing Blocks of Data**

Log records are read, compared to the registrations, deconstructed, and the data assembled to provide the changed data. The changed data is passed to the PowerExchange Logger. Unit of recovery data is kept in memory until it is either complete or abandoned, so that appropriate checkpoint or abort calls can be made to the PowerExchange Logger. This data is also logged for restart purposes.

**Waiting for Data**

The collector process is constantly trying to obtain new logs to process. When the IMS log-based ECCR reaches the IMS log position that was current at the beginning of this collector run, a “no more log data” return code is presented. When no data is returned, the process waits until either:

- An interrupt from an event has occurred.
- The parameterized wait period has elapsed.

**Relationships with Other PowerExchange Components**

PowerExchange for IMS change data capture is shipped with the standard PowerExchange software.

The IMS log-based ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- The IMS log-based ECCR must log all changes to a single PowerExchange Logger running on the same MVS system.
- The PowerExchange Logger and PowerExchange Agent must run on the same MVS system as the IMS log-based ECCR.
- Operational issues in the PowerExchange Logger can cause the IMS log-based ECCR to enter a wait state, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the IMS log-based ECCR continues the capture and recording of change data without any loss of data.

You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.
Configuring IMS for Log-Based CDC

Before you can use IMS log-based change data capture, verify the following for each IMS database for which you want to capture changes:

- The DBD source for the database specifies the EXIT parameter.
- The database is registered with DBRC.

Specifying the EXIT Parameter in the DBD Source

The following changes have to be made to the IMS DBD to allow for the writing of the IMS capture log records. On the DBD for which data is to be captured, specify the EXIT parameter.

For example:

```
DBD NAME=DBFSAMD3, ACCESS=DEDB, RMNAME=DBFHDC40,
EXIT=(*, KEY, PATH, (CASCADE, KEY, PATH), LOG)
```

The EXIT statement causes IMS to create log record type x'99' for data logged for the segment. The IMS log-based ECCR uses x'99' record to obtain the changed data. Using the EXIT statement will increase the number of log records for IMS online and batch regions.

To reduce the amount of x'99' records, modify the EXIT= statement and change PATH to NOPATH. PATH causes the logging of the entire hierarchical path for the segment and NOPATH causes the logging of the segment only. Only use NOPATH if you are registering a table represented by a single segment for capture. NOPATH incurs less logging overhead as IMS does not log the entire hierarchical path.

For more information on specifying the EXIT parameter, see *IBM IMS Utilities Reference: System*.

Configuring the IMS Log-Based ECCR

Prior to starting the IMS log-based ECCR, you must:

- Configure the IMS log-based ECCR parameters.
- Add the PowerExchange LOAD library to the APF list.
- Configure the IMS log-based ECCR JCL.
- Create at least one capture registration for an IMS source.
Configuring the IMS Log-Based ECCR Parameters

Configure the IMS log-based ECCR parameters in the CAPTIMS parameter member of the RUNLIB library. The IMS log-based ECCR has the following parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>BYPASS_VERSION_CHECKING</td>
<td>Specifies whether the IMS log-based ECCR should verify that the IMS version it is using matches the IMS version of the DBRC RECON data sets. Specify Y for this parameter if you upgrade your RECON data sets to a higher release of IMS in preparation for upgrading IMS. Otherwise, specify N.</td>
<td>Y or N. Default is N.</td>
</tr>
<tr>
<td>CAPT_STATS</td>
<td>If set to Y, CAPT_STATS causes a WTO to be written when an SLDS has been processed. After each SLDS is processed the stats regarding each registration, number of inserts, updates and deletes appear in the ECCR. On shutdown the overall totals are seen.</td>
<td>Y or N. Default is N.</td>
</tr>
<tr>
<td>COLDSTART</td>
<td>Indicates whether the ECCR is to perform a cold or warm start. There are three methods of invoking a cold start of the IMS log-based ECCR: - Start the ECCR using a new PowerExchange Logger that IMS log-based ECCR has never connected to. - Change the value of ECCRNAMED in the HLQ.RUNLIB(CAPTIMS) member. This should be pointed to by the DTLCACFG DD in the JCL. - Code COLDSTART=Y in the RUNLIB(CAPTIMS) member. This should be pointed to by the DTLCACFG DD in the JCL.</td>
<td>Y. Directs the ECCR to perform a cold start, which means it starts processing the next IMS log file created. N. Directs the ECCR to perform a warm start, which means it will continue processing where it left off. Default is N.</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Defines the type of collector to be used, such as IMS.</td>
<td>IMS</td>
</tr>
<tr>
<td>DBID</td>
<td>Defines the IMS RECON Identifier parameter of the Registration Group for which this instance of the collector is being run.</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| ECCRNAME    | Required. The ECCR name for the IMS log-based ECCR. The ECCR name value must be unique within a PowerExchange Logger group. **Warning:** If you change the ECCRNAME value, the ECCR cannot warm start from the last stopped position. The IMS log-based ECCR uses the value specified for the following purposes:  
  - The ECCR name that connects to the PowerExchange Logger to write change data  
  - The member name that joins the XCF group of the PowerExchange Logger  
  - As part of the ECCR-UOW field in the control information for each change record written to PowerExchange Logger log files  
  **Tip:** Informatica recommends that you use the same value for the ECCRNAME parameter and the IMS log-based ECCR started task or job name. This practice allows you to easily identify the IMS log-based ECCR when reviewing messages and data from the PowerExchange Logger. | 1 through 8 alphanumeric characters. Default is PWXIMSEC. |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR_LOG</td>
<td>This parameter controls the behavior of the IMS log-based ECCR when it encounters an IMS log which is marked in the RECON data set as one of the following:</td>
<td>- <strong>ABEND</strong>. This value is the default. When the IMS ECCR encounters a log marked “in error,” it ends with a WTO to the MVS console and messages that indicate that the log that was in error. The messages include the start and stop times of the log in question. The ECCR ends in such a way that it can be restarted after the log in question is resolved by the user. - <strong>SKIP</strong>. The IMS ECCR skips the log “in error.” Use care when using this value. Changes can be missed, which can invalidate the accuracy of the capture target (the Data Warehouse). When this option is used, messages such as the start and stop time and name are issued to indicate which logs have been skipped. - <strong>WAIT</strong>. When encountering an IMS log “in error,” the IMS ECCR issues informational messages to indicate the status of the IMS LOG, and then it sleeps. The IMS ECCR wakes up periodically (according to the NO_DATA_WAIT2 value) to check the log status. Once resolved, it continues processing. The user has the option to change the status of the log by performing the relevant IMS steps or to remove the log from the RECON data set. The user must ensure that no changes are lost. - <strong>WTOR</strong>. Stops the IMS ECCR from continuing and issues a WTO to ask for the option to use. - <strong>No response</strong>. The ISM ECCR waits forever. Again messages are required to detail the reason, log in question.</td>
</tr>
<tr>
<td>IMSID</td>
<td>Defines the IMS subsystem ID (SSID), the DBDLIB data set, and the RECON data sets. Enables PowerExchange to register IMS data sources for CDC.</td>
<td>Syntax: <code>IMSID=(ims_ssid, DBD_lib, recon, recon, ...)</code> Where: - <code>ims_ssid</code> is the IMS SSID. Specify a 1- to 8-character alphanumeric string. - <code>DBD_lib</code> is the name of the IMS DBDLIB data set that contains the DBD load modules. - <code>recon</code> is an IMS RECON data set. Use commas to separate RECON data sets.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MSGLVL</td>
<td>Specifies whether or not to provide detailed messages about the processing of IMS logs from the RECON data sets as well as statistical information about change data capture.</td>
<td>- 0. Detailed messages are not reported.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1. Detailed messages are reported. This value is the recommended setting.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Default is 0.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>Specifies the number of seconds for the ECCR to wait before doing another read after the ECCR received an end-of-log condition.</td>
<td>A number from 1 through 99999999. Default is 60.</td>
</tr>
<tr>
<td></td>
<td>On receiving another end-of-log condition on the first read following the previous end-of-log the ECCR will wait NO_DATA_WAIT_2 before retrying another read.</td>
<td></td>
</tr>
<tr>
<td>NO_DATA_WAIT_2</td>
<td>Specifies the number of seconds for the ECCR to wait before doing another read after the ECCR received an end-of-log condition after waiting for COLL_NO_DATA_WAIT. The ECCR will wait NO_DATA_WAIT_2 and retry another read over and over again. The RECON is read to check whether a new LOG data set has been registered.</td>
<td>A number from 1 through 99999999. Default is 600.</td>
</tr>
<tr>
<td>RECID</td>
<td>This value relates to the record ID of the start marker written to the IMS log by the utility DTLCUIML. The record prevents data capture from a point prior to the required start point.</td>
<td>A0 to FF. Default is A0.</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>Enables you to direct the IMS log-based ECCR to start processing change records (ISRT/REPL/DLET) from IMS Logs after the specified start time. STARTTIME is only used is during a cold start of the ECCR.</td>
<td>Example: STARTTIME=&quot;05/01/01 09:00:00&quot;</td>
</tr>
<tr>
<td>WRITE_RESTART_SECS</td>
<td>Controls how often, in seconds, a special restart UOW is written to the PowerExchange Logger when nothing of interest has occurred since the last special restart UOW was written. This value affects how far back the PowerExchange Logger searches to find the restart point when the ECCR is restarted.</td>
<td>A decimal number of 0 or greater. Default is 600 seconds.</td>
</tr>
</tbody>
</table>

**Note:** While the IMS log-based ECCR and PowerExchange Condense parameters have similar names, two different members are required. PowerExchange Condense parameters are found in the CAPTIMSS member.

**Related Topics:**

- “Configuring PowerExchange Condense Parameters” on page 84
Adding the PowerExchange LOAD Library to the APF List

Verify that the PowerExchange LOAD library is APF-authorized. The PowerExchange LOAD library should have been added to the APF list as a part of the PowerExchange installation process.

Configuring the IMS Log-Based ECCR JCL

You must configure the IMS log-based ECCR JCL.

PowerExchange provides sample JCL in the xxxIMSEC member in the PowerExchange PROCLIB library, where xxx is the PowerExchange Agent / Logger Prefix value that you specified in the MVS Installation Assistant.

The IMS log-based ECCR can run as a batch job or as a started task. To run the ECCR as a started task, convert the ECCRIMS JCL to a PROC and copy it to the system PROCLIB library that is for started tasks.

Specify the job-step program for the IMS log-based ECCR based on the IMS version you are running. The following table shows the program names for each supported IMS version:

<table>
<thead>
<tr>
<th>IMS Version</th>
<th>Program Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>DTLCCIM8</td>
</tr>
<tr>
<td>9.1</td>
<td>DTLCCIM9</td>
</tr>
<tr>
<td>10</td>
<td>DTLCCIMA</td>
</tr>
</tbody>
</table>

The ECCRIMS JCL contains the following statements:

```
//*ECCRIMS EXEC PGM=DTLCCIM8,TIME=NOLIMIT,REGION=0M (V7)
//*ECCRIMS EXEC PGM=DTLCCIM8,TIME=NOLIMIT,REGION=0M (V8)
//*ECCRIMS EXEC PGM=DTLCCIM9,TIME=NOLIMIT,REGION=0M (V9)
//*ECCRIMS EXEC PGM=DTLCCIMA,TIME=NOLIMIT,REGION=0M (V10)
```

The MVS Installation Assistant configures the JCL and removes the comment for the EXEC card that matches the IMS version specified. Verify that the correct EXEC card is chosen.

IMS Log-Based ECCR DD Statements

Running the IMS log-based ECCR requires a variety of DD statements for data sets to be present. Some of the data sets are allocated at installation time while others are created dynamically by the IMS log-based ECCR.

The IMS log-based ECCR requires the following DD statements and data sets:

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAMAP</td>
<td>This data set contains the data maps used by the PowerExchange Listener. These are the standard data maps used for normal NRDB access to data.</td>
</tr>
<tr>
<td>DTLAMCPR</td>
<td>This data set contains the actual registration information. It is used by both the PowerExchange Listener and the IMS log-based ECCR. The PowerExchange Listener opens the data set in read/write mode whereas the ECCR only reads it.</td>
</tr>
<tr>
<td>DTLACFG</td>
<td>This data set contains the configuration parameters for the IMS log-based ECCR.</td>
</tr>
<tr>
<td>DTLCFG</td>
<td>This is an existing parameter file for PowerExchange. Some of the parameters are also applicable to the IMS log-based ECCR.</td>
</tr>
<tr>
<td>DD Name</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DTLKEY</td>
<td>The PowerExchange License key file, containing the license key for PowerExchange and the various options used.</td>
</tr>
<tr>
<td>DTLLOG and DTLLOG01</td>
<td>The PowerExchange logging files. These SYSOUT files contain various messages reporting the status and events for the IMS log-based ECCR.</td>
</tr>
<tr>
<td>DTLMSG</td>
<td>This data set contains the PowerExchange messages.</td>
</tr>
</tbody>
</table>

### Managing IMS Log-Based CDC

You can start, stop, and control the IMS log-based ECCR using commands.

When you register databases for CDC, you must restart the IMS log-based ECCR to activate the new or changed capture registrations.

#### Starting the IMS Log-Based ECCR

The PowerExchange Listener, PowerExchange Agent, and PowerExchange Logger must be running prior to starting the IMS ECCR.

You must first configure the IMS log-based ECCR JCL.

If you run the IMS log-based ECCR as a batch job, submit the configured JCL. If you run the IMS log-based ECCR as a started task, issue the MVS START command.

**RELATED TOPICS:**

- “Configuring the IMS Log-Based ECCR JCL” on page 204

#### Controlling the IMS Log-Based ECCR

PowerExchange IMS log-based ECCR has an operator interface. The interface allows the installation to operate and manage the PowerExchange IMS log-based ECCR by using the installation-specific system management tools.

You can enter multiple commands or parameters at the same time. As the command separator, use a space or a comma (,). You can enter the commands in both lowercase and uppercase.

Use the MVS MODIFY command to issue the commands. For example:

```
F jobname,command
```
The *jobname* variable is the name of the IMS log-based ECCR batch job or started task. The *command* variable is one of the following ECCR commands:

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY TRACE</td>
<td>Displays the tracing status.</td>
</tr>
<tr>
<td>CLOSE</td>
<td>Requests a shutdown of the PowerExchange IMS log-based ECCR.</td>
</tr>
<tr>
<td>TRACEOFF</td>
<td>Turns off all traces.</td>
</tr>
</tbody>
</table>
| TRACEON trace,severity | Turns on tracing.  
  For example, the following command turns on "trace1" when severity is less than or equal to 3: 
  TRACETRACEON trace1,3 |

**Stopping Change Data Capture for IMS Sources**

Use one of the following methods to stop change data capture for IMS sources:

<table>
<thead>
<tr>
<th>To Stop Capturing Changes for</th>
<th>Use This Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>All registered IMS databases</td>
<td>Stop the IMS log-based ECCR.</td>
</tr>
<tr>
<td>A specific registered IMS database</td>
<td>Deactivate or delete the capture registration, and restart the IMS log-based ECCR.</td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**

- “Stopping the IMS Log-Based ECCR” on page 206
- “Deactivating or Deleting Registrations” on page 206

**Stopping the IMS Log-Based ECCR**

Stop the IMS log-based ECCR using the MVS STOP command. Stopping the IMS log-based ECCR stops the capture of changes until the ECCR is restarted. For more information about IMS log-based ECCR commands, see the *PowerExchange Command Reference*.

When you stop the IMS log-based ECCR, it disconnects from the PowerExchange Logger and displays a set of messages, including the number and type of changes captured since the last time the data set was opened. For example:

```
14:07:37.56  FWKXDM17289I Change Capture counts for INLIMS1IMSVXCP1100000: Insert=3, Update=0, Delete=0
14:07:38.12  FWKXDM17289I Left XCF group 'DOCL' as member 'DLUSRIN'
14:07:38.12  FWKXDM17289I EDM ECCR sent 3 records to Logger DOCL (3 change records)
```

**Deactivating or Deleting Registrations**

Use PowerExchange Navigator to delete or deactivate capture registrations. To activate capture registration changes, you must restart the IMS log-based ECCR.
Using the DTLCUIML Utility to Create Restart Points

This utility is used to write user-defined records to the IMS log. When the IMS log-based ECCR encounters one of the user-defined records, it triggers a message in the PowerExchange Logger to create a marker in the change stream for the affected registration tags.

The tokens represented by the marker can be used to define the start point for an extraction in the PWXPC restart token file or in the DTLUAPPL utility for ODBC extractions.

There is no limit or restriction on the number of markers being set in the change stream. The IMS log record ID chosen has to be unique for the individual installation. Specify the IMS log record ID chosen in the RECID parameter for the IMS log-based ECCR.

The utility runs as a standard IMS application program. There is no need to provide a specific PSB. The utility can use any PSB as long as the first PCB in the PSB is an IOPCB. The utility uses the IMS LOG Call to write IMS log records.

This utility has to run as an IMS BMP job. This ensures that the IMS Log record is written into the IMS logs and that the associated log is read by the IMS log-based ECCR. Sample JCL is supplied in member IMSLOGW in the RUNLIB library.

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

Refreshing the IMS Log-based ECCR

When you register databases for CDC, refresh the IMS ECCR to activate the new or changed capture registration.

1. Restart the log-based ECCR.
2. If you are preparing capture registrations for a condense, recycle the Condense task after you refresh the IMS ECCR.

Managing IMS Schema Changes

If you need to make schema changes for an IMS database that is registered for change capture, use the following procedure. The procedure indicates how to make schema changes for an IMS database in a manner that enables change capture for data of the new format and that retains access to the historically captured data.

1. Stop all update activity against the IMS database.
2. Verify that the IMS log-based ECCR has captured all changes for the current schema.
3. Stop the IMS log-based ECCR.
4. Extract all of the captured changes for the IMS database.
5. Make the schema changes to the IMS database.
6. Create a new capture registration that reflects the schema changes.
7. Restart the IMS log-based ECCR.
8. Allow update activity to the IMS database.
**Introduction to IMS Change Data Capture**

PowerExchange for IMS change data capture (CDC) captures changes made to IMS databases and logs those changes to the PowerExchange Logger. You can use PowerCenter to extract the captured changed data from the PowerExchange Logger or from condensed files and apply it to one or more target databases.

PowerExchange for IMS provides the following CDC options:

- **Synchronous IMS CDC** captures the changes as they occur and logs them to the PowerExchange Logger. The IMS synchronous ECCR runs as separate subtasks in IMS regions such as the control region, DBCTL, DL/1, and DBB batch jobs.

- **Log-based IMS CDC** reads the changes from the IMS archive logs and logs them to the PowerExchange Logger. The IMS log-based ECCR runs in a separate address space and can be either a started task or a batch job.

The following table describes the functional differences between the IMS synchronous ECCR and the IMS log-based ECCR implementations of CDC:

<table>
<thead>
<tr>
<th>Description</th>
<th>IMS Synchronous ECCR</th>
<th>IMS Log-Based ECCR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerExchange IMS interface modules install into IMS RESLIB.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>DBD for every database for which capture is required needs EXIT statement added.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Uses IMS external subsystem to communicate with IMS ECCR.</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
IMS Synchronous Change Data Capture

The IMS synchronous ECCR captures changes made by IMS transactions as they occur and passes the captured changes to the PowerExchange Logger. After the changes are logged, control is returned to the transaction and the changed data is available for extraction. IMS synchronous capture provides real-time changed data capture and near real-time extraction capability for changed data.

During the installation of IMS synchronous capture, you link PowerExchange code into the IMS RESLIB. The IMS synchronous ECCR uses this code to gain control during database OPEN processing to perform registration checks. Registration check processing communicates with the PowerExchange Agent to determine if the database being opened is registered for capture.

The IMS synchronous ECCR runs as separate subtasks in the IMS control region, IMS DBCTL region, or in DL/I and DBB batch regions. In addition to the modifications to the IMS RESLIB, you must also update the IMS region JCL. The PowerExchange CRG load library must be included in the STEPLIB for all IMS online and batch regions where capture is required. During the initialization of the IMS region, PowerExchange dynamically installs the IMS interface and initializes the IMS synchronous ECCR to capture changes.

The IMS synchronous ECCR captures changes made to IMS databases and logs those changes to the PowerExchange Logger. You must perform the following tasks before capturing changes for IMS databases:

- Start the PowerExchange Listener, PowerExchange Agent, and PowerExchange Logger.
- Install modification to IMS DBRC.
- Configure the IMS region and DBRC JCL.
- Configure an IMS external subsystem.
- Start the IMS subsystem.
- Create capture registrations for IMS database segments using the PowerExchange Navigator.
- Open databases to begin capturing changed data.

Once the IMS synchronous ECCR is active, you can activate new registrations by closing the database using the IMS DBR command and opening the database using the IMS START command. You can communicate with the ECCR using IMS external subsystem commands.
IMS CDC Operational Considerations

Review the following operational considerations and restrictions when planning your IMS CDC environment.

IMS Environments

The IMS synchronous ECCR operates in the following IMS environments:

- DBCTL
- DB/DC
- Batch IMS

IMS Synchronous ECCR Restrictions

The IMS synchronous ECCR has the following restrictions:

- **z/Architecture® mode.** If you use the PowerExchange CRG software, you must IPL the MVS system on which the IMS synchronous ECCR runs in z/Architecture mode.
  
  PowerExchange provides the CRG software in the CRG.LOAD library, which contains components from BMC Software products CHANGE RECORDING FACILITY and DATABASE INTEGRITY PLUS. The PowerExchange CRG software is based on version 4.6.00 of these products from BMC Software and requires that z/OS run in z/Architecture mode.
  
  If you have these BMC Software products, you can use them instead of the PowerExchange CRG software.

- The IMS synchronous ECCR does not capture changes made to IMS databases in the following situations:
  - When you run IMS migration, initialization, reorganization, or recovery utilities
  - If you specify PROCOPT=L in the program specification block (PSB)
  - If you have user data in secondary indexes
  - If an update request does not change any data in the segment that it updates

- The IMS synchronous ECCR does not support change data capture for the following IMS database types:
  - Hierarchical Sequential Access Method (HSAM) databases
  - Simple Hierarchical Sequential Access Method (SHSAM) databases
  - Generalized Sequential Access Method (GSAM) databases
  - Main Storage databases (MSDBs)
  - IMS Fast Path sequential dependent (SDEP) segments
  - Any IMS databases after an XRF failover
  - Block-level data-sharing IMS databases that are not in a sysplex

**RELATED TOPICS:**

- “Compatibility with BMC Software Products” on page 212
**IMS Synchronous ECCR Considerations**

When using the IMS synchronous ECCR, consider the following items:

- If you are capturing changes involving paired logical children, do so according to the following principles:
  - For virtual pairings, propagate changes from the physical child.
  - For physical pairings, use the child that contains the physical dependent segments from which you plan to propagate changes.

- In an online environment, the IMS synchronous ECCR runs as an IMS external subsystem. In this environment, IMS does not support the SETS function. IMS does support the SETU and ROLS functions if your application accepts the SC and RC status codes, respectively. If your application accepts the SC and RC status codes, the IMS synchronous ECCR captures changed data from SETU and ROLS functions.

- You can capture changes to both keyed and non-keyed segments.
  
  For non-keyed or non-uniquely keyed segments, the IMS synchronous ECCR generates an 8-byte field containing the relative byte address (RBA) of the segment. This RBA value is passed to the PowerExchange Logger where it is logged along with the changed data.
  
  To use this RBA value to create a unique key field for the segment, you must create a new field in the data map for the segment. You need to use the GetIMSRBAByLevel expression to populate this field with the captured RBA value. Use this altered data map to create the capture extraction map.
  
  Reorganizing the IMS source database changes the RBA value of segments. To ensure that the generated RBA value in the target is associated with the correct source data record, rematerialize the target table from the source table if it is reorganized.

**Relationships with Other PowerExchange Components**

The IMS synchronous ECCR uses other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent. Consider the following operational factors:

- The IMS synchronous ECCR must log all changes to a single PowerExchange Logger running on the same MVS image.

- The PowerExchange Logger and PowerExchange Agent must run on the same MVS image as the IMS synchronous ECCR.

- In configurations where updates to an IMS database occur on multiple MVS images, you must configure PowerExchange on each MVS image. PowerExchange requires an IMS synchronous ECCR, PowerExchange Logger, and PowerExchange Agent on each MVS image. The PowerExchange Logger on all MVS images must be configured to use Post-Log Merge.

- Operational issues in the PowerExchange Logger can cause IMS transactions to wait, which would prevent further capture and recording of change data until the issues are resolved. After you resolve the operational issues in the PowerExchange Logger, the waiting transactions continue and PowerExchange captures and records the change data without any loss of data.

  You must carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

**Related Topics:**

- “Monitoring the PowerExchange Logger for MVS” on page 51
- “Using Post-Log Merge” on page 69
Configuring the IMS Synchronous ECCR

The IMS synchronous ECCR captures changes made to IMS databases. Before you can capture changes, install the modifications to DBRC and to the IMS region JCL. Then activate the IMS synchronous ECCR in the appropriate IMS regions.

Depending upon the configuration options you chose, JCL for the following IMS regions may need to be modified:

- IMS control regions
- MPP and BMP dependent regions
- DBCTL regions
- DL/I and DBB batch regions

Compatibility with BMC Software Products

To use the IMS synchronous ECCR, PowerExchange requires components from the BMC Software CHANGE RECORDING FACILITY and DATABASE INTEGRITY PLUS products. If you use other BMC products such as CONCURRENT REORG or BMC MAXM Online/Reorg for IMS, then you have the CHANGE RECORDING FACILITY and DATABASE INTEGRITY PLUS products installed on your system.

Note: You do not need to have the BMC Software products installed to use PowerExchange. The PowerExchange installation process creates a data set called hlq.CRG.LOAD that contains the necessary CHANGE RECORDING FACILITY and DATABASE INTEGRITY PLUS components.

If you have these BMC Software products installed, verify that you have meet the following minimum requirements:

<table>
<thead>
<tr>
<th>BMC Software Product</th>
<th>Minimum Version that PowerExchange Requires</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHANGE RECORDING FACILITY</td>
<td>- 4.5.04 to support IMS Version 8.1 or 9.1</td>
</tr>
<tr>
<td></td>
<td>- 4.6.00 to support IMS Version 10.1</td>
</tr>
<tr>
<td>DATABASE INTEGRITY PLUS</td>
<td>- 4.5.04 to support IMS Version 8.1 or 9.1</td>
</tr>
<tr>
<td></td>
<td>- 4.6.00 to support IMS Version 10.1</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> DATABASE INTEGRITY PLUS 4.6.00 requires that the system on which it runs be IPLed in z/Architecture mode.</td>
</tr>
</tbody>
</table>

If you are unsure of the installed version of the CHANGE RECORDING FACILITY and DATABASE INTEGRITY PLUS products, browse the BMC load library and select the CRGLEVEL and DBILEVEL load modules. The version is displayed on the last line, after the date. If you need assistance, call BMC Software Technical Support.

If these BMC Software products are installed and meet the minimum release requirements, use the BMC Software load libraries instead of the PowerExchange hlq.CRG.LOAD library. Then, configure the IMS region JCL.

If the installed version of these products is earlier than the recommended minimum versions, you must upgrade before you activate the IMS synchronous ECCR. After the upgrade, continue configuring the IMS synchronous ECCR.

**Related Topics:**

- “Configuring IMS Region JCL” on page 214
Configuring IMS DBRC

PowerExchange requires modification to DBRC to allow the IMS synchronous ECCR to operate. This modification consists of including PowerExchange modules into the IMS RESLIB. The PowerExchange modules perform capture registration checks from the IMS DBRC address space.

**Note:** If the BMC Software product DATABASE INTEGRITY PLUS is installed, you do not need to install the PowerExchange version of this code. Verify that DATABASE INTEGRITY PLUS meets the minimum version requirement and then configure the IMS region JCL.

If the BMC Software product DATABASE INTEGRITY PLUS is not installed, you must install the PowerExchange modification to DBRC. The PowerExchange modification creates a new load module by including load module DBICRXvr with IMS DBRC load module DSPCRTR0. The new load module, DBICRYvr, resides in the IMS RESLIB (SDFSRESL). The variable vr represents the version and release of the IMS system. In addition, load module DBICRT00 replaces DSPCRTR0 in the IMS RESLIB.

The following table lists the `hlq.CRG.LOAD` load modules for each IMS version that are included with load module DSPCRTR0 to create the DBICRYvr load module:

<table>
<thead>
<tr>
<th>IMS Version.Release</th>
<th>CRG.LOAD Module Name</th>
<th>DBICRYvr Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>DBICRX81</td>
<td>DBICRY81</td>
</tr>
<tr>
<td>9.1</td>
<td>DBICRX91</td>
<td>DBICRY91</td>
</tr>
<tr>
<td>10.1</td>
<td>DBICRXA1</td>
<td>DBICRYA1</td>
</tr>
</tbody>
</table>

Informatica strongly recommends using SMP/E to install the DBRC modifications. Using SMP/E instead of manual link-edits ensures that the appropriate modules are included when you apply IMS maintenance and prevents any interruption in change data capture operation.

PowerExchange provides a sample job to use SMP/E called CRGUMOD in `hlq.SAMPLIB`. This sample job contains two SMP/E USERMODs:

- **USERMOD MODDBI1** includes DBICRXvr from `hlq.CRG.LOAD` and DSPCRTR0 from the IMS RESLIB to create the DBICRYvr load module in the IMS RESLIB.
- **USERMOD MODDBI2** includes DBICRT00 from `hlq.CRG.LOAD` to replace DSPCRTR0 in the IMS RESLIB.

**Warning:** A full IMS SYSGEN will regress the PowerExchange modifications to DBRC regardless of whether SMP/E is used or not. Prior to doing the SYSGEN, remove these USERMODs by using member CRGUREM in `hlq.SAMPLIB`. CRGUREM is sample JCL that contains SMP/E RESTORE and REJECT commands. After the SYSGEN, reapply the USERMODs to DBRC before restarting the IMS subsystem.

PowerExchange provides member CRGCLINK in `hlq.SAMPLIB`, which can be used instead of the SMP/E USERMODs. This sample JCL manually link-edits the DBICRXvr and the DBICRT00 modules to create the necessary combination load modules. The job places the resulting load modules in `hlq.CRG.LOAD`.

**Note:** The CRGCLINK JCL exists to allow temporary installation without modifying the IMS RESLIB. This JCL is useful for tests such as a proof of concept. Use the SMP/E method for permanent installation of the modifications.

**Related Topics:**

- “Compatibility with BMC Software Products” on page 212
- “Configuring IMS Region JCL” on page 214
Configuring IMS Region JCL

You must modify IMS JCL before you can activate the IMS synchronous ECCR.

Perform the following configuration tasks:
1. Verify the installed version of the BMC products.
2. Add the CRG.LOAD library to the IMS region JCL.
3. Add the CRG.LOAD library to the DBRC JCL.
4. Add the remaining PowerExchange libraries.
5. Configure the IMS external subsystem.
6. Provide access to the external subsystem modules.

RELATED TOPICS:
- “Verifying Installed Version of BMC Products” on page 214
- “Adding the CRG.LOAD Library to the IMS Region JCL” on page 214
- “Adding the CRG.LOAD Library to the DBRC JCL” on page 215
- “Adding Remaining PowerExchange Libraries ” on page 215
- “Configuring the IMS External Subsystem” on page 215
- “Providing Access to the External Subsystem Modules” on page 217

Verifying Installed Version of BMC Products

PowerExchange has minimum version requirements for the BMC Software products CHANGE RECORDING FACILITY and DATABASE INTEGRITY PLUS. If you have these products, verify that the installed version is the same as or later than the version that PowerExchange requires.

Then, perform one of the following actions:
- If the installed version is earlier than the recommended version, upgrade before proceeding.
- If the installed version meets the minimum requirements, add the remaining PowerExchange libraries.

RELATED TOPICS:
- “Compatibility with BMC Software Products” on page 212
- “Adding Remaining PowerExchange Libraries ” on page 215

Adding the CRG.LOAD Library to the IMS Region JCL

Complete this step if you do not have BMC Software product CHANGE RECORDING FACILITY installed. Add hlq.CRG.LOAD to the STEPLIB concatenation for the following IMS region JCL:

- IMS control region
- DBCTL region
- DL/I and DBB batch regions

This library must precede the IMS RESLIB. For example:

```
//STEPLIB DD DISP=SHR,DSN=hlq.CRG.LOAD
// DD DISP=SHR,DSN=IMS.SDFSRESL
```
Adding the CRG.LOAD Library to the DBRC JCL

Complete this step if you do not have the DATABASE INTEGRITY PLUS product installed. Verify that the PowerExchange modifications have been applied to DBRC. If not, configure the IMS DBRC before proceeding.

If the DBRC modifications are installed, add hlq.CRG.LOAD to the STEPLIB concatenation in the DBRC region JCL. This library must precede the IMS RESLIB. For example:

```
//STEPLIB DD DISP=SHR,DSN=hlq.CRG.LOAD
// DD DISP=SHR,DSN=IMS.SDFSRESL
```

Alternatively, you can customize and execute member DBICOPY in the hlq.SAMPLIB library. DBICOPY copies the required DATABASE INTEGRITY PLUS load modules from the hlq.CRG.LOAD to the IMS RESLIB.

**RELATED TOPICS:**

* "Configuring IMS DBRC" on page 213

Adding Remaining PowerExchange Libraries

You must add the EDMPARMS DD statement and update the STEPLIB DD statement with the PowerExchange hlq.LOAD library in the following IMS region JCL:

- IMS control region
- DBCTL region
- DBRC region
- DL/I and DBB batch regions

The EDMPARMS DD statement references the PowerExchange USERLIB data set containing the EDMSDIR module. For example:

```
//EDMPARMS DD DISP=SHR,DSN=hlq.logger_id.USERLIB
```

Add hlq.LOAD to the STEPLIB concatenation. This library must precede the IMS RESLIB. For example:

```
//STEPLIB DD DISP=SHR,DSN=hlq.CRG.LOAD
// DD DISP=SHR,DSN=IMS.SDFSRESL
// DD DISP=SHR,DSN=hlq.LOAD
```

Alternatively, you can copy the entire hlq.LOAD library into an existing library in the STEPLIB concatenation.

Configuring the IMS External Subsystem

The IMS synchronous ECCR operates as an IMS external subsystem.

When you configure the IMS external subsystem, specify a command recognition character (CRC) so that you can communicate with the IMS synchronous ECCR by using IMS /SSR commands.
To configure the external subsystem, perform the following tasks:

- If you do not have any external subsystems defined, add the SSM parameter to the EXEC statement in the IMS region JCL. Alternatively, specify the SSM parameter in the DFSPBxxx member, where xxx is the RGSUF value in the IMS region JCL.

- Add or modify the member in the IMS procedure library that defines the external subsystem. The member name must be the four-character IMS system ID followed by the value of the SSM parameter.
  - If you also specify the SSM parameter in MPP or BMP regions, change the members that contain the external subsystem definitions for both the control region and the dependent regions.
  - If you use the SSM parameter in the IMS control region, all the MPP and BMP dependent regions have access to the subsystems defined in the member. If you plan to use SSM parameter in both the IMS control region and the dependent regions, you must change both SSM members because the dependent region only has access to the subsystems that are defined in both members.

- Do not include the external subsystem in any SSM member used by DL/I batch procedures and jobs.

- Within the member, you can use positional parameters to define the external subsystem. The following table describes these parameters. Separate the parameters with a comma (,).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSN</td>
<td>Yes</td>
<td>Alphanumeric MVS subsystem name for the external subsystem. This name can be up to four characters long. It must match the value of the PowerExchange Agent AgentID configuration parameter.</td>
</tr>
<tr>
<td>LIT</td>
<td>Yes</td>
<td>Alphanumeric parameter that specifies the language interface token. This value can be up to four characters long. It must match the value of the PowerExchange Agent AgentID configuration parameter. SSN and LIT must have the same value.</td>
</tr>
<tr>
<td>ESMT</td>
<td>Yes</td>
<td>Alphanumeric parameter that specifies the name of the external subsystem module table. This value must be EDMCESMT.</td>
</tr>
<tr>
<td>RTT</td>
<td>No</td>
<td>Alphanumeric parameter that specifies the name of the resource translation table. PowerExchange does not use this field. Because the fields are positional, you must include a comma as a placeholder for this field.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| REO       | Yes      | One character region error option code. The option specified determines the action IMS takes when an application program issues a request for external subsystem services before connection to the external subsystem is complete or if problems are encountered with the external subsystem. Valid values are:  
- R, the default  
- Q  
- A  
PowerExchange requires A, which means that IMS abnormally terminates the application program with an abend code of U3047 and discards the transaction input. |
| CRC       | No       | One character command recognition character that allows external subsystem commands from IMS terminals or automated operator interface (AOI) applications. Any EBCDIC value except ‘/’ is permitted. The ‘/’ character is reserved for IMS. Issue external subsystem commands by entering /SSR, followed by the command recognition character specified here, followed by the external subsystem command. Note: The external subsystem may require IMS user IDs and LTERM names to allow authorization for issuing external subsystem commands. For information on command authorization requirements, see the IBM IMS documentation. PowerExchange provides four IMS external subsystem commands. |

The following example shows the fields that define the external subsystem for the IMS synchronous ECCR using the positional format:

PWXA, PWXA, EDMCESMT, , A, X

In this example, the PowerExchange AgentID is PWXA, the required REO value is A, and the CRC selected for the external subsystem commands is X.

The following example specifies an equivalent statement for the external subsystem using the keyword format:

SST=DB2, SSN=PWX, LIT=PWX, ESMT=EDMCSTM, CRC=X

You must specify SST=DB2 when using the keyword format.

Providing Access to the External Subsystem Modules

The IMS synchronous ECCR requires access to the IMS external subsystem modules in the IMS dependent regions. The DFSESL DD statement specifies the library that contains the external subsystem modules. At minimum, the DD statement must contain the following libraries:

- IMS RESLIB  
- PowerExchange hlq.LOAD

You do not need to add hlq.logger_name.USERLIB or hlq.CRG.LOAD to the DFSESL concatenation. All libraries in the DFSESL concatenation must be APF-authorized.

The IMS synchronous ECCR concatenates the data sets in the DFSESL DD statement in the control region and the data sets in the ESLLIB parameter to the data sets specified in the DFSESL DD statement in the dependent region. If necessary, the IMS synchronous ECCR allocates the DFSESL DD statement in the dependent region.
Use one or more of the following methods, which are listed in the order of search, to construct the DFSESL concatenation for the dependent regions:

- Include the DFSESL DD statement in the JCL of any IMS MPP and BMP dependent regions that update databases registered for capture.
- Include the DFSESL DD statement in the IMS control region JCL.
- Specify the libraries in the ESLLIB parameter of the EDMSDIR default options module.

The EDMSDIR module is created during the installation of PowerExchange. If you change the ESLLIB parameter in EDMSDIR, reassemble and reliant the module by editing and running the job in the XICDC600 member of hlq.RUNLIB. In order for the IMS synchronous ECCR to use the updated EDMSDIR, you must stop and restart the affected IMS online regions.

The following example demonstrates a DFSESL concatenation constructed by the IMS synchronous ECCR. In this example, the following statement is specified in the IMS control region:

```
/DFSESL DD DSN=IMS.SDFSRESL,DISP=SHR
```

The EDMSDIR specifies ESLLIB=(hlq.LOAD). The dependent region contains no DFSESL DD statement. The IMS synchronous ECCR concatenates this information to produce the following DFSESL concatenation in the IMS dependent region:

```
/DFSESL DD DSN=IMS.SDFSRESL,DISP=SHR
// DDF DSN=hlq.LOAD,DISP=SHR
```

### MVS LNKLST Concatenation

Informatica strongly recommends against including the PowerExchange load libraries in the MVS LNKLST concatenation as unexpected job abends can occur. For example, IMS jobs that start prior to the PowerExchange Logger and PowerExchange Agent address spaces initializing may fail.

If the PowerExchange hlq.LOAD and hlq.CRG.LOAD libraries are included in the LNKLST concatenation, then:

- You must include the IMS RESLIB and it must be included after hlq.CRG.LOAD.
- The library containing EDMSDIR must be included.
- EDMSDIR should specify the option CCERR=CONT as access to any IMS database causes PowerExchange software to get control. If CCERR=ABEND is coded in EDMSDIR, access fails if the PowerExchange Agent is not active.

Source for EDMSDIR is supplied in member XICDC600 in the hlq.RUNLIB library. Change and rerun this job if changing the CCERR parameter is necessary. To use CCERR=ABEND, add the EDMPARMS DD in any batch job that updates IMS files to be captured.

If you have added the hlq.LOAD library to the LNKLST concatenation, you can stop an ECCR from capturing changes for a specific job by including the following DD statement:

```
//EDMNOCAP DD DUMMY
```

### Activating the IMS Synchronous ECCR

Before you activate the IMS synchronous ECCR, you must complete the customization by configuring the IMS DBRC and configuring the IMS region JCL.
In addition, review the following considerations:

- If you activate the ECCR and open a database before you activate the capture registrations, you must close the database by using the IMS DBR command and reopen using the IMS START command to capture changes.
- The IMS synchronous ECCR activates in IMS regions containing the PowerExchange modules in the STEPLIB concatenation. You can prevent the ECCR from capturing changes in a specific job or region by adding the following DD statement to that JCL:

  ```plaintext
  //EDMNOCAP DD DUMMY
  ```

To activate the IMS synchronous ECCR:

1. Start the PowerExchange tasks. Start the PowerExchange Listener, PowerExchange Agent, and the PowerExchange Loggers. These tasks must be active before the IMS subsystem is started or no changed data is captured.

2. **Start the IMS subsystem.** The IMS synchronous ECCR starts during the IMS subsystem initialization and generates a report beginning with message PWXEDM172852I in the EDMMSG SYSOUT data set. The report lists the default options that are in effect. If the IMS synchronous ECCR is running in an online region, the report also contains allocation options for the DFSESL DD statement.

3. **Verify activation.** Check the system messages to verify that the IMS synchronous ECCR activated. The following messages are issued when using the PowerExchange hlq.CRG.LOAD. The messages issued may differ slightly when using the BMC Software DATABASE INTEGRITY PLUS and CHANGE RECORDING FACILITY products.

- In the DBRC region, verify that the job log (JESMGLG) contains the following messages:

  ```plaintext
  BMC27001 NO VALID DB1 PASSWORD FOUND
  BMC440011 DI+ INITIALIZATION COMPLETE
  BMC440081 DI+ LABEL PROCESSING SUSPENDED
  DFS36313 - DRC TCB INITIALIZATION COMPLETE  imsid
  ```

  The variable `imsid` is the IMS subsystem name.

  Message BMC440011 indicates that the DBRC modification required by the IMS synchronous ECCR requires is installed.

- In the IMS control region, verify that the job log (JESMSGLG) contains the following messages:

  ```plaintext
  *DFS08001I AWAITING NOTIFICATION FROM SUBSYS ssid  imsid
  BMC250011I CRF V4600 12/21/07 INITIALIZATION COMPLETED, RC=0, RSN=00000000
  BMC9048W CHANGE RECORDING FACILITY COMPONENT NOT INSTALLED
  F imsid_job,SNSTimsssid
  ```

  The variable `ssid` is the IMS external subsystem and `imsid` is the IMS subsystem name.

  You can ignore message BMC90488W. Message BMC250011I indicates that the CHANGE RECORDING FACILITY (CRF) product required by the IMS synchronous ECCR has initialized. PowerExchange generates the MVS MODIFY command following CRF activation to notify IMS that the external subsystem is active and ready to connect.

  The following messages in the EDMMSG SYSOUT data set indicate that the IMS synchronous ECCR connected successfully to the PowerExchange Logger and completed initialization:

  ```plaintext
  PWXEDM1728181 Joined XCF group 'logger_id' as member 'imsid'
  PWXEDM1728411 EDM ECCR imsid connected to EDM logger logger_id, Log RBA=x'0000000011680000'
  PWXEDM1728521 DFSESL DD allocation options:
    DSNs to allocate to DFSESL DD. . . . . . . : user.data.set1
    : IMS910.DFSFESREL
    : DSN910.DSNLOAD
    : user.data.set2
  PWXEDM1728201 Change Capture initialized for IMS Online on V9.1.0 - imsid
  ```

**RELATED TOPICS:**

- “Configuring IMS DBRC” on page 213
- “Configuring IMS Region JCL” on page 214
Output from the IMS ECCR

The IMS synchronous ECCR generates a report at startup that displays the default options that are in effect for the ECCR. When the IMS synchronous ECCR ends, the reports displays the number of captured changes for each segment of each database. These reports are written to the EDMMSG SYSOUT data set in the IMS region.

The following example shows sample messages from EDMMSG for an IMS control region:

```
PWXEDM1728521 Options in effect:
Load Library containing EDMSDIR. . . . . : EDM.QA.I24L.USERLIB
EDMSDIR assembly date/time . . . . . . . : 20071023 19.54
Product distribution date. . . . . . . . . : 20060831
Product distribution level . . . . . . . . : 2.4.05
Agent Id . . . . . . . . . . . . . . . . : I24A
Logger Id . . . . . . . . . . . . . . . . : I24L
SYSOUT class . . . . . . . . . . . . . . . : *
Action if ECCR error encountered . . . . : Abend

PWXEDM1728201 Change Capture initialized for IMS Online on V9.1.0 - EDMA

PWXEDM1728081 Change Capture active for IMS DBD/DSN DBLOG5OF/EDM.EDM.SSLOG5
Segment-DB#AASEG SegCode=1 Edition=000000000000000000 EDMNAME=IMS.DBLOG5OF.DBAASEG
Segment-DB#BASEG SegCode=2 Edition=000000000000000000 EDMNAME=IMS.DBLOG5OF.DBBASEG
Segment-DB#CASEG SegCode=3 Edition=000000000000000000 EDMNAME=IMS.DBLOG5OF.DBBCASEG
Segment-DB#BESEG SegCode=4 Edition=000000000000000000 EDMNAME=IMS.DBLOG5OF.DBEBASEG

PWXEDM1728531 Change Capture counts for IMS DBD DBLOG5OF
Segment-DB#AASEG ISRT-0 REPL-0 DLET-0
Segment-DB#BASEG ISRT-0 REPL-0 DLET-0
Segment-DB#CASEG ISRT-0 REPL-0 DLET-0
Segment-DB#BESEG ISRT-0 REPL-0 DLET-0

PWXEDM1728411 EDM ECCR EDM disconnected from EDM Logger I24L, Log RBA-X'0000000011680000'
PWXEDM1728181 Left XCF group 'I24L' as member 'EDMA'
PWXEDM1728201 EDM ECCR sent 0 records to Logger I24L (0 change records)
```

Managing IMS Synchronous CDC

This section describes how to refresh the IMS synchronous ECCR, issue commands to control ECCR processing, and stop IMS change capture processing. It also includes recovery considerations.

Refreshing the IMS Synchronous ECCR

When you register databases for CDC, refresh the IMS ECCR to activate the new or changed capture registration.

- Close, and reopen the IMS database. For more information, see your IBM documentation.

Controlling the IMS Synchronous ECCR

You can use the following types of commands to control IMS synchronous ECCR processing:

- IMS commands to stop and start the external subsystem and to display the names of the databases files registered for changed data capture.
- IMS external subsystem commands, which are routed through the command /SSR to the ECCR for processing.
IMS Console Commands

You can use the following IMS console commands to control the IMS external subsystem.

Controlling the IMS External Subsystem

The following table describes IMS commands that you can use to control IMS synchronous ECCR processing:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| /STOP SUBSYS ssid   | Stops the IMS synchronous ECCR external subsystem specified by ssid. When you stop the external subsystem, PowerExchange takes the following action based on the CCERR setting in the EDMSDIR options module:  
                      - If CCERR=CONT, the IMS synchronous ECCR ceases logging changes in the PowerExchange Logger. Transactions run normally.  
                      - If CCERR=ABEND, transactions that process segments registered for capture abend with a U4094. You set the CCERR parameter in the XICDC600 job when installing PowerExchange. |
| /START SUBSYS ssid  | Starts the IMS synchronous ECCR external subsystem, ssid. Change data capture begins when the START command is completed.                     |
| /DISPLAY SUBSYS ssid| Displays the status of the IMS external subsystem specified by ssid.                                                                          |

**RELATED TOPICS:**

- “EDMSDIR Module Options” on page 22

IMS External Subsystem Commands

You can issue external subsystem commands through the subsystem routing command, /SSR. Use these commands to dynamically change how the IMS control region reacts when the IMS synchronous ECCR cannot capture changes for an IMS database or to produce snapshot reports of the IMS synchronous ECCR.

The following table describes the external subsystem commands. In the commands, substitute the variable x with the command recognition character (CRC) that you specified when defining the external IMS subsystem:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
</table>
| /SSR xEDP-ABORT     | Overrides the CCERR parameter option of the EDMSDIR module to ABEND. The ABEND action:  
                      - Causes transactions to pseudo-abend with a message U4094 if the external subsystem is stopped or if the PowerExchange Logger terminates.  
                      - Remains in effect until a process or command terminates the IMS control region, or until another SSR command supersedes the current command. |
| /SSR xEDP-CONTINUE   | Overrides the CCERR parameter option of the EDMSDIR module to CONT  
                      The CONT action:  
                      - Instructs the IMS ECCR to take no action if the PowerExchange Logger or the external subsystem have been terminated. If these conditions occur, changes are lost.  
                      - Remains in effect until a process or command terminates the IMS control region, or until another SSR command supersedes the current command. |
IMS Command Examples

The following examples demonstrate how to issue IMS external subsystem commands.

Example 1. /DISPLAY SUBSYS

This example shows the /DISPLAY SUBSYS command and resulting output for an IMS external subsystem called I24A:

```
R 89,/DISPLAY SUBSYS I24A
IEE6001 REPLY TO 89 IS;/DISPLAY SUBSYS I24A
DFS0001 SUBSYS CRC REGID PROGRAM LTERM STATUS EDMA
DFS0001 I24A # CONN EDMA
DFS0001 *07/10/211738* EDMA
```

The output shows the command recognition character (CRC) assigned to the I24A external subsystem. This CRC is needed when issuing /SSR commands to the IMS synchronous ECCR external subsystem.

Example 2. /SSR xEDP-ABORT

This example shows the EDP-ABORT /SSR command and resulting output:

```
R 93,/SSR #EDP-ABORT.
DFS0581 SSR COMMAND COMPLETED EDMA
PNXEDM1728890 Action if ECCR error encountered has been set to ABORT
```

This command changes the CCERR option to ABEND.

Example 3. /SSR xEDP-CONTINUE

This example shows the EDP-CONTINUE /SSR command and resulting output:

```
R 94,/SSR #EDP-CONTINUE.
DFS0581 SSR COMMAND COMPLETED EDMA
PNXEDM1728890 Action if ECCR error encountered has been set to CONTINUE
```

This command changes the CCERR option to CONT.

Example 4. /SSR xEDP-STATWTO

This example shows the EDP-STATWTO1 /SSR command and resulting output:

```
R 95,/SSR #EDP-STATWTO1.
DFS0581 SSR COMMAND COMPLETED EDMA
PNXEDM172890W There are no open databases registered for capture
```

This example indicates that no capture registrations exist for any open databases.
Example 5. /SSR xEDP-STAT

This example shows the output from the EDP-STAT /SSR command, which is written to the EDMMSG SYSOUT data set:

```
FWXEDM1728531 Change Capture counts for IMS DBD DBLOG5OF
Segment=DB#ASEG ISRT=0 REPL=0 DLET=0
Segment=DB#BASEG ISRT=0 REPL=0 DLET=0
Segment=DB#CASEG ISRT=0 REPL=0 DLET=0
Segment=DB#RSEG ISRT=0 REPL=0 DLET=0
```

In this example, a single database with four segments is registered for capture. The IMS synchronous ECCR has captured no changes for this database.

Stopping Change Data Capture for IMS Sources

You can stop change data capture for IMS sources at different levels. For example, you can stop capture for a specific database or stop all synchronous capture activity of the ECCR.

The following table lists the methods of stopping the change capture by level:

<table>
<thead>
<tr>
<th>Level at Which to Stop Change Capture</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>An IMS database</td>
<td>Close the database or data set. Alternatively, you can also stop the IMS synchronous ECCR.</td>
</tr>
<tr>
<td>Synchronous capture</td>
<td>Stop the IMS synchronous ECCR.</td>
</tr>
<tr>
<td>Log-based capture</td>
<td>Stop the log-based ECCR.</td>
</tr>
<tr>
<td>Any registered data object</td>
<td>Deactivate or delete the corresponding data-resource registration. Then, close or stop the database or data set, as appropriate, and refresh the ECCR.</td>
</tr>
</tbody>
</table>

RELATED TOPICS:

- “Closing an IMS Database” on page 223
- “Stopping the IMS Synchronous ECCR” on page 223
- “Stopping the IMS Log-Based ECCR” on page 224

Closing an IMS Database

When you close a source database or data set, the IMS synchronous ECCR no longer captures changes associated with that source. For information about closing a database or data set, see the appropriate IBM documentation.

Stopping the IMS Synchronous ECCR

Stopping the IMS ECCR requires that you issue the IMS external subsystem command:

```
/STOP SUBSYS ssid
```

The variable `ssid` designates the subsystem ID.

Before you can issue the IMS external command, you must set the value for the option CCERR to CONTINUE. You can also change the value by issuing the EDP_CONTINUE command of the IMS synchronous ECCR external subsystem.
Stopping the IMS Log-Based ECCR

Stopping the IMS log-based ECCR stops the capture of changes. The ECCR disconnects from the PowerExchange Logger and displays a set of messages, including the number and type of changes captured since the last time the data set was opened. The following code shows a sample:

```plaintext
PWXEDM1728091 Change Capture counts for IMLIMSIIMSVXP1100000: Insert=3, Update=0,
Delete=0
RBA=X'00000004EA570000'
PWXEDM1728181 Left XCF group 'DOCL' as member 'DTLUSRIM'
PWXEDM1728291 EDM ECCR sent 3 records to Logger DOCL (3 change records)
```

Deactivating or Deleting Registrations

Use PowerExchange Navigator to delete or deactivate PowerExchange registrations.

Then close and reopen the IMS databases.

Application Recovery Considerations

The following section describes batch execution and recovery issues to consider when using the IMS synchronous ECCR. You may have to change existing operational recovery procedures to accommodate PowerExchange CDC.

Using Point-in-Time Recovery

Point-in-time recovery for IMS databases invalidates any change data captured to the PowerExchange Logger.

If point-in-time recovery is necessary, complete the following tasks:

1. Stop all PowerCenter sessions extracting change data for the source database.
2. Recover the source database to the correct point-in-time.
   Leave the database in read-only mode.
3. Rematerialize all targets that apply change data from that source database.
4. Use the DTLUAPPL utility to generate a new restart token for all extractions using the source database. Then, update the restart token file of all PowerCenter sessions extracting change data for the source database with the new restart token.
5. Reset the source database to read-write mode and resume normal operation.
6. Cold start all affected PowerCenter sessions.

MVS Checkpoint/Restart

You cannot use MVS Checkpoint/Restart in an IMS synchronous ECCR job.

IMS Batch Backout Utility

If a DL/I batch jobs fails and the IMS Batch Backout utility is used, consider the following:

- If the DL/I job step does not issue IMS checkpoints, recover the IMS database by:
  - Executing the Batch Backout utility.
  - Restoring an image copy taken prior to the failed job executing.
- If the DL/I job step issues IMS checkpoints:
  - Run the Batch Backout utility to remove uncommitted records caused by the failure of the job step. Using an image copy or point-in-time recovery requires synchronizing the source and target databases again.
  - Resume execution of the job step from the failed checkpoint.
Managing IMS Schema Changes

If changes are made to an IMS database that is registered for capture, use the following procedure to allow the new format of data to be captured while retaining the ability to access historically captured data.

1. Stop all update activity against the IMS database.
2. Ensure that PowerExchange has processed all changes that occurred under the old schema.
3. Change the IMS database.
4. Create a new PowerExchange registration that reflects the schema changes.
5. Restart PowerExchange processing.
6. Allow update activity to the IMS database to resume.
Part IV: Change Data Extraction

This part contains the following chapters:

- Introduction to Change Data Extraction, 227
- Extracting Change Data, 247
- Managing Change Data Extractions, 262
- Monitoring and Tuning Options, 270
CHAPTER 15

Introduction to Change Data Extraction

This chapter includes the following topics:

- Change Data Extraction Overview, 227
- Extraction Modes, 228
- PowerExchange-Generated Columns in Extraction Maps, 228
- Restart Tokens and the Restart Token File, 231
- Recovery and Restart Processing for CDC Sessions, 233
- Group Source Processing in PowerExchange, 238
- Commit Processing with PWXPC, 241
- Offload Processing, 246

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 228
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 PowerExchange Logger for MVS log files 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 on MVS 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 MVS.
- 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 284
- “Extracting Change Data Captured on a Remote System” on page 290

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 z/OS 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>
</tbody>
</table>
| DTL__CAPXRESTART2 | 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. | VARBIN   | 255    |
| DTL_CAPXRRN     | For DB2 on i5/OS only, the relative record number. | DECIMAL  | 10     |
| DTL__CAPXUOW    | A binary value that represents the position in the change stream of the start of the UOW for the change record. | VARBIN   | 255    |
| DTL__CAPXUSER   | 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. | VARCHAR  | 255    |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Datatype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTL__CAPXTIMESTAMP</td>
<td>The timestamp for when the change was made to the data source, as recorded by the source DBMS in the following format: YYYYMMDDhhmmsnnnnnn. Where: - YYYYMMDD is the date in year (YYYY), month (MM), and day (DD) format. - hhmmsnnnnnn is the time in hours (hh), minutes (mm), seconds (ss), and microseconds (nnnnnn) format. <strong>Note:</strong> Oracle does not support microseconds in the timestamp.</td>
<td>CHAR</td>
<td>20</td>
</tr>
<tr>
<td>DTL__CAPXACTION</td>
<td>A single character that indicates the type of change operation. Valid values are: - I. INSERT operation. - D. DELETE operation. - U. UPDATE operation.</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>DTL__CAPXCASDELIND</td>
<td>For DB2 for z/OS sources only, a single character that indicates whether DB2 has deleted the row because the table specifies the ON DELETE CASCADE clause. Valid values are: - Y. Indicates that DB2 deleted this row because of a cascade delete rule. - N. Indicates that DB2 did not delete this row because of a cascade delete rule.</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>DTL__BI_columnname</td>
<td>For UPDATE operations, the value of the before image of the selected column in the change record.</td>
<td>Datatype of the source column</td>
<td>Length of the source column</td>
</tr>
<tr>
<td>DTL__CI_columnname</td>
<td>For UPDATE operations, a single character that indicates whether the selected column was changed. Valid values are: - Y. Indicates that the column changed. - N. Indicates that the column did not changed. - Null value. Indicates an INSERT or DELETE operation.</td>
<td>CHAR</td>
<td>1</td>
</tr>
<tr>
<td>DTL__columnname_CNT</td>
<td>Binary count column. PowerExchange generates this column for variable length columns of types VARCHAR and VARBIN to determine the length of the column during change data extraction processing. <strong>Note:</strong> By default, binary count columns are not selected in an extraction map. You must edit an extraction map to select these columns.</td>
<td>NUM32U</td>
<td>0</td>
</tr>
<tr>
<td>DTL__columnname_IND</td>
<td>Null indicator column. PowerExchange generates this column for nullable columns to indicate the nullable value for the column. <strong>Note:</strong> By default, null indicator columns are not selected in an extraction map. You must edit an extraction map to select these columns.</td>
<td>BIN</td>
<td>1</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.

On MVS, PowerExchange generates an event mark in the PowerExchange Logger log files that can be used to create restart tokens for the following sources:

- **DB2 for z/OS.** The DB2 ECCR generates an event mark when it reads a quiesce point from the DB2 logs. DB2 creates quiesce points when you use the DB2 QUIESCE utility.
- **IMS.** The IMS log-based ECCR generates an event mark when it reads the records that the DTLCUIML utility creates in the IMS logs.
- **Adabas.** The Adabas ECCR generates an event mark when it finishes reading an Adabas PLOG data set.

The PowerExchange Logger writes message PWXEDM172774I, which contains the restart point of the event mark, to its EDMMSG data set. The event marks that these ECCRs generate are the same as those that the EDMXLUTL utility creates.

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.

---

**Recovery and Restart Processing for CDC Sessions**

233
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>
<tbody>
<tr>
<td>All MVS sources</td>
<td>Oldest condense file, as recorded in the CDCT.</td>
<td>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.</td>
</tr>
<tr>
<td>DB2 for i5/OS</td>
<td>Oldest condense file, as recorded in the CDCT.</td>
<td>Oldest journal receiver still attached on the journal receiver chain.</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>Oldest PowerExchange Logger for Linux, UNIX, and Windows log file, as recorded in the CDCT.</td>
<td>Current log position at the time the PowerExchange capture catalog was created.</td>
</tr>
</tbody>
</table>
PowerExchange uses the default restart point only if all sources in a CDC session have null restart tokens. If some sources have non-null restart tokens, PWXPC assigns the oldest restart point from those tokens to any sources for which no restart tokens are specified.

For example, a new CDC session contains the sources A, B, and C. The restart token file contains restart tokens for sources A and B. The restart point for source A is older than that for source B. Source C does not have existing or supplied restart tokens. Because some sources in the CDC session have explicit restart points, PWXPC does not assign null restart tokens to source C. Instead, PWXPC assigns the restart point for source A to source C because this restart point is the oldest one supplied.

**Determining the Restart Tokens for Cold Start Processing**

When you cold start a CDC session, PWXPC uses the restart token file to determine the restart tokens for all sources. PWXPC ignores any entries in the state tables or state file for the sources in the CDC session.

More specifically, PWXPC uses one of the following methods to determine the restart tokens:

- If the restart token file is empty or does not exist, PWXPC assigns null restart tokens to all sources in the CDC session.
- If the restart token file contains only explicit override statements, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the oldest supplied restart point to any sources for which an explicit override statement was not specified.
- If the restart token file contains only the special override statement, PWXPC assigns the restart tokens in the special override statement to all sources.
- If the restart token file contains a special override statement and explicit override statements, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the restart tokens in the special override statement to all remaining sources.

**Determining the Restart Tokens for Warm Start Processing**

When you warm start a CDC session, uses the state tables and state file, in conjunction with restart token file, to determine the restart tokens for all sources.
More specifically, PWXPC uses one of the following methods to determine the restart tokens:

- If the restart token file is empty or does not exist and there is no matching entry in a state table or state file, PWXPC assigns null restart tokens to all sources in the session.
- If the restart token file is empty or does not exist and if some but not all sources have a matching entry in a state table or a state file, PWXPC performs the following processing:
  - Assigns any restart tokens found in a state table and state file to the appropriate sources.
  - Assigns the oldest available restart point to all sources that do not have restart tokens.
- If the restart token file is empty or does not exist and if all sources have an entry in a state table or state file, PWXPC uses the restart tokens from the state tables or state file.
- If the restart token file contains explicit override statements and no sources have a matching entry in a state table or no state file, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the oldest supplied restart point to all sources that do not have restart tokens.
- If the restart token file contains explicit override statements and if some but not all sources have a matching entry in a state table or a state file, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns restart tokens from a state table or state file to the appropriate sources, provided that the tokens have not been supplied in the restart token file.
  - Assigns the oldest available restart point to all sources that do not have restart tokens supplied in the restart token file or from a state table or state file.
- If the restart token file contains explicit override statements and if all sources have an entry in a state table or a state file, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the restart tokens from state tables or the state file to all remaining sources that do not have restart tokens supplied in the restart token file.
- If the restart token file contains only the special override statement, PWXPC assigns the restart tokens in the special override statement to all sources.
- If the restart token file contains a special override statement and explicit override statements, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the restart tokens in the special override statement to all remaining sources.

**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:

**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 255
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 milli-seconds</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 milli-seconds**
- **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 milli-seconds**, 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 Integration Service machine, then multithreaded processing may provide increased throughput.
Chapter 16

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 to the 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 250
- "Creating Restart Tokens for Extractions" on page 257
- "Configuring the Restart Token File" on page 258
- "Monitoring and Tuning Options" on page 270
- "Starting PowerCenter CDC Sessions" on page 262
- "Testing Change Data Extraction" on page 248

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>- CAPXRT. Real-time extraction mode or continuous extraction mode.</td>
</tr>
<tr>
<td></td>
<td>- CAPX. 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>
<tr>
<td></td>
<td>In the statement, a table is identified in the following format:</td>
</tr>
<tr>
<td></td>
<td>Schema.RegName_TableName</td>
</tr>
<tr>
<td></td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td>- Schema is schema for the extraction map.</td>
</tr>
<tr>
<td></td>
<td>- RegName is the name of the capture registration that corresponds to the extraction map.</td>
</tr>
<tr>
<td></td>
<td>- TableName is the table name of the data source.</td>
</tr>
</tbody>
</table>

**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>
</tbody>
</table>
### Configuring Application Connection Attributes

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

#### Image Type

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 the after-image data for updated row as a SQL UPDATE operation.

You can also configure one or more data columns in an extraction map with before-image (BI) columns. Use the PowerExchange Navigator to update the extraction map with before-image columns, which adds additional columns to the extraction map with the name of DTL\_BI\_columnname. If you use BI columns, select **AI** for the **Image Type** attribute. PowerExchange then includes before-image data in any BI columns, along with the after-image data, in a single SQL UPDATE operation.

When you configure BI columns, you can make decisions about UPDATE operations in a mapping because the before and after-image data is contained in a single record. For example, you can use BI columns to handle update operations that change the value of a key column of a row. Some relational databases, such as DB2 for z/OS, allow update operations to key columns. The RDBMS understands that this operation is equivalent to deleting the row and then re-adding it with a new primary key and logs the change as an update.
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 use 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:

```plaintext
[PWXPC_10072] [INFO] [CCDCdispatcher] 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 the 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 241

## 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.

You can also construct restart tokens by using the RBA or LRSN of an event mark record that is written to the PowerExchange Logger log files. Use the EDMXLUTL utility to generates event marks. Certain PowerExchange ECCRs also generates event marks in the following situations:

- **DB2 ECCR.** Generates an event mark when it reads a quiesce point from the DB2 logs. DB2 creates quiesce points when you use the DB2 QUIESCE utility.
- **IMS log-based ECCR.** Generates an event mark when it reads records that the DTLCUIML utility creates in the IMS logs.
- **Adabas ECCR.** Generates an event mark when it reads an Adabas PLOG data set.

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:

```
excution_map_name=sequence_token
execution_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:

```xml
<!-- Restart Tokens for existing tables -->
restart1=000000AD775600000000000AD7756000000000000
restart2=C1E4E2D3404F000000AD5F2C00000000
<!-- Restart Tokens for the Table: rrtrb0001_RRTB_SRC_001 -->
didsn.rrtrb0001_RRTB_SRC_001=000060D1DB20000000000060D1DB200000000000000000
<!-- Restart Tokens for the Table: rrtrb0001_RRTB_SRC_002 -->
didsn.rrtrb0001_RRTB_SRC_002=0000AD37195000000000000AD3719500000000000000000
<!-- Restart Tokens for the Table: rrtrb0002_RRTB_SRC_002 -->
didsn.rrtrb0002_RRTB_SRC_002=C1E4E2D3404F0000000968FC600000000
<!-- Restart Tokens for the Table: rrtrb0004_RRTB_SRC_004 -->
didsn.rrtrb0004_RRTB_SRC_004=000060D84E780000000000060D84E780000000000000000
<!-- Restart Tokens for the Table: rrtrb0006_RRTB_SRC_006 -->
didsn.rrtrb0006_RRTB_SRC_006=0000AD37195000000000000AD3719500000000000000000
<!-- Restart Tokens for the Table: rrtrb0007_RRTB_SRC_007 -->
didsn.rrtrb0007_RRTB_SRC_007=C1E4E2D3404F0000000968FC600000000
```

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:

```
| Extraction Map Name | Restart Token 1 | Restart Token 2 | Source
|---------------------|----------------|----------------|--------
| didsn.rrtrb0001_RRTB_SRC_001 | 000060D1DB20000000000060D1DB200000000000000000 | C1E4E2D3404F0000000968FC600000000 | Restart file |
| didsn.rrtrb0002_RRTB_SRC_002 | 0000AD37195000000000000AD3719500000000000000000 | C1E4E2D3404F0000000968FC600000000 | Restart file |
| didsn.rrtrb0003_RRTB_SRC_003 | 0000AD37195000000000000AD3719500000000000000000 | C1E4E2D3404F0000000968FC600000000 | Restart file (special override) |
| didsn.rrtrb0004_RRTB_SRC_004 | 000060D84E780000000000060D84E780000000000000000 | C1E4E2D3404F000000060D8E1600000000 | Restart file |
| didsn.rrtrb0005_RRTB_SRC_005 | 0000AD37195000000000000AD3719500000000000000000 | C1E4E2D3404F000000060D8E1600000000 | Restart file (special override) |
| didsn.rrtrb0006_RRTB_SRC_006 | 0000AD37195000000000000AD3719500000000000000000 | C1E4E2D3404F000000060D8E1600000000 | Restart file (special override) |
| didsn.rrtrb0007_RRTB_SRC_007 | 0000AD37195000000000000AD3719500000000000000000 | C1E4E2D3404F000000060D8E1600000000 | Restart file (special override) |
```

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.
Chapter 17

Managing Change Data Extractions

This chapter includes the following topics:
- Starting PowerCenter CDC Sessions, 262
- Stopping PowerCenter CDC Sessions, 264
- Changing PowerCenter CDC Sessions, 266
- Recovering PowerCenter CDC Sessions, 268

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 stoptask 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 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.

5. Edit the restart token file to add the new source and its tokens.

   Add eight zeros to the end of the Sequence value to create the sequence value for the restart token file.

   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 represents the current end of the change stream.

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:

   PWXPC writes the following messages in the session log:

   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:

   The PRINT command produces the following output:

   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.
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]

```
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>dian8.rtt0054_RBTB_SRC_004</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0059_RBTB_SRC_009</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0055_RBTB_SRC_005</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0066_RBTB_SRC_006</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0058_RBTB_SRC_008</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0001_RBTB_SRC_002</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0001_RBTB_SRC_002</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0001_RBTB_SRC_002</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></td>
</tr>
<tr>
<td>dian8.rtt0001_RBTB_SRC_002</td>
<td>00000FC95B410000000000000002E020A00000000FFFFFF</td>
<td>C1E4E2540A0000002181A500000000 GMD storage</td>
<td></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 [00000FC95B41000000000000002E020A00000000FFFFFF]" : Restart 2 [C1E4E2540A0000002181A500000000]
to: "Restart 1 [00000FC95B41000000000000002E020A00000000FFFFFF]" : Restart 2 [C1E4E2540A0000002181A500000000].

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, 270
- Tuning Change Data Extractions, 276
- CDC Offload and Multithreaded Processing, 281

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.
- **LISTTASK command output.** You can use the 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
PRGIN=100
```

When a CDC session that has a session name of `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 greater for **Worker Threads** on the connection. Otherwise, PowerExchange does not use multithreaded processing or produce statistics messages.

Valid values are from 10000 through 50000000.
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 IO time: 4% (avg: 235 min: 51 max: 1021 usecs)
PWX-31257 Parse time: 79% (avg: 4551 min: 4102 max: 5495 usecs)
PWX-31258 External time: 20% (avg: 1145 min: 618 max: 3287 usecs)
PWX-31259 Delay time: 0% (avg: 7 min: 4 max: 165 usecs)
PWX-31254 PowerExchange threading stats for last 100000 rows. Cycle (array) size is 25 rows. 0 out of array occurred.
PWX-31255 Cycle time: 99% (avg: 5706 min: 4735 max: 7790 usecs)
PWX-31256 IO time: 4% (avg: 234 min: 51 max: 950 usecs)
PWX-31257 Parse time: 79% (avg: 4549 min: 4108 max: 5425 usecs)
PWX-31258 External time: 20% (avg: 1144 min: 616 max: 3242 usecs)
PWX-31259 Delay time: 0% (avg: 7 min: 4 max: 115 usecs)
```

### LISTTASK Command Output

Issue the PowerExchange Listener LISTTASK command to display the CDC sessions that are active in the PowerExchange Listener.

You can issue the command from the MVS operator console or an interface such as SDF by using the MVS MODIFY (F) command. Alternatively, issue a pwxcmd listtask command from a Linux, UNIX, or Windows system to a PowerExchange Listener on a z/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 TaskId=1, Partner=10.10.10.01, Port=2480, PwrCntrSess=intserv1/workflow1/cdc_sess1, Application=appl_name1, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=
PWX-00712 TaskId=2, Partner=10.10.10.02, Port=2480, PwrCntrSess=intserv2/workflow2/cdc_sess2, Application=appl_name2, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=
```
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.

**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 \( [\text{restart}1], [\text{restart}2] \) because the UOW Count \( [\text{count}] \) is reached
- PWXPC_10082 [INFO] [CDCDispatcher] raising real-time flush with restart tokens \( [\text{restart}1], [\text{restart}2] \) because Real-time Flush Latency \( [\text{latency}] \) is reached
- PWXPC_12128 [INFO] [CDCDispatcher] raising real-time flush with restart tokens \( [\text{restart}1], [\text{restart}2] \) because the Maximum Rows Per commit \( [\text{count}] \) 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 message displays the source-based commit statistics.

For more information about tuning CDC sessions, see the *PowerCenter Performance Tuning Guide*.

**RELATED TOPICS:**

- “Using Connection Options to Tune CDC Sessions” on page 279
- “Tuning Commit Processing” on page 280
- “Viewing Performance Details in Workflow Monitor” on page 273

**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 10 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 the 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>
<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>
<tr>
<td>Performance Counter Field</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------</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. |
| 3.2 Timestamp On Last Target Commit | The capture timestamp, DTL__CAPXTIMESTAMP, from the last UOW committed to the target. |
### Performance Counter Field

<table>
<thead>
<tr>
<th>Performance Counter Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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:
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.

 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.

 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:

 - MSQ
 - 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.

**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.

**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 DBMOVVER configuration files on all systems.

**RELATED TOPICS:**

- “Using Connection Options to Tune CDC Sessions ” on page 279

## 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>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>Connection Option</td>
<td>Description</td>
<td>Tuning Suggestion</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</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 281.</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 281.</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 280
- “CDC Offload and Multithreaded Processing” on page 281

**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.

  For more information about tuning buffer memory, see the *PowerCenter Performance Tuning Guide*.

- **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
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:

<table>
<thead>
<tr>
<th>Connection Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Specifies 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.</td>
</tr>
</tbody>
</table>
| 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. Select one of the following values:  
- 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 | Specifies 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>
</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 290

**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 MVS 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 MVS 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 are issued. You can issue pwxcmd commands from a Linux, UNIX, or Windows system to a PowerExchange Condense process running on a z/OS system.

This service name must match the 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 is 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 is displayed for the registration group in the PowerExchange Navigator.</td>
</tr>
</tbody>
</table>
| Datacom           | One of the following values:  
|                   | - The MUF Name value that is displayed for the registration group in the PowerExchange Navigator.  
|                   | - For Datacom synchronous CDC, the MUF parameter value in the DTLINPUT data set specified in the MUF JCL.  
<p>|                   | - For Datacom table-based CDC, the REG_MUF parameter value in the ECCRDMP member of the RUNLIB library. |</p>
<table>
<thead>
<tr>
<th>CDC Source Type</th>
<th>DBID Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 for i5/OS</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>Instance</strong> name that is displayed for</td>
</tr>
<tr>
<td></td>
<td>the registration group in the</td>
</tr>
<tr>
<td></td>
<td>PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- The INST parameter value in the</td>
</tr>
<tr>
<td></td>
<td>AS4J CAPI_CONNECTION statement in the</td>
</tr>
<tr>
<td></td>
<td>DBMOVER member of the CFG file.</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>The <strong>Database</strong> name that is displayed 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 <strong>Instance</strong> name that is displayed for</td>
</tr>
<tr>
<td></td>
<td>the registration group in the</td>
</tr>
<tr>
<td></td>
<td>PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- The RN parameter value from the DB2 statement</td>
</tr>
<tr>
<td></td>
<td>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 <strong>LogsId</strong> value that is displayed for</td>
</tr>
<tr>
<td></td>
<td>the registration group in the</td>
</tr>
<tr>
<td></td>
<td>PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- The LOGSID parameter value in the ECCRIDL</td>
</tr>
<tr>
<td></td>
<td>P member of the RUNLIB library.</td>
</tr>
<tr>
<td>IMS</td>
<td>One of the following values:</td>
</tr>
<tr>
<td></td>
<td>- The <strong>IMSID</strong> value that is displayed for</td>
</tr>
<tr>
<td></td>
<td>the registration group in the</td>
</tr>
<tr>
<td></td>
<td>PowerExchange Navigator.</td>
</tr>
<tr>
<td></td>
<td>- For IMS log-based CDC, the first parameter</td>
</tr>
<tr>
<td></td>
<td>of the IMSID statement in the</td>
</tr>
<tr>
<td></td>
<td>CAPTIMS member of the RUNLIB library.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>The <strong>Instance</strong> name that is displayed for the</td>
</tr>
<tr>
<td></td>
<td>registration group in the PowerExchange</td>
</tr>
<tr>
<td></td>
<td>Navigator.</td>
</tr>
<tr>
<td>Oracle</td>
<td>ORCL and UOWC</td>
</tr>
<tr>
<td>VSAM</td>
<td>The <strong>Instance</strong> name that is displayed for the</td>
</tr>
<tr>
<td></td>
<td>registration group in the PowerExchange</td>
</tr>
<tr>
<td></td>
<td>Navigator.</td>
</tr>
</tbody>
</table>

**EPWD**

A deprecated parameter. Use CAPTURE_NODE_EPWD instead. If both CAPTURE_NODE_EPWD and EPWD are specified, CAPTURE_NODE_EPWD takes precedence.

**EXT_CAPT_MASK**

Specifies an existing path and unique prefix to be used for generating the PowerExchange Logger log files.

**PWD**

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

**RESTART_TOKEN** and **SEQUENCE_TOKEN**

Optionally, specifies a restart point for starting change data processing when the PowerExchange Logger is cold started.

The format of the restart tokens varies based on data source type and, if specified, 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 cold started.
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 you capture the change data 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.

**UID**

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

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

288 Chapter 18: Monitoring and Tuning Options
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>MVS sources</td>
<td>LRAP and UOWC</td>
</tr>
<tr>
<td>Oracle</td>
<td>ORCL and UOWC</td>
</tr>
</tbody>
</table>

For MVS sources, remove MVS-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 z/OS source to PowerExchange Logger log files on a remote system, use a PWX DB2zOS 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 MVS by Using Offload Processing - Example

In this example, a CDC session that uses real-time connections to extract change data from an MVS source is changed to use CDC offload processing. The source change data remains on MVS but all column-level and UOW Cleanser processing is moved to the PowerCenter Integration Service machine.

The MVS 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=MV2UOWC,
  TYPE=(UOWC,CAPINAME=M2_LRAP,RSTRADV=600,HMCCACHE=20480,DATACLAS=UOWC))
CAPI_CONNECTION=(NAME=MV2_LRAP,
  TYPE=(LRAP,LOG=MV2L,AGENT=MV2A))
```

To extract change data from MVS by 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 LRAP CAPI_CONNECTION statements from the DBMOVER member on MVS to the dbmover.cfg configuration file on the PowerCenter Integration Service machine. Remove any MVS-specific parameters from the UOWC CAPI_CONNECTION. In this example, the following CAPI_CONNECTION statements are copied into the dbmover.cfg and the DATACLAS parameter is removed:

```
CAPI_CONNECTION=(NAME=MV2UOWC,
    TYPE=(UOWC, CAPINAME=M2_LRAP, RSTRADV=600, MEMCACHE=20480))
CAPI_CONNECTION=(NAME=MV_2_LRAP,
    TYPE=(LRAP, LOG=MV2L, AGENT=MV2A))
```

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 MV2UOWC.
4. Restart the CDC session.

Capturing and Extracting Change Data from MVS on UNIX - Example

In this example, change data for DB2 for z/OS sources is captured on a UNIX machine by the PowerExchange Logger for Linux, UNIX, and Windows. 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 MVS system where the change data was originally captured.

The MVS 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=MV2UOWC,
    TYPE=(UOWC, CAPINAME=M2_LRAP, RSTRADV=600, MEMCACHE=20480, DATACLAS=UOWC))
CAPI_CONNECTION=(NAME=MV_2_LRAP,
    TYPE=(LRAP, LOG=MV2L, AGENT=MV2A))
```

The DB2 subsystem on the MVS system that contains the tables that are registered for capture is called DSN9.

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 MVS on UNIX:

1. Configure the PowerExchange Logger for Linux, UNIX, and Windows on the UNIX system by performing the following actions:
   - Configure the pwxccl.cfg configuration file.
   - Configure the dbmover.cfg configuration file on the PowerExchange Logger machine.

In this example, the dbmover.cfg has the following parameters:

```
/*
 * dbmover.cfg
 */
LISTENER=(unix1,TCPIP,2480)
NODE=(MVS2,TCPIP,prodms2,2480)
...
logpath=/px/logs/mvscond
CAPI_XTRA=/px/wx/capture/mvscond/camaps
CAPI_PATH=/px/wx/capture/mvscond
/*
 * Source-specific CAPI Connection
 CAPI_CONNECTION=(NAME=MV2UOWC,
    TYPE=(UOWC, CAPINAME=M2_LRAP, RSTRADV=600, MEMCACHE=20480))
 CAPI_CONNECTION=(NAME=MV_2_LRAP,
    TYPE=(LRAP, LOG=MV2L, AGENT=MV2A))
/*
 * CAPX CAPI Connection for continuous extraction
 CAPI_CONNECTION=(NAME=CAPXDSN9,TYPE=(CAPX, DFLTINST=DSN9, FILEWAIT=60, RSTRADV=600))
```
In this example, the pwxccl.cfg file has the following parameters:

```c
/*
 /* pwxccl.cfg
 /*
 DBID=DSN9
 DB_TYPE=DB2
 CONN_QVR=DB2U0WC
 CAPTURE_NODE=MVS2
 EXT_CAPT_MASK=/pwx/capture/mvscond/condense
 CHKPT_NUM=3
 CHKPT_BASENAME=/pwx/capture/mvscond/condense.chkpt
 COND_CDCT_RET_R=50
 COLL_END_LOG=0
 NO_DATA_WAIT=5
 NO_DATA_WAIT2=2
 FILE_SWITCH_VAL=20000
 FILE_SWITCH_CRIT=R
 CAPT_IMAGE=BA
 SIGNALLING=N
 OPER_WTO=N
 VERBSE=Y
*/
```

2. After you configure the dbmover.cfg and the pwxccl.cfg configuration files, start the PowerExchange Listener and PowerExchange Logger on the UNIX system.

3. 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 MVS system

   In this example, the following statements are added to the dbmover.cfg on the Integration Service machine:

   ```
   NODE=(unix1,TCPIP,unix1,2480)
   NODE=(MVS2,TCPIP,prodMVS2,2480)
   ```

4. Create and configure the PowerCenter mapping, session, and workflow to extract the change data.

5. To extract the change data from the UNIX systems, configure a PWX DB2zOS 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 MVS 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 MVS2
   - For the CAPI Connection Name option, specify CAPXDSN9

6. Cold start the CDC session to extract the change data from the PowerExchange Logger log files on the UNIX system.

**Related Topics:**

- “Configuring pwxccl.cfg” on page 284
- “Configuring dbmover.cfg on the PowerExchange Logger Machine” on page 288
APPENDIX A

CDC for z/OS Troubleshooting

This appendix includes the following topics:
- CDC for z/OS Troubleshooting Overview, 293
- Problems Related to Propagating Data, 293

CDC for z/OS Troubleshooting Overview

This chapter provides general troubleshooting information to assist you when problems occur when you use PowerExchange.

If you cannot resolve the problem, contact Informatica Global Customer Support.

Problems Related to Propagating Data

If your system is not propagating data, perform the verifications listed in this section.

Change Data Capture Components

If PowerExchange is not successfully capturing changes from the source, verify the following items:
- Verify that the PowerExchange Agent is active.
- Verify that the PowerExchange Logger is active and connected to the correct PowerExchange Agent.
- Verify that the appropriate ECCR is active.
- Verify that the ECCR is capturing the appropriate data, as follows:
  - For IMS and VSAM, check message number PWXEDM172849I, which provides information about the PowerExchange repository and the capture process. PowerExchange writes this informational message to the log data set for the PowerExchange Agent each time the change interface component (CIC) checks the repository to determine whether to capture changed data for a specific file or database.
  - For DB2, check message number PWXEDM172808I, which lists the source tables from which the ECCR is capturing changes.
- For DB2, verify that the source tables are defined with the DATA CAPTURE CHANGES option.
- Verify that your sources are registered correctly in the PowerExchange Navigator.
- Verify that the correct PowerExchange Agent repository is being used.
To determine which PowerExchange repository is allocated to the PowerExchange Agent, check the EDMSLOG associated with the PowerExchange agent's startup procedure. Search for message PWXEDM172119I to find the name of the PowerExchange repository that the PowerExchange agent is accessing.

- Verify that the source is being updated with changes.

**Gathering Operating Environment Information**

Use this procedure and the worksheet in this section to gather information before contacting Informatica Global Customer Support.

**Before You Begin**

The following information is need for problem rectification:

<table>
<thead>
<tr>
<th>System Component</th>
<th>Information Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>The type of CPU you are using</td>
</tr>
<tr>
<td>Operating System</td>
<td>The type of operating system you are using</td>
</tr>
<tr>
<td>OS Level</td>
<td>The release of the operating system you are using</td>
</tr>
<tr>
<td>SMS</td>
<td>Whether SMS is being used or is not being used</td>
</tr>
<tr>
<td>PowerExchange Version</td>
<td>The version number of the PowerExchange product</td>
</tr>
<tr>
<td>MVS</td>
<td>MVS version and release</td>
</tr>
<tr>
<td></td>
<td>MVS maintenance level</td>
</tr>
<tr>
<td>Security</td>
<td>Security product</td>
</tr>
<tr>
<td></td>
<td>Security package version and release</td>
</tr>
<tr>
<td>CICS</td>
<td>CICS version and release</td>
</tr>
<tr>
<td>IMS</td>
<td>IMS version and release</td>
</tr>
<tr>
<td>DB2 for z/OS</td>
<td>DB2 version and release</td>
</tr>
<tr>
<td>Oracle</td>
<td>Oracle version and release</td>
</tr>
<tr>
<td>MS SQL Server</td>
<td>MS SQL Server version, release, and service pack level</td>
</tr>
<tr>
<td>Sybase</td>
<td>Sybase version and release</td>
</tr>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>UDB version and release</td>
</tr>
<tr>
<td>UNIX</td>
<td>UNIX variant version and release</td>
</tr>
<tr>
<td>Windows</td>
<td>Windows version, release, and service pack level</td>
</tr>
<tr>
<td>System Component</td>
<td>Information Needed</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>PowerExchange Version</td>
<td>Version number</td>
</tr>
<tr>
<td></td>
<td>Source database type</td>
</tr>
<tr>
<td></td>
<td>Target database type</td>
</tr>
<tr>
<td>PowerCenter Version</td>
<td>Version number</td>
</tr>
<tr>
<td></td>
<td>Target database type</td>
</tr>
<tr>
<td>PowerExchange Agent</td>
<td>Copy of all output</td>
</tr>
<tr>
<td>PowerExchange Logger</td>
<td>Copy of all output</td>
</tr>
<tr>
<td>Any component</td>
<td>Description of the problem</td>
</tr>
<tr>
<td></td>
<td>Record of all console messages</td>
</tr>
<tr>
<td></td>
<td>Description of your troubleshooting procedure</td>
</tr>
</tbody>
</table>
INDEX

A
activating the CICS VSAM ECCR 124
active batch ECCR 117
ADASEL_DSN 108
adding active log data set definitions 57
Agent Security 34
AGENTGEN 25
AgentID 25
AGENTID 22
allocating restart data sets 56
altering the DB2 table structure 178
APPBUFSIZE
  configuration parameter for PowerExchange Listener 276
application name
  configuring for CDC sessions 254
application names 235
application recover considerations 121
APPLID
  EDWKOPER option 124
archive log rules and guidelines 53
archive_options statement 43
AVERAGE DELAY 169

B
backing up PowerExchange Condense files 102
BackToBackDelay 27
batch job requirements 117
batch mode 79
BYPASS 136

C
CA NAME 158
Cache1 27
Cache2 27
calculating the data set size 55
canceling the condense job 99
c API connection statements
LRAP parameters 13
  UOWC parameters 15
CAPT_IMAGE 85
CAPT_STATS 200
CAPTPARM 82, 84, 96
catalog tables
data capture changes 174
CCERR 22
CCT
DTLMCPR 81
CCVACTIVE 25
CDC
  CICS/VSAM 122
CDC data map
  extraction map 259
CDC sessions
  commit processing 241
default restart points 236
methods of starting 236, 262
offload processing 246
recovery example 288
restart points for warm starts 237
restart token file 259
stopping 264
CDC tables
  Datacom 141
CDCL program 141
CDCM program 141
CDCU program 141
CENTURY 22
changing IDMS structures 195
changing the size of existing active log data sets 58
changing VSAM structures 121, 129
cHECKPOINT files 82, 85
cHECKPOINT record type 82
CHKPT_BASENAME 85, 96
CHKPT_NUM 85
CHKPT_PRIM_ALLOC 85
CHKPT_SCND_ALLOC 85
CHKPT_VOLSERS 85
CHKSCHEM 160
CI_INTV 169
CI_PSEC 169
CI_TOT 169
CICS/VSAM
  CDC 122
CICS/VSAM ECCR
  controlling 127
  stopping 128
CLOSE 205
close (pwxcmd) 19
closeforce (pwxcmd) 19
closing a VSAM dataset 120, 128
CmdAuthCheck 25
CmdPrefix 25
cold start 96
COLDSTART 108, 200
COLL_END_LOG 85, 108
commands
  DISPLAY SUBSYS 222
  IMS 222
  SSR xEDP-ABORT 221
  SSR xEDP-CONTINUE 221
  SSR xEDP-STAT 221
  SSR xEDP-STATWTO 221
commit processing
  configuring for CDC sessions 255
  controlling with connection attributes 242
  examples 244
  in CDC sessions 241
data set size determination 54
data spaces
  SCOPE=COMMON 21
Datacom table-based CDC 139
DATAMAP 204
DATE 22
DB_TYPE 85, 108, 200
DB2 V8
data capture changes requirement 157, 174
DB2-LOG LOCATION 169
DB2-LOG TIMESTAMP 169
DBID 85, 108, 200
DCOMDLF 136
DCOMPLF 136
DCPARMFLF 136
DCT records 82
DDASSOR1 106
DDATAR1 106
DDWORKR1 106
deactivating
  registrations 179
define statement
  syntax overview 40
define_log command 63
defining
  log data sets 46, 63
deleting
  log data sets 64
  registrations 179
dispatching priorities 74
DISPLAY
  EDMC keyword 127
PowerExchange Agent command 118
DISPLAY TRACE 205
displaystatus (pwxcmd) 102
DRAIN 32
DSPACE_ID 133, 134
DTL__CAPXRESTART1
  sequence token 258
DTL__CAPXRESTART2
  restart token 258
DTLADKSD 106
DTLAMNOM
  CCT 81
DTLACDC 81
DTLCACFG 106, 108, 111, 204
DTLCFG
  PowerExchange configuration file 81
DTLCUIML 207
DTLIDLC
  DTLULONGC utility 189
DTLIDLL
  DTLULONGC utility 189
DTLINPUT parameters 133
DTLKEY 204
DTLLOG 204
DTLMSG 204
DTLOUT 84
DTLUAPPL
  displaying restart tokens 258
DTLUCSR2
  Utility scan program for SR2 and SR3 records 190
DTLULCAT catalog utility 189
DTLULONGC utility 189
DTLUTSK utility 264
EXT_CAPT_MASK 85
extraction map columns, PowerExchange-generated
  DTL__BI__columnname 228
  DTL__CAPXACTION 228
  DTL__CAPXCASEDELIND 228
  DTL__CAPXRESTART1 228
  DTL__CAPXRESTART2 228
  DTL__CAPXRNN 228
  DTL__CAPXTIMESTAMP 228
  DTL__CAPXUOW 228
  DTL__CAPXUSER 228
  DTL__CI__columnname 228
  PowerExchange-generated columns 228

F
FAKEIT 136
FILE_SWITCH_CRIT 79, 85
FILE_SWITCH_VAL 79, 85
FILE_TYPE
  DTLULCAT utility 189
FILESWITCH 82
fileswitch (pwxcmd) 82, 102
FILESWITCH command 79
formatting log data sets 62

G
GetIMSRBAByLevel
  synchronous ECCR 211
  group source
    description 238
    processing CDC data for multiple source definitions 240
  GROUPDEFS 85

H
HELP
  EDMC keyword 127

I
idle time
  configuring for a CDC session 253
  description 253
IDMS_VERSION
  DTLULCAT utility 189
IFI30OPT 160
IGNORENOCHANGEUPDATES 108
IMS databases
  GSAM 210
  HSAM 210
  MSDB 210
  SHSAM 210
IMS fast path 210
IMS.schema changes 207, 225
IMSID 200
INIT1
  EDMC keyword 127
InitAuthCheck 25
INSTANCE_IDENTIFIER
  DTLULCAT utility 189

J
JCL 107
JOBLIB 28
jobname
  EDMKOPER option 124

K
KEY_CHANGE_ALW 85

L
LAST DELAY 169
  linkage index 21
  listtask (pwxcmd) 19, 270, 272
  LISTTASK command 19, 272
  LNKLST 124
Local Mode
  adding log restrictions 185
  Location 27
  LOCATION 134
  log catalog
    adding Logs in Order 185
  Log Feeder 136
  log start sample report
  CICS VSAM ECCR 126
log-based
  ECCR 197
  LogBuffLimit 25
  LogClass 25
  LOGCLOSE 32
  LOGGER 22
  logging_options statement 45
  LogHold 25
  LogLimit 25
  LOGOPEN 32
  LOGRGRP 22
  LOGSPIN 32
  long names
    PowerExchange restrictions 158
  LRAP CAPI_CONNECTION parameters
    parameters and syntax 13

M
manage schema changes 138
managing
  log and restart data sets 52
  maximum row count
    configuring for a CDC session 255
MEDIA_CONTENT
  DTLULCAT utility 189
MEDIA_TYPE
  DTLULCAT utility 189
message log 31
  migrating to a DB2 data-sharing environment 176
minimum row count
  configuring for a CDC session 255
  MNT table 141
moving
  log data sets to other devices 67
  MSG_INTV 169
  MSG_PSEC 169
  MSG_TOT 169
Index

restart token file
  example 261
  explicit override 259
  overview 231
  special override 260
  syntax 259
restart tokens
  displaying with DTLUAPPL 258
  DTL__CAPXRESTART1 258
  DTL__CAPXRESTART2 258
  null 236
  overview 231
  recovery state file 235
  recovery state table 234
RESTART_TOKEN 96
restarting processing
  EDMCMUOW DD statement 118
restartInterval 27
restrictions
  IFI306OPT processing 175
  XPEDITOR CICS 123
restrictions, post-log merge 70
RESUME 32
RUNLIB 200
running as part of a SHADOW-Plex 135

S
sample CICS ECCR log start 126
sample JCL procedure for the PowerExchange Logger 48
Sample Messages 30
schema change 147, 180
schema changes 195
schema verification 179
sequence token
  DTL__CAPXRESTART1 258
  SEQUENCE_TOKEN 96
shutcond (pwxcmd) 102
SHUTCOND command 79, 98
SHUTDOWN 32, 82
shutdown (pwxcmd) 82, 102
SHUTDOWN command 79, 98
shutting down
  STOP command 99
shutting down Condense 98
SIGNALLING 85
size and number of active log data sets 53
SR2OUT
  DTLUCSR2 DD card 190
SR2TOTAL
  DTLUCSR2 DD card 190
SRT record 82
START
  PowerExchange Agent command 118
START COLD 158
START keyword 160
START WARM 158
starting the PowerExchange Logger 48
STARTTIME 200
Startup 25
STARTUP 28
STARTUP WARM 171, 195
STAT LEV 160
STATUS 134
STEPLIB 28
STOP
  PowerExchange Agent command 118
STOP command
  shutting down 99
  stopping a VSAM ECCR 128
  stopping change capture 138, 178
  stopping the ECCR 120, 167
  stopping the PowerExchange Logger 49
stoptask (pwxcmd) 19
STOPTASK command
  CDC sessions, stopping 264
synchronous ECCR
  considerations 211
  IMS Batch Backout utility 224
  MVS checkpoint and restart 224
  non-keyed segments 211
  overview 209
  recovery considerations 224
  restrictions 210
SYOUT 22
SYSPRINT 28
SYSTEM LNKLST 124
system requirements 69
system_options statement 41

T
TABLE_NAME 169
TaskLimit 25
TERM 168
TERM1
  EDMC keyword 127
terminating conditions
  idle time for CDC sessions 253
testing PowerExchange for Adabas CDC installation 111
TIME 22
timed checkpoint considerations for dormant member Loggers 73
TRACE 160
TRACEOFF 205
TRACEON 205
tracks per cylinder and bytes per track 55
TSN table 141

U
unique connection identifier
  each CICS region 124
UOW count
  configuring for a CDC session 255
UOWC CAPI_CONNECTION parameters
  parameters and syntax 15
UpdateInterval 27
URID 168
using post-log merge 69

V
VERBOSE 85
VSAM batch change capture 114
VSAM batch ECCR
  output 119
  stopping 120

W
warm starts
  CDC session restart points 237
WRITE_RESTART_SECS 200 restrictions 123

X

XCF groups 39
XPEDITER CICS