Informatica® PowerExchange
10.0

CDC Guide for z/OS

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), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (http://www.opensource.org/licenses/bsd-license.php), the MIT License (http://www.opensource.org/licenses/mit-license.php), the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0) and the Initial Developer's Public License Version 1.0 (http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/).

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 product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.


DISCLAIMER: Informatica LLC provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica LLC 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.

Publication Date: 2018-07-05
# Table of Contents

**Preface** ................................................................. 14

Informatica Resources .................................................. 14
  Informatica My Support Portal ....................................... 14
  Informatica Documentation ........................................... 14
  Informatica Product Availability Matrixes ....................... 15
  Informatica Web Site .................................................. 15
  Informatica How-To Library .......................................... 15
  Informatica Knowledge Base ......................................... 15
  Informatica Support YouTube Channel ........................... 15
  Informatica Marketplace .............................................. 16
  Informatica Velocity .................................................. 16
  Informatica Global Customer Support ............................. 16

**Part I: PowerExchange Change Data Capture Introduction** .................. 17

**Chapter 1: Change Data Capture Overview** .......................... 18

  PowerExchange CDC Overview ...................................... 18
  PowerExchange Components for CDC .................................. 20
    PowerExchange Agent .............................................. 20
    PowerExchange Environmental Change Capture Routine (ECCR) .............................................. 20
    PowerExchange Listener ........................................ 21
    PowerExchange Logger for MVS ................................. 21
    PowerExchange Condense ........................................ 22
    PowerExchange Navigator ........................................ 22
  PowerExchange CDC for MVS Data Sources ..................... 22
    Adabas Change Data Capture .................................. 22
    Datacom Change Data Capture ................................ 23
    DB2 for z/OS Change Data Capture .......................... 23
    IDMS Change Data Capture ................................... 23
    IMS Change Data Capture ................................... 23
    VSAM Change Data Capture ................................... 23
  PowerExchange Message Log Data Sets .......................... 24
  PowerExchange Integration with PowerCenter .................... 25
  Summary of CDC Implementation Tasks .......................... 26

**Part II: CDC Components Configuration and Management** ............... 28

**Chapter 2: PowerExchange Listener** ................................... 29

  PowerExchange Listener Overview ................................ 29
  Configuring the PowerExchange Listener for CDC .................. 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configuring the PowerExchange Listener JCL</td>
<td>30</td>
</tr>
<tr>
<td>Configuring CAPI_CONNECTION Statements</td>
<td>30</td>
</tr>
<tr>
<td>Managing the PowerExchange Listener</td>
<td>36</td>
</tr>
<tr>
<td>Starting the PowerExchange Listener</td>
<td>37</td>
</tr>
<tr>
<td>Stopping the PowerExchange Listener</td>
<td>37</td>
</tr>
<tr>
<td>Controlling PowerExchange Listener Tasks</td>
<td>37</td>
</tr>
<tr>
<td>Chapter 3: PowerExchange Agent</td>
<td>38</td>
</tr>
<tr>
<td>PowerExchange Agent Overview</td>
<td>38</td>
</tr>
<tr>
<td>Running Multiple Instances of the PowerExchange Agent</td>
<td>39</td>
</tr>
<tr>
<td>Configuring MVS for the PowerExchange Agent</td>
<td>39</td>
</tr>
<tr>
<td>Linkage Indexes and Common Data Spaces</td>
<td>40</td>
</tr>
<tr>
<td>Global Enqueue for PowerExchange CDC</td>
<td>40</td>
</tr>
<tr>
<td>Configuring the PowerExchange Agent</td>
<td>40</td>
</tr>
<tr>
<td>EDMSDIR Options Module</td>
<td>41</td>
</tr>
<tr>
<td>Customizing EDMSDIR Module Options</td>
<td>45</td>
</tr>
<tr>
<td>Configuring AGENTCTL Module Options</td>
<td>45</td>
</tr>
<tr>
<td>Configuring AGENTREP Module Options</td>
<td>47</td>
</tr>
<tr>
<td>Customizing the PowerExchange Agent JCL</td>
<td>48</td>
</tr>
<tr>
<td>PowerExchange Agent Message Log</td>
<td>51</td>
</tr>
<tr>
<td>Managing the PowerExchange Agent</td>
<td>51</td>
</tr>
<tr>
<td>Starting the PowerExchange Agent</td>
<td>51</td>
</tr>
<tr>
<td>Stopping the PowerExchange Agent</td>
<td>52</td>
</tr>
<tr>
<td>Controlling the PowerExchange Agent</td>
<td>52</td>
</tr>
<tr>
<td>Managing Capture Registration Caching</td>
<td>53</td>
</tr>
<tr>
<td>Controlling Security for the PowerExchange Agent</td>
<td>54</td>
</tr>
<tr>
<td>Controlling Access to PowerExchange Agent Services</td>
<td>55</td>
</tr>
<tr>
<td>Controlling Access to PowerExchange Agent Commands</td>
<td>55</td>
</tr>
<tr>
<td>Controlling Access to PowerExchange Components</td>
<td>56</td>
</tr>
<tr>
<td>Chapter 4: PowerExchange Logger for MVS</td>
<td>57</td>
</tr>
<tr>
<td>PowerExchange Logger for MVS Overview</td>
<td>57</td>
</tr>
<tr>
<td>Multiple Instances of the PowerExchange Logger for MVS</td>
<td>58</td>
</tr>
<tr>
<td>Planning for the PowerExchange Logger for MVS</td>
<td>59</td>
</tr>
<tr>
<td>XCF Groups</td>
<td>59</td>
</tr>
<tr>
<td>Recall of Archived Log Data Sets</td>
<td>59</td>
</tr>
<tr>
<td>PowerExchange Logger Configuration Considerations</td>
<td>59</td>
</tr>
<tr>
<td>Configuring the PowerExchange Logger for MVS</td>
<td>60</td>
</tr>
<tr>
<td>Configuring the EDMUPARM Options Module</td>
<td>60</td>
</tr>
<tr>
<td>Verifying That the Active Log and Emergency Restart Data Sets Were Created Correctly</td>
<td>66</td>
</tr>
<tr>
<td>Customizing the PowerExchange Logger JCL</td>
<td>67</td>
</tr>
<tr>
<td>Managing the PowerExchange Logger for MVS</td>
<td>69</td>
</tr>
<tr>
<td>Starting the PowerExchange Logger for MVS</td>
<td>69</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Stopping the PowerExchange Logger for MVS</td>
<td>69</td>
</tr>
<tr>
<td>Controlling the PowerExchange Logger for MVS</td>
<td>69</td>
</tr>
<tr>
<td>Overriding Log-Read API Timed Defaults</td>
<td>70</td>
</tr>
<tr>
<td>Resolving In-Doubt Units of Work</td>
<td>71</td>
</tr>
<tr>
<td>Monitoring the PowerExchange Logger for MVS</td>
<td>72</td>
</tr>
<tr>
<td>Performance Rules and Guidelines</td>
<td>72</td>
</tr>
<tr>
<td>Managing Log and Restart Data Sets</td>
<td>73</td>
</tr>
<tr>
<td>Archive Log Rules and Guidelines</td>
<td>73</td>
</tr>
<tr>
<td>Size and Number of Active Log Data Sets</td>
<td>74</td>
</tr>
<tr>
<td>Data Set Size Determination</td>
<td>75</td>
</tr>
<tr>
<td>Number of Data Sets</td>
<td>76</td>
</tr>
<tr>
<td>Allocating Restart Data Sets</td>
<td>77</td>
</tr>
<tr>
<td>Adding Active Log Data Set Definitions to the Restart Data Set</td>
<td>78</td>
</tr>
<tr>
<td>Changing the Size of Active Log Data Sets</td>
<td>79</td>
</tr>
<tr>
<td>Formatting Log Data Sets</td>
<td>83</td>
</tr>
<tr>
<td>Defining Log Data Sets to the ERDS</td>
<td>84</td>
</tr>
<tr>
<td>Deleting Log Data Sets from the ERDS</td>
<td>85</td>
</tr>
<tr>
<td>Recovering Damaged Active Log Data Sets</td>
<td>86</td>
</tr>
<tr>
<td>Recovering Damaged Restart Data Sets</td>
<td>88</td>
</tr>
<tr>
<td>Moving Log Data Sets to Other Devices</td>
<td>89</td>
</tr>
<tr>
<td>Using Post-Log Merge</td>
<td>90</td>
</tr>
<tr>
<td>Post-Log Merge System Requirements</td>
<td>91</td>
</tr>
<tr>
<td>Post-Log Merge Restrictions</td>
<td>91</td>
</tr>
<tr>
<td>Post-Log Merge Configuration</td>
<td>92</td>
</tr>
<tr>
<td>Performance Considerations</td>
<td>94</td>
</tr>
<tr>
<td>Recovery Scenarios</td>
<td>95</td>
</tr>
<tr>
<td>Post-Log Merge Job Commands</td>
<td>97</td>
</tr>
</tbody>
</table>

**Chapter 5: PowerExchange Condense**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerExchange Condense Overview</td>
<td>99</td>
</tr>
<tr>
<td>Configuring PowerExchange Condense</td>
<td>100</td>
</tr>
<tr>
<td>Enabling Capture Registrations for PowerExchange Condense Use</td>
<td>100</td>
</tr>
<tr>
<td>PowerExchange Condense Job</td>
<td>100</td>
</tr>
<tr>
<td>Condense Operational Modes</td>
<td>101</td>
</tr>
<tr>
<td>Configuring PowerExchange Condense JCL</td>
<td>102</td>
</tr>
<tr>
<td>Condense Input Files</td>
<td>103</td>
</tr>
<tr>
<td>Condense Output Files</td>
<td>104</td>
</tr>
<tr>
<td>PowerExchange Message Data Sets</td>
<td>106</td>
</tr>
<tr>
<td>Configuring PowerExchange Condense Parameters</td>
<td>107</td>
</tr>
<tr>
<td>Parameter Descriptions</td>
<td>107</td>
</tr>
<tr>
<td>Controlling Allocation Attributes of Condense Data Sets</td>
<td>121</td>
</tr>
<tr>
<td>Configuring Condense Group Definitions</td>
<td>125</td>
</tr>
<tr>
<td>Starting and Stopping PowerExchange Condense</td>
<td>127</td>
</tr>
</tbody>
</table>
Starting Condense. .......................................................... 127
Shutting Down Condense. .................................................. 130
Condense Job Message Output. .......................................... 130
Controlling PowerExchange Condense. ............................... 134
Backing Up PowerExchange Condense Output Files. ............... 134

Part III: CDC Sources Configuration and Management. ............... 136

Chapter 6: Adabas Change Data Capture. .............................. 137
Adabas CDC Overview. .................................................. 137
Planning and Implementation Considerations. ........................ 138
  Gathering Information About Your CDC Environment. .......... 138
  Operational Considerations. ........................................ 139
  Change Capture from Multiple Adabas Databases. .............. 139
  Change Capture from Adabas Spanned Records. ................. 140
Configuring Adabas PLOG Archiving JCL. ............................ 140
  Customizing the PowerExchange Sample SAMPUEX2 Exit. ...... 141
Configuring the Adabas ECCR. ....................................... 142
  Configuring the Adabas ECCR Parameters. ....................... 142
  Configuring the Adabas ECCR JCL. ............................... 149
Testing Adabas CDC Installation and Configuration. ................ 150
Managing Adabas CDC. ............................................... 151
  Starting the Adabas ECCR. ....................................... 151
  Stopping the Adabas ECCR. ...................................... 151
  Adding an Adabas Capture Registration. ......................... 152
  Deleting an Adabas Capture Registration. ....................... 152
  Suspending Change Capture for Registered Adabas Sources Temporarily. 153
Using the Adabas PCAT Utility (DTLCCADW) to Manage the PCAT File. 154

Chapter 7: Batch VSAM Change Data Capture. ....................... 155
Batch VSAM CDC Overview. ......................................... 155
  ECCR Relationships with Other PowerExchange Components. .... 155
  Batch VSAM ECCR Restrictions. .................................... 156
Configuring Batch VSAM Jobs for CDC. ................................ 158
  Making the Batch VSAM ECCR Available to Batch Jobs. ....... 158
  Activating and Loading the Batch VSAM ECCR Interface. ....... 158
Restoring VSAM Data Sets When Using the Batch VSAM ECCR. .... 159
Managing Batch VSAM Change Data Capture. ........................ 159
  Controlling the Batch VSAM ECCR. ............................... 160
  Output from the Batch VSAM ECCR. ............................... 160
  Stopping Change Data Capture for VSAM Sources. ............. 161
  Stopping the Batch VSAM ECCR. .................................. 161
  Refreshing the Batch VSAM ECCR. ................................ 162
Chapter 8: CICS/VSAM Change Data Capture ........................................ 164
CICS/VSAM CDC Overview .......................................................... 164
Planning for CICS/VSAM CDC ...................................................... 164
  CICS/VSAM CDC Requirements and Restrictions ......................... 165
  CICS/VSAM CDC Use of CICS Global and Task-Related Exit Points .... 165
  CICS/VSAM ECCR Relationships with Other PowerExchange Components. 168
Configuring CICS for CDC ............................................................ 169
Starting the CICS/VSAM ECCR .................................................... 172
Managing CICS/VSAM CDC .......................................................... 172
  Output from the CICS/VSAM ECCR ........................................... 173
  Using the EDMC Transaction and Keywords to Manage the CICS/VSAM ECCR. 173
  Displaying the VSAM Data Sets from Which Changes Are Captured .... 174
  Changing CDC Override Options for VSAM Data Sets .................... 175
  Migrating an ESDS Data Set from VSAM Batch CDC to CICS/VSAM CDC. 176
  Stopping the CICS/VSAM ECCR ............................................... 176
  Stopping Change Capture for a Specific VSAM Data Set .................. 177
  Refreshing Capture Registrations for the ECCR .............................. 177
  Managing VSAM Schema Changes .............................................. 177

Chapter 9: Datacom Table-Based Change Data Capture ..................... 178
Datacom Table-Based CDC Overview ............................................ 178
  Relationships with Other PowerExchange Components .................. 178
Implementing Datacom Table-Based CDC ...................................... 179
Architectural Overview .............................................................. 179
  Datacom CDC Components .................................................... 179
  Datacom Table-Based ECCR .................................................... 180
Configuring Datacom for CDC ..................................................... 180
Configuring the Datacom Table-Based ECCR ................................. 181
  Datacom Table-Based ECCR Parameters .................................... 181
  JCL for the Datacom Log-Based ECCR ...................................... 191
Managing Datacom Table-Based CDC ............................................ 193
  Starting the Datacom Table-Based ECCR ................................... 193
  Stopping the Datacom Table-Based ECCR .................................. 193
  Adding a Datacom Capture Registration ..................................... 193
  Deleting a Datacom Capture Registration ................................... 194
  Suspending Change Capture for Registered Datacom Sources Temporarily. 194
  Changing a Datacom Source Table Definition .............................. 195

Chapter 10: DB2 for z/OS Change Data Capture ............................. 196
DB2 for z/OS CDC Overview ....................................................... 196
# Table of Contents

- DB2 CDC Considerations .......................................................... 197
- DB2 for z/OS Datatypes Supported for CDC .................................. 197
- DB2 CDC Operational Considerations ........................................... 198
- Handling Compressed DB2 Table Spaces ....................................... 199
- FIELDPROC and EDITPROC Exit Routines .................................... 200
- How the DB2 ECCR Interacts with Other PowerExchange Components ........................................ 200
- DB2 ECCR Capture Directory Tables ............................................. 201
- Running Multiple DB2 ECCRs ..................................................... 203
- DB2 Data-Sharing Considerations ................................................ 204
- If You Migrate from DB2 for z/OS Version 9.1 to Version 10 ............ 204
- If You Migrate to DB2 for z/OS Version 11 ................................... 205

## Configuring DB2 for CDC
- Activating Change Data Capture for DB2 Catalog Tables .................. 207
- Managing DB2 Logs ................................................................. 207
- DB2 Logging in a Data Sharing Environment ................................... 207

## Configuring the DB2 ECCR
- DB2 ECCR Usage Guidelines ..................................................... 208
- DB2 ECCR Access to DB2 Catalog Tables ....................................... 209
- DB2 ECCR Control Statements in the REPL2CTL DD Data Set ............ 209
- DB2 ECCR Configuration Statements in the REPL2OPT DD Data Set .... 210
- Configuring the DB2 ECCR JCL .................................................. 216

## Defining Restart Tokens for a DB2 Target Table Materialized from an Image Copy
- Starting the DB2 ECCR ............................................................. 218
- Managing DB2 CDC ................................................................. 219
- Stopping the DB2 ECCR ............................................................ 219
- Commands for Controlling DB2 ECCR Processing .......................... 220
- DB2 ECCR Reports ................................................................. 221
- Recovering the DB2 ECCR ......................................................... 224
- Altering DB2 System Tables for DATA CAPTURE CHANGES ............... 224
- DB2 ECCR Capture Directory Table Upgrades ................................ 225
- Reducing the Amount of Data Sent to the DB2 ECCR by Using the IFI306 OPT Statement .................. 227
- Replacing a Table with Another Table That Has the Same Name .......... 228
- Migrating to a DB2 Data Sharing Environment ............................... 229
- Migrating from a DB2 Data Sharing Environment ............................ 229
- Stopping DB2 Change Data Capture ............................................ 230

## Managing DB2 Schema Changes
- Schema Verification ................................................................. 231
- Changing the Schema of DB2 Source Tables .................................. 232
- Recovering from Unplanned Schema Changes to DB2 Source Tables .... 232
- Altering Columns in DB2 Source Tables ....................................... 233
- Changing the Qualifiers of DB2 Table Spaces ................................ 234
<table>
<thead>
<tr>
<th>Chapter 11: IDMS Log-Based Change Data Capture</th>
<th>235</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDMS Log-Based CDC Overview</td>
<td>235</td>
</tr>
<tr>
<td>PowerExchange IDMS Log-Based CDC Components</td>
<td>236</td>
</tr>
<tr>
<td>IDMS Log-Based ECCR Operational Considerations</td>
<td>237</td>
</tr>
<tr>
<td>PowerExchange Log Catalog for IDMS Log-Based CDC</td>
<td>238</td>
</tr>
<tr>
<td>Configuring IDMS Log Catalog Procedures</td>
<td>238</td>
</tr>
<tr>
<td>Running DTLULCAT.</td>
<td>239</td>
</tr>
<tr>
<td>Running DTLULLOGC.</td>
<td>240</td>
</tr>
<tr>
<td>Configuring and Starting the IDMS Log-Based ECCR</td>
<td>241</td>
</tr>
<tr>
<td>Configuring IDMS Log-Based ECCR Parameters</td>
<td>241</td>
</tr>
<tr>
<td>Configuring the IDMS Log-Based ECCR JCL</td>
<td>248</td>
</tr>
<tr>
<td>Detecting Matching SR2 and SR3 Records for ECCR Capture of Relocated Records</td>
<td>249</td>
</tr>
<tr>
<td>Starting the IDMS Log-Based ECCR</td>
<td>251</td>
</tr>
<tr>
<td>Configuring IDMS Log-Based ECCR Parameters</td>
<td>241</td>
</tr>
<tr>
<td>Configuring the IDMS Log-Based ECCR JCL</td>
<td>248</td>
</tr>
<tr>
<td>Configuring and Starting the IDMS Log-Based ECCR</td>
<td>241</td>
</tr>
<tr>
<td>Configuring IDMS Log-Based ECCR Parameters</td>
<td>241</td>
</tr>
<tr>
<td>Configuring the IDMS Log-Based ECCR JCL</td>
<td>248</td>
</tr>
<tr>
<td>Detecting Matching SR2 and SR3 Records for ECCR Capture of Relocated Records</td>
<td>249</td>
</tr>
<tr>
<td>Starting the IDMS Log-Based ECCR</td>
<td>251</td>
</tr>
<tr>
<td>Configuring IDMS Log-Based ECCR Parameters</td>
<td>241</td>
</tr>
<tr>
<td>Configuring the IDMS Log-Based ECCR JCL</td>
<td>248</td>
</tr>
<tr>
<td>Detecting Matching SR2 and SR3 Records for ECCR Capture of Relocated Records</td>
<td>249</td>
</tr>
<tr>
<td>Starting the IDMS Log-Based ECCR</td>
<td>251</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chapter 12: IMS Log-Based Change Data Capture</th>
<th>258</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS CDC Overview</td>
<td>258</td>
</tr>
<tr>
<td>IMS Log-Based Change Data Capture</td>
<td>259</td>
</tr>
<tr>
<td>IMS Log-Based ECCR Processing Phases</td>
<td>260</td>
</tr>
<tr>
<td>Relationships with Other PowerExchange Components</td>
<td>261</td>
</tr>
<tr>
<td>Configuring IMS for Log-Based CDC</td>
<td>261</td>
</tr>
<tr>
<td>Specifying the EXIT Parameter in the DBD Statement</td>
<td>261</td>
</tr>
<tr>
<td>Configuring the IMS Log-Based ECCR</td>
<td>262</td>
</tr>
<tr>
<td>IMS Log-Based ECCR Programs</td>
<td>262</td>
</tr>
<tr>
<td>Configuring the IMS Log-Based ECCR Parameters</td>
<td>263</td>
</tr>
<tr>
<td>APF-Authorize the Libraries in the STEPLIB Concatenation</td>
<td>272</td>
</tr>
<tr>
<td>Configuring the IMS Log-Based ECCR JCL</td>
<td>272</td>
</tr>
<tr>
<td>Configuring the IMS Log-Based ECCR JCL</td>
<td>272</td>
</tr>
<tr>
<td>Managing IMS Log-Based CDC</td>
<td>274</td>
</tr>
<tr>
<td>Starting the IMS Log-Based ECCR</td>
<td>274</td>
</tr>
<tr>
<td>Stopping IMS Log-based Change Data Capture</td>
<td>275</td>
</tr>
<tr>
<td>Using the DTLCUIML Utility to Create Restart Points</td>
<td>275</td>
</tr>
<tr>
<td>Configuring and Starting the IMS Log-Based ECCR</td>
<td>274</td>
</tr>
<tr>
<td>Configuring the IMS Log-Based ECCR Parameters</td>
<td>275</td>
</tr>
<tr>
<td>APF-Authorize the Libraries in the STEPLIB Concatenation</td>
<td>272</td>
</tr>
<tr>
<td>Configuring the IMS Log-Based ECCR JCL</td>
<td>272</td>
</tr>
<tr>
<td>Managing IMS Log-Based CDC</td>
<td>274</td>
</tr>
<tr>
<td>Starting the IMS Log-Based ECCR</td>
<td>274</td>
</tr>
<tr>
<td>Stopping IMS Log-based Change Data Capture</td>
<td>275</td>
</tr>
<tr>
<td>Using the DTLCUIML Utility to Create Restart Points</td>
<td>275</td>
</tr>
<tr>
<td>Adding an IMS Capture Registration</td>
<td>276</td>
</tr>
<tr>
<td>Deleting an IMS Capture Registration</td>
<td>276</td>
</tr>
<tr>
<td>Suspending Change Capture for Registered IMS Sources Temporarily</td>
<td>277</td>
</tr>
<tr>
<td>Changing IMS Source Schema</td>
<td>278</td>
</tr>
</tbody>
</table>
Chapter 16: Extracting Change Data. 322
Overview of Extracting Change Data. 322
Task Flow for Extracting Change Data. 323
Testing an Extraction Map. 324
Configuring PowerCenter CDC Sessions. 325
    Changing Default Values for Session and Connection Attributes. 326
    Configuring Application Connection Attributes. 327
    Examples of Controlling Commit Processing. 335
Recovery and Restart Processing for CDC Sessions. 337
    PowerCenter Recovery Tables for Relational Targets. 338
    PowerCenter Recovery Files for Nonrelational Targets. 339
    Application Names. 340
    Restart Processing for CDC Sessions by Start Type. 340
Creating Restart Tokens for Extractions. 343
Displaying Restart Tokens. 344
Configuring the Restart Token File. 344
    Restart Token File Statements. 345
    Example Restart Token File. 347

Chapter 17: Managing Change Data Extractions. 349
Starting PowerCenter CDC Sessions. 349
    Cold Start Processing. 350
    Warm Start Processing. 350
    Recovery Processing. 351
Stopping PowerCenter CDC Sessions. 351
    Stop Command Processing. 352
    Terminating Conditions. 352
Changing PowerCenter CDC Sessions. 353
    Examples of Adding Sources and Creating Restart Tokens. 353
Recovering PowerCenter CDC Sessions. 355
    Example of Session Recovery. 356

Part V: Monitoring and Tuning. 358

Chapter 18: Monitoring CDC Sessions. 359
Monitoring Overview. 359
Monitoring CDC Sessions in PowerExchange. 359
    Read Progress Messages. 360
    Extraction Statistics Messages. 360
Preface

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

This guide covers the following PowerExchange CDC data sources:

- Adabas®
- CA Datacom®
- CA IDMS™
- DB2® for z/OS®
- IMS™
- VSAM and sequential data sets

In this guide, 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 My Support Portal

As an Informatica customer, the first step in reaching out to Informatica is through the Informatica My Support Portal at https://mysupport.informatica.com. The My Support Portal is the largest online data integration collaboration platform with over 100,000 Informatica customers and partners worldwide.

As a member, you can:

- Access all of your Informatica resources in one place.
- Review your support cases.
- Search the Knowledge Base, find product documentation, access how-to documents, and watch support videos.
- Find your local Informatica User Group Network and collaborate with your peers.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base, find product documentation, access how-to documents, and watch support videos.
Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at https://mysupport.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 https://mysupport.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 Support YouTube Channel

You can access the Informatica Support YouTube channel at http://www.youtube.com/user/INFASupport. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.
Informatica Marketplace

The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at http://www.informaticamarketplace.com.

Informatica Velocity

You can access Informatica Velocity at https://mysupport.informatica.com. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support.

Online Support requires a user name and password. You can request a user name and password at http://mysupport.informatica.com.

Part I: PowerExchange Change Data Capture Introduction

This part contains the following chapter:

- Change Data Capture Overview, 18
CHAPTER 1

Change Data Capture Overview

This chapter includes the following topics:

• **PowerExchange CDC Overview, 18**
• **PowerExchange Components for CDC, 20**
• **PowerExchange CDC for MVS Data Sources, 22**
• **PowerExchange Message Log Data Sets, 24**
• **PowerExchange Integration with PowerCenter, 25**
• **Summary of CDC Implementation Tasks, 26**

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:

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 ECCRs 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 for MVS**

The PowerExchange Logger for MVS receives change data from the ECCRs that connect to it and stores the change data in log data sets.

Real-time extraction sessions or PowerExchange Condense jobs can then extract data from the log data sets. Alternatively, you can configure the PowerExchange Logger for Linux, UNIX, and Windows on a remote system to read read change data from the PowerExchange Logger for MVS log files and relog that data on the Linux, UNIX, or Windows system.

The PowerExchange Logger for MVS stores change data in an active log data set. When the active log data set becomes full, the PowerExchange Logger for MVS 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 most CDC data source types, the maximum length of a record for which PowerExchange can capture and process change data is 128,000 bytes. For Adabas spanned records, PowerExchange CDC supports the Adabas maximum spanned record size. For Datacom sources, other Datacom record length limits might apply. For more information, see the 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 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. You must have Datacom Release 11 SP4 or later.

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 Message Log Data Sets

PowerExchange writes messages to multiple message data sets on z/OS. PowerExchange also writes some WTO messages to the system operator console.

**Note:** You can determine which, if any, of these locations to route a particular message by using the message destination override capability. For more information, see the *PowerExchange Reference Manual*.

PowerExchange uses the following message log data sets:

**DTLLOG**

When alternative logging is enabled, the DTLLOG data set contains only the messages up to the point when the alternative logging subtask initializes. Usually, this information consists of only the DBMOVER statements. If you enabled traces with the TRACE statement in the DBMOVER configuration member, the data set also includes diagnostic trace information.

When alternative logging is not enabled, the DTLLOG data set is the primary message log data set for run-time messages from the PowerExchange programs and components, including the PowerExchange Agent, PowerExchange Condense, PowerExchange Listener, ECCRs, and PowerExchange utilities.

If you set the CAPT_STATS parameter to Y for the Adabas, Datacom, IDMS, or IMS log-based ECCR, the data set also contains ECCR capture statistics by capture registration.

All message lines begin with a time stamp.

**Note:** Alternative logging is enabled by default in the TRACING statement in the DBMOVER configuration member.

**DTLLOGnn**

When alternative logging is enabled, PowerExchange sends run-time messages from PowerExchange components, programs, and commands to a set of DTLLOGnn data sets that are used in a rotating manner. If you set the CAPT_STATS parameter to Y for the Adabas, Datacom, IDMS, or IMS log-based ECCR, the DTLLOGnn log data sets contain ECCR capture statistics messages by capture registration. The DTLLOGnn data sets do not include trace information.

The data set names end with a sequential number nn. When a log file reaches its specified size, PowerExchange switches to the next log file and begins overwriting any data in that file. Informatica strongly recommends that you use alternative logging on z/OS to help improve logging performance and be able to customize the amount of data that is logged for long-running jobs.

To allocate the DTLLOGnn data sets, you can either add DTLLOGnn DD statements in the JCL for a PowerExchange component that logs messages to these data sets or configure the TRACING statement to dynamically allocate the data sets. To send the output to a JES2 or JES3 SYSOUT file rather than a data set that you specify, enter a single DTLLOG01 DD statement in the JCL that specifies the SYSOUT parameter. By using SYSOUT, you can keep the output from a single PowerExchange Listener execution with the rest of the job output. With dynamic allocation, PowerExchange dynamically creates a set of log data sets in a separate directory for each PowerExchange process. The generated data set names vary by component type.

If you use an extended sequential data set as the DTLLOGnn data set format, PowerExchange writes only one message on each track. If you use a normal sequential data set, PowerExchange writes one message to each data block.

**Note:** On z/OS, you cannot see log records in an alternative log data set until the data set is closed. Informatica recommends that you specify VIEW=Y in the TRACING statement to periodically close and reopen an alternative log data set based on the FLUSH interval so that you can view the log records. On
operating systems other than z/OS, you do not need to set VIEW=Y because you can view the log records whenever PowerExchange flushes them to disk.

**DTLOUT**

DTLOUT is a dynamically allocated data set that usually contains normal output from PowerExchange programs. When alternative logging is disabled, this data set contains most of the same run-time messages as in the DTLLOG data sets but without the time stamps at the beginning of each message line.

If you set the CAPT_STATS parameter to Y for the Adabas, Datacom, IDMS, or IMS log-based ECCR, the DTLOUT data set also contains ECCR capture statistics messages by capture registration.

If alternative logging is enabled and you use PowerExchange Condense, the DTLOUT data set contains messages only if condense file allocation errors occurred.

**DTLERR**

DTLERR is a dynamically allocated data set that contains error and diagnostic messages from PowerExchange programs.

**EDMMSG**

The EDMMSG data set contains PWXEDM messages from the PowerExchange Agent, ECCRs, PowerExchange Logger for MVS, Log Read API (LRAPI), and the Log Write API (LWAPI). The data set is allocated based on the EDMMSG DD statement in the JCL for a component, or if you do not include the EDMMSG DD in the JCL, the EDMMSG data set is dynamically allocated.

The EDMMSG data set includes Logger messages that are generated when CDC workflows run. If you use PowerExchange Condense, this data set also includes messages that indicate the PowerExchange Logger and PowerExchange Agent to which a Condense job attaches and the starting point at which to begin, which is passed to the Logger.

---

**PowerExchange Integration with PowerCenter**

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

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

PowerCenter provides transformation and data cleansing capabilities, which you can use in your CDC sessions.
The following figure shows the data flow for processing change data that PowerExchange captured from z/OS data sources:

In this data flow, PowerExchange ECCR components capture change data and send it to the PowerExchange Logger. Optionally, PowerExchange Condense reads data from the PowerExchange Logger log files and writes it to condense files. When a CDC session runs on the PowerCenter Integration Service machine, PWXPC uses the PWX SCLI interface to communicate with the PowerExchange Listener on the z/OS system to retrieve change data.

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

Summary of CDC Implementation Tasks

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

The following table identifies the tasks that you perform to implement change data capture and extraction processing for a z/OS data source:

<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 Guide</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- &quot;Configuring the PowerExchange Listener for CDC&quot; on page 29</td>
</tr>
<tr>
<td>2</td>
<td>Start the PowerExchange Listener.</td>
<td>&quot;Managing the PowerExchange Listener&quot; on page 36</td>
</tr>
<tr>
<td>3</td>
<td>Configure the PowerExchange Agent.</td>
<td>&quot;Configuring the PowerExchange Agent &quot; on page 40</td>
</tr>
<tr>
<td>4</td>
<td>Start the PowerExchange Agent.</td>
<td>&quot;Managing the PowerExchange Agent&quot; on page 51</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>&quot;PowerExchange Logger Configuration Considerations&quot; on page 59</td>
</tr>
<tr>
<td>6</td>
<td>Start the PowerExchange Logger.</td>
<td>&quot;Managing Log and Restart Data Sets&quot; on page 73</td>
</tr>
<tr>
<td>7</td>
<td>Configure the appropriate PowerExchange ECCR for the data source.</td>
<td>&quot;CDC Sources Configuration and Management&quot; on page 136</td>
</tr>
<tr>
<td>8</td>
<td>Create a data map using the PowerExchange Navigator. This step is required for nonrelational sources.</td>
<td>PowerExchange Navigator User Guide</td>
</tr>
<tr>
<td>9</td>
<td>For DB2 sources that require user-defined fields and expressions, create a data map using the PowerExchange Navigator.</td>
<td>PowerExchange Navigator User Guide</td>
</tr>
<tr>
<td>10</td>
<td>Define and activate capture registrations and extraction maps for the data source using the PowerExchange Navigator.</td>
<td>PowerExchange Navigator User Guide</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>&quot;Change Data Extraction&quot; on page 309</td>
</tr>
<tr>
<td>13</td>
<td>Start the ECCR.</td>
<td>&quot;CDC Sources Configuration and Management&quot; on page 136</td>
</tr>
<tr>
<td>14</td>
<td>(Optional) Configure PowerExchange Condense.</td>
<td>&quot;Configuring PowerExchange Condense&quot; on page 100</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, 29
- PowerExchange Agent, 38
- PowerExchange Logger for MVS, 57
- PowerExchange Condense, 99
CHAPTER 2

PowerExchange Listener

This chapter includes the following topics:

- PowerExchange Listener Overview, 29
- Configuring the PowerExchange Listener for CDC, 29
- Managing the PowerExchange Listener, 36

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>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLAMCPR</td>
<td>Required. Points to the VSAM CCT data set, which contains the capture registrations.</td>
</tr>
<tr>
<td>DTLCACDC</td>
<td>Optional. Points to the VSAM CDCT 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>Required. Points to the VSAM CDEP 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>Required. Points to the VSAM DTLCAMAP data set, which contains the extraction maps.</td>
</tr>
<tr>
<td>EDMPARMS</td>
<td>Required. 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>

**Note:** If you want to override the default time that the Log Read API (LRAPI) waits for a response after sending a command to the PowerExchange Logger for MVS, you can include the EDMLRPRM DD statement with the appropriate parameters in the PowerExchange Listener JCL. The parameters then pertain to all LRAPI instances and extractions. Alternatively, you can specify the parameters for a specific LRAPI instance by specifying the EDMLRPRM DD in the job that issues the Log-Read API (LRAPI) calls to the PowerExchange Logger. For more information, see "Overriding Log-Read API Timed Defaults" on page 70.

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.

**CAPI_CONNECTION - LRAP Statement**

The LRAP CAPI_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses 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.

**Data Sources:** Adabas, CA Datacom/DB, CA IDMS/DB, DB2 for z/OS, IMS, and VSAM

**Related Statements:** CAPI_CONNECTION - UOWC

**Required:** Yes for z/OS CDC
Syntax:

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

Parameters:

**DLLTRACE=trace_id**

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

**NAME=capi_connection_name**

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

**TRACE=trace_name**

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

**TYPE=[LRAP, ... ]**

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

**AGENT=agent_id**

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

**EOF=[N|Y]**

Optional. Controls whether PowerExchange stops change data extractions after reaching the end-of-log (EOL). 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.

Default is `N`.

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, enter 0 for the `Idle Time` attribute on the PWX DB2zOS CDC Real Time application connections.
- For PowerExchange Condense, enter 1 in the `COLL_END_LOG` statement in the CAPTPARM configuration member.
- For CDC sessions that use ODBC connections, enter 0 for the `WAITTIME` parameter in the ODBC data source.
The UOWC CAPI_CONNECTION statement specifies a named set of parameters that the Consumer API (CAPI) uses for the UOW Cleanser. Cleanser reconstructs the intermingled changes read from the change stream into complete UOWs in chronological order based on end time.

Data Sources

Related Statements

Syntax

<table>
<thead>
<tr>
<th>Required</th>
<th>Yes, for CDC for the specified sources</th>
</tr>
</thead>
</table>

**LOG=logger_id**

Required. The PowerExchange Logger ID. This value 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)**

Optional. 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: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.

Default is UID.

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

**CAPI_CONNECTION - UOWC Statement**

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

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

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

**Data Sources:** DB2 for i5/OS CDC sources, Oracle CDC with LogMiner sources, and z/OS CDC sources

**Related Statements:** CAPI_CONNECTION - AS4J, CAPI_CONNECTION - LRAP, and CAPI_CONNECTION - ORCL

**Required:** Yes, for CDC for the specified sources

**Syntax:**

```
CAPI_CONNECTION={[DLLTRACE=trace_id]
 ,NAME=capi_connection_name
 [,TRACE=trace_name]
 ,TYPE=UOWC
 ,CAPINAME=source_capi_name
 [,BKSIZE=block_size]
 [,DATACLASS=data_class]
 [,LARGEOPS=number_of_operations]
 [,MEMCACHE=[cache_size|1024]]
 [,MONITORINT={minutes|5}]
 [,RSTRADV=seconds]
 [,SPACEFRI={primary_space|50}]
 [,SPACETYPE={BLK|TRK|CYL}]
 [,SPILLKEEP=number_of_spill_files]
 [,STORCLASS=storage_class]
 [,TIMESTAMP=LOG|COMMIT]}
```
Parameters:

**DLLTRACE=trace_id**

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

**NAME=capi_connection_name**

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

**TRACE=trace_name**

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

**TYPE=(UOWC, ...)**

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

**CAPINAME=capi_name**

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

- AS4J CAPI_CONNECTION statement for DB2 for i5/OS sources
- ORCL CAPI_CONNECTION statement for Oracle CDC with LogMiner sources
- LRAP CAPI_CONNECTION statement for data sources on z/OS

**BLKSIZE=block_size**

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

The following table shows valid values by CDC source type:

<table>
<thead>
<tr>
<th>Data Source Type</th>
<th>Valid Values</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 for i5/OS</td>
<td>A number from 8 through 32760</td>
<td>32760</td>
</tr>
<tr>
<td>Oracle</td>
<td>A number from 8 through 65535</td>
<td>32768</td>
</tr>
<tr>
<td>z/OS data sources</td>
<td>A number from 8 through 32760</td>
<td>18452</td>
</tr>
</tbody>
</table>

**DATACLASS=data_class**

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

**LARGEOPS=number of operations**

Optional. Overrides the default value that PowerExchange uses to identify transactions as large transactions for reporting purposes. Enter the number of DML operations (inserts, updates, and deletes), in thousands, that a transaction must contain to be considered a large transaction.
PowerExchange issues status messages for large transactions that meet this criteria. If PowerExchange issues too many messages, you can increase this value to limit the number of messages.

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

**MEMCACHE=(cache_size|1024)**

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

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

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

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

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

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

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

- For i5/OS, PowerExchange uses CRTPF command to create a physical file for UOW spill files. PowerExchange names the UOW spill files using the C/C++ tmpnam() function.
- For Linux and UNIX, PowerExchange uses the current directory by default for UOW spill files. To use a different directory, specify the TMPDIR environment variable. PowerExchange names the UOW spill file names using the prefix “dtlq” and the operating system function tempnam.
  
  **Note:** The UOW spill files are temporary files that are deleted when PowerExchange closes them. These files are not visible in the directory while they are open.
- For Windows, PowerExchange uses the current directory by default for UOW spill files. To use a different directory, specify the TMP environment variable. PowerExchange names the UOW spill file using the prefix “dtlq” and the Windows _tempnam function.
- For z/OS, PowerExchange uses dynamic allocation to allocate temporary data sets for the UOW spill files. Generally, SMS controls the location of temporary data sets. If you do not use SMS to control temporary data sets, the UNIT parameter controls the location for the UOW spill files. Because PowerExchange allocates temporary data sets for the UOW spill files, z/OS assigns these files system-generated data set names, which begin with SYSyyddd.Tthmmss.RA000.jobname.
**Warning:** PowerExchange allocates the cache size for each extraction operation. If you use a large MEMCACHE value and run many concurrent extraction sessions, memory constraints can occur.

**MONITORINT=minutes**

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

**RSTRADV=seconds**

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

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

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

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

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

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

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

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

**SPACEPRI={primary_space|50}**

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

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

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

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

SPACETYP={BLK|TRK|CYL}
Optional. On z/OS, the type of units in which the primary space for UOW Cleanser allocation of UOW spill files is expressed.
Options are:
- BLK. Blocks.
- CYL. Cylinders.
- TRK. Tracks.
Default is BLK.

SPILLKEEP=number_of_spill_files
Optional. The number of spill files that the UOW Cleanser retains for re-assignment. The UOW Cleanser retains spill files instead of deallocating them so that the files are available to be reassigned to new transactions. This feature is intended to prevent excessive file deallocation and allocation activity.
Valid values are 0 through 999. On z/OS and i5/OS, the default is 3. On Linux, UNIX, and Windows, the default is 0.

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

TIMESTAMP={LOG|COMMIT}
Optional. The type of timestamp that PowerExchange records in the generated DTL__CAPXTIMESTAMP column of each change record for a transaction. Specify this parameter only if you want to display the commit timestamp instead of the timestamp from the source logs or data sets.
Options are:
- LOG. The timestamp of a change on the source database, as recorded by the DBMS in the source database logs or data sets near the time when the change is made. For more information, see Appendix B, “DTL__CAPXTIMESTAMP Time Stamps” on page 387.
- COMMIT. The timestamp of the transaction commit on the source database. Specify this option if you use the timestamp to calculate latency.
Default is LOG.

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

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

You can run the PowerExchange Listener as a started task or batch job. Because the PowerExchange Listener is long running, usually the preferred method is to run the PowerExchange Listener as a started task.

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

To start the PowerExchange Listener, issue the MVS START command followed by the name of the started task or batch job. For example:

```
START listener_task_name
```

Stopping the PowerExchange Listener

To stop the PowerExchange Listener, you can use the MVS MODIFY command (F) followed by the PowerExchange CLOSE or CLOSE FORCE command.

Use the following syntax:

```
F listener_task_name,CLOSE
```

Command descriptions:

- **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. This command has the same result as the MVS STOP (P) command.

**Note:** You can use the pwxcmd program to issue the close or closeforce command from a Linux, UNIX, or Windows system.

Controlling PowerExchange Listener Tasks

PowerExchange supplies several commands that you can use to control PowerExchange Listener tasks.

Enter PowerExchange Listener commands with the z/OS MODIFY (F) command. Use the following syntax:

```
F listener_task_name,command
```

Use the following commands to list or stop PowerExchange Listener tasks:

- **LISTTASK**
  Lists all active PowerExchange Listener tasks.
- **STOPTASK**
  Stops a specified PowerExchange Listener task.

Alternatively, you can use the pwxcmd program to issue listtask and stoptask commands from a remote Linux, UNIX, or Windows system to a PowerExchange Listener running on a z/OS system. The pwxcmd commands provide the same results.

For more information about PowerExchange Listener commands, see the *PowerExchange Command Reference*. 
This chapter includes the following topics:

- **PowerExchange Agent Overview, 38**
- **Configuring MVS for the PowerExchange Agent, 39**
- **Configuring the PowerExchange Agent, 40**
- **Managing the PowerExchange Agent, 51**
- **Controlling Security for the PowerExchange Agent, 54**

### PowerExchange Agent Overview

The PowerExchange is a PowerExchange CDC component that provides services to other PowerExchange CDC components. It runs as a started task in a separate address space.

The PowerExchange Agent has the following characteristics:

- The PowerExchange Agent interacts with the following PowerExchange CDC components:
  - PowerExchange Listener
  - Environmental Change Capture Routines (ECCRs)
  - PowerExchange Logger for MVS
- The PowerExchange Agent provides the following services to the other CDC components:
  - Gets and manages global queues for the PowerExchange CDC components.
  - Gets new or modified capture registrations from the PowerExchange Listener.
  - Manages data flow between PowerExchange CDC components that run in different address spaces.
  - Manages requests from ECCRs for capture registration information.
  - Provides access to authorized users.
  - Provides a common message log.

- Start the PowerExchange Agent started task after you start the PowerExchange Listener and before you start the PowerExchange Logger and ECCRs. The following start order is recommended:
  1. PowerExchange Listener
  2. PowerExchange Agent
  3. PowerExchange Logger
  4. ECCRs
• The PowerExchange Agent connects to a single PowerExchange Listener. By default, the PowerExchange Agent gets capture registration information from the PowerExchange Listener.

• If you run more than one PowerExchange Listener on the z/OS system and create, edit, or delete capture registrations, make sure that you use the PowerExchange Listener on z/OS that interacts with the PowerExchange Agent. This requirement applies to registration changes that you make from the PowerExchange Navigator and with the DTLUCBRG and DTLURDMO utilities. The PowerExchange Agent can then refresh its memory cache with information from the CCT file that reflects the registration changes.

• The PowerExchange Agent uses the AgentID parameter in the AGENTCTL member, to which the EDMSCTL DD in the Agent JCL points, to create its MVS subsystem. Use the AgentID to communicate with the PowerExchange Agent address space.

• You can control certain aspects of PowerExchange Agent processing by issuing commands from the MVS system console.

• The PowerExchange Agent cannot run as a batch job.

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 AGENTCTL 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 might need to increase the values of the NSYSLX and MAXCAD parameters in the EASYSxx member of the SYS1.PARMLIB library.

To determine whether to increase these values, use the following guidelines:

- 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 the SYS1.PARMLIB to enable increased usage of common data spaces.
  If you change the NSYSLX and MAXCAD parameter, you must IPL the z/OS system for the changes to take effect.

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 z/OS 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 Options Module

Set configuration options for the PowerExchange Agent in the EDMSDIR options module.

When you run the XICDC600 job during installation, the PowerExchange installer assembles and link-edits the EDMSDIR options module and writes it to the PowerExchange USERLIB data set for CDC. The USERLIB is created when you run the SETUPCC1 job during installation. The installer enters values for some EDMSDIR options based on your entries in the z/OS Installation Assistant.

The EDMSDIR options apply to any PowerExchange CDC component that points to this USERLIB library. You can modify the EDMSDIR options, if necessary.
The following table describes the EDMSDIR options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default Value</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENTID</td>
<td>Specifies the name of the default PowerExchange Agent.</td>
<td>EDMA</td>
<td>• Four characters, beginning with a letter, #, @, or $</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• A value that does not conflict with an existing MVS subsystem</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> The value of AGENTID and the LOGGER cannot be the same.</td>
</tr>
<tr>
<td>CCERR</td>
<td>Specifies the action to take when a DB2, IMS synchronous, batch VSAM, or CICS/VSAM ECCR is unable to capture changes for a data source.</td>
<td>CONT</td>
<td>• CONT stops change capture but lets the job continue. Changes to the data resource are not captured. If a /STOP subsys is issued from IMS, work continues but the data to be captured is not logged.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• ABEND ends the job. The transaction does not update the resource. For IMS synchronous capture, the BMP or MPP ends abnormally but the control region continues to function.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Notes:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• With ABEND, if the CICS/VSAM ECCR encounters a serious error or abends during initialization, PowerExchange ends and backs out in-flight CICS transactions on VSAM source files during syncpoint processing. Or if that action is not possible, PowerExchange shuts down the CICS region to ensure data integrity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• If the PowerExchange Logger abends or shuts down, it cannot receive updates from the ECCR. In this case, the CICS/VSAM ECCR causes the CICS update transaction to abend with abend code ASP7 at the transaction syncpoint. Because the transaction does not write updates to the VSAM files that are registered for change capture, PowerExchange does not miss any changes.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Similarly, if the registration status of a file cannot be determined when the file is opened, the CICS/VSAM ECCR abends transactions that update the file, typically with abend code ASP7 at the transaction syncpoint. This situation might occur when the PowerExchange Agent is down or repository access through the PowerExchange Agent has been stopped. Because no updates are written to the files with the uncertain registration status, PowerExchange does not miss any changes.</td>
</tr>
<tr>
<td>CENTURY</td>
<td>Specifies whether to include the century when the PowerExchange CDC components display the date.</td>
<td>Y</td>
<td>• Y displays the century.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• N displays the date without the century.</td>
</tr>
<tr>
<td>Option</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>
| 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. |
| ESLLIB | Specifies the data sets to be concatenated to existing DFSESL DD statements in the 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 data sets, enclosed within parentheses.  
- If you specify multiple data sets, separate them with commas.  
- You can specify up to five data sets. |
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Default Value</th>
<th>Valid Values</th>
</tr>
</thead>
</table>
| LGWAITTO | When CCERR=CONT, specifies the maximum number of seconds that an ECCR waits to write change data records to the PowerExchange Logger queue. Define this option only under the following conditions:  
- ECCR requests to write change data to the PowerExchange Logger queue are blocked because the queue has become full. The queue might become full, for example, because the PowerExchange Logger does not have enough resources to offload data from its active log data sets.  
- You prefer for user applications to continue without error, even though some change data loss might occur when the timeout interval is exceeded.  
- You confirm with Informatica Global Customer Support that this option is appropriate for use in your environment.  
If you use the VSAM/CICS ECCR, this option can be particularly useful. With this ECCR, a blocked queue can cause the CICS region to stop, which potentially affects many users.  
If you enter a non-zero value, PowerExchange issues message PWXEDM172895I at ECCR initialization to indicate that this timeout is in effect. When the timeout interval elapses, PowerExchange issues PWXEDM172895I again and discards change records as long as the timeout persists. When the PowerExchange Logger resumes reading records from its queue, PowerExchange issues message PWXEDM172834I. | 0             | 30 to 21,600 seconds  
A value that is not a multiple of 10 is rounded to the nearest 10-second multiple.  
The default value of 0 disables the LGWAITTO timeout. |
| 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. |
| LOGRGRP  | 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. |
### Customizing EDMSDIR Module Options

After PowerExchange installation, you can modify the values of EDMSDIR module options.

1. Customize and run the JCL in the XICDC600 member of the RUNLIB library.
2. Stop any PowerExchange CDC component that specifies the USERLIB library that contains the EDMSDIR module.
   - These components include:
     - Environmental Change Capture Routines (ECCRs)
     - PowerExchange Agent
     - PowerExchange Condense jobs
     - PowerExchange Listener
     - PowerExchange Logger for MVS
3. Modify the EDMSDIR options.
4. Restart the PowerExchange CDC components that you stopped.

### RELATED TOPICS:
- "EDMSDIR Options Module" on page 41

### Configuring AGENTCTL Parameters

The PowerExchange installation process generates the AGENTCTL member that contains the PowerExchange Agent AGENTCTL parameters. This topic describes these parameters.

The EDMSCTL DD statement in the PowerExchange Agent JCL points to the AGENTCTL parameters.

After installation, you can edit the AGENTCTL parameters by editing the AGENTCTL member in the RUNLIB library. If the AGENTCTL member does not exist, view the EDMSCTL DD statement in the PowerExchange Agent JCL to find the member with these parameters.

**Note:** You must restart the PowerExchange Agent for any change to the AGENTCTL parameters to take effect.
The following table describes the AGENTCTL 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, if the agents are on different z/OS systems. This value must match the value of the AGENTID parameter in the EDMSDIR module.</td>
<td>EDMA</td>
<td>- Four characters, beginning with a letter, #, @, or $.&lt;br/&gt;- A value that does not conflict with a z/OS subsystem.</td>
</tr>
<tr>
<td>CCVACTIVE</td>
<td>Optional. Specifies whether to activate the Batch VSAM ECCR during startup of the PowerExchange Agent.</td>
<td>No</td>
<td>- Yes. Activates the Batch VSAM ECCR during startup.&lt;br/&gt;- No. Does not activate the Batch VSAM ECCR 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.&lt;br/&gt;- No. The PowerExchange Agent does not check authorization.</td>
</tr>
<tr>
<td>CmdPrefix</td>
<td>Optional. An MVS command prefix to use for all PowerExchange Agent commands. This value must not conflict with existing MVS or PowerExchange Agent commands.</td>
<td>Value of AgentID parameter.</td>
<td>One to eight characters. The first must be a letter or one of the following symbols: &lt;br/&gt;¢ . &lt; +</td>
</tr>
<tr>
<td>InitAuthCheck</td>
<td>Optional. Whether to check authorization by issuing a RACROUTE authorization macro whenever anyone makes a request to initialize a PowerExchange Agent service.</td>
<td>No</td>
<td>- Yes. The PowerExchange Agent checks authorization.&lt;br/&gt;- No. The PowerExchange Agent does not check authorization.</td>
</tr>
<tr>
<td>LogBuffLimit</td>
<td>Optional. The data space size to allocate as an integration area for EDMSLOG messages. PowerExchange stores the message log in a data space and not in common storage. Estimate the space in terms of number of messages. For each message, allow 216 bytes.</td>
<td>2000</td>
<td>A number from 1000 through 10000.</td>
</tr>
<tr>
<td>LogClass</td>
<td>Required. The EDMSLOG SYSOUT class.</td>
<td>-</td>
<td>Any valid SYSOUT class.</td>
</tr>
<tr>
<td>LogHold</td>
<td>Optional. Specifies whether the EDMSLOG SYSOUT data is allocated with HOLD=YES.</td>
<td>No</td>
<td>- Yes. Uses HOLD=YES.&lt;br/&gt;- No. Does not use HOLD=YES.</td>
</tr>
<tr>
<td>LogLimit</td>
<td>Optional. The EDMSLOG line limit. When this limit is reached, the PowerExchange Agent allocates another log.</td>
<td>10000</td>
<td>A number from 5000 through 100000.</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>
<tr>
<td>Refreshsscvt</td>
<td>Optional. Causes the system to build a SSCVT. The parameter specifies the current SSCVT address that you want to refresh because it is no longer usable. Use this parameter if all of the following conditions occur: - PowerExchange issues message PWXEDM172020E. - The STARTUP parameter is set to COLD. - You do not need to IPL because of the failure.</td>
<td>-</td>
<td>An eight-character hexadecimal address that you get from message PWXEDM172020E.</td>
</tr>
<tr>
<td>RepositoryDSN</td>
<td>Required. The data set name that the PowerExchange Agent repository uses for either the AGENTREP data set or CCT data set.</td>
<td>-</td>
<td>A valid cataloged data set name.</td>
</tr>
<tr>
<td>RepositoryMode</td>
<td>Required. The type of repository.</td>
<td>-</td>
<td>Valid values are DETAIL or EDP. Use DETAIL for PowerExchange CDC.</td>
</tr>
<tr>
<td>Startup</td>
<td>Optional. Whether, during startup, the PowerExchange Agent creates a data space or uses an existing data space, if one exists.</td>
<td>WARM</td>
<td>- WARM. Reuses a data space if one exists. - COLD. Create a data space.</td>
</tr>
<tr>
<td>TaskLimit</td>
<td>Optional. The amount of data space storage used as an integration area for concurrent PowerExchange Agent tasks. Specify this limit in terms of the maximum number of concurrent task control blocks (TCBs) that can request services from the PowerExchange Agent. Allow 128 bytes for each control block.</td>
<td>500</td>
<td>A number from 150 through 1500.</td>
</tr>
</tbody>
</table>

**Configuring AGENTREP Parameters**

The AGENTREP data set, which is created during the installation, specifies PowerExchange Agent parameters that are related to controlling the capture registration subtask.

**Note:** The AGENTREP data set is created as a sequential data set. Do not change it to a PDS member.

The AGENTREP data set name is specified in the RepositoryDSN parameter in the AGENTCTL parameters, as follows:

```
RepositoryDSN=hlq.AGENTREP
```

The *hlq* variable is the PowerExchange high-level qualifier that is specified in the z/OS Installation Assistant during installation.

Alternatively, you can specify the name of the PowerExchange CCT data set in the RepositoryDSN parameter, as follows:

```
RepositoryDSN=hlqvs.CCT
```

The *hlqvs* variable is the PowerExchange high-level qualifier for VSAM, which is specified in the z/OS Installation Assistant.
For improved performance and resource usage, Informatica recommends that you use the AGENTREP data set rather than the CCT data set as the PowerExchange Agent repository.

- If 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 no changes have occurred.
- If 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 any changes have occurred. This activity results in unnecessary I/O activity and CPU overhead in the PowerExchange Agent address space.

The following table describes the AGENTREP parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BackToBackDelay</td>
<td>Determines the minimum time interval between update notifications. You can use this parameter to reduce or eliminate the number of registration change messages in environments where repositories are modified frequently. When messages are suppressed, you can use the Repository Display command to display the latest change information. Default is 0, which does not suppress any messages.</td>
</tr>
<tr>
<td>Cache1</td>
<td>Copy 1 of the sequential cache data set. No default value.</td>
</tr>
<tr>
<td>Cache2</td>
<td>Copy 2 of the sequential cache data set. No default value.</td>
</tr>
<tr>
<td>Location</td>
<td>The name of the PowerExchange Listener that is retrieved from the PowerExchange configuration member. No default value.</td>
</tr>
<tr>
<td>RestartInterval</td>
<td>Interval at which the Agent subtask that interrogates the PowerExchange Listener for capture registration changes is restarted. This interval is expressed as the number of UpdateInterval intervals. Restarting effectively frees memory that was allocated to the TCP/IP layer. Default is 60.</td>
</tr>
<tr>
<td>UpdateInterval</td>
<td>The interval, in minutes, at which PowerExchange checks for registration changes. PowerExchange issues messages in the Agent output when it checks for changes. Default is 1.</td>
</tr>
</tbody>
</table>

Customizing the PowerExchange Agent JCL

The PowerExchange Agent runs as a started task. You must customize the JCL for your installation and copy it to a system procedure library (PROCLIB) for started tasks.

PowerExchange provides sample JCL for the PowerExchange Agent. The XIZZZ998 cleanup job in the RUNLIB library, which runs during PowerExchange installation, moves the PowerExchange Agent JCL to the PowerExchange PROCLIB library.

The name of the PowerExchange Agent JCL member in the PROCLIB library is the value that was specified in the Agent / Logger Prefix field in the z/OS Installation Assistant followed by the letter A. Based on the default Agent / Logger Prefix value of PWX, the default member name for the PowerExchange Agent JCL in the PROCLIB library is PWXA.
The following table describes the PowerExchange Agent JCL statements and parameters:

<table>
<thead>
<tr>
<th>JCL Statements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>The PGM parameter in the EXEC statement must specify the PowerExchange Agent module name EDMSTART.</td>
</tr>
<tr>
<td>START</td>
<td>Controls how the PowerExchange Agent starts. Include the STARTUP symbolic parameter to control whether the PowerExchange Agent WARM or COLD starts. This parameter overrides the installation option for warm or cold starting the PowerExchange Agent. For a cold start, specify: START agent_proc_name,STARTUP=COLD} The variable agent_proc_name is the name that was assigned to the PowerExchange Agent procedure at installation. If you start the PowerExchange Agent without the STARTUP parameter, it starts with the options that you specified during installation. For a WARM start, the PowerExchange Agent uses an existing Agent environment, if one exists. For a COLD start, the Agent creates a new Agent environment and completes startup processing that is the same as for a first-time start. Use the following syntax to start the PowerExchange Agent with all of the installation options: START agent_proc_name To start the PowerExchange Agent with all of the installation options except the option that determines whether the Agent COLD or WARM starts, use the following syntax: START agent_proc_name,STARTUP={COLD</td>
</tr>
<tr>
<td>STEPLIB or JOBLIB DD</td>
<td>Includes the PowerExchange load libraries, hlq.LOAD and hlq.LOADLIB. This statement is required even if you specify the load libraries in the LNKLIST concatenation. The PowerExchange Agent loads certain modules from the STEPLIB or JOBLIB.</td>
</tr>
<tr>
<td>EDMPARMS DD</td>
<td>Specifies the name of the user library, your.USERLIB, that contains the EDMSDIR options module that is 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 get the configuration options.</td>
</tr>
<tr>
<td>EDSMCTL DD</td>
<td>Specifies the data set that contains the PowerExchange Agent startup parameters. Informatica 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>Specifies 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. The installation process customizes this JCL with the values that you specify in the z/OS Installation Assistant.

The sample PowerExchange Agent PROC is in the AGENTSTP member of the RUNLIB. This member is copied to the PROCLIB library using a member name that consists of the value that you specified in the PowerExchange Agent / Logger Prefix field during installation followed by the letter A.

The following sample JCL is for the PowerExchange Agent:

```bash
//PWXA  PROC STARTUP=WARM,HLQ=YOUR.INSTALL.HLQ,
//       RUNLIB=YOUR.INSTALL.HLQ. RUNLIB,
//       LOGGER=PWXL
//*/ PowerExchange Agent
//*/
```
Sample Messages from Starting the PowerExchange Agent

The following sample text shows the PowerExchange Agent startup messages:

PWXEDM1720021 EDMINIT0: ChangeDataMove. Version 2.4.04. Release date: 20031015
PWXEDM1720081 EDMINIT0: EDM Agent Configuration Parameters:
PWXEDM1720101 EDMINIT0: AgentID=PWXA
PWXEDM1720101 EDMINIT0: LogClass=* 
PWXEDM1720101 EDMINIT0: LogHold=NO
PWXEDM1720101 EDMINIT0: LogLimit=5000
PWXEDM1720101 EDMINIT0: LogBuffLimit=2000
PWXEDM1720101 EDMINIT0: TaskLimit=500
PWXEDM1720101 EDMINIT0: LSNPort=0
PWXEDM1720101 EDMINIT0: CmdPrefix=PWXA
PWXEDM1720101 EDMINIT0: RepositoryDN=EDMSR.DETAIL.VB11.AGENTREP
PWXEDM1720101 EDMINIT0: RepositoryMode=Detail
PWXEDM1720101 EDMINIT0: InitAuthCheck=No
PWXEDM1720101 EDMINIT0: CmdAuthCheck=No
PWXEDM1720101 EDMINIT0: CCVActive=YES
PWXEDM1720101 EDMINIT0: SysplexLogDays=0
PWXEDM1720101 EDMINIT0: STARTUP=WARM <==== PARM ON STARTUP CMD
PWXEDM1720101 EDMINIT0: ServiceModule=EDMSDUMY
PWXEDM1720101 EDMINIT0: ServiceModule=EDMSQIO
PWXEDM1720101 EDMINIT0: ServiceModule=EDMSCTQ
PWXEDM1720101 EDMINIT0: DeloldPMods=0
PWXEDM1720101 EDMINIT0: EDMAgentTrace=off
PWXEDM1720101 EDMINIT0: TRACEOPTIONS=None <==== DEFAULT
PWXEDM1720101 EDMINIT0: PATROLKM=NO <==== DEFAULT
PWXEDM1720101 EDMINIT0: PXDATASPACEINIT=100 <==== DEFAULT
PWXEDM1720101 EDMINIT0: PXDATASPACEMAX=500 <==== DEFAULT
PWXEDM1720101 EDMINIT0: MSGPREFIX=PWX <==== DEFAULT
PWXEDM1720241 EDMINIT2: New SSCVT built for EDM Agent PWXA. Addr=00C16328
PWXEDM1720241 EDMINIT2: New SAST built for EDM Agent PWXA. Addr=00C16210
PWXEDM1720641 EDMINIT4: EDM Agent Dataspace created.
Name=00018EDM, STOKEN=8001FD100000056,Blocks=234
PWXEDM1720691 EDMEXEC0: Subtask ATTACHd. Module=EDMSCCV0,TaskID=CCV,RC=0
PWXEDM1720711 EDMEXEC0: Subtask initialization completed. TaskId=CCV
PWXEDM1720691 EDMEXEC0: Subtask ATTACHd. Module=EDMSDIS0,TaskId=DIS,RC=0
PWXEDM1720231 EDMEXEC0: Active= 1, Inactive=0. PWXA
PWXEDM1720711 EDMEXEC0: Subtask initialization completed. TaskId=DIS
PWXEDM1720691 EDMEXEC0: Subtask ATTACHd. Module=EDMSREPO,TaskId=REP,RC=0
PWXEDM1720711 EDMEXEC0: Subtask initialization completed. TaskId=REP
PWXEDM1720691 EDMEXEC0: Subtask ATTACHd. Module=EDMSDPP0,TaskId=DSF,RC=0
PWXEDM1720711 EDMEXEC0: Subtask initialization completed. TaskId=DSF
PWXEDM1720691 EDMEXEC0: Subtask ATTACHd. Module=EDMSLOG0,TaskId=LOG,RC=0
Managing the 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 45

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 EASYSxx PARMLIB member. The SCOPE=COMMON data space limit is specified in the MAXCAD parameter in the MVS EASYSxx PARMLIB member.

Stopping the PowerExchange Agent

PowerExchange Agent commands use the MVS command prefix defined by the CmdPrefix statement in the PowerExchange AGENTCTCL 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>Command</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</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>
<tr>
<td>START</td>
<td>START DIS starts the DIS subtask, which processes DISPLAY commands.</td>
</tr>
<tr>
<td></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></td>
<td>START REP starts the REP subtask, which retrieves PowerExchange repository information.</td>
</tr>
<tr>
<td>STOP</td>
<td>STOP DIS stops the DIS subtask, which processes DISPLAY commands.</td>
</tr>
<tr>
<td></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></td>
<td>STOP REP stops the REP subtask, which retrieves PowerExchange repository information.</td>
</tr>
</tbody>
</table>

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

The PowerExchange Agent cache data sets are created during installation when the job in the SETUPCC1 member of the RUNLIB library runs.

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 REPSTATUS 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 EDMREPO: REPSTATUS command accepted by EDM Agent USA
PWXEDM1812161 DTERIOM: Repository status follows:
PWXEDM1812171 DTERIOM: PWX-10052 last refresh attempt Tue Jan 22 15:23:39 2008
PWXEDM1812171 DTERIOM: PWX-10053 current change identifier 20080122152344
PWXEDM1812171 DTERIOM: PWX-10055 configuration type repository AUSQA.PWX.AGENTREP
PWXEDM1812171 DTERIOM: PWX-10057 location node1
PWXEDM1812171 DTERIOM: PWX-10058 cache (1) AUSQA.PWX.C1.CACHE
PWXEDM1812171 DTERIOM: PWX-10058 cache (2) AUSQA.PWX.C2.CACHE
PWXEDM1812181 DTERIOM: End of repository status
```

If the cache data sets are not specified in the AGENTREP parameters, the REPSTATUS 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 REOPEN 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.

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_ID.REGISTER 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, 57**
- **Planning for the PowerExchange Logger for MVS, 59**
- **Configuring the PowerExchange Logger for MVS, 60**
- **Managing the PowerExchange Logger for MVS, 69**
- **Monitoring the PowerExchange Logger for MVS, 72**
- **Managing Log and Restart Data Sets, 73**
- **Using Post-Log Merge, 90**

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.
The following figure shows the PowerExchange Logger data flow and 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 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).

#### PowerExchange Logger Configuration Considerations

Before you begin configuration, review the following considerations related to PowerExchange Logger usage.

- A PowerExchange Logger can log data from multiple ECCRs that operate on the same z/OS system. By using Post-Log Merge, you can access changes from multiple MVS systems as if they were stored in a single PowerExchange Logger environment.

- If you use multiple PowerExchange Loggers, you need a copy of the EDMSDIR default options module for each PowerExchange Logger instance. Because you cannot rename the EDMSDIR module, you must allocate a separate USERLIB for each copy of EDMSDIR. To reduce the chance of data loss, use 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 of the changes that have been captured but not yet applied. You must reinitialize the PowerExchange processes that consume data from the PowerExchange Logger. If you restart these processes in the normal manner, PowerExchange uses the last-read PowerExchange
Logger RBA to generate the restart point. However, reinitialization of the PowerExchange Logger invalidates the last-read RBA.

- On a z/OS version 1.12 system, PowerExchange Logger active log data sets cannot be larger than 4 GB. Also, you must apply IBM APAR OA34369. Otherwise, the PowerExchange Logger might abend when it tries to use data-in-virtual (DIV) services to access its log data sets.

### Configuring the PowerExchange Logger for MVS

To use the PowerExchange Logger for MVS for CDC, you must complete the following configuration tasks:

- Configure the EDMUPARM options module.
- Customize the PowerExchange Logger JCL, and copy the JCL to the PROCLIB to run the PowerExchange Logger as a started task.
- Verify that the active log data sets and emergency restart data sets were created at installation.
- Define the active and archive log data sets to the emergency restart data set.

### Configuring the EDMUPARM Options Module

You can specify PowerExchange Logger options in the EDMUPARM options module, which is in the USERLIB library. This module is created by the SETUPCC2 job in the RUNLIB library during PowerExchange installation.

Before you configure the EDMUPARM options module, consider the following issues:

- If you use dual logging and dual emergency restart data sets, allocate the primary and secondary data sets to different volumes. This practice makes data recovery possible when a disk failure occurs.
- To create an effective logging configuration, balance the following guidelines:
  - Size the input and output buffers based on the volume of captured change data.
  - Define the number of active log data sets based on the volume of captured change data and how rapidly the data can be archived. Minimum is 3, and maximum is 31.
  - Size the active log data set based on the volume of change data and the size requirements of the archive media.
  - Size the archive log data set based on the active log data set size, the block size of the archive data sets, and the type of device to which you are archiving.

### RELATED TOPICS:

- "Size and Number of Active Log Data Sets" on page 74

### DEFINE Statement

Use the DEFINE statement to configure PowerExchange Logger system, archive, and logging options. This statement is required.

**Syntax:**

The DEFINE statement has the following general syntax:

```plaintext
DEFINE
   LOGGER_TITLE=name
```
Substatements:
The following table describes the substatements:

<table>
<thead>
<tr>
<th>Substatement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGGER_TITLE</td>
<td>Required. Specifies a PowerExchange Logger name of up to 16 characters in length,</td>
</tr>
<tr>
<td>SYSTEM_OPTIONS</td>
<td>Optional. Specifies configuration options for the PowerExchange Logger.</td>
</tr>
<tr>
<td>ARCHIVE_OPTIONS</td>
<td>Optional. Specifies configuration options for the archive log data sets.</td>
</tr>
<tr>
<td>LOGGING_OPTIONS</td>
<td>Optional. Specifies configuration options for the active and archive log data sets.</td>
</tr>
</tbody>
</table>

Usage Notes:
Enter the substatements in a single DEFINE statement. If you omit a substatement, the PowerExchange Logger uses its default value.

The SYSTEM_OPTIONS, ARCHIVE_OPTIONS, and LOGGING_OPTIONS substatements each have unique parameters. You must specify at least one substatement with at least one parameter.

SYSTEM_OPTIONS Parameters
In the SYSTEMS_OPTIONS substatement of the DEFINE statement, you can set PowerExchange Logger system parameters, such as those that control the Logger name, checkpoint processing, and tracing.

Syntax:

```
SYSTEM_OPTIONS
  [LOGGER_NAME=id,]
  [CHKPT_FREQUENCY=nnnn,]
  [START_TRACE=Y|N,]
  [SUFFIX=s,]
  [TIMER_INTERVAL=nnnn,]
  [TIME_CHKPT_FREQ=nn]
```

Parameters:
The following table describes the SYSTEM_OPTIONS parameters:

<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 PowerExchange Logger, the value must match the LOGGER parameter in the PowerExchange Agent EDMSDIR options module and the LOG parameter on LRAPI 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 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. For the trace output to be received, the EDMTRACE DD statement must be in the Logger JCL.</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. The value Y causes additional overhead in the Logger. Enter Y 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 operations, such as freeing unused virtual storage or detecting inactive tasks that need to be POSTed.</td>
<td>An interval in hundredths of seconds in the following range:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Minimum is 50 (.5 seconds).</td>
</tr>
<tr>
<td></td>
<td></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 Post-Log Merge environment. This parameter is used only when running Post-Log Merge.</td>
<td>The checkpoint frequency expressed in number of elapsed TIMER_INTERVAL 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. If you use the default TIMER_INTERVAL value of 100 hundredths of a second with the default of 30 for this parameter, a time-based checkpoint record is written every 30 seconds (100 * 1/100 * 30).</td>
</tr>
</tbody>
</table>

**Usage Notes:**

You must specify at least one parameter.

If you specify multiple parameters, use a comma (,) as a separator character. Do not put a comma at the end of the last parameter.
ARCHIVE_OPTIONS Parameters

In the ARCHIVE_OPTIONS substatement of the DEFINE statement, you can set parameters for allocating and managing archive log data sets.

Syntax:

```
ARCHIVE_OPTIONS
    [PREFIX_COPY1=prefix,]
    [PREFIX_COPY2=prefix,]
    [ARCHIVE_BLKSIZE=number,]
    [ARCHIVE_DACL=sms_dataclas,]
    [ARCHIVE_DACL2=sms_dataclas,]
    [ARCHIVE_MGCL=sms_mgmtclas,]
    [ARCHIVE_MGCL2=sms_mgmtclas,]
    [ARCHIVE_RTPD=number_of_days,]
    [ARCHIVE_RTPD2=number_of_days,]
    [ARCHIVE_STCL=sms_storclas,]
    [ARCHIVE_STCL2=sms_storclas,]
    [ARCHIVE_UNIT=unit_name,]
    [ARCHIVE_UNIT2=unit_name,]
    [ARC_UNIT_CNT=number,]
    [PRIM_SPACE=number,]
    [SEC_SPACE=number,]
    [SPACE_ALLOC=type_of_units]
```

Parameters:

The following table describes the ARCHIVE_OPTIONS parameters:

<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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value can be up to 17 alphanumeric characters long and must follow MVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>data set name rules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The value can be up to 17 alphanumeric characters long and must follow MVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>data set name rules.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ARCHIVE_UNIT parameter.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<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>A number from 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>A number from 0 through 9999. 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 eight alphanumeric characters long. Informatica 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. Use this parameter only if you want to set the value differently for the second data set.</td>
<td>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 Specify a device type or unit name up to 8 alphanumeric characters long.</td>
</tr>
</tbody>
</table>
Parameter | Description | Valid Values
---|---|---
ARC_UNIT_CNT | Specifies the number of DASD units to use for archiving. | 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.

PRIM_SPACE | Specifies the primary space allocation for DASD data sets in the unit type specified by SPACE_ALLOC. | A number greater than 0. Default is 4320 blocks.

SEC_SPACE | Specifies the secondary space allocation for DASD data sets in the unit type that you specify in SPACE_ALLOC. | A number greater than 0. Default is 540 blocks.

SPACE_ALLOC | Specifies the type of units in which primary and secondary space are allocated. | - BLK. Allocates space in blocks. - CYL. Allocates space in cylinders. - TRK. Allocates space in tracks. Default is BLK.

Usage Notes:
You must specify at least one parameter for this statement.
If you specify multiple parameters, use a comma (,) as a separator character. The last parameter must not end in a comma.

LOGGING_OPTIONS Parameters
In the LOGGING_OPTIONS substatement of the DEFINE statement, you can set logging parameters for the PowerExchange Logger.

Syntax:
```
LOGGING_OPTIONS
 [LOG_INBUFF=number,]
 [LOG_OUTBUFF=number,]
 [ACTIVE_LOG_MODE=mode,]
 [ARCHIVE_LOG_MODE=mode,]
 [ERDS_LOG_MODE=mode]
```

Parameters:
The following table describes the LOGGING_OPTIONS parameters:

Parameter | Description | Valid Values
---|---|---
LOG_INBUFF | Defines the number of 4 KB buffers used for reading the active and archive logs. | 1 through 60 (decimal). Default is 28.

LOG_OUTBUFF | 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. | 1 through 50 (decimal).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
</table>
| ACTIVE_LOG_MODE | Specifies whether the PowerExchange Logger writes to one or two active log data sets at a time. | • **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.  
Default is DUAL. Informatica strongly recommends that you use dual logging. |
| 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.  
Default is DUAL. Informatica strongly recommends that you use dual logging. |
| 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.  
Default is DUAL. Informatica strongly recommends that you use dual logging. |

**Usage Notes:**

You must specify at least one parameter in the statement.

If you specify multiple parameters, use a comma (,) as a separator character. The last parameter must not end in a comma.

**END Statement**

Use the END statement to indicate the end of input for the DEFINE statement.

This statement has no substatements or parameters.

**Verifying That the Active Log and Emergency Restart Data Sets Were Created Correctly**

PowerExchange creates the PowerExchange Logger active log data sets and emergency restart data sets (ERDS) at installation when you run the XICDC500 in the RUNLIB library.

The active logs are VSAM linear data sets that are defined using IDCAMS. The ERDS data sets are VSAM KSDS data sets.

Verify that these data sets exist and were defined in accordance with the following guidelines:

• Do not specify secondary allocation.
• Specify a single VOLSER in the VOLUME parameter.
• Do not 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 your environment uses SMS, use an SMS STORCLAS that does not specify GUARANTEED SPACE=YES.
Active and Archive Log Entries in the ERDS

Active and archive log data sets must have entries in the ERDS for the PowerExchange Logger to access them.

PowerExchange defines the active logs at installation when you run the SETUPCC2 job that is in the RUNLIB library. This job runs the PowerExchange Logger in batch mode to create the EDMUPARM options module and define the active logs to the ERDS.

During archive processing, the PowerExchange Logger automatically defines archive logs to the ERDS.

Also, you can use the DEFINE_LOG command to define the active and archive logs to the ERDS.

RELATED TOPICS:
- "Defining Log Data Sets to the ERDS" on page 84

Customizing the PowerExchange Logger JCL

The PowerExchange Logger for MVS 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 long running. You must customize the PowerExchange Logger JCL for your installation and then copy it to a system PROCLIB library for started tasks.

PowerExchange provides sample JCL for the PowerExchange Logger. The XIZZZ998 cleanup job in the RUNLIB library, which runs during installation, moves the PowerExchange Logger JCL to the PowerExchange PROCLIB library.

The name of the PowerExchange Logger JCL member in the PROCLIB library is the value that you specify for the PowerExchange Agent / Logger Prefix field in the z/OS Installation Assistant followed by the letter L. Based on the default PowerExchange Agent / Logger Prefix value of PWX, the default name for the PowerExchange Logger JCL member in the PROCLIB library is PWXL.

The PowerExchange Logger JCL includes the following statements and parameters:

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

Invokes the PowerExchange Logger.

The PARM parameter can contain the following required and optional positional parameters:

- **logger_id**
  The PowerExchange Logger identifier that is 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. The option for running the PowerExchange Logger in batch mode to perform maintenance activities. 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 on the z/OS system where it runs. If this SMF ID value is not unique within the Post-Log Merge group, you can use this parameter to provide a unique SMF ID 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=0,M,TIME=NOLIMIT,
// PARM='&LOGNAME,,,&SMFID',ACCT=XXX
```

Valid values are 1 through 4 alphanumeric characters in length.

**JOBLIB or STEPLIB DD**

Defines the LOAD library that contains the PowerExchange Logger load modules. This library must be APF-authorized.

**EDMPARMS DD**

Defines the user library, USERLIB, that contains the EDMUPARM options module that is associated with the PowerExchange Logger.

If you do not include an EDMPARMS DD statement in the JCL, or if you specify a library that does not contain the EDMUPARM options module, PowerExchange uses the JOBLIB or STEPLIB concatenation to get the Logger configuration options.

**ERDS01 DD**

Defines the name of the primary emergency restart data set.

**ERDS02 DD**

Optional. Defines the name of the dual emergency restart data set when DUAL is specified for the ERDS_LOG_MODE parameter in the EDMUPARM options module.

**SYSPRINT DD**

Defines the output data set for MVS system messages.

**EDMTRACE DD**

Defines the output data set for the common services trace.

Include this DD statement only at the request of Informatica Global Customer Support.

### Sample JCL Procedure for the PowerExchange Logger

PowerExchange provides a sample PROC for running the PowerExchange Logger for MVS. The installation process customizes this JCL with the values that you specify in the z/OS Installation Assistant.

The sample PowerExchange Logger PROC is provided in the LOGERSTP member, which is copied to the PROCLIB library. The member name is comprised of the value that was entered in the **PowerExchange Agent / Logger Prefix** field during installation followed by the letter L.

The following sample JCL is for the PowerExchange Logger:

```
//PWXL PROC HLQ=PWX,LOGGER=PWXL,
// HLQVSMS=PWX
//*-------------------------------------------------------------
//LOGGER EXEC PGM=EDMLC000,REGION=0,M,TIME=NOLIMIT,
// PARM=&LOGGER,ACCT=XXX
//STEPLIB DD DISP=SHR,DSN=&HLQS..LOAD
//EDMPARMS DD DISP=SHR,DSN=&HLQS..&LOGGER..USERLIB
//SYSPRINT DD SYSOUT=*
//SYSDUMP DD SYSOUT=*```
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 PWXL
```

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:

```
STOP PWXL
```

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

Controlling the PowerExchange Logger for MVS

Use PowerExchange commands to control the PowerExchange Logger for MVS and display information about its processing.

Enter a PowerExchange Logger command with the MVS MODIFY (F) command. Use the following syntax:

```
F logger_proc_name,command
```

The following table describes each PowerExchange Logger command:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFINE_LOG</td>
<td>Adds PowerExchange Logger log definitions to the restart data set. You can add definitions for the following types of log data sets:</td>
</tr>
<tr>
<td></td>
<td>- Additional active log definitions</td>
</tr>
<tr>
<td></td>
<td>- Replacement active log definitions</td>
</tr>
<tr>
<td></td>
<td>- Replacement archive log definitions</td>
</tr>
<tr>
<td>DELETE_LOG</td>
<td>Deletes all information about a specified PowerExchange Logger log data set from the restart data set. Run this command periodically to delete information about obsolete archive log data sets.</td>
</tr>
<tr>
<td>DISPLAY OBJECT=CONNECTION</td>
<td>Displays information about tasks connected to the PowerExchange Logger.</td>
</tr>
<tr>
<td>DISPLAY OBJECT=LOG</td>
<td>Displays information about the active or archive log data sets.</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRINT</td>
<td>Prints log records to a dynamically allocated SYSOUT data set.</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>
<tr>
<td>STOP</td>
<td>Stops the PowerExchange Logger. The MVS STOP command can also be used.</td>
</tr>
</tbody>
</table>

For more information about these commands, including the syntax and parameters, see the PowerExchange Command Reference.

### Overriding Log-Read API Timed Defaults

After the Log-Read API (LRAPI) sends commands to the PowerExchange Logger for MVS, it waits a fixed amount of time for a response. In some customer environments, the default wait time for LRAPI commands might be too short. You can override the amount of time that the LRAPI waits for a response to any request type by specifying parameters in the EDMLRPRM DD statement.

### EDMLRPRM Parameters

You can specify the EDMLRPRM DD statement in the JCL for the job that issues the Log-Read API (LRAPI) calls to the PowerExchange Logger. The parameters can be specified in-stream or in a sequential data set.

Use the following DCB attributes if you specify the parameters in a sequential data set that is referenced by the EDMLRPRM DD rather than instream:

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

Specify one parameter statement per record or line. For a comment, enter an asterisk (*) or a hash (#) character in column one. Use the following general syntax for a parameter entry:

```
parameter=parm_value
```

The following table describes the EDMLRPRM parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTLST</td>
<td>Specifies the time LRAPI spends waiting for the PowerExchange Logger to respond to a Resource Interest List command. This wait period starts after the PowerExchange Logger issues the PWXEDM172791I message. Default is 6000 hundredths of seconds (60 seconds).</td>
</tr>
<tr>
<td>REQTRN</td>
<td>Specifies the time LRAPI spends waiting for the PowerExchange Logger to start sending data. This wait period starts after the PowerExchange Logger issues the PWXEDM263011I message. Default is 24000 hundredths of seconds (240 seconds).</td>
</tr>
<tr>
<td>SIGNON</td>
<td>Specifies the time LRAPI spends trying to connect to the PowerExchange Logger. This time period starts after the PowerExchange Logger issues the PWXEDM263010I message. Default is 6000 hundredths of seconds (60 seconds).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>STPTRN</td>
<td>Specifies the time LRAPI spends waiting for the PowerExchange Logger to stop sending more data. This wait period starts after the PowerExchange Logger issues the PWXEDM 263014I message. Default is 12000 hundredths of seconds (120 seconds).</td>
</tr>
<tr>
<td>TERM</td>
<td>Specifies the time LRAPI spends disconnecting from the PowerExchange Logger. This time period starts after the PowerExchange Logger issues the PWXEDM263012I message. Default is 4500 hundredths of seconds (45 seconds).</td>
</tr>
</tbody>
</table>

Usually, the Request Data Transfer (REQTRN) command is the command that is most likely to require additional time. When processing a REQTRN command, the PowerExchange Logger might 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 4 minutes and provide the data to the LRAPI, the LRAPI request times out, returns reason code 0x0A0E0062 (LoggerDidNotRespondToCommand), and ends the extraction request. In some environments, the LRAPI might frequently encounter this situation because of operational issues. In these environments, use the REQTRN command to extend the wait time.

**Note:** You can set these parameter values in an EDMLRPRM DD statement in the PowerExchange Listener JCL. However, they then affect each instance of the LRAPI, and all extractions use the same values.

The following example specifies a value of 3 minutes for the REQTRN parameter:

```c
/*
 //** Set REQTRN timeout value to 3 minutes (i.e. 3*60*100 )
 //**
 //EDMLRPRM DD *
 #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:

PWXEDM172672I 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 loggeruid.

  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 for MVS, consider the following rules and guidelines:

- The PowerExchange Logger is a high-performance started task. Informatica 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 74

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 73
• “Size and Number of Active Log Data Sets” on page 74
• “Data Set Size Determination” on page 75
• “Number of Data Sets” on page 76
• “Defining Log Data Sets to the ERDS” on page 84
• “Deleting Log Data Sets from the ERDS ” on page 85
• “Allocating Restart Data Sets” on page 77
• “Adding Active Log Data Set Definitions to the Restart Data Set” on page 78
• “Changing the Size of Active Log Data Sets” on page 79
• “Formatting Log Data Sets” on page 83
• “Recovering Damaged Restart Data Sets” on page 88
• “Moving Log Data Sets to Other Devices” on page 89

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.
• 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 Parameters " on page 63

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.

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.

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:

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

For the overhead rate, use 5-10 percent.
• **Formula 2.** To convert the active log data set size from bytes to tracks:
  \[
  \text{active log data set size in cylinders} = \frac{\text{active log data set size in tracks}}{\text{number of tracks per cylinder}}
  \]

• **Formula 3.** To convert the active log data set size from tracks to cylinders:
  \[
  \text{active log data set size in tracks} = \frac{\text{active log data set size in cylinders} \times \text{number of usable bytes per track}}{\text{number of tracks per cylinder}}
  \]

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:

<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>

**Note:** This table applies only to the PowerExchange Logger and is based on the fact that the PowerExchange Logger writes 4 KB blocks.

### 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:
   \[
   302,400,000 / 49,152 = 6,152 \text{ tracks}
   \]
   \[
   6,152 / 15 = 410 \text{ cylinders}
   \]

### 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). You can use this procedure to expand the restart data sets.

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 options module. To help distinguish restart data sets for different PowerExchange Logger subsystems, include the Logger ID as part of these data sets.

Use the following sample JCL in the #DEFRDS member of the hlq.SAMPLIB library, where hlq is the high-level qualifier that you specified at installation, to define the restart data set in dual mode:

```haskell
// 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. REPLACED ????? 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.**
// **---------------------------------------------------------------**
// ALLOCRES EXEC PGM=IDCAMS,REGION=4M
// SYSPRINT DD SYSOUT=*
// SYSDUMP DD SYSOUT=*
// SYSD * DELETE (YOUR.???? .ERDS01) ERASE
// DELETE (YOUR.???? .ERDS02) ERASE
// SET MAXCC = 0
// DEFINE CLUSTER
// (NAME (YOUR.???? .ERDS01) -
// VOLUMES (VVVVVV) -
// SHAREOPTIONS (2,3) -
// DATA -
// (NAME (YOUR.???? .ERDS01.DA ) -
// RECORDS (200) -
// RECORDSIZE (4089 4089) -
// CONTROLLINTERVALSIZE (4096) -
// FRESSPACE (0 20) -
// KEYS (4 0) ) -
// INDEX -
// (NAME (YOUR.???? .ERDS01 .INDEX) -
// RECORDS (5 5) -
// CONTROLLINTERVALSIZE (1024) }
// DEFINE CLUSTER
// (NAME (YOUR.???? .ERDS02) -
// VOLUMES (VVVVVV) -
// SHAREOPTIONS (2,3) -
// DATA -
// (NAME (YOUR.???? .ERDS02 .DATA) -
// RECORDS (200) -
// RECORDSIZE (4089 4089) -
// CONTROLLINTERVALSIZE (4096) -
// FRESSPACE (0 20) -
// KEYS (4 0) ) -
// INDEX -
// (NAME (YOUR.???? .ERDS02 .INDEX) -
// RECORDS (5 5) -
// CONTROLLINTERVALSIZE (1024) }
// **---------------------------------------------------------------**
```

To allocate restart data sets:

1. Make a working copy of the sample #DEFRDS member. Then 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.

**RELATED TOPICS:**
- "Data Set Size Determination" on page 75

**Adding Active Log Data Set Definitions to the Restart Data Set**

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.

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

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.

Use the following sample JCL in the #ADDLOGS member of the hlq.SAMPLIB library, where hlq the high-level qualifier that you specified during installation, to add active log data sets:

```jcl
//
//  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=EDMLC000,PARM='????,BATCH'
//STEP1B DD DISP=SHR,DSN=HLQ.LOAD == PX LOAD
//EDMPARMS DD DISP=SHR,DSN=YOUR.USERLIB == EDMSDIR,EDMPARM
//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
  DSNNAME=YOUR.????,PRILOG.DS03,
  COPY=PRILOG
END
DEFINE_LOG
  DSNNAME=YOUR.????,SECLOG.DS03,
  COPY=SECLOG
END
/*
```
Note: In this 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.

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

1. Make a working copy of the sample #ADDLOGS member. Then, edit the copy as required.

The following table describes the JCL statements:

<table>
<thead>
<tr>
<th>JCL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specify the EDMLC000 program.</td>
</tr>
<tr>
<td>PARM</td>
<td>Include the Logger name, followed by BATCH.</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.</td>
</tr>
<tr>
<td>EDMPARMS DD</td>
<td>Specify the name of the user library (YOUR.USERLIB) that contains the PowerExchange Logger EDMUPARMS module options associated with the PowerExchange Logger that uses these data sets.</td>
</tr>
<tr>
<td>ERDS01 DD</td>
<td>Specify the data set name of the primary restart data set. Make sure that this name matches the name you used when you created this data set.</td>
</tr>
<tr>
<td>ERDS02 DD</td>
<td>Specify the data set name of the backup restart data set. Ensure that this name matches the name you used when you created this data set.</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 PowerExchange Logger command, DEFINE_LOG.</td>
</tr>
</tbody>
</table>

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

Related Topics:
- “Sample JCL Procedure for the PowerExchange Logger” on page 68
- “Size and Number of Active Log Data Sets” on page 74

Changing the Size of Active Log Data Sets

You can change the size of existing active log data sets.

First 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: You must shut down the PowerExchange Logger and stop all capture and extraction tasks.

1. Make a copy of the sample #SIZELOG member in the h/lq.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 EDMLUTIL0 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 LNKLIST 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 h/lq.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, DNAM='* END
```

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

```
  LOG START
  PWXEDM172502I EDM Logger BATCH initialization in-progress product level V2.4.04
  10/15/2003
  PWXEDM172638I EDM Logger system timestamp for ERDS = 2006.241 16:08:25.95
  DISPLAY OBJECT=LOG, TYPE=ACTIVE, DNAM='* END
  PWXEDM172572I EDM Logger input commands accepted execution started
  PWXEDM172679I EDM Logger LOG ACTIVE report follows:
  | Start RBA | End RBA | Log Dsnme | Status
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>000001FA4000</td>
<td>000002A2FFFE</td>
<td>EDMUSR.PWX.PRILOG.DS01</td>
<td>REUS</td>
</tr>
<tr>
<td>000002A30000</td>
<td>0000034BFFFF</td>
<td>EDMUSR.PWX.PRILOG.DS02</td>
<td>REUS, IN-USE</td>
</tr>
<tr>
<td>000001518000</td>
<td>000001FA3FFF</td>
<td>EDMUSR.PWX.PRILOG.DS03</td>
<td>REUS</td>
</tr>
<tr>
<td>000002A30000</td>
<td>000002A2FFFE</td>
<td>EDMUSR.PWX.SECLOG.DS03</td>
<td>IN-USE</td>
</tr>
<tr>
<td>000001518000</td>
<td>000001FA3FFF</td>
<td>EDMUSR.PWX.SECLOG.DS03</td>
<td>REUS</td>
</tr>
</tbody>
</table>
```

Chapter 4: PowerExchange Logger for MVS
To change the size of the active log data sets, run the customized #SIZELOG job.

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 active 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 75

**Example #SIZELOG Member**

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

```plaintext
//PWXLOGR JOB (MYJOB), 'EXPAND LOGS', CLASS=A, MSGCLASS=X,
// MSGLEVEL=(1,1), NOTIFY=SYSUID
//*---------------------------------------------------------------------*
//RENAME EXEC PGM=IDCMAS,REGION=QM
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER PWX.PRILOG.DS01 -
   NEWNAME (PWX.TEMPLOG1.DS01)
ALTER PWX.PRILOG.DS01.DATA -
   NEWNAME (PWX.TEMPLOG1.DS01.DATA)
ALTER PWX.SECLOG.DS01 -
   NEWNAME (PWX.TEMPLOG2.DS01)
ALTER PWX.SECLOG.DS01.DATA -
   NEWNAME (PWX.TEMPLOG2.DS01.DATA)
ALTER PWX.PRILOG.DS02 -
   NEWNAME (PWX.TEMPLOG1.DS02)
ALTER PWX.PRILOG.DS02.DATA -
   NEWNAME (PWX.TEMPLOG1.DS02.DATA)
ALTER PWX.SECLOG.DS02 -
   NEWNAME (PWX.TEMPLOG2.DS02)
ALTER PWX.SECLOG.DS02.DATA -
   NEWNAME (PWX.TEMPLOG2.DS02.DATA)
```
/*
*-------------------------------------------------------------
*/
//ALLOCS EXEC PG=IDCMSS,REGION=0M,COND=(0,LT)
//SYSLOG DD SYSUT*=
//SYSIN DD *
DEFINE CLUSTER -
  (NAME(PWX.PRLOG.DS01) -
   LINEAR -
   STORCLAS(SMSPool) -
   CYL(300)) -
   DATA -
   (NAME(PWX.PRLOG.DS01.DATA) }
DEFINE CLUSTER -
  (NAME(PWX.SECLOG.DS01) -
   LINEAR -
   STORCLAS(SMSPool) -
   CYL(300)) -
   DATA -
   (NAME(PWX.SECLOG.DS01.DATA) }
DEFINE CLUSTER -
  (NAME(PWX.PRLOG.DS02) -
   LINEAR -
   STORCLAS(SMSPool) -
   CYL(300)) -
   DATA -
   (NAME(PWX.PRLOG.DS02.DATA) }

COMMENTS:
  *-------------------------------------------------------------
//REPRO EXEC PG=IDCMSS,REGION=0M,COND=(0,LT)
//SYSLOG DD SYSUT*=
//SYSIN DD *
REPRO IN DATASET(PWX.TEMPLOG1.DS01) -
   OUTDATASET(PWX.PRLOG.DS01)
REPRO IN DATASET(PWX.TEMPLOG2.DS01) -
   OUTDATASET(PWX.SECLOG.DS01)
REPRO IN DATASET(PWX.TEMPLOG1.DS02) -
   OUTDATASET(PWX.PRLOG.DS02)
REPRO IN DATASET(PWX.TEMPLOG2.DS02) -
   OUTDATASET(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.
*-------------------------------------------------------------
*/
//FORMAT EXEC PG=EMDLUTL0,REGION=0M,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=PWX.LOAD
//PRLOG DD DISP=OLD,DSN=PWX.PRLOG.DS01
//-------------------------------------------------------------
//FORMAT EXEC PG=EMDLUTL0,REGION=0M,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=PWX.LOAD
//PRLOG DD DISP=OLD,DSN=PWX.SECLOG.DS01
//-------------------------------------------------------------
//FORMAT EXEC PG=EMDLUTL0,REGION=0M,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=PWX.LOAD
//PRLOG DD DISP=OLD,DSN=PWX.PRLOG.DS02
//-------------------------------------------------------------
//FORMAT EXEC PG=EMDLUTL0,REGION=0M,COND=(0,LT)
//STEPLIB DD DISP=SHR,DSN=PWX.LOAD

82  Chapter 4: PowerExchange Logger for MVS
Formatting Log Data Sets

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.

Use the following sample JCL in the #EDMLFMT member of the hlq.SAMPLIB library. This JCL formats four log data sets: two primary data sets and two secondary data sets.

```
//PRILOG DD DISP=OLD,DSN=PMX.SECLOG.DS02
//*================================================================================================**

/* PowerExchange CDC - FORMAT ACTIVE LOG DATA SETS FOR LOGGER
/* REPLACE THE FOLLOWING ITEMS WITH PROPER INSTALLATION VALUES
/* 1. JCL DATA SET NAMES
/*================================================================================================***/
/DEFLG01 EXEC PGM=EDMLUTL0
/STEPLIB DD DISP=SHR,DSN=HLQ.LOAD  ===> PMX LOAD
/PRILOG DD DISP=SHR,DSN=YOUR????..PRILOG.DS01  ===> PRI LOG #1
/*================================================================================================***/
/DEFLG02 EXEC PGM=EDMLUTL0
/STEPLIB DD DISP=SHR,DSN=HLQ.LOAD  ===> PMX LOAD
/PRILOG DD DISP=SHR,DSN=YOUR????..SECLOG.DS01  ===> SEC LOG #1
/*================================================================================================***/
/DEFLG03 EXEC PGM=EDMLUTL0
/STEPLIB DD DISP=SHR,DSN=HLQ.LOAD  ===> PMX LOAD
/PRILOG DD DISP=SHR,DSN=YOUR????..PRILOG.DS02  ===> PRI LOG #2
/*================================================================================================***/
/DEFLG04 EXEC PGM=EDMLUTL0
/STEPLIB DD DISP=SHR,DSN=HLQ.LOAD  ===> PMX LOAD
/PRILOG DD DISP=SHR,DSN=YOUR????..SECLOG.DS02  ===> SEC LOG #2
```

Note: In this 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.

To format log data sets:
1. Make a copy of the sample #EDMLFMT JCL procedure, and edit the copy as required.
2. The following table describes the statements that you must specify for each log data set:

<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 LNKLIST 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>

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. Include the subsystem name in a log data set name to distinguish between the log data sets.

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

**Defining Log Data Sets to the ERDS**

The PowerExchange Logger for MVS requires entries for active and archive logs in the emergency restart data set (ERDS) to access the log data sets.

Use the DEFINE_LOG command to define active and archive logs to the ERDS.

Also, PowerExchange defines the active logs at installation when you run the SETUPCC2 job in the RUNLIB library. This job runs the PowerExchange Logger in batch mode to create the EDMUPARM options module and define the active logs to the ERDS.

**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
   DSN=dsname,
   COPY={PRILOG|SECLOG},
   [STARTRBA='start_rba',ENDRBA='end_rba']
END
```

The DEFINE_LOG command has the following syntax for archive logs:

```
DEFINE_LOG
   DSN=dsname,
   STARTRBA='start_rba',ENDRBA='end_rba'
END
```

The following table describes the DEFINE_LOG parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNAME</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. SECLOG indicates that you are defining a secondary log (backup copy).</td>
</tr>
<tr>
<td>Parameter</td>
<td>Definition</td>
<td>Valid Values</td>
</tr>
<tr>
<td>-----------</td>
<td>------------</td>
<td>--------------</td>
</tr>
<tr>
<td>STARTRBA</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>ENDRBA</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td>Indicates that the input for this command is complete. This parameter is required.</td>
<td></td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**
- “Adding Active Log Data Set Definitions to the Restart Data Set” on page 78

**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).

You can run the DELETE_LOG command as part of a batch job or interactively, whenever you need to delete a log. For example, run this command to delete outdated archive log data sets.

**Syntax**

To issue the DELETE_LOG command with the MVS MODIFY command, use the following syntax:

```
F proc_name,DELETE_LOG DSNAMED=logical_dataset_name
```

To issue the DELETE_LOG command as part of a batch job, use the following syntax:

```
DELETE_LOG DSNAMED=logical_dataset_name END
```
The following table describes the DELETE_LOG parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Valid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSNAME</td>
<td>Specifies the fully qualified data set name for the log data set for which</td>
<td>A data set name up to 44 characters</td>
</tr>
<tr>
<td></td>
<td>to delete information from the ERDS.</td>
<td>in length.</td>
</tr>
<tr>
<td>END</td>
<td>Indicates that the input for this command is complete. Required if you</td>
<td>Not applicable.</td>
</tr>
<tr>
<td></td>
<td>include the command in a batch job.</td>
<td></td>
</tr>
<tr>
<td>proc_name</td>
<td>Specifies the PowerExchange Logger procedure name. Required if you</td>
<td>A valid proc.</td>
</tr>
<tr>
<td></td>
<td>issue the command interactively.</td>
<td></td>
</tr>
</tbody>
</table>

**Usage Notes**

- If you use the MVS MODIFY command to run the DELETE_LOG command in interactive mode, the PowerExchange Logger can continue running.
- If you run the DELETE_LOG command as part of a batch job, you must stop the PowerExchange Logger before the batch job runs. Also stop any ECCR that is running against data sources for which the PowerExchange Logger logs changes.

**Sample JCL**

The following sample JCL deletes an archive log data set in batch mode. Replace the question marks (????) with the PowerExchange Logger ID.

```
//jobname JOB
//DEPLG EXEC PGM=EDMLC00G,PARM='????',BATCH
//STEPLIB DD DISP=SHR,DSN=hlq.LOAD
//EDMPARMS DD DISP=SHR,DSN=YOUR.USERLIB
//ERDS01 DD DISP=SHR,DSN=YOUR.????..ERDS02 DD DISP=SHR,DSN=YOUR.????..ERDS02
//SYSPRINT DD SYSPRINT=*
//SYIN DD *
DELETE_LOG
   DSNNAME=archive_log_dataset_name
END
```

**Recovering Damaged Active Log Data Sets**

To recover damaged active log data sets, you can delete the damaged set and replace it with a copy of the corresponding backup log data set. 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. After recovery, restart the PowerExchange Logger.

Use the following sample JCL in the #RCOVADS member of the hlq.SAMPLIB library, where hlq is the high-level qualifier that you specified at installation, to recover damaged active log data sets:

```
//JOB
/*------------------------------------------------------------------------------*/
/* PowerExchange Change Data Capture - RECOVER PRIMARY LOG FROM SECONDARY LOG */
/*------------------------------------------------------------------------------*/
/* 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. */
/*------------------------------------------------------------------------------*/
```
To recover damaged active log data sets:

1. Make a working copy of the sample #RCOVADS member. Then 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>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.</td>
</tr>
</tbody>
</table>

2. Stop the PowerExchange Logger.
3. Run the JCL procedure or job.
4. Restart the PowerExchange Logger.
Recovering Damaged Restart Data Sets

If a restart data set is damaged, delete it and then copy and rename the corresponding backup restart data set to create a replacement data set.

Use the following sample JCL in the #RCOVRDS member of the hlq.SAMPLIB library to delete the damaged restart data set and copy the backup:

```jcl
// 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=IDCMS,REGION=0M
//SYSPRINT DD SYSOUT=*                                                                                           
//SYSPRINT DD *                                                                                                      
//SYSIN DD *                                                                                                           
// DEDELETE YOUR.????..ERDS01) ERASE
SET MAXCC = 0
DEFINE CLUSTER
  (NAME=YOUR.????..ERDS01) -
  VOLUMES(volser) -
  SHAREOPTIONS(2 3) -
DATA
  (NAME=YOUR.????..ERDS01.DATA) -
  RECORDS(100) -
  RECORDSIZE(4096 4096)
  CONTROLINTERVALSIZE(4096) -
  FREESPACE(0 20) -
  KEYS(0 0) -
INDEX
  (NAME=YOUR.????..ERDS01.INDEX) -
  RECORDS(5 5)
  CONTROLINTERVALSIZE(1024) -
/*-------------------------------------------------------------------*
//REPRORDS EXEC PGM=IDCMS,REGION=0M
//SYSPRINT DD SYSOUT=*                                                                                               
//SYSPRINT DD *                                                                                                          
//SYSPRINT DD *                                                                                                           
// SYSPRINT DD SYSOUT=*                                                                                                    
// REPRO IN DATASET(YOUR.????..ERDS02) -
// OUT DATASET(YOUR.????..ERDS01)                                                                                         
/*
```

To recover damaged restart data sets:

1. Make a working copy of the sample #RCOVRDS member, 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>
</tbody>
</table>
2. Stop the PowerExchange Logger.
3. Run the #RCOVRS job.
4. Restart the PowerExchange Logger.

Moving Log Data Sets to Other Devices

You can move log data sets to another device if necessary. You must stop the PowerExchange Logger and then run a JCL procedure or job that allocates space on the target device and moves the data sets.

Use the following sample JCL in the #MOVELOG member of the hlq.SAMPLIB library, where hlq is the high-level qualifier that you specified at installation:

```jcl
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(CC) ) -
DATA -
(NAME(YOUR.????..PRILOG.DS01.DATA) )
/*
/*---------------------------------------------------------------*/
/*REPROLOG EXEC PGM=IDCAMS,REGION=0M */
/*SYSPRINT DD SYSOUT=* */
/*SYSIN DD */
REPRO INDASSET(YOUR.????..TEMPLOG.DS01) -
OUTASSET(YOUR.????..PRILOG.DS01)
/*
/*---------------------------------------------------------------*/
/* NOTE: THE NEXT STEP WILL *NOT* DESTROY THE DATA THAT WAS JUST */
/* COPIED INTO THE PRILOG DATASET. INSTEAD, THE UTILITY DETECTS */
/* WHETHER ANY PART OF THE DATASET HAS BEEN ALLOCATED, BUT NOT */
```
Make a working copy of the sample #MOVELOG member. Then edit the copy as required.

To move log data sets to other devices:

1. Make a working copy of the sample #MOVELOG member. Then 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 LNKLST 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>SYSSIN 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.

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 CDC environments that use Post-Log Merge:

- Change capture for synchronous data sources must run on the z/OS system where the changes are made. Synchronous data sources include IMS, Batch VSAM, and CICS/VSAM.

  Run the ECCR for a synchronous data source on each z/OS system in the sysplex where the changes are made. Also run a PowerExchange Agent on each system where the ECCR runs, and run a Post-Log Merge member Logger on one of the z/OS systems. 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 z/OS system as the Post-Log Merge job. Log readers include the PowerExchange Listener, netport jobs, Condense jobs, and the DTLUAPPL utility.
- For the DTLUAPPL utility and Condense jobs, ensure that the Post-Log Merge member Logger runs 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 environment:

![Post-Log Merge Environment 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)
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:

- **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.
- **TIME_CHKPT_FREQ=** in SYSTEM_OPTIONS should be reviewed and changed if necessary.
- **TIMER_INTERVAL=** in SYSTEM_OPTIONS should be reviewed and changed if necessary.

**Note:** In environments with member Loggers that are occasional less active than others, you need to carefully consider the values specified for **TIME_CHKPT_FREQ=** and **TIMER_INTERVAL=**. Lower values reduce extraction latency in a Post-Log Merge environment.

5. Customize the PowerExchange Logger JCL, as necessary.

If your MVS systems do not have unique SMF IDs, update the PowerExchange Logger JCL for those systems to override the non-unique SMF ID with a unique value.

This completes the additional installation customization required for Post-Log Merge.

**RELATED TOPICS:**

- “**SYSTEM_OPTIONS Parameters**” on page 61
- “**Customizing the PowerExchange Logger JCL**” on page 67
- “**Performance Considerations**” on page 94

**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
// *---------------------------------------------------------------------*
```
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.

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 components that are involved in the Post-Log Merge environment and describes
the result of component failure:

<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>
</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 describes considerations for moving extraction components in a Post-Log Merge environment to another z/OS system in the sysplex:

<table>
<thead>
<tr>
<th>Component</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>Considerations</td>
</tr>
</tbody>
</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 describes considerations for moving capture components in a Post-Log Merge environment to another z/OS 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 the 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.                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| PowerExchange Logger | - If no PowerExchange Logger runs on the destination MVS system, then you must also move the related PowerExchange Agent from the failed MVS system.  
- 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.                                                                                                                                                                                                                                                                                                                                 |
| PowerExchange Listener | If you use the PowerExchangeListener on the failed MVS system to extract change data, then also move the Post-Log Merge job to the destination MVS system.                                                                                                                                                                                                                                                                                                                                                   |

**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:

```
MODIFY job_name,DISPLAY
f 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:

```
MODIFY job_name, QUIT
```

The following table describes the commands that you can use to manage a Post-Log Merge job:

<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, 99
- Configuring PowerExchange Condense, 100
- Configuring PowerExchange Condense Parameters, 107
- Starting and Stopping PowerExchange Condense, 127
- Controlling PowerExchange Condense, 134
- Backing Up PowerExchange Condense Output Files, 134

PowerExchange Condense Overview

PowerExchange Condense captures change data from PowerExchange Logger for MVS log files and writes that data to condense files. You can make the change data in the condense files available for extraction processing at user-defined intervals.

PowerExchange Condense can perform full or partial condense processing. You specify the type of condense processing when you create a capture registration in the PowerExchange Navigator by setting the Condense option to one of the following values:

**Full**

Full condense processing. During a full condense cycle, PowerExchange Condense records only the latest image of the change data to keyed VSAM condense data sets. If multiple changes are made to the same record, later changes supersede earlier changes, leaving only the latest change. Use full condense processing only for tables or data maps that specify key columns. You cannot use full condense processing for Adabas and IDMS log-based CDC. Full condense processing does not maintain transactional integrity, but it can decrease the amount of change records that are processed and extracted.

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

**Part**

Partial condense processing. During a partial condense cycle, PowerExchange Condense writes successfully completed changes to sequential condense files, in chronological order based on end time. PowerExchange Condense does not eliminate any of the changes. This condense type maintains transactional integrity.
Also, you can run a PowerExchange Condense job in batch mode or continuous mode. CDC sessions extract change data from the condense files in batch extraction mode.

## Configuring PowerExchange Condense

To configure PowerExchange Condense, you must define a CAPTPARM configuration member for each source type and instance that is 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. For the **Full** option to be selectable, the source table or data map must identify at least one column as a key column.

**Restriction**: PowerExchange does not support full condense processing for Adabas or IDMS log-based CDC.

If you want 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**:
- [“PowerExchange Condense Job” on page 100](#)
- [“Condense Operational Modes” on page 101](#)
- [“Configuring PowerExchange Condense JCL” on page 102](#)
- [“Condense Input Files” on page 103](#)
- [“Configuring PowerExchange Condense Parameters” on page 107](#)
- [“Configuring Condense Group Definitions” on page 125](#)
- [“Enabling Capture Registrations for PowerExchange Condense Use” on page 100](#)

## 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** setting of **Part** or **Full**.

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 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.

#### 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 i/f: Read times out after 10 seconds
PWX-09967 CAPI i/f: End of log for time 06/08/16 16:59:43 reached
PWX-06415 Condense: Condense completed. Total Records=584, Data=289, UOWs=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.CPKPTV0 time=06/08/16 17:00:05
PWX-06420 Condense: Checkpoint done. Sequence=0000035CAA140000000000035CAA1400000000
PowerExchange Logfile=C5C4D4D34D0400000035C5EEB0000000
```

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, a Condense job on a z/OS system runs for a long period, perhaps on a 24-hour basis. In this mode, the condense subtask sleeps after each condense operation.

The next condense cycle is triggered by one of the following events:

- The number of minutes defined by the NO_DATA_WAIT parameter elapses.
- A CONDENSE command is issued from the command line, or a pwxcmd condense command is issued from a remote Linux, UNIX, or Windows system.
- A FILES ITCH command is issued from the command line, or a pwxcmd fileswitch command is issued from a remote Linux, UNIX, or Window system.
- A SHUTC O ND command is issued from the command line, or a pwxcmd shutcond command is issued from a remote Linux, UNIX, or Windows system.

**Note:** In continuous mode, the Condense job does not shut down automatically. To shut down the Condense job down, you must issue the SHUTDOWN or SHUTC O ND command.

PowerExchange performs a file switch when the file switch criteria defined by the FILE_SWITCH_CRIT and FILE SWITCH VAL parameters are met or when you issue a FILES WITCH command.
After a file switch, condense files become available to change data extraction processes. File switch processing closes open condense files if they contain data and then opens a new set of condense files for new changes. Extractions can process only closed condense files.

**Note:** A file switch does not occur if the current condense file does not contain data. PowerExchange next tries the file switch when the FILE_SWITCH_CRIT and FILE_SWITCH_VAL criteria are met. If the condense file still does not contain data, PowerExchange Condense continues retrying the file switch at the set intervals until data is available.

To determine the status of PowerExchange Condense processing in continuous mode, look for the following messages:

- PWX-06455 Command Handler: received CAPTURE_STARTUP_COMPLETE event.
- PWX-06417 Condense: Start to Condense because initialization complete
- PWX-09357 CAP1 i/f: Read times out after 10 seconds
- PWX-09367 CAP1 i/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
- PWX-06418 Condense: Closed file EDMUSR.D081.CND.CP060816.T1720146
- PWX-06136 Checkpoint taken to file=EDMUSR.D811.CHKPTV1 time=06/08/16 17:20:10
- PWX-06420 Condense: Checkpoint done. Sequence=0000363595600000000000036359560000000
- PowerExchange Logger=C5C4B40340000000363043000000
- PWX-06415 Condense: Condense completed. Total Records=1356, Data=672, UOWs=12
- PWX-06421 Condense: 06/08/16 17:20:21 Starting wait on commands for 5 minute

In this example, the Condense job is running in continuous mode. The condense task runs periodically and then waits for the NO_DATA_WAIT interval.

- Message PWX-06417 indicates that the condense task starts after initialization or after the NO_DATA_WAIT interval expires.
- Message PWX-06420 indicates that PowerExchange Condense processed changes and has taken a checkpoint.
- Message PWX-06421 indicates that PowerExchange Condense is starting the NO_DATA_WAIT interval of 5 minutes.

### Configuring PowerExchange Condense JCL

The PowerExchange installation provides sample Condense jobs in the RUNLIB library.

- **CONDB2.** Runs the Condense job as a batch job.
- **PCNDB2.** Runs the Condense job as a started task.

The following example JCL is from the PCNDB2 member in RUNLIB:

```plaintext
//PCNDB2  PROC SCERUN=CBE.SCEERUN,  
  //   HLQ=YOUR_HLQ,  
  //   LOGGER=PWXL,  
  //   HLQVS=YOUR_HLQVS,  
  //   RUNLIB=YOUR_HLQ.RUNLIB  
  //   RUN  EXEC PGM=DTLCACON,PARM=""  
  //  */  
  //STEPLIB DD DISP=SHR,DSN=+HLQ..LOADLIB  
  //   DD DISP=SHR,DSN=+HLQ..LOAD  
  //   DD DISP=SHR,DSN=+SCERUN  
  //  */  
  //EDMPARMS DD DISP=SHR,DSN=+HLQ..LOGGER..USERLIB  
  //  */ SYSTCPD EXPLICITLY IDENTIFIES WHICH DATA SET IS TO BE USED TO  
  //  */ OBTAIN THE PARAMETERS DEFINED BY TCPIP.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.TCPIP.DATA,DISP=SHR  
  //  */  
  //  */ CDC DATASETS FOLLOW - WITH SPECIFIC PARMS
```
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 at the hlq.logger.USERLIB data set, which is created during the installation of PowerExchange. This data set contains the EDMDSDIR 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 describe PowerExchange Condense output files, including the condense files, checkpoint files, CDCT file, and message data sets.

DTLCACDC (CDCT)

The Condense task writes tracking information about condense files to the hlq.CDCT file, which is a VSAM KSDS data set. The PowerExchange Listener reads the CDCT file for extraction processes.

During installation, PowerExchange creates the CDCT file and initializes it with a high values (9s) record.

After each file switch, PowerExchange Condense completes the following processing:

1. Closes the condense files.
2. Inserts keyed tracking records with information about each condense file into the CDCT file.
   The keyed tracking records contain the following information:
   - The condense file name
   - Whether the file is a partial or full condense file
   - The start and end times
   - Whether before images are included
   - The number of records in the file
   - Other control information
3. Writes a new checkpoint to the checkpoint data set. Also writes the CDCT tracking records to the checkpoint data set.

Each time a Condense job warm starts, PowerExchange Condense checks the tracking records in the checkpoint data set, and, if necessary, inserts or deletes tracking records in the CDCT file.

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 EXT_CAPT_MASK and CONDF_FULL_FILE_CTL parameters in the CAPTPARM file determine the names of these data sets.
The following table lists the data set type and the format of the data set name for each type of condense file:

<table>
<thead>
<tr>
<th>Condense File Type</th>
<th>Data Set Type</th>
<th>Format of the Data Set Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial</td>
<td>Variable-blocked</td>
<td>The data set name has the following format:</td>
</tr>
<tr>
<td></td>
<td>sequential</td>
<td>hlq.CND.CPyymmdd.Thhmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- hlq is an EXT_CAPT_MASK value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- yymmdd is a date</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- hhmmss is a time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- n is a sequence number, starting at 1, to establish a unique ID</td>
</tr>
<tr>
<td>Full</td>
<td>VSAM KSDS</td>
<td>The cluster data set name has the following format:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>hlg.CND.CPyymmdd.Thhmmss</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Where:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- hlq is an EXT_CAPT_MASK value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- yymmdd is a date</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- hhmmss is a time</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- n is a sequence number, starting at 1, to establish a unique ID</td>
</tr>
</tbody>
</table>

The PowerExchange Listener or a netport job uses the CAPX access method to read condense files. You can use the PowerExchange Navigator to view the data in closed condense files. Run a database row test against the extraction map in the appropriate extraction group.

You can use a variety of methods, including PowerCenter sessions and workflows, to extract and process the condense change data.

**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 two 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 time stamp, indicates that a checkpoint has been taken:

```
PWX-06136 Checkpoint taken to file=hlg.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, another 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 DTLLOGnn, where nn is a number from 01 to 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 DTLLOG 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

Configure PowerExchange Condense parameters in the CAPTPARM configuration member for your data source type. Also, ensure that the DTLCACFG DD statement in the PowerExchange Condense job or started task JCL points to this member.

The following table identifies the CAPTPARM configuration member that is available for each data source type in the RUNLIB library:

<table>
<thead>
<tr>
<th>Data Source</th>
<th>Member</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adabas</td>
<td>CAPTADA1</td>
</tr>
<tr>
<td>DB2 for z/OS</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 and have unique checkpoint file and condense file names.

Parameter Descriptions

In the CAPTPARM member, you can define PowerExchange Condense parameters.

The following table summarizes the PowerExchange Condense parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPT_IMAGE</td>
<td>The data image type that PowerExchange Condense captures to condense files.</td>
</tr>
<tr>
<td>CHKPT_BASENAME</td>
<td>The high-level data set name qualifiers for generating the checkpoint data sets.</td>
</tr>
<tr>
<td>CHKPT_FILE_CTL</td>
<td>The template file that contains the IDCAMS DEFINE CLUSTER control statements for the checkpoint files.</td>
</tr>
<tr>
<td>CHKPT_NUM</td>
<td>The number of checkpoint data sets.</td>
</tr>
<tr>
<td>CHKPT_PRIM_ALLOC</td>
<td>The primary space allocation for checkpoint files.</td>
</tr>
<tr>
<td>CHKPT_SCND_ALLOC</td>
<td>The secondary space allocation for checkpoint files.</td>
</tr>
<tr>
<td>CHKPT_VOLSERS</td>
<td>The DASD volume serial numbers (VOLSERs) where checkpoint data sets are allocated.</td>
</tr>
<tr>
<td>COLL_END_LOG</td>
<td>The operational mode of the Condense job.</td>
</tr>
<tr>
<td>COND_CDCT_RET_P</td>
<td>The number of days to keep CDCT records and condense files.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CONDENSE_SHUTDOWN_TIMEOUT</td>
<td>The maximum number of seconds that PowerExchange Condense waits after receiving a SHUTDOWN command before stopping.</td>
</tr>
<tr>
<td>CONDENSENAME</td>
<td>The name for the command-handling service for a PowerExchange Condense process to which pwxcmd commands are issued.</td>
</tr>
<tr>
<td>CONDF_FULL_FILE_CTL</td>
<td>The template file that contains the IDCAMS DEFINE CLUSTER control statements for the full condense files.</td>
</tr>
<tr>
<td>CONDF_PART_BLKSZ</td>
<td>The block size for partial condense files.</td>
</tr>
<tr>
<td>CONDF_PART_DATACLAS</td>
<td>The SMS DATACLAS value for partial condense files.</td>
</tr>
<tr>
<td>CONDF_PART_LRECL</td>
<td>The logical record length (LRECL) value for partial condense files.</td>
</tr>
<tr>
<td>CONDF_PART_MGMTCLAS</td>
<td>The SMS MGMTCLAS value for partial condense files.</td>
</tr>
<tr>
<td>CONDF_PART_STORCLAS</td>
<td>The SMS STORCLAS value for partial condense files.</td>
</tr>
<tr>
<td>CONDF_PRIM_ALLOC</td>
<td>The primary space allocation for condense files.</td>
</tr>
<tr>
<td>CONDF_SCND_ALLOC</td>
<td>The secondary space allocation for condense files.</td>
</tr>
<tr>
<td>CONDF_TYPE</td>
<td>The space unit type for condense files.</td>
</tr>
<tr>
<td>CONDF_UNIT</td>
<td>The unit for condense files.</td>
</tr>
<tr>
<td>CONDF_VOL</td>
<td>The VOLSER for condense files.</td>
</tr>
<tr>
<td>CONN_OVR</td>
<td>The CAPI_CONNECTION name to use when running PowerExchange Condense.</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>The data source type.</td>
</tr>
<tr>
<td>DBID</td>
<td>The instance name. When used with DB_TYPE, it defines selection criteria for capture registrations in the CCT file.</td>
</tr>
<tr>
<td>EXT_CAPT_MASK</td>
<td>The unique high-level qualifier (HLQ) that PowerExchange Condense uses to allocate condense data sets.</td>
</tr>
<tr>
<td>FILE_SWITCH_CRIT</td>
<td>Controls whether to use minutes or records for determining when to do an automatic file switch.</td>
</tr>
<tr>
<td>FILE_SWITCH_VAL</td>
<td>The number of FILE_SWITCH_CRIT units at which to do a file switch.</td>
</tr>
<tr>
<td>GROUPDEFS</td>
<td>The fully qualified data set name for the Condense Group Definitions file that defines condense groups.</td>
</tr>
<tr>
<td>KEY_CHANGE_ALW</td>
<td>Controls whether changes to the source key columns are allowed for full condense.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>The number of minutes to wait between condense operations when running in continuous mode.</td>
</tr>
<tr>
<td>NO_DATA_WAIT2</td>
<td>The number of seconds to wait for additional data to be received after the end-of-log is reached, indicated by the PWX-09967 message.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OPER_WTO</td>
<td>Controls whether condense file close WTO messages are issued.</td>
</tr>
<tr>
<td>RESTART_TOKEN</td>
<td>The restart point for starting change data processing when PowerExchange</td>
</tr>
<tr>
<td></td>
<td>Condense is cold started.</td>
</tr>
<tr>
<td>SEQUENCE_TOKEN</td>
<td>The restart point for starting change data processing when PowerExchange</td>
</tr>
<tr>
<td></td>
<td>Condense is cold started.</td>
</tr>
<tr>
<td>SIGNALLING</td>
<td>Controls whether PowerExchange Condense handles abnormal end conditions,</td>
</tr>
<tr>
<td></td>
<td>such as ABEND 0C4, SIGSEGV, SIGABEND.</td>
</tr>
<tr>
<td>VERBOSE</td>
<td>Controls whether PowerExchange Condense issues verbose or terse messages</td>
</tr>
<tr>
<td></td>
<td>for frequent condense activities such as cleanup, checkpoint, condense, and</td>
</tr>
<tr>
<td></td>
<td>file switch processing.</td>
</tr>
</tbody>
</table>

**CAPT_IMAGE Parameter**

The type of data image that PowerExchange Condense captures to condense files.

PowerExchange Condense can capture after images only or both before and after images. The capture image type must be consistent with the image type delivered to the target during extraction processing.

**Syntax:**

```
CAPT_IMAGE={AI|BA}
```

**Valid Values:**

- **AI.** After images only.
  - If you enter this value, the following limitations apply:
    - You cannot extract before images to the target.
    - You cannot use DTL_BI columns in extraction maps.
    - If you add DTL_CI columns to extraction maps, any insert or delete operations result in Null values in these columns.
- **BA.** Before and after images.
  - Informatica recommends that you specify this value so that you have the flexibility to use either **AI** or **BA** for the PowerCenter Image Type connection attribute for extraction processing.

The z/OS Installation Assistant adds the recommended value of **BA** to the configuration member unless you specify another value. If you do not define this parameter, the default of **AI** is used.

**CHKPT_BASENAME Parameter**

The high-level qualifier for generating PowerExchange Condense checkpoint data set names.

Checkpoint data sets are VSAM KSDS clusters. To create the full checkpoint VSAM KSDS cluster name, PowerExchange appends \( V_n \) to the last qualifier, where \( n \) is a number from 0 to the value of **CHKPT_NUM-1**. 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 \( .I \).

**Related Parameters:** **CHKPT_NUM**
Syntax:

`CHKPT_BASENAME=hlq`

**Value:** For the `hlq` variable, enter a high-level qualifier for generating checkpoint data sets. Maximum length is calculated as 41 - (CHKPT_NUM-1).

**Example:** Enter the following high-level qualifier:

`INFA.D.CHKPT`

PowerExchange appends V0:

`INFA.D.CHKPTV0`

PowerExchange also appends .D because the name is for a data component of the VSAM KSDS cluster:

`INFA.D.CHKPTV0.D`

**CHKPT_FILE_CTL Parameter**

The template file that contains the IDCAMS DEFINE CLUSTER control statements for PowerExchange Condense checkpoint files.

PowerExchange supplies a sample template, TMLCHKPT, in the RUNLIB library.

**Syntax:**

`CHKPT_FILE_CTL={dataset_name|"pds_member"}`

**Valid Values:** A fully qualified sequential data set name, or a PDS member name that is enclosed in double quotation marks (").

**Usage Notes:** If you use this parameter, do not also specify the following parameters:

- `CHKPT_PRIM_ALLOC`
- `CHKPT_SCND_ALLOC`
- `CHKPT_VOLSERS`

**CHKPT_NUM Parameter**

The number of PowerExchange Condense checkpoint files.

**Related Parameters:** CHKPT_BASENAME

**Syntax:**

`CHKPT_NUM={number|3}`

**Value:** For the `number` variable, enter a number from 1 to 999999.

Default is 3.

**Usage Notes:** If you decrease the CHKPT_NUM value and warm start PowerExchange Condense, PowerExchange Condense might restart from an incorrect location. In this situation, you must do a cold start.

**CHKPT_PRIM_ALLOC Parameter**

The amount of primary space that is allocated for PowerExchange Condense checkpoint files.

**Related Parameters:** CHKPT_SCND_ALLOC, CHKPT_VOLSERS
Syntax:

\[
\text{CHKPT_PRIM_ALLOC} = \text{number}
\]

Value: For the \textit{number} variable, enter a number greater than 0.

Usage Notes: If you specify this parameter, do not also specify the \text{CHKPT_FILE_CTL} parameter.

**CHKPT_SCND_ALLOC Parameter**

The amount of secondary space that is allocated for PowerExchange Condense checkpoint files.

Related Parameters: \text{CHKPT_PRIM_ALLOC}, \text{CHKPT_VOLSERS}

Syntax:

\[
\text{CHKPT_SCND_ALLOC} = \text{number}
\]

Value: For the \textit{number} variable, enter a number greater than 0.

Usage Notes: If you specify this parameter, do not also specify the \text{CHKPT_FILE_CTL} parameter.

**CHKPT_VOLSERS Parameter**

The DASD volume serial numbers (VOLSERs) where PowerExchange Condense checkpoint data sets are allocated.

Syntax:

\[
\text{CHKPT_VOLSERS} = \text{volser1, volser2, volser3}
\]

Valid Values: The \textit{volser1}, \textit{volser2}, and \textit{volser3} variables are valid MVS VOLSER values on your system. You must define all three variables, even if they specify the same value.

Example: The following statement specifies three valid VOLSERs on your system:

\[
\text{CHKPT_VOLSERS} = \text{DSK100, DSK101, DSK102}
\]

**COLL_END_LOG Parameter**

The operational mode of the PowerExchange Condense job.

Syntax:

\[
\text{COLL_END_LOG} = \{0|1\}
\]

Valid Values:

- 0. Continuous mode. After each Condense run, PowerExchange Condense waits for the number of minutes that is specified in the \text{NO_DATA_WAIT} parameter before starting another Condense cycle.
- 1. Batch mode. PowerExchange Condense shuts down after a single Condense run. For example, use batch mode if Condense is scheduled to run after a particular batch update job and then shut down.

Default is 0.

**COND_CDCT_RET_P Parameter**

The number of days to retain CDCT records and condense files.

Syntax:

\[
\text{COND_CDCT_RET_P} = \{\text{days}|60\}
\]
Condense files that are older than this retention period and their corresponding CDCT records are automatically deleted during cleanup processing. Cleanup processing occurs during startup, fileswitch, or shutdown processing.

**Value:** For the days variable, enter a number greater than 0.

The z/OS Installation Assistant enters a value of 50 in the PowerExchange Condense configuration member unless you specify another value. If this parameter is not defined, the default of 60 is used.

**Usage Notes:**

Enter a time interval that is long enough for change data to be extracted from condense files before the files are deleted.

**CONDENSE_SHUTDOWN_TIMEOUT Parameter**

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

**Syntax:**

```
CONDENSE_SHUTDOWN_TIMEOUT={seconds|600}
```

**Value:** For the seconds variable, enter a number from 0 to 2147483647.

Default is 600.

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

**CONDENSENAME Parameter**

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

**Syntax:**

```
CONDENSENAME=service_name
```

**Value:** For the service_name variable, enter a character string up to 64 characters in length.

No default value.

**Usage Notes:** The service name that you specify in this parameter must match the service name that you specify in the associated SVCNODE statement in the DBMOVER configuration file.

**CONDF_FULL_FILE_CTL Parameter**

The template file that contains the IDCAMS DEFINE CLUSTER control statements for PowerExchange Condense full condense files.

PowerExchange supplies a sample template, TMLCONF, in the RUNLIB library.

**Syntax:**

```
CONDF_FULL_FILE_CTL={dataset_name|"pds_member_name"}
```

**Valid Values:** A fully qualified sequential data set name, or a PDS member name that is enclosed in double quotation marks (").

**Usage Notes:**

- Do not specify the CONDF_FULL_FILE_CTL parameter with the CONDF_UNIT or CONDF_VOL parameter for a full condense. If you do, PowerExchange issues error message PWX-06308.
Do not specify both the CONDF_FULL_FILE_CTL and CONDF_PART_STORCLAS parameters with either the CONDF_UNIT or CONDF_VOL parameter for a full or partial condense. If you do, PowerExchange issues error message PWX-06308.

CONDF_PART_BLK_SZ Parameter
The block size for PowerExchange Condense partial condense files.

Syntax:
CONDF_PART_BLK_SZ={number|0}

Value: For the number variable, enter a number from 0 to 32760.
Default is 0.

CONDF_PART_DATACLAS Parameter
The SMS data class for PowerExchange Condense partial condense files.

Syntax:
CONDF_PART_DATACLAS=sms_dataclas

Value: For the sms_dataclas variable, enter a valid SMS DATACLAS value.

CONDF_PART_LRECL Parameter
The logical record length (LRECL) for PowerExchange Condense partial condense files.

Syntax:
CONDF_PART_LRECL=bytes

Value: For the bytes variable, enter a number of bytes from 4044 to 147444. Default is 147444.
If you enter 32756 or less, PowerExchange Condense uses RECFM=VB to create the condense files. You can then read the condense files by using the Interactive System Productivity Facility (ISPF) and standard IBM utilities such as IDCAMS.
If you enter a value greater than 32756, PowerExchange Condense uses RECFM=VS to create the condense files. You can then read the condense files only by using specialized utilities such as the IBM Data Interfile Transfer, Testing and Operations Utility (DITTO) with the DB command.
Usage Notes: If you have 3390 disks, you can usually achieve efficient disk space usage by setting CONDF_PART_BLK_SZ=27998 to write two blocks per track and by setting the CONDF_PART_LRECL parameter either to 27994 or to a value greater than 32756.
Informatica recommends that you set the CONDF_PART_LRECL parameter based on the following guidelines:
• If the maximum record size is 27994 bytes or less, set the CONDF_PART_LRECL parameter to 27994. This value causes RECFM=VB to be used.
• If the maximum record size exceeds 32756 bytes, which might occur with Adabas spanned data, set the CONDF_PART_LRECL parameter to 147444. This value causes RECFM=VS to be used.
• If the maximum record size is a value from 27995 to 32756 bytes, use one of the following settings:
  • Set CONDF_PART_LRECL=32764 to use RECFM=VS.
  • Set CONDF_PART_LRECL=27994 and CONDF_PART_BLK_SZ=27998 to use RECFM=VB. In this case, PowerExchange handles spanned records.
CONDF_PART_MGMTCLAS Parameter

The SMS management class for PowerExchange Condense partial condense files.

Syntax:

CONDF_PART_MGMTCLAS=sms_mgmtclas

Value: For the sms_mgmtclas variable, enter a valid SMS MGMTCLAS.

CONDF_PART_STORCLAS Parameter

The SMS storage class for PowerExchange Condense partial condense files.

Syntax:

CONDF_PART_STORCLAS=sms_storclas

Value: For the sms_storclas variable, enter a valid SMS STORCLAS.

Usage Notes:

• Do not specify the CONDF_PART_STORCLAS parameter with the CONDF_UNIT or CONDF_VOL parameter for a partial condense. If you do, PowerExchange issues error message PWX-06308.

• Do not specify both the CONDF_PART_STORCLAS and CONDF_FULL_FILE_CTL parameters with either the CONDF_UNIT or CONDF_VOL parameter for a full or partial condense. If you do, PowerExchange issues error message PWX-06308.

CONDF_PRIM_ALLOC Parameter

The amount of primary space that is allocated for PowerExchange Condense condense files. The CONDF_TYPE parameter indicates whether the units are cylinders or tracks.

Related Parameters: CONDF_FULL_FILE_CTL and CONDF_TYPE

Syntax:

CONDF_PRIM_ALLOC={1|number}

Value: For the number variable, enter a number greater than 0.

The z/OS Installation Assistant enters 10 for this parameter in the PowerExchange Condense configuration member unless you specify another value. If this parameter is not defined, the default of 1 is used.

Usage Notes: If you specify the CONDF_FULL_FILE_CTL parameter, the CONDF_PRIM_ALLOC parameter is ignored for full condense files.

CONDF_SCND_ALLOC Parameter

The amount of secondary space that is allocated for PowerExchange Condense condense files. The CONDF_TYPE parameter indicates whether the units are cylinders or tracks.

Related Parameters: CONDF_FULL_FILE_CTL and CONDF_TYPE

Syntax:

CONDF_SCND_ALLOC={1|number}

Value: For the number variable, enter a number greater than 0.

Default is 1.

Usage Notes: If you specify the CONDF_FULL_FILE_CTL parameter, the CONDF_SCND_ALLOC parameter is ignored for full condense files.
CONDF_TYPE Parameter

The unit type for defining primary space and secondary space for PowerExchange Condense condense files.

**Related Parameters:** CONDF_PRIM_ALLOC, CONDF_SCND_ALLOC, and CONDF_FULL_FILE_CTL

**Syntax:**

```
CONDF_TYPE={CYL|TRK}
```

**Valid Values:**

- **CYL**. Cylinders.
- **TRK**. Tracks.

Default is CYL.

**Usage Notes:** If you specify the CONDF_FULL_FILE_CTL parameter, the CONDF_TYPE parameter is ignored for full condense files.

CONDF_UNIT Parameter

The unit name of the device where PowerExchange Condense condense files reside.

**Related Parameters:** CONDF_FULL_FILE_CTL

**Syntax:**

```
CONDF_UNIT=unit_name
```

**Value:** For the `unit_name` variable, enter a valid z/OS generic or esoteric unit name, such as 3390 or SYSDA.

**Usage Notes:**

- Do not specify the CONDF_UNIT parameter with the CONDF_FULL_FILE_CTL parameter for a full condense. If you specify both parameters, PowerExchange issues error message PWX-06308.
- Do not specify the CONDF_UNIT parameter with CONDF_PART_STORCLAS parameter for a partial condense. If you specify both parameters, PowerExchange issues error message PWX-06308.

CONDF_VOL Parameter

The volume serial number (VOLSER) for condense files.

**Related Parameters:** CONDF_FULL_FILE_CTL

**Syntax:**

```
CONDF_VOL=volser
```

**Value:** For the `volser` variable, enter a z/OS VOLSER.

**Usage Notes:**

- Do not specify the CONDF_VOL parameter with the CONDF_FULL_FILE_CTL parameter for a full condense. If you specify both parameters, PowerExchange issues error message PWX-06308.
- Do not specify the CONDF_VOL parameter with the CONDF_PART_STORCLAS parameter for a partial condense. If you specify both parameters, PowerExchange issues error message PWX-06308.
CONN_OVR Parameter
The CAPI_CONNECTION name to use when you run PowerExchange Condense.

Syntax:
```
CONN_OVR=capi_connection_name
```

Value: For the `capi_connection_name` variable, enter a valid source CAPI connection name.
If you do not specify this name, PowerExchange Condense uses the default connection.

DB_TYPE Parameter
For PowerExchange Condense, the data source type.

Related Parameters: DBID

Syntax:
```
DB_TYPE=database_type
```

Valid Values: For the `database_type` variable, enter one of the following values:
- ADA for Adabas
- DB2 for DB2 for z/OS
- DCM for Datacom
- IDL for IDMS log-based
- IMS for IMS
- VSM for VSAM

DBID Parameter
For PowerExchange Condense, the instance name for capture registrations. When used with the DB_TYPE parameter, it defines selection criteria for capture registrations in the CCT file.

Related Parameters: DB_TYPE

Syntax:
```
DBID=instance_name
```

Value: For the `instance_name` variable, enter the instance name for capture registrations.

Usage Notes:
- 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.

EXT_CAPT_MASK Parameter
The unique high-level qualifier (HLQ) that PowerExchange Condense uses to allocate condense data sets.

Syntax:
```
EXT_CAPT_MASK=hlq
```

Value: For the `hlq` variable, enter a high-level qualifier (HLQ) value. 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 characters.

To create condense data sets, PowerExchange appends the following information for VSAM full condense data sets:

\[ .CND.CPyyymmdd.Thhmmnnn \]

PowerExchange appends the following information for sequential partial condense data sets:

\[ .CND.CPyyymmdd.Thhmmnnn \]

Where:

- \( yy \) is year.
- \( mm \) is month.
- \( dd \) is day.
- \( hh \) is hour.
- \( mm \) is minutes.
- \( nnn \) is a sequence number starting from 001.

**Example:** For example, you might specify the following HLQ:

\[ \text{EXT\_CAPT\_MASK} = \text{INFA.D} \]

For sequential partial condense data sets, PowerExchange appends the following additional information to the mask:

\[ \text{INFA.D.CNP080718.T1545001} \]

**Warning:** Do not use the same \( \text{EXT\_CAPT\_MASK} \) 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 \( \text{EXT\_CAPT\_MASK} \) value until the Condense task has finished processing all condense data sets that match the mask.

**FILE\_SWITCH\_CRIT Parameter**

For PowerExchange Condense, defines whether to use minutes or records for determining when to do an automatic file switch.

**Related Parameters:** FILE\_SWITCH\_VAL

**Syntax:**

\[ 
\text{FILE\_SWITCH\_CRIT} = \{M|R\} \]

**Valid Values:**

- \( M \) Minutes.
- \( R \) Records.

Default is \( M \).

**FILE\_SWITCH\_VAL Parameter**

For PowerExchange Condense, the number of FILE\_SWITCH\_CRIT units at which to do a file switch.

**Related Parameters:** FILE\_SWITCH\_CRIT

**Syntax:**

\[ 
\text{FILE\_SWITCH\_VAL} = \{\text{number}\} \]

**Value:** For the \( \text{number} \) variable, enter any number greater than 0.
Default is 30.

**Example:** To configure the Condense task to complete a file switch every 30 records, define the following parameters:

```
FILE_SWITCH_VAL=30
FILE_SWITCH_CRIT=R
```

To configure the Condense task to complete a file switch every 30 minutes, define the following parameters:

```
FILE_SWITCH_VAL=30
FILE_SWITCH_CRIT=M
```

**Usage Notes:** If a condense file contains no data when the FILE_SWITCH_VAL limit is reached, the file switch does not occur.

**GROUPDEFS Parameter**

The fully qualified data set name for the Condense Group Definitions file that defines condense definition groups for PowerExchange Condense.

To use condense definition groups, this parameter is required.

**Syntax:**

```
GROUPDEFS={dataset_name|"pds_member_name"}
```

**Valid Values:**

- **dataset_name.** Any fully qualified sequential data set name or PDS member name.
- **pds_member_name.** Any fully qualified PDS member name enclosed in quotation marks (").

For example:

```
GROUPDEFS="DTLUSR.V810.RUNLIB(CONDGRP)"
```

**KEY_CHANGE_ALW Parameter**

Controls whether Condense jobs fail or continue when PowerExchange Condense detects a change to the source key columns during full condense processing.

This parameter applies only to full condense processing, which is enabled by selecting **Full** for the **Condense** option in the capture registration.

**Syntax:**

```
KEY_CHANGE_ALW={N|Y}
```

**Valid Values:**

- **N.** The Condense job fails when the change to the key columns is detected.
- **Y.** The Condense job ignores the change to the key during full condense processing and continues.

Default is N.

**Usage Notes:**

- If you have a DB2 for z/OS source, you can do an update operation to change any or all key columns in a row.
- This parameter does not apply to partial condense processing.
NO_DATA_WAIT Parameter

When PowerExchange Condense runs in continuous mode, the number of minutes that it waits before starting the next condense cycle.

Related Parameters: FILE_SWITCH_VAL

Syntax:

\[ \text{NO_DATA_WAIT} = \{ \text{minutes} | 60 \} \]

Value: For the \textit{minutes} variable, enter a number greater than 0.

The z/OS Installation Assistant enters 5 for this parameter in the PowerExchange Condense configuration member unless you specify another value. If this parameter is not defined, the default of 60 is used.

Usage Notes:

- If the FILE_SWITCH_CRIT parameter is set to M, for minutes, and the FILE_SWITCH_VAL parameter value is less than the NO_DATA_WAIT parameter value, PowerExchange Condense uses the FILE_SWITCH_VAL value instead.
- If the COLL_END_LOG parameter is set to 1, PowerExchange Condense runs in batch mode and the NO_DATA_WAIT parameter is ignored.

NO_DATA_WAIT2 Parameter

The number of seconds that PowerExchange Condense waits after it reaches the end-of-log to receive more data.

This parameter sets the Consumer API (CAPI) interface timeout value, which appears in message PWX-09967.

Syntax:

\[ \text{NO_DATAWAIT2} = \{ \text{seconds} | 600 \} \]

Value: For the \textit{seconds} variable, enter any number greater than 0.

The z/OS Installation Assistant enters 60 for this parameter in the ECCR configuration member unless you specify another value. If this parameter is not defined, the default of 600 is used.

Usage Notes:

- The completion of a condense cycle occurs when the number of seconds specified in the NO_DATA_WAIT2 parameter expires and PowerExchange Condense has not received any data from the PowerExchange Logger.
- The optimal value for the parameter varies according to change data activity on the system:
  - If you set the parameter 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 you set the parameter too low and the PowerExchange Logger encounters a large unit of work for a source that is not being condensed, the condense operation might also end prematurely because no data is being returned.
  - If you set the parameter too high, an individual condense operation might never end.
**OPER_WTO Parameter**

Defines whether PowerExchange Condense issues WTO messages when a condense file closes.

**Syntax:**

```
OPER_WTO=[N|Y]
```

**Value:**

- **N.** When a condense file closes, PWX-06418 messages are written to the PowerExchange log.
- **Y.** When a condense file closes, PWX06418I WTOs are issued. You can use these messages with an automation product. The PWX-06418 messages are also written to the PowerExchange log.

Default is **N.**

**Usage Notes:** File switch processing does not occur for empty condense files.

**RESTART_TOKEN Parameter**

A token value that works with the SEQUENCE_TOKEN value to define the restart point for PowerExchange Condense change data processing when you cold start PowerExchange Condense.

**Related Parameters:** SEQUENCE_TOKEN

**Syntax:**

```
RESTART_TOKEN=restart_token
```

**Valid Values:**

- A specific restart token value.
- Zeroes for the entire 32-digit token value. For example:
  
  ```
  RESTART_TOKEN=00000000000000000000000000000000
  ```
- Not specified.

**Usage Notes:** Based on how you set the RESTART_TOKEN and SEQUENCE_TOKEN parameters, PowerExchange Condense processing starts from one of the following restart points during a cold start:

- If you enter specific restart token and sequence token values other than all zeroes, processing starts from the restart point that these token values specify.
- If you enter only zeroes for both parameters, processing starts from the beginning of the PowerExchange Logger for MVS active log files.
- If you do not specify these parameters, processing starts from the current end-of-log position.

**SEQUENCE_TOKEN Parameter**

A token value that works with the RESTART_TOKEN value to define the restart point for PowerExchange Condense change data processing when you cold start PowerExchange Condense.

**Related Parameters:** RESTART_TOKEN

**Syntax:**

```
SEQUENCE_TOKEN=sequence_token
```

**Valid Values:**

- A specific sequence token value.
• Zeros for the entire 40-digit token value. For example:

```
SEQUENCE_TOKEN=00000000000000000000000000000000
```

• Not specified.

**Usage Notes**: Based on how you set the `SEQUENCE_TOKEN` and `RESTART_TOKEN` parameters, PowerExchange Condense processing starts from one of the following restart points during a cold start:

- If you enter specific restart token and sequence token values other than all zeroes, processing starts from the restart point that these token values specify.
- If you enter only zeroes for both parameters, processing starts from the beginning of the PowerExchange Logger for MVS active log files.
- If you do not specify these parameters, processing starts from the current end-of-log position.

**SIGNALLING Parameter**

Defines whether PowerExchange Condense handles abnormal end conditions, such as an ABEND 0C4, SIGSEGV, and SIGABEND.

**Syntax**:

```
SIGNALLING={N|Y}
```

**Valid Values**:

- **N**: PowerExchange Condense does no automatic trapping of errors. Instead, the default error handling of the operating system is used. This error handling usually reports the offending program line and dump memory.
- **Y**: PowerExchange Condense takes automatic action in the event of certain abnormal end conditions such as memory corruption and SOC4 abends and attempts to shut down in an orderly manner.

Default is **N**.

**VERBOSE Parameter**

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

**Syntax**:

```
VERBOSE={N|Y}
```

**Value**:

- **Y**: For each condense cycle and file switch, PowerExchange Condense logs multiple messages.
- **N**: For each condense cycle and file switch, PowerExchange Condense amalgamates information into a single terse message.

Default is **Y**.

**Usage Notes**: If a condense file contains no data when the VERBOSE limit is reached, the file switch does not occur.

### Controlling Allocation Attributes of Condense Data Sets

You can set parameters in the CAPTPARM configuration member to define allocation attributes for the data sets that the Condense job creates.

You can set parameters that control the allocation of checkpoint files and of partial or full condense 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 the CHKPT_FILE_CTL Parameter

Use the CHKPT_FILE_CTL parameter to specify the template file that contains the IDCAMS DEFINE CLUSTER control statements for checkpoint files.

You can customize DEFINE CLUSTER control statements to control allocation attributes for checkpoint files. For example, you might want to customize the following control statements:

- If you use SMS, customize the DATACLASS, STORAGECLASS, and MANAGEMENTCLASS statements based on the SMS data classes, storage classes, and management classes that are defined at your site.
- To change the default suffix of .D for the Data component, customize the DATA statement.
- To change the default suffix of .I for the Index component, customize the INDEX statement.
- To override the default control interval size of 32768, customize the CONTROLINTERVALSIZE statement.

The following sample template is provided in the TMLCHKPT member in the RUNLIB library:

```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) 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) -
DATA -
  (NAME(<<name>>.D)) -
INDEX -
  (NAME(<<name>>.I)) -
```

**Note:** The PowerExchange installer adds the initial DATACLASS, STORAGECLASS, MANAGEMENTCLASS, and VOLUMES values based on the values that you enter in the z/OS Installation Assistant. You can customize these values in the TMLCHKPT member.
If you customize the template, use the following guidelines:

- Make sure that the DEFINE CLUSTER control statements are valid IDCAMS control statements because PowerExchange passes these statements to IDCAMS as is, with the exception of the NAME control statements.
- Use uppercase to define the DEFINE CLUSTER control statements.
- Do not start control statements in column 1.
  The maximum number of lines is 35 lines.
- You must define the <<name>> variable in the NAME parameter of the DEFINE CLUSTER, DATA, and INDEX control statements. PowerExchange populates this variable with the value that you specify in the EXT_CAPT_MASK parameter of the CAPTPARM member. Make sure that the EXT_CAPT_MASK prefix, when combined with any changes made to the suffix for the DATA and INDEX statements, does not exceed 44 characters.
- Specify the KEYS parameter exactly as shown in the template.
- Start comments with a forward slash and an asterisk (/*) and consistently place them before or after the control statements.

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:

- EXT_CAPT_MASK
- CONDF_PART_DATACLAS
- CONDF_PART_STORCLAS
- CONDF_PART_LRECL
- CONDF_PART_BLKSZ
- CONDF_PRIM_ALLOC
- CONDF_SCND_ALLOC
- CONDF_VOL
- CONDF_UNIT
- CONDF_TYPE

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_PART_LRECL. Default is (blocksize - 4).
- CONDF_PART_BLKSZ. Default is 0.
- CONDF_PRIM_ALLOC. Default is from DBMOVER SPACE= parameter, if specified.
- CONDF_SCND_ALLOC. Default is from DBMOVER SPACE= parameter, if specified.
- 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 CAPTPARM or 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

You can control the allocation of full condense files by specifying allocation parameters in the CAPTPARM configuration member or in a file that is referenced from the CAPTPARM member.

Use the following methods to define allocation attributes:

- Specify the following allocation parameters in the CAPTPARM configuration member:
  - EXT_CAPT_MASK. The high-level qualifier for the full condense data sets.
  - CONDF_PRIM_ALLOC. The primary space allocated for condense files. Default is 1.
  - CONDF_SCND_ALLOC. The secondary space allocated for condense files. Default is 1.
  - CONDF_VOL. A volser. If you omit this parameter, allocation of full condense files might succeed based on the z/OS and SMS system configuration.
  - CONDF_TYPE. Type of space allocation units. Default is CYL for cylinders.
- Specify IDCAMS DEFINE CLUSTER control statements in the TMLCONDF template file, and use the CONDF_FULL_FILE_CTL parameter in the CAPTPARM member to point to this file.

For both methods, EXT_CAPT_MASK is a required parameter. Use any combination of the other parameters and statements.

Using the CONDF_FULL_FILE_CTL Parameter

Use the CONDF_FULL_FILE_CTL parameter to specify the template file that contains the IDCAMS DEFINE CLUSTER control statements for the full condense files.

You can customize DEFINE CLUSTER control statements to control allocation attributes for full condense files. For example, you might want to customize the following control statements:

- If you use SMS, customize the DATACLASS, STORAGECLASS, and MANAGEMENTCLASS statements based on the SMS data classes, storage classes, and management classes that are defined at your site.
- To change the default suffix of .D for the Data component, customize the DATA statement.
- To change the default suffix of .I for the Index component, customize the INDEX statement.
- To override the default control interval size of 32768, customize the CONTROLINTERVALSIZE statement.

The following sample template is provided in the TMLCONDF member in the RUNLIB library:

```plaintext
/* 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) -
   STORAGECLASS(storclas) -
   MANAGEMENTCLASS(mgmtclas) -
   TRACKS (5 5) -
   VOLUMES(volser) -
   REUSE -
   FREESPAC (20 20) -
   SHAREOPTIONS (2 3)) -
DATA -
  (NAME(<<name>>.D)) -
INDEX -
  (NAME(<<name>>.I))
```
**Note:** The PowerExchange installer adds the initial DATACLASS, STORAGECLASS, MANAGEMENTCLASS, and VOLUMES values based on the values that you enter in the z/OS Installation Assistant. You can customize these values in the TMLCONDF member.

If you customize the template, use the following guidelines:

- Make sure that the DEFINE CLUSTER control statements are valid IDCAMS control statements because PowerExchange passes these statements to IDCAMS as is, with the exception of the NAME control statements.
- Use uppercase to define the DEFINE CLUSTER control statements.
- Do not start control statements in column 1.
  - The maximum number of lines is 35 lines.
  - You must define the `<<name>>` variable in the NAME parameter of the DEFINE CLUSTER, DATA, and INDEX control statements. PowerExchange populates the variable with the value that you specify in the EXT_CAPT_MASK parameter of the CAPTPARM member.
  - Make sure that the EXT_CAPT_MASK prefix, when combined with any changes made to the suffix for the DATA and INDEX statements, does not exceed 44 characters.
- Specify the KEYS parameter as shown in the template.
- Start comments with a forward slash and an asterisk (`/*`). Place comments before or after 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 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 table lists the registrations and tables that this example uses:

<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:

```
GROUP=(Personnel, DTLUSR.PERCOND)
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

Different methods of starting and stopping PowerExchange Condense are available. Choose the method that best suits your needs in a given situation.

Starting Condense

On z/OS, you can run the Condense job as a batch job or a started task.

Generally, you use a started task to run a Condense task in continuous mode for a long time, and use a batch job to run a Condense job in batch mode as part of a scheduled batch job.

To start a Condense job as a batch job, submit the job to the MVS Job Scheduler by using products such as TSO/E, a job scheduler, or automation. PowerExchange provides sample JCL for running Condense as a batch job in the RUNLIB(CONDDB2) member.

To run the Condense job as a started task, place its PROC into the system PROCLIB. Then use the MVS START command to start the Condense started task. PowerExchange provides sample JCL for running Condense as a started task in the RUNLIB(PCNDDDB2) member.

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

Before you start the Condense job, verify that the following conditions are met:

- The PowerExchange Logger and Agent are started.
- The checkpoint files are in the correct state for the start type:
  - For a cold start, verify that no checkpoint files exist with the CHKPT_BASENAME mask that is defined in the CAPTPARM member.
  - For a warm start, verify that all of the checkpoint files from the last Condense job exist and are available.
- The capture registrations that are required for the Condense run are defined for the DBTYPE and DBID in the PowerExchange Navigator. If required, you can delete or deactivate registrations in the PowerExchange Navigator.

Cold Start Processing

When you start the Condense job, it checks for existing checkpoint files that have the high-level qualifier that is specified by the CHKPT_BASENAME parameter in the CAPTPARM member.

Note: The DTLCACFG DD statement points to the CAPTPARM member.
If the Condense job does not find a checkpoint file, it cold starts and issues the following Write-To-Operator-with-Reply (WTOR) message:

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

To continue with the cold start, reply Y to the PWX06101A message. The Condense job then issues the following WTOR message to indicate that the request to cold start was accepted:

```
PWX06103I Cold Start accepted
```

To cancel the cold start, reply N to PWX06101A message. The Condense job ends immediately with the following message:

```
PWX06104W Cold Start declined
```

**Note:** The Condense job writes messages to the PowerExchange DTLLOG member of the LOG file in the PowerExchange `datalib` library, by default. If alternative logging is enabled, Condense writes messages to a `DTLLOGnn` log member in the `job_number` file in the `datalib` library.

For each checkpoint file, the Condense job writes the following message to the PowerExchange message log:

```
PWX-06365 Warning: Checkpoint file chkptasenameVn could not be read and was ignored:
Checkpoint FILE chkpt_basenameVn Does not exist. OPEN retcodes 368/4/5896
```

This message indicates that the Condense job ignored the specified checkpoint file because it does not exist.

The point from which the Condense job starts getting change data from the PowerExchange Logger depends on the `RESTART_TOKEN` and `SEQUENCE_TOKEN` parameters in `CAPTPARM` member. Based on the values of these parameters, the following processing occurs:

- If the `RESTART_TOKEN` and `SEQUENCE_TOKEN` parameters are not present in the `CAPTPARM` member, PowerExchange Condense starts from the current end-of-log position in the PowerExchange Logger logs.

- If the `RESTART_TOKEN` and `SEQUENCE_TOKEN` parameters are set to all zeroes, PowerExchange Condense starts from the earliest available point in the PowerExchange Logger log files. In a Post-Log Merge environment, the PowerExchange Logger goes back to the oldest available RBA or timestamp. This process might be time-consuming depending on the number and size of available Logger archive logs. The following messages are written to the PowerExchange message log to indicate that only zeroes are specified for the restart tokens:

```
PWX-06100 Sequence token 00000000000000000000000000000000
PWX-06100 Logger token 00000000000000000000000000000000
```

- If the `RESTART_TOKEN` and `SEQUENCE_TOKEN` parameters are set to specific token values to define a specific restart point, PowerExchange Condense starts getting data from this restart point if it is a valid restart point in the Logger log files. The following messages in the PowerExchange message log identify the restart token values:

```
PWX-06100 Sequence token sequence_token_value
PWX-06100 Logger token restart_token_value
```

To determine the specific restart token values to use, you can use the DTLUAPPL or DTLUCDEP utility or review previous Condense job runs. In error recovery situations, Informatica Global Customer Support might provide these token values.

At this point in the initialization process, the controller task starts the dump, command, and condense subtasks of the Condense job.

PowerExchange Condense issues the following message to echo the restart tokens that are being used to define the start point for data extraction from the PowerExchange Logger:

```
PWX-06413 Condense: Highest Restart Token. Sequence=sequence_token_value
PowerExchange Logger=restart_token_value
```
After the restart point is established, PowerExchange Condense cleans up condense files and CDCT entries that have expired as a result of the cold start and writes checkpoint information to the current checkpoint file. Initialization is then complete as indicated by the following messages in the PowerExchange log:

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

PowerExchange Condense then triggers the first condense operation.

**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 DTLLGnn if using alternative logging) for each Checkpoint data set that is found:

- PWX-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_basenameVn could not be read and was ignored: Checkpoint FILE chkpt_basenameVn Does not exist. OPEN retcodes 268/4/5896

**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_basenameVn.

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-06413 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: TotMs=nnnnn Diff=nnnnn

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 shut down. Each of these
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.

**SHUTCOND**

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

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.

**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.

**Condense Job Message Output**

Review PowerExchange Condense job messages in the message log to determine the status of Condense
processing.

The following example messages are for a Condense job that uses continuous extraction mode. The job ran
and then was cold started from the earliest point in the change stream, as indicated by the restart tokens that
are composed only of zeroes.

PWX-21605 Connection selected CHANGES found from covr< > tag< > typeDB2> int<FALSE>
method<CONN_NAME>.  
PWX-21605 Connection selected CHANGES found from covr< > tag< > typeDB2> 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
Starting and Stopping PowerExchange Condense
PWX-06420 Condense: Checkpoint done. Sequence=00000260A94C00000000000260A94C00000000
PowerExchange Logger=C5C4D4D34040000026091AB00000000
PWX-09967 CAPI 1/f: End of log for time 06/08/21 19:30:11 reached
PWX-06415 Condense: Condense completed. Total Records=144696, Data=103251, UOWs =9275
PWX-06421 Condense: 06/08/21 19:32:19 Starting wait on commands for 5 minute
Command=SHUTDOWN
PWX-06463 Command Handler: Close Condense request is now queued.
PWX-06464 Command Handler: Shutdown will occur shortly.
PWX-06105 Controller: Executing command Setting STOPTASK to CAPI.
PWX-06109 Controller: Warning During shutdown, ignored event=11 (CMD_TO_CONT).
PWX-06453 Command Handler: shutting down.
PWX-06454 Command Handler: has stopped.
PWX-06110 Unloaded module 1 (COMMAND_HANDLER).
PWX-06060 Controller: subtask Command Handler ended.
PWX-06416 Condense: Shutting down because SHUTDOWN event received
PWX-06418 Condense: Closed file EDMUSR.D811.CND.CP060821.T1931003
PWX-06495 Dump: task got an event event_num=2.
PWX-06491 Dump: ending.
PWX-06060 Controller: subtask Dump ended.
PWX-06110 Unloaded module 4 (DUMP).
PWX-06136 Checkpoint taken to file=EDMUSR.D811.CHKPTVO time=06/08/21 19:32:27
PWX-06420 Condense: Checkpoint done. Sequence=00000364AF1400000000000364AF1400000000
PowerExchange Logger=C5C4D4D34040000036464641500000000
PWX-06414 Condense: Closing down CAPI
PWX-06401 Condense: Ending successfully.
PWX-06110 Unloaded module 3 (CONDENSE).
PWX-06060 Controller: subtask Condense ended.
PWX-06107 Controller: All subtasks shut down.
PWX-06065 Controller: Condensing ended. Last checkpoint time 06/08/21 19:32:27
PWX-06039 Controller: Ending.

The following table describes key messages 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 statement that is used. In the example message, the covr value is blank, which indicates the CONN_OVR parameter in not used. Instead, the CAPI_CONNECTION statement is taken from the DBMOVER file that the DTLCFG DD points to.</td>
</tr>
<tr>
<td>PWX-06365</td>
<td>Indicates that none of the checkpoint data sets were found.</td>
</tr>
<tr>
<td>PWX-06100</td>
<td>Shows the restart tokens that were used for restart.</td>
</tr>
<tr>
<td>PWX-06103</td>
<td>Indicates that the operator responded Y to the PWX06101A WTOR message.</td>
</tr>
</tbody>
</table>
| PWX-06118, PWX-06119, PWX-6412 | Lists information about each capture registration. The PWX-06119 and PWX-06412 messages list the registration tag names. The PWX-06118 message includes:  
  • DBID / instance (DBName)  
  • Registration name and version  
  • Creator  
  • Table  |
<p>| PWX-06049         | Indicates that the cold start completed successfully.                       |
| PWX-06112         | Reports that the Controller task is starting the Command Handler, Condense, and Dump subtasks. |
| PWX-06404, PWX-06405 | 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. |</p>
<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWX-06413</td>
<td>Lists the highest restart tokens across all registration tags. Restart tokens consist of the following token types: · Sequence token (20 bytes), which contains UOW and sub-UOW sequences. · Logger token (16 bytes), which contains the PowerExchange 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 that is done before any processing starts. This file is a result of merging any checkpoint data brought forward from the last run (if a warm start is used) and any new data added or deleted from the CCT file.</td>
</tr>
<tr>
<td>PWX-06111</td>
<td>Reports that all subtasks have successfully completed 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 Condense job started and the reason it started. Reasons are: · Initialization completed. · A timeout occurred while PowerExchange Condense was waiting for commands. · A CONDENSE command was issued to the Condense job to end the wait period: F jobyname,CONDENSE · A pwxcmd condense command was issued to the PowerExchange Condense job from a Linux, UNIX, or Windows system.</td>
</tr>
<tr>
<td>PWX-09957</td>
<td>Indicates the first read from the Consumer API (CAPI). The example message indicates that Condense stopped because no data was received for 10 seconds, which is the maximum wait period specified by the NO_DATA_WAIT2 statement in the CAPTPARM configuration member.</td>
</tr>
<tr>
<td>PWX-06419</td>
<td>Indicates that a file switch occurred and the reason it occurred. Reasons are: · When FILE_SWITCH_CRIT=R, the number of records that is specified by the FILE_SWITCH_VAL parameter was reached. · When FILE_SWITCH_CRIT=M, the number of minutes that is specified by the FILE_SWITCH_VAL parameter was reached. · A FILESITCH command was received. · A pwxcmd fileswitch command was received.</td>
</tr>
<tr>
<td>PWX-06418</td>
<td>Identifies the data set names of the condense files that were closed during the file switch.</td>
</tr>
<tr>
<td>PWX-09967</td>
<td>Indicates that PowerExchange Condense read all of the changes that were available at the start the condense cycle. That is, PowerExchange Condense read up to the point that was the end-of-log (EOL) when the condense cycle started. The NO_DATA_WAIT2 wait interval now takes effect. If PowerExchange Condense receives no additional changes, the condense task stops. Use this message to determine if PowerExchange Condense has captured committed changes for registered tables of interest. Look for this message if a condense file does not receive change data within the time period you expect. Delays can occur for various reasons.</td>
</tr>
<tr>
<td>PWX-06415</td>
<td>Indicates the end of the condense cycle. Reports the number of insert, update, and delete records (Data=) and the number of UOWs that were processed. This message is issued only if records were processed.</td>
</tr>
</tbody>
</table>
### Message

<table>
<thead>
<tr>
<th>Message</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWX-06421</td>
<td>Indicates that the condense task is going into a sleep state. Condense waits for the NO_DATA_WAIT period to expire or until a CONDENSE or pwxcmd condense command is received before starting the next condense cycle.</td>
</tr>
<tr>
<td>PWX-06463, PWX-06404</td>
<td>Indicates that a SHUTDOWN or pwxcmd shutdown command was issued and is being processed.</td>
</tr>
<tr>
<td>PWX-06401</td>
<td>Indicates that the condense subtask successfully shut down after closing open condense files and taking a final checkpoint.</td>
</tr>
<tr>
<td>PWX-06039, PWX-06065</td>
<td>Indicates that the Condense job is ending and reports the timestamp of the final checkpoint.</td>
</tr>
</tbody>
</table>

### Controlling PowerExchange Condense

You can use the PowerExchange Condense commands to control PowerExchange Condense processing or display the status of PowerExchange Condense tasks.

The following table describes these commands:

<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>

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, 137
- Batch VSAM Change Data Capture, 155
- CICS/VSAM Change Data Capture, 164
- Datacom Table-Based Change Data Capture, 178
- DB2 for z/OS Change Data Capture, 196
- IDMS Log-Based Change Data Capture, 235
- IMS Log-Based Change Data Capture, 258
- IMS Synchronous Change Data Capture, 279
- Remote Logging of Data, 297
Adabas Change Data Capture

This chapter includes the following topics:

- **Adabas CDC Overview, 137**
- **Planning and Implementation Considerations, 138**
- **Configuring Adabas PLOG Archiving JCL, 140**
- **Configuring the Adabas ECCR, 142**
- **Managing Adabas CDC, 151**

Adabas CDC Overview

PowerExchange change data capture (CDC) for Adabas captures change data from Adabas archived PLOG data sets. PowerExchange maintains a catalog, called the PCAT file, of the PLOG data sets from which change data is captured.

The following figure shows the general architecture of Adabas CDC:

The Adabas ECCR, PowerExchange Logger for MVS, and PowerExchange Agent must run on the same z/OS system.
The PowerExchange Adabas ECCR reads change data from archived PLOG data sets that have entries in the PowerExchange PCAT file. The ECCR calls the Adabas ADASEL utility to extract records from the PLOG files and decompress the records. The PCAT utility, DTLCCADW, runs concurrently with the ECCR to keep the PCAT file up-to-date.

The ECCR passes the change data to the PowerExchange Logger for MVS. The ECCR must log all changes to a single PowerExchange Logger. The PowerExchange Logger stores the change data in its log files. If you use the optional PowerExchange Condense component, PowerExchange Condense reads the change data from the Logger log files and writes it to condense files.

When you run a CDC session in PowerCenter, PowerExchange works with PWXPC and PowerCenter to extract change data from the PowerExchange Logger log files or PowerExchange Condense files and to write that data to one or more targets.

To configure Adabas CDC in the PowerExchange Navigator, you must first create a data map to get metadata for the Adabas database. Then, create a capture registration for each Adabas source file. PowerExchange generates a corresponding extraction map.

If you use PowerExchange Condense, you must configure PowerExchange Condense parameters in the RUNLIB(CAPTPARM) member. Because PowerExchange Condense does not support Full condense processing for Adabas sources, you must select Part for the Condense option in the Adabas capture registrations. You can use either continuous extraction mode or batch extraction mode.

Before you start the Adabas ECCR, customize the ECCR JCL and ADAECRP1 options member. Also, populate the PCAT file with information about the latest archived PLOG data sets. To populate the PCAT file, customize the sample JCL in the SAMPUEX2 member and perform a PLOG file switch. The JCL runs the PCAT DTLCCADW utility internally to populate the PCAT file.

If you want to capture change data for spanned records in an Adabas 8.2.2 or later database, you must complete some additional configuration tasks.

Planning and Implementation Considerations

Review the following topics to plan your CDC implementation.

Gathering Information About Your CDC Environment

When planning your CDC implementation, gather the following information about your CDC environment:

- How often do PLOG switches occur?
- How often do you need to extract change data?
- What is the volume of changes on the Adabas source database?
- Do you need to capture changes from spanned records in an Adabas 8.2.2 or later database?
- How will you complete the initial target load prior to starting CDC?
Operational Considerations

Review the following Adabas CDC operational considerations:

- PowerExchange imports Long Alpha (LA) fields with a default length of 1,024 bytes. You can override this default length by editing the data map in the PowerExchange Navigator. 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 utility, DTLCCADW, can read archived Adabas PLOG records from tape data sets, including data sets that have a block size greater than 32,760. The Adabas ECCR can then capture change data from those PLOG records.

- If the Adabas File Description Table (FDT) for a source file is password protected in Adabas, the Adabas FDT password is required for a database row test of the Adabas extraction map and for connecting to the Adabas source file during a PowerCenter CDC session. Enter the Adabas FDT password at the following locations:
  - In the PowerExchange Navigator, enter the password in the ADABAS File Password field of the CAPXRT Advanced Parameters dialog box for a row test.
  - In PowerCenter Task Manager, edit the session. On the Mapping tab of the Edit Tasks dialog box, under Sources, click the Adabas source. In the right pane, under Properties, enter the FDT password in the ADABAS Password attribute.

- If you use Adabas 8.2.2 or later, PowerExchange can capture change data from Adabas spanned records up to their maximum size. The Adabas maximum size depends on the device type.

  Tip: If you capture change data from Adabas spanned records larger than 32 KB, PowerExchange might allocate a large number of spill files during the extraction of change data, depending on the MEMCACHE parameter setting in the UOWC CAPI_CONNECTION statement of the DBMOVER member. This situation can slow down subsequent extraction processing. To reduce the number of spill files, increase the MEMCACHE parameter value in the UOWC CAPI_CONNECTION statement.

- PowerExchange Logger operational issues can cause the Adabas ECCR to enter a wait state, which stalls change data capture until the Logger issues are resolved. After the Logger issues are resolved, the Adabas ECCR can resume change data capture without losing change data.

  Tip: Monitor the PowerExchange Logger carefully so that change data capture can proceed without interruption.

Change Capture from Multiple Adabas Databases

To capture changes from multiple Adabas databases, configure an Adabas ECCR for each database.

The JCL for each Adabas ECCR must reference unique versions of the following files and data sets:

- The PowerExchange Adabas ECCR configuration file to which the DTLCACFG DD statement in the JCL points
- The PowerExchange PLOG Catalog (PCAT) file to which the DTLADKSD DD statement in the JCL points
- The Adabas database data sets to which the DDASSOR1, DDDATAR1, and DWORKR1 DD statements in the JCL point
Change Capture from Adabas Spanned Records

In an Adabas 8.2.2 or later database, the Adabas ECCR can capture changes from Adabas spanned records. A spanned record is a logical record that is composed of a single physical primary record and up to four physical secondary records. Each record is stored in a separate data storage block. The block size depends on the Adabas device type.

Before you start the Adabas ECCR, perform the following PowerExchange and Adabas configuration tasks that are required to capture change data from source files with spanned records:

- In the PowerExchange Adabas ECCR JCL, add the following PARM to the EXEC statement:
  EXEC PGM=DTLCCADA, PARM=(ADA82)

  The ECCR JCL is usually in the PROCLIB member named prefixAD1EC, which was created during installation. If you do not include PARM=(ADA82), the ECCR does not capture changes for the sources files that contain spanned records.

- Apply the following Adabas SAG ZAPs to your Adabas load libraries:
  - AU823101 (ADA823)
  - AU824072 (ADA824)
  - AU825047 (ADA825)
  - AU826017 (ADA826)

- In Adabas, specify the following SRLOG=ALL parameter for the Adabas nucleus:
  ADARUN SRLOG=ALL

  The SRLOG=ALL parameter causes Adabas to log before and after images for the entire primary record and the entire secondary records that contain changes to the PLOG data sets.

- In Adabas, verify that record spanning is explicitly enabled for each Adabas file.

To check if an Adabas file contains spanned records, you can generate a report by using one of the following Adabas utilities:

- If you use Adabas 8.2.3 or later, use the ADAREP utility to generate a database report that indicates whether the Spanned Record option is set for the database and whether a specific file contains spanned records.
- Use the ADADBS utility SPANCOUNT function to display counts of primary records, secondary records, and non-spanned records for a file.

Configuring Adabas PLOG Archiving JCL

PowerExchange provides sample members to configure the Adabas PLOG archiving JCL. When this JCL runs, PowerExchange populates the PCAT file with information about the last archived PLOG data set.

Use one of the following sample members in the DTLEXPL library:

- SAMPUEX2. Contains PLOG archiving JCL that is submitted from within the Adabas UEX2 exit. If you use this JCL, the Adabas DBA must modify, assemble, link, stop, and start the Adabas nucleus.
- SAMPEXTU. Contains PLOG archiving JCL that you can submit as a job, outside of the Adabas UEX2 exit.
Note: PowerExchange creates the PCAT file during installation if you select Adabas CDC on the Data Sources page in the z/OS Installation Assistant.

1. If you use SAMPUEX2, complete the following steps:
   a. Customize the Adabas ADARUN parameters for your environment.
      For example:
      
      ```
      ADARUN DB=200,DE=3390,SVC=249,PROG=ADASEL
      ```
   b. Customize the contents of SAMPUEX2 and use it to modify your current UEX2 exit.
   c. Perform an Adabas PLOG file switch.

   Adabas calls the UEX2 exit, and PowerExchange adds the former PLOG to the PCAT file.

2. If you use SAMPEXTU, complete the following steps:
   a. Modify your current PLOG archiving JCL to reflect the contents of SAMPEXTU.
   b. Verify that the Adabas ADARUN parameters in the user JCL reflect the correct settings for your environment.
      For example:
      
      ```
      ADARUN DB=dbid,DE=3390,SVC=249,PROG=ADASEL
      ```

   The `dbid` variable is the database identifier.

Customizing the PowerExchange Sample SAMPUEX2 Exit

If you use the sample SAMPUEX2 exit for populating the PCAT file, use these directions to customize it.

1. In the JCL for PROLOG flips, before the comment block * CLOSE THE INTERNAL READER, add the following highlighted statements:

   ```
   CLI 0(4),EOJ LAST CARD PROCESSED ?
   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' *STR-01*
   BNE CLOSE i.e. it's a CLOG *STR-01*
   LA 4,1,(4) Skip over first EOJ mark *STR-01*
   SUBMIT2 DS OH *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 *
   *
   CLOSE DS OH *STR-01*
   CLOSE (INTRDR2) CLOSE INTERNAL READER.
   ```

2. Immediately before the comment * READER DCB, add the following JCL cards:

   ```
   * BELOW ARE PWX ADDITIONAL CARDS
   DC CL50'/PLOGCNTL EXEC PGM=DTLCCADW,COND=(4,LT),'
   DC CL50'/ // parm=(A)'
   DC CL50'/STIFLIB DD DSN=sceerun,DISP=SHR'
   DC CL50'/ // DD DSN=hlq.LOADLIB,DISP=SHR'
   DC CL50'/ // DD DSN=*,COPY.DSISIAUS1,DISP=SHR'
   DC CL50'/ // DD DSN=hlq.DBdbid.PCAT,'
   DC CL50'/ // disp=SHR'
   DC CL50'/ // DD DSN=hlq.RUNLIB(DBUPMOVE),'
   DC CL50'/ // disp=SHR'
   DC CL50'/ // DD DSN=hlq.DTLMSG,'
Configuring the Adabas ECCR

To configure the Adabas ECCR, configure the ECCR parameters and the ECCR JCL. Then test the ECCR installation.

Configuring the Adabas ECCR Parameters

Configure the Adabas ECCR parameters in the RUNLIB(ADAECRP1) member to which the DTLACFG DD statement in the ECCR JCL points.

Based on your input during installation, the z/OS Installation Assistant adds values for some parameters to the ADAECRP1 member. You can change these values if necessary.

The ADAECRP1 member can contain the following parameters:

- DBID=Adacollection_id
- DB_TYPE=ADA
- ECCRNAME=Ad1EC
- [NO_DATA_WAIT=minutes]
- [NO_DATA_WAIT2=seconds]
- [COLL_END_LOG=<xx_collendlog_ada>]
- [ADASEL_DSN=data_set_name]
- [CAPT_STATS=Y|N]
- [CAPT_STATS_INTVL=minutes]
- [CAPT_STATS_TERSE=Y|N]
- [COLDSTART=Y|N]
- [IGNORENOCHANGEUPDATES=Y|N]
- [ON_SUSPENSION_ERROR_CONTINUE=N|Y]
- [REFRESH_ALLOWED=Y|N]

The following table summarizes the Adabas ECCR parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBID</td>
<td>Required</td>
<td>The collection ID for the Adabas source. This parameter is customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Required</td>
<td>The database type, which must be ADA for Adabas.</td>
</tr>
<tr>
<td>ECCRNAME</td>
<td>Required</td>
<td>The Adabas ECCR name.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>Optional</td>
<td>The number of minutes that the Adabas ECCR waits after processing all PLOG entries in the PCAT file before next checking for new PLOG entries. If the ECCR finds no new entries, the NO_DATA_WAIT2 wait interval takes effect. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>NO_DATA_WAIT2</td>
<td>Optional</td>
<td>After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the ECCR waits after processing all PLOG entries in the PCAT before checking for new PLOG entries. If COLL_END_LOG is set to 0 and NO_DATA_WAIT is set to a value greater than 0, the NO_DATA_WAIT2 wait and retry cycle remains in effect as long as no new entries are found. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>COLL_END_LOG</td>
<td>Optional</td>
<td>Controls whether the ECCR must process a specific number of PLOGs before it can shut down. Used in conjunction with NO_DATA_WAIT and NO_DATA_WAIT2. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>ADASEL_DSN</td>
<td>Required</td>
<td>The name of a data set that contains the Adabas ADASEL parameters.</td>
</tr>
<tr>
<td>CAPT_STATS</td>
<td>Optional</td>
<td>Controls whether PowerExchange writes ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the Adabas log-based ECCR finishes processing a PLOG.</td>
</tr>
<tr>
<td>CAPT_STATS_INTVL</td>
<td>Optional</td>
<td>The interval, in minutes, for which the Adabas log-based ECCR collects and reports the number of inserts, deletes, updates, and commits that were captured. The ECCR also reports the timestamp in the log up to which changes were processed.</td>
</tr>
<tr>
<td>CAPT_STATS_TERSE</td>
<td>Optional</td>
<td>Controls whether the Adabas log-based ECCR prints PWX-06153 messages with capture statistics only for registered sources for which the ECCR captured changes.</td>
</tr>
<tr>
<td>COLDSTART</td>
<td>Optional</td>
<td>Controls whether the Adabas ECCR cold starts or warm starts.</td>
</tr>
<tr>
<td>IGNORENOCHANGEUPDATES</td>
<td>Optional</td>
<td>Controls whether the Adabas ECCR ignores records for which update operations did not change data.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ON_SUSPENSION_ERROR_CONTINUE</td>
<td>Optional</td>
<td>If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether the ECCR ends or continues when a UOW that contains change records to be discarded or captured started at an invalid point in the change stream relative to the suspension window.</td>
</tr>
<tr>
<td>REFRESH_ALLOWED</td>
<td>Optional</td>
<td>Controls whether you can use the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command refreshes the list of registered Adabas files that the ECCR uses for change capture processing.</td>
</tr>
</tbody>
</table>

**Note:** If a parameter has a default value or is not required, it is marked as optional. A default value is the value that PowerExchange uses if the parameter is not defined. For some parameters, the z/OS Installation Assistant provides recommended values, which you can accept or change.

More detailed parameter descriptions follow.

### ADASEL_DSN Parameter

The name of a data set that contains the Adabas ADASEL utility parameters.

The Adabas ECCR calls ADASEL to read the PLOGs. When the DTLCCADA function of the DTLCCADW utility updates the PCAT file with the last archived PLOG, PowerExchange prepends the ADASEL parameters to the parameters that the DTLCCADA function generates.

**Syntax:**

```
ADASEL_DSN=dsn
```

**Value:** For the `dsn` variable, enter the data set name that contains the ADASEL parameters.

### CAPT_STATS Parameter

Controls whether PowerExchange writes ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the Adabas log-based ECCR finishes processing a PLOG.

The ECCR issues PWX-06153 statistics messages that report the number of inserts, deletes, and updates that were captured for each registration, grouped by PLOG. The WTO messages notify the system operator that a PLOG was closed and also provide capture counts.

Regardless of the CAPT_STATS setting, the ECCR always reports the total number of inserts, deletes, updates, and commits across all of the PLOGs at the end of the ECCR run.

**Related Parameters:** CAPT_STATS_INTVL, CAPT_STATS_TERSE

**Syntax:**

```
CAPT_STATS={N|Y}
```

**Valid Values:**

- **N.** Do not write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing a PLOG.
• Y. Write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing a PLOG.

Default is N.

Usage Notes:
• If you do not set the global CAPT_STATS parameter to Y, you can issue the STATISTICS ON command after the ECCR is started to enable statistics reporting for each PLOG.
• If you also specify the CAPT_STATS_INTVL parameter or run the STATISTICS minutes command, the ECCR also reports the total number of inserts, deletes, updates, and commits for each interval.

For more information about the STATISTICS command and its parameters, see the PowerExchange Command Reference.

CAPT_STATS_INTVL Parameter

The interval, in minutes, for which the Adabas log-based ECCR collects and reports change capture statistics.

If you specify an interval, the ECCR prints a PWX-06181 message each time the interval elapses. The message reports the total number of inserts, deletes, updates, and commits that the ECCR processed during the interval and the last log position.

You can use this ECCR parameter to print statistics messages at a specific frequency, for example, every 60 minutes.

For the ECCR to print capture statistics, you must set the CAPT_STATS parameter to Y in the RUNLIB(ADAECRP1) member or run the ECCR STATISTICS ON command.

Related Parameters: CAPT_STATS, CAPT_STATS_TERSE

Syntax:

CAPT_STATS_INTVL=minutes

Value: For the minutes variable, enter a number from 1 through 1440. No default is provided.

Usage Notes:
• If you set the CAPT_STATS_INTVL parameter to 0, PowerExchange issues the error message PWX-00967.
• After you start the ECCR, message PWX-07805 identifies the collection interval that is defined.
• If you issue the STATISTICS minutes command, the number of minutes that is specified in the command overrides the CAPT_STATS_INTVL value for the duration of the ECCR run.

CAPT_STATS_TERSE Parameter

Controls whether the Adabas log-based ECCR prints PWX-06153 messages only for registered sources for which the ECCR captured changes. If no inserts, updates, or deletes occurred on a registered source, the ECCR does not report capture counts for it.

A PWX-06153 message reports the number of inserts, deletes, and updates that were captured for a registered source. The message is printed when the ECCR finishes processing a PLOG and at the end of the ECCR run.

For the ECCR to print statistics, you must set the CAPT_STATS=Y parameter in the RUNLIB(ADAECRP1) member or run the ECCR STATISTICS ON command.

Related Parameters: CAPT_STATS, CAPT_STATS_INTVL
Syntax:

\[
\text{CAPT\_STATS\_TERSE} = \{\text{N|Y}\}
\]

Valid Values:

- **N.** Print statistics for all registered sources, including sources without any change activity.
- **Y.** Print statistics only for registered sources for which the ECCR captured changes.

Default is **N**.

Usage Notes:

- If you set the **CAPT\_STATS\_TERSE** parameter to **N** and then issue the **STATISTICS SINCE TERSE** command, the command overrides the **CAPT\_STATS\_TERSE** setting for the SINCE period. PWX-06153 messages are then printed only for registered sources for which changes were captured.

**COLDSTART Parameter**

Controls whether the Adabas ECCR cold starts or warm starts.

Syntax:

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

Valid Values:

- **N.** The ECCR warm starts. The change capture process starts from where it last left off without loss of change data.
- **Y.** The ECCR cold starts. The change capture process starts from the oldest log in the PCAT.

Default is **N**.

Usage Notes: Regardless of how you set the **COLDSTART** parameter, the ECCR cold starts in the following circumstances:

- You use a new PowerExchange Logger to which the Adabas ECCR has not previously connected.
- You change the **ECCHRNAME** value in the **RUNLIB(ADAECRP1)** member.

**COLL\_END\_LOG Parameter**

Controls whether the Adabas ECCR shuts down after a specific number of PLOGs are processed. Used in conjunction with the **NO\_DATA\_WAIT** and **NO\_DATA\_WAIT2** parameters.

Related Parameters: **NO\_DATA\_WAIT** and **NO\_DATA\_WAIT2**

Syntax:

\[
\text{COLL\_END\_LOG} = \{\text{0|number}\}
\]

Valid Values:

- **0.** The number of PLOGs processed does not affect when the ECCR shuts down.
- A **number** greater than 0. The minimum number of PLOGs that the ECCR must process before it shuts down.

The z/OS Installation Assistant enters **1** for this parameter in the ECCR configuration member unless you specify another value. If this parameter is not defined, the default of **0** is used.
**DB_TYPE Parameter**

Required. The database type.

**Related Parameters:** DBID

**Syntax:**

```
DB_TYPE=ADA
```

**Value:** The value must be "ADA" for the Adabas ECCR.

**DBID Parameter**

Required. The collection ID for the Adabas source.

**Related Parameters:** DB_TYPE

**Syntax:**

```
DBID=collection_ID
```

**Value:** For the `collection_ID` variable, enter the collection ID that you entered for the registration group.

**Usage Note:** In conjunction with the DB_TYPE parameter, this parameter controls the registrations in the CCT file that the ECCR uses.

**ECCRNAME Parameter**

Required. A name for the Adabas ECCR.

**Syntax:**

```
ECCRNAME={ecr_name|PWXAD1EC}
```

**Value:** For the `ecr_name` variable, enter a 1- to 8-character alphanumeric string.

No default. However, the z/OS Installation Assistant generates an ECCR name that begins with the **PowerExchange Agent / Logger Prefix** value followed by AD1EC, for example, PWXAD1EC.

**Usage Notes:**

- The Adabas ECCR uses this parameter value for the following purposes:
  - To connect to the PowerExchange Logger to write change data
  - As 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 ECCR name value must be unique within a PowerExchange Logger group.
- If you change the ECCRNAME value, the ECCR cannot warm start from where it last left off.
- Informatica recommends that you use the same value for the ECCRNAME parameter and the Adabas ECCR started task or job name. This practice lets you to easily identify the Adabas ECCR when reviewing messages and data from the PowerExchange Logger.

**IGNORENOCHANGEUPDATES Parameter**

Controls whether the Adabas ECCR ignores records for which update operations did not change the data.

**Syntax:**

```
IGNORENOCHANGEUPDATES={N|Y}
```
Valid Values:

- **N.** The Adabas ECCR passes all records to the PowerExchange Logger, including the records with unchanged data.
- **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. Use this setting to reduce the number of records that are sent to the PowerExchange Logger.

Default is N.

Usage Notes:

- Use this parameter to configure the Adabas ECCR to ignore the many unchanged records that the ADAORD utility typically produces for online reorder operations.
- When you REORDER Adabas files, Adabas logs the before and after images of unchanged records to PLOG files. Unless you configure the ECCR to ignore these records, the ECCR captures the unchanged records from the PLOG files.

NO_DATA_WAIT Parameter

The number of minutes that the Adabas ECCR waits after processing all PLOG entries in the PCAT file before it next checks for new PLOG entries to process. If the ECCR finds no new entries, the NO_DATA_WAIT2 wait interval takes effect.

**Related Parameters:** COLL_END_LOG and NO_DATA_WAIT2

**Syntax:**

```
NO_DATA_WAIT= {0|minutes}
```

The z/OS Installation Assistant enters 5 for this parameter in the ECCR configuration member unless you specify another value. If this parameter is not defined, the default of 60 is used.

**Valid Values:**

- **0.** Shuts down the ECCR after it processes all PLOG entries in the PCAT.
- **A number** greater than 0. Specifies the number of minutes that the ECCR waits before checking for new PCAT entries. After this initial wait period expires without new changes, the NO_DATA_WAIT2 parameter controls subsequent waiting.

NO_DATA_WAIT2 Parameter

After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the Adabas ECCR waits after processing all PLOG entries in the PCAT file before checking for new PLOG entries to process.

If the COLL_END_LOG parameter is 0 and the NO_DATA_WAIT parameter is greater than 0, the Adabas ECCR retries the NO_DATA_WAIT2 wait and retry cycle on a continuing basis, until the ECCR is shut down or finds new entries in the PCAT.

**Related Parameters:** COLL_END_LOG and NO_DATA_WAIT

**Syntax:**

```
NO_DATA_WAIT2= {seconds|600}
```

**Value:** For the seconds variable, enter a number greater than 0.

The z/OS Installation Assistant enters 60 for this parameter in the ECCR configuration member unless you specify another value. If this parameter is not defined, the default of 600 is used.
ON_SUSPENSION_ERROR_CONTINUE Parameter

Optional. If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether
the Adabas ECCR ends or continues when a UOW that contains change records to be discarded or captured
started at an invalid point in the change stream relative to the suspension window.

Syntax:

ON_SUSPENSION_ERROR_CONTINUE={N|Y}

Valid Values:

• N. The ECCR issues an error message and ends.
• Y. The ECCR issues a warning and continues processing.

Default is N.

Usage Notes: If you use the PWXUCREG utility, this parameter controls whether the ECCR ends or continues
in the following situations:

• When discarding change records for a suspended registrations, the ECCR determines that the associated
UOW started before the beginning of the suspension window.
• When capturing change records for an activated registration, the ECCR determines that the associated
UOW started before the end of the suspension window.

The suspension window is the time period between the suspension timestamp and reactivation timestamp.
For more information about the PWXUCREG utility, see the PowerExchange Utilities Guide.

REFRESH_ALLOWED Parameter

Controls whether PowerExchange users can issue the ECCR REFRESH command. This command refreshes
the list of Adabas files with active capture registrations that the Adabas log-based ECCR uses to capture
change data.

When this parameter is set to Y, users can issue the REFRESH command after adding or deleting capture
registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The
REFRESH command updates the list of registered sources that the ECCR uses, without shutting down and
restarting the ECCR.

Syntax:

REFRESH_ALLOWED={N|Y}

Valid Values:

• N. Do not allow users to issue the REFRESH command. This option is intended for users of
PowerExchange versions earlier than 9.5.0, when the REFRESH command was not available. This option
maintains the previous behavior, which requires a restart of the ECCR after registration changes.
• Y. Allow users to issue the REFRESH command.

Default is N.

Configuring the Adabas ECCR JCL

You can run the Adabas ECCR as a started task or batch job. PowerExchange supplies a sample PROC for
running the ECCR as a started task.

The sample ECCR PROC is in the ECCRADA member of the RUNLIB library. The XIIZZZ998 installation job
copies the ECCRADA member to the PowerExchange PROCLIB library as xxxAD1EC. The xxx variable is the
PowerExchange Agent / Logger Prefix value that you specified in the z/OS Installation Assistant.
To configure the PROC for your environment, you must customize some of the DD statements in the PROCLIB(XXXAD1EC) member and the ADARUN parameters in the RUNLIB(ADACARD1) member.

**Note:** You must configure a separate Adabas ECCR for each Adabas database from which you capture change data.

1. In the RUNLIB(ADACARD1) member, verify that the ADARUN parameters reflect the correct settings for your environment.
   
   **For example:**
   
   ADARUN DB=DBID,DE=3390,SVC=249,PROG=ADASEL
   
   The `dbid` variable is the database ID.

2. In the PROCLIB(XXXAD1EC) member, complete the following steps:
   
   a. Customize the following DD statements for the data sets that the ECCR started task requires:
      
      ```
      //DTLACFGF DD DISP=SHR,DSN=RUNLIB(ADACRP1)
      //DTLADKSD DD DISP=SHR,DSN=HLQVSD.BDBID.PCAT
      //DDAASSR1 DD DISP=SHR,DSN=adabas.ASSOR
      //DDDATARA1 DD DISP=SHR,DSN=adabas.DATA
      //DDMINWORKR1 DD DISP=SHR,DSN=adabas.WORK
      ```
   
   b. If you plan to capture changes from Adabas spanned records, enter the `PARM=(ADA82)` option in the EXEC statement:
      
      ```
      EXEC PGM=DTLCCADA,PARM=(ADA82)
      ```
      
      **Note:** If you do not specify this PARM value, PowerExchange does not capture changes from the source files that include spanned records.

3. In the RUNLIB(ADACRP1) member, verify that the Adabas ECCR DBID parameter value is correct.
   
   The DBID value must match the collection identifier that is specified in the registration group that includes the capture registrations for the CDC sources.

4. If you use PowerExchange Condense, verify that the DBID parameter in the RUNLIB(CAPTADA1) member is correct.
   
   The DBID parameter must match the collection identifier that is specified in the registration group.

### Testing Adabas CDC Installation and Configuration

After you install and configure Adabas CDC, test it.

1. Update the Adabas file that you registered in the PowerExchange Navigator.
2. Complete a PLOG switch.
3. Review the PLOG switch job output to verify that condition code 0 was received on both the PLOG Copy and PCAT population steps.
   
   Record the name of the newly created archived PLOG data set name.
4. Review the Adabas ECCR job output to verify that change data occurred. In particular, look for message PWXEDM172808I in the EDMMSG data set.
   
   **Note:** The ECCR captures change from archived PLOGs and move the data to the PowerExchange Logger if new PCAT entries exist when the following events occur:
   
   - The ECCR is first started.
   - The NO_DATA_WAIT or NO_DATA_WAIT2 interval elapses.
5. Review the PowerExchange Logger output to verify that the ECCR read an archived PLOG.
Look for message PWXEDM172774I in the EDMMSG data set:

PWXEDM172774I Event Mark generated by ECCR xxxAD1EC for:
Finished with Plog copy ADABAS.DB00199.PLOG.G0022V00

Note the archived PLOG data set name.

6. If you do not use PowerExchange Condense, run a database row test in the PowerExchange Navigator:
   a. Open the extraction map.
   b. Click File > Database Row Test.
   c. In the DB_Type field, enter CAPXRT.
   d. In the Application field, enter an application name.
   e. Click Go.

7. If you use PowerExchange Condense, complete the following steps:
   a. Enter the fileswitch command to make the condense file available for extraction processing.
   b. Review the PowerExchange Condense job output to determine if records were added to the condense file.
   c. Review the PowerExchange message log file. Look for message PWX-06415, which contains information about the condense.
   d. In the PowerExchange Navigator, run a database row test.
      Enter CAPX in DB_Type field.

Managing Adabas CDC

You can manage the Adabas ECCR and PCAT files.

Starting the Adabas ECCR

To start the Adabas ECCR, issue the MVS START command with the name of the started task, such as:

START PWXAD1EC

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
Adding an Adabas Capture Registration

You might need to add a capture registration for a new or existing Adabas file from which you want to start capturing change data. In this case, you can use the REFRESH command to refresh the list of registered Adabas files for the Adabas log-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(ADAECRP1) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. If you need to begin capturing changes for the new registration from a specific point, stop any change activity on the source file.
2. In the PowerExchange Navigator, create the capture registration and set the Status field to **Active**.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:
   ```
   F eCCR_task_name,REFRESH
   ```
   The newly registered source is added to the list of registered sources for the ECCR.
5. Enable change activity on the source to resume.
6. If you use PowerExchange Condense, restart it.

Deleting an Adabas Capture Registration

You might need to delete a capture registration that has been used for change capture processing. In this case, you can use the REFRESH command to refresh the list of registered Adabas files for the Adabas log-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(ADAECRP1) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. Stop applications and other activities that update the source file that is associated with the registration that you are deleting.
2. Ensure that the ECCR has processed all of the Adabas PLOGs that contain changes for the source that is associated with the registration that you are deleting. Also ensure that the source data has been extracted and applied to the target. Then stop all workflows that extract change data for the source.
   **Note:** The ECCR cannot access an active PLOG until it is closed.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. In the PowerExchange Navigator, open the capture registration and set the Status field to **History**. Then delete the registration.
5. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:
   ```
   F eCCR_task_name,REFRESH
   ```
6. Enable change activity on the source to resume.
7. If you use PowerExchange Condense, restart it.
8. Restart extraction processing.
Suspending Change Capture for Registered Adabas Sources Temporarily

Use this task flow to suspend change capture processing for registered Adabas sources temporarily.

You perform some tasks with the PWXUCREG utility and other tasks outside of the utility on the z/OS system.

Before you begin, ensure that the REFRESH_ALLOWED=Y parameter is specified in the RUNLIB(ADAECRP1) member to which the DTLCACFG DD statement in the ECCR JCL points. You must have the authority to issue a REFRESH command after each registration status change.

1. Stop database activity for the registered source or sources for which you want to suspend capture registrations.

2. To suspend the capture registrations, use the PWXUCREG utility to issue the SUSPEND_REGISTRATION command.

   The suspension window opens. The utility sets the suspension timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLLOG log to report the registration status change.

   For each suspended registration, the PowerExchange Navigator Resource Inspector displays Suspended in the Status field and the suspension timestamp in the Suspend Time field. The Suspend Time value is not adjusted for the local time.

3. Perform a PLOG switch.

   This step ensures that all of the changes up to the point of the PLOG switch are captured for the active registration.

4. Enter the ECCR REFRESH command with the MVS MODIFY (F) command:

   F eccr_task_name,REFRESH

   The ECCR becomes aware of the registration status change and suspension timestamp. When the ECCR encounters the first change record to discard, it issues message PWX-07752. The ECCR discards change records that have a timestamp later than the suspension timestamp.

5. Run the jobs or processes that generate the changes that you do not want to capture for the source or sources that are associated with the suspended registrations.

6. To reactivate the capture registrations, use the PWXUCREG utility to issue the ACTIVATE_REGISTRATION command.

   The suspension window closes. The utility sets the activation timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLLOG log to report the registration status change.

   For each reactivated registration, the PowerExchange Navigator Resource Inspector displays Active in the Status field and the activation timestamp in the Active Time field. The Active Time value is not adjusted for the local time.

7. Perform a PLOG switch.

   This step ensures that all of the changes that occur during the suspension window up to the PLOG switch are discarded for the suspended registration.

8. Enter the ECCR REFRESH command with the MVS MODIFY (F) command again.

   The ECCR becomes aware of the registration status change and activation timestamp.

9. Enable database activity to resume on the registered source or sources.

   The ECCR starts capturing change records that have timestamps later than the activation timestamp.

   The ECCR issues message PWX-07753 when it encounters the first change record in the change stream after the end of the suspension window.

Note: You can automate this processing if appropriate for your environment.
Using the Adabas PCAT Utility (DTLCCADW) to Manage the PCAT File

The Adabas PCAT Utility, DTLCCADW, provides functions for populating, reporting on, and manipulating the PCAT file. The PCAT file stores information about the Adabas PLOG files for CDC.

PowerExchange uses the utility functions internally. However, occasionally, you might need to manually override the default DTLCCADW processing. For assistance in determining when to use the utility, contact Informatica Global Customer Support.

The PCAT utility is controlled by parameters on the PARM option in the EXEC statement. PowerExchange provides example JCL for each DTLCCADW function in the DTLEXPL library. The example member names have the format DTLCCAD\(x\), where \(x\) corresponds to a function identifier.

For more information about the utility, see the *PowerExchange Utilities Guide*. 
Batch VSAM Change Data Capture

This chapter includes the following topics:

- Batch VSAM CDC Overview, 155
- Configuring Batch VSAM Jobs for CDC, 158
- Managing Batch VSAM Change Data Capture, 159
- Managing VSAM Schema Changes, 162

Batch VSAM CDC Overview

PowerExchange batch change data capture (CDC) for VSAM synchronously captures changes made to registered VSAM data sets by batch jobs.

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 MVS for logging. After the batch program opens the VSAM data set, PowerExchange records a single unit of work (UOW) in the PowerExchange Logger for all changes that the batch program makes to that VSAM data set. PowerExchange commits the UOW that contains the changes for the VSAM data set when the batch program closes the VSAM data set.

ECCR Relationships with Other PowerExchange Components

The batch VSAM ECCR interacts with PowerExchange components such as the PowerExchange Logger for MVS and PowerExchange Agent to perform CDC.

Consider the following the following relationships:

- The batch VSAM ECCR, PowerExchange Logger, and PowerExchange Agent must run on the same z/OS system.
- The batch VSAM ECCR must log all changes to a single PowerExchange Logger.
If you use the Post-Log Merge option of the PowerExchange Logger, you can capture changes that originate on different z/OS systems. In this case, you must run a PowerExchange Logger on each z/OS system where changes to the source VSAM data sets occur.

PowerExchange Logger operational issues can cause the batch CDC job to enter a wait state, which could prevent further capture and recording of change data. After you resolve the Logger operational issues, PowerExchange continues to capture and record change data without data loss.

**Tip:** Carefully monitor the PowerExchange Logger to ensure that change data capture can proceed without interruption.

**RELATED TOPICS:**
- "Monitoring the PowerExchange Logger for MVS" on page 72
- "Using Post-Log Merge" on page 90

### Batch VSAM ECCR Restrictions

Batch VSAM ECCR processing is subject to the following restrictions:

- The batch VSAM ECCR does not capture change data for the following items:
  - Environments with multiple task control blocks (TCBs)
  - Natively updated alternate indexes
  - Records that are larger than 32,660 bytes
  - Spanned ESDSs
  - Paths defined over ESDSs
  - Control interval (CI) mode updates
  - VSAM data sets that are 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 that have certain names or prefixes from change data capture. The exclude table contains the following types of entries:
  - Complete load module names
  - Prefixes for load module names
  - Prefixes for data set names

Based on the exclude table, the batch VSAM ECCR does not capture change data for the following VSAM data sets:

- Data sets that begin with any data set prefix in the internal exclude table.
- Data sets that are opened by load modules that match the specific load module names or prefixes in the internal exclude table.
The following table lists the load module names and prefixes in the 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>
<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>BNJLINTX</td>
<td>Specific</td>
<td>IBM Tivoli NetView for z/OS</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>DUIFT000</td>
<td>Specific</td>
<td>IBM Tivoli NetView 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>EKGTC000</td>
<td>Specific</td>
<td>IBM Tivoli NetView for z/OS</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>IEFIIC</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 with the prefix UCC1</td>
</tr>
</tbody>
</table>
Configuring Batch VSAM Jobs for CDC

For batch jobs to use the batch VSAM ECCR, you must edit the batch job JCL to add the PowerExchange libraries and to activate the batch VSAM ECCR interface.

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 included 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 Interface Automatically

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:

```
cmddm_prelxSTVSMCECCRT
```

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:

```
/EDCMCUOW 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.
Controlling the Batch VSAM ECCR

You can control the batch VSAM ECCR interface by using PowerExchange Agent commands.

These commands have the following syntax:

```
cmd_prefix keyword VSAMECCCR
```

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 z/OS 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 VSAMECCCR/RELOAD to load a new batch VSAM ECCR interface module into Extended Common Storage Area (ECSA). The module is placed at the beginning of the LPA queue in an active state. <strong>Warning:</strong> This command affects all Batch VSAM ECCRs on the same z/OS 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. <strong>Warning:</strong> This command affects all Batch VSAM ECCRs on the same z/OS system.</td>
</tr>
</tbody>
</table>

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 EDMMMSG 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 . . . . . . . . . . . . . . . : AUSL
  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'
PWXEDM1728091 Change Capture active for VSAM file AUSQ.AVSAM.VSMD01 Edition=4A80B00000000001, EDMMNAME=VSAM.AUSQ.AVSAM.VSMD01
PWXEDM1728091 Change Capture counts for AUSQ.AVSAM.VSMD01: 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)
```

**Note:** This report also includes message PWXEDM1728861, which indicates any load module replacements that have been applied.
Stopping Change Data Capture for VSAM Sources

You can stop change data capture for all VSAM data sets or a specific data set.

To stop change capture processing for all VSAM data sets, stop the batch VSAM ECCR interface.

To stop change capture processing for a specific registered VSAM data set, deactivate or delete the capture registration and close the data set.

**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:

```
PWXEDM172818I Joined XCF group 'AUSL' as member 'AUSVSUPD'
PWXEDM172841I EDM ECCR AUSVSUPD connected to EDM Logger AUSL, Log RBA=X'0000560078040000'
PWXEDM1728081 Change Capture active for VSAM file AUSQA.VSAM.VSMDM01
Edition=C4E3D30000000001, EDNAME=AUSMAUQA.VSAM.VSMDM01
PWXEDM172809I Change Capture counts for AUSQA.VSAM.VSMDM01: Insert=0, Update=5, Delete=0
PWXEDM172841I EDM ECCR AUSVSUPD disconnected from EDM Logger AUSL, Log RBA=X'0000560084000000'
PWXEDM172818I Left XCF group 'AUSL' as member 'AUSVSUPD'
PWXEDM172829I 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 VSAEECCR
```

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*. 
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

Consider these batch execution and recovery issues for PowerExchange CDC.

You might need to change some operational recovery procedures to accommodate change data propagation.

Point-in-Time Recovery

Point-in-time recovery invalidates the change data that the PowerExchange Logger logged and that the batch job recorded.

Standard point-in-time recovery does not indicate that the PowerExchange Logger data is invalid to the processors of that data.

A processor of PowerExchange Logger data must perform the following processing if you need to use point-in-time recovery:

- Recover the source to the correct point-in-time.
- Recover the target to the correct point-in-time. This recovery might require rematerialization of the targets. If you use PowerExchange Condense, wait until the Condense process captures all outstanding changes from the PowerExchange Logger. Then regenerate the restart tokens for the PowerCenter CDC sessions that update the targets.
- Reset the change processor to restart processing when the recovery is complete.

DFSMSdfp Checkpoint/Restart

PowerExchange for VSAM CDC does not support DFSMSdfp Checkpoint/Restart.

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.
This chapter includes the following topics:

- **CICS/VSAM CDC Overview, 164**
- **Planning for CICS/VSAM CDC, 164**
- **Configuring CICS for CDC, 169**
- **Starting the CICS/VSAM ECCR, 172**
- **Managing CICS/VSAM CDC, 172**

## CICS/VSAM CDC Overview

PowerExchange change data capture (CDC) for CICS/VSAM synchronously captures changes that CICS transactions make to registered VSAM data sets.

The CICS/VSAM ECCR runs in the CICS region. To capture changes, the ECCR uses CICS global user exits (GLUE) and a PowerExchange task-related user exit (TRUE).

The ECCR passes captured changes to the PowerExchange Logger for MVS. The PowerExchange Logger logs the changes in its log files. PowerExchange, in conjunction with PWXPC and PowerCenter workflows, can then extract the changes from the PowerExchange Logger log files for propagation to targets in near real time.

Review the CDC configuration and management information that is specific to CICS/VSAM data sources. For other implementation tasks, see "Summary of CDC Implementation Tasks" on page 26. For example, you must create data maps and capture registrations in the PowerExchange Navigator, and define a PWX NRDB connection in PowerCenter.

## Planning for CICS/VSAM CDC

Before you configure CICS/VSAM CDC, review this planning information.
CICS/VSAM CDC Requirements and Restrictions

Before implementing CICS/VSAM CDC, consider its requirements and restrictions.

- The CICS/VSAM ECCR can capture changes only from local VSAM ESDS, KSDS, RRDS, or VRRDS data sets and CICS-maintained data tables.

- If you specify CCERR=ABEND in the EDMSDIR options module and the ECCR abends or encounters an error during initialization, PowerExchange performs one of the following actions to ensure data integrity:
  - Ends and backs out in-flight CICS transactions on VSAM source files during syncpoint processing.
  - If necessary, shuts down the CICS region, as if you had issued the CICS command CEMT PERFORM SHUTDOWN IMMEDIATE NORESTART.

  **Tip:** In production environments where data integrity is important, specify CCERR=ABEND. If you specify CCERR=CONTinue instead, data integrity might not be maintained.

- In CICS, define VSAM source data sets as recoverable by using the RECOVERY(BACKOUTONLY) or RECOVERY(ALL) option. Alternatively, you can define VSAM data sets other than ESDS data sets as nonrecoverable by using the RECOVERY(NONE) option under either of the following circumstances:
  - You specify CCERR=CONTinue in the EDMSDIR options module and specify the AllowRecoveryNone option in the INITPARM statement for the EDMKOPER module.
  - You specify CCERR=ABEND in the EDMSDIR options module.

  **Note:** You must define ESDS data sets as recoverable for the ECCR to properly handle backouts of WRITE requests during CDC processing. If you define an ESDS file with the RECOVERY(NONE) option and specify the AllowRecoveryNone option in the INITPARM statement, the ECCR captures change data for the ESDS data set but cannot process backouts. CICS does not perform backout processing for nonrecoverable data sets when a transaction abend occurs or a SYNCPOINT ROLLBACK request is made.

- The CICS/VSAM ECCR must be active in each CICS region that owns VSAM files from which you capture changes.

- If a CICS transaction updates CICS/VSAM files and other data sources outside of the CICS region in the same unit of work, for example, DB2 tables or IMS databases, the CICS/VSAM ECCR captures only the changes to the CICS/VSAM files.

- To apply changes from multiple data source types to targets in the order that the changes were made by a CICS transaction, use a staging table. For each data source type, extract the changes and insert them into the staging table. Include the PowerExchange-generated DTL__CAPXTIMESTAMP column. Then, extract changes from the staging table, in sequential order based on the DTL__CAPXTIMESTAMP values, and apply the changes to the target tables in that order.

- The CICS/VSAM ECCR can capture change data from ESDS data sets that use both the 32-bit relative byte addressing (RBA) and 64-bit extended relative byte addressing (XRBA). However, the ECCR does not capture change data for the following types of ESDS items:
  - Spanned ESDSs
  - Paths defined over ESDSs
  - An alternate Index (AIX) that points to an ESDS base cluster

  **Note:** In the log records that the ECCR generates, the RBA is always stored right-justified in an 8-byte field, regardless if it is a 4-byte 32-bit address RBA or an 8-byte long 64-bit extended RBA.

CICS/VSAM CDC Use of CICS Global and Task-Related Exit Points

PowerExchange uses several CICS global exit points and a single task-related user exit point during CICS/VSAM CDC.
CICS Global Exit Points

PowerExchange uses the following global exit points:

**XFCFRIN**

Exit point for invoking the PowerExchange EDMKIRnn exit program before a CICS File Control Domain request such as READ, WRITE, DELETE, or REWRITE. This exit enables the CICS/VSAM ECCR to capture changes to VSAM files that are registered for CDC. Use the XFCFRIN exit with the XFCFROUT exit. However, these exit points do not support the processing of backouts for recoverable ESDS data sets. For backout processing of ESDS data sets, use the XFCLDEL and XFCBOUT exit points.

If the XFCFRIN exit detects any DELETE operation that uses the RIDFLD operand, the program reads the record as an UPDATE and then issues another DELETE without the RIDFLD operand. The XFCFROUT exit can then capture and log all of the required information for the deletion.

*Note:* The suffix nn in the exit program names corresponds to the first two digits of the CICS TS Internal Release level.

CICS TS V5.2 corresponds to level 690 and the exit program at XFCFRIN is EDMKIR69.

**XFCFROUT**

Exit point for invoking the PowerExchange EDMKIRnn exit program after a CICS File Control Domain request. This exit enables the CICS/VSAM ECCR to capture changes to VSAM files that are registered for CDC and transmit the changes to the PowerExchange Logger for MVS. Use the XFCFROUT exit with the XFCFRIN exit. However, this exit point does not support the processing of backouts for recoverable ESDS data sets. For backout processing of ESDS data sets, PowerExchange uses the XFCLDEL and XFCBOUT exit points.

**XFCSREQ**

Exit point for the PowerExchange exit program that is called before a data set OPEN request is processed. At this CICS exit point, the CICS/VSAM ECCR determines whether the data set that is being opened is registered for change data capture. If the data set is registered, change data capture will be active for this data set.

**XFCSREQC**

Exit point for the PowerExchange exit program that is called after a successful file OPEN or CLOSE request with a return code of 4 or lower and after a failed OPEN request. If an OPEN request is successful and the data set is registered for change data capture, the program retains the Change Capture Directory entry for the data set. If the OPEN request fails, the program removes the Change Capture Directory entry for the data set.

**XFCLDEL**

Required only for recoverable ESDS source data sets in an online CICS TS environment. Exit point for the following exit programs that are required to process transaction backouts for a recoverable ESDS data set that is registered for change data capture, when a transaction abend or syncpoint rollback occurs:

- A *user-defined* program that marks the backout records as logically deleted and then writes them back to the ESDS data set. You must logically delete backout records because CICS TS does not provide a mechanism to directly delete these records from an ESDS data set. To define this program, customize the IBM-supplied sample program in the DFHSLDEL member of the CICS SAMPLIB library. Then install the customized backout exit program at the XFCLDEL exit point using the TBEXITS system initialization parameter. Typically, a logical deletion is indicated by setting the first character (or byte) of the record to X'FF'. When a record is marked as logically deleted, the CICS/VSAM ECCR is able to determine that the before and after images of the record are different and generate an appropriate change record. The change record is then sent to the Power Exchange Logger for MVS.
• The PowerExchange EDMKLDnn program that is invoked whenever a WRITE operation on a VSAM ESDS data set is backed out. In this program name, nn represents the CICS TS version. The program retrieves the after image of the backed-out record that was logically deleted and then generates an UPDATE record that contains both the before and after images. The generated record is written to the Change Capture Log. In this manner, the CICS/VSAM ECCR processes the logically deleted record as an UPDATE to prevent transaction backout failures and the generation of numerous error messages. This processing avoids data integrity issues in the CDC environment that are caused by backout failures. PowerExchange provides a sample EDMKLDnn program in the CICS TS SAMPLIB library.

**Important:** This exit program must be the last program enabled at the XFCLDEL exit point. You can use the EDMC XPGM command to verify that this program is last.

Use the XFCLDEL exit point with the XFCBOUT exit point.

**XFCBOUT**

Required only for recoverable ESDS source data sets in an online CICS TS environment. At CICS/VSAM ECCR initialization in the CICS region, the EDMKBOnn program is installed at this global exit point. In the program name, nn represents the CICS TS version. This program captures the before image of each record in a recoverable ESDS data set that is to be backed out because of a transaction abend or syncpoint rollback. The program runs before CICS attempts to back out each record. PowerExchange CDC uses the EDMKBOnn program at the XFCBOUT exit point with the EDMKLDnn program at the XFCLDEL exit point to get both the before and after images for the backed out record.

The following considerations apply to using these global exit points:

• To enable and activate all of the PowerExchange CDC exit programs at the global exit points at CICS initialization, you can specify an entry for the EDMKOPER program in the CICS PLTPI system initialization parameter. The EDMKOPER program enables these exit programs during the second phase of program load table (PLT) processing at CICS startup.

• All of the PowerExchange CDC exit programs at the global exit points can process only uncompressed data records.

• If you issue the CICS/VSAM ECCR command EDMC INIT, the EDMC transaction both initializes the ECCR and dynamically installs the appropriate CICS/VSAM CDC exit programs at the XFCFRIN, XFCFROUTE, XFCREQ, XFCREQE, XFCREQC, XFCBOUT, and XFCLDEL exit points.

• If other exit programs in the CICS region are installed at the same global exit points that CICS/VSAM ECCR uses, the PowerExchange-supplied exits might not get control in the correct order. In this case, the CICS/ VSAM ECCR might not capture change data properly. Ensure that the other exits do not affect the processing of the PowerExchange-supplied exits.

**Note:** CICS gives control to the exits based on the order in which they are enabled in CICS.

• To determine whether the CICS region has other exit programs installed at one of these global exit points, use the CICS CECI transaction with the following system commands to browse the exit list:

```
INQUIRE EXITPROGRAM EXIT(global_exit_point_identifier) START
INQUIRE EXITPROGRAM NEXT
INQUIRE EXITPROGRAM END
```

For more information about the CECI transaction and INQUIRE EXITPROGRAM command, see the IBM CICS Transaction Server system programming reference.

Alternatively, you can use the CICS/VSAM ECCR command EDMC XPGM or EDMC EXITPGMS to display the global exit points and task-related user exit points that are used by the PowerExchange exit programs and any other exit programs that are enabled at the same exit points. For more information, see the **PowerExchange Command Reference.**
• If the CICS/VSAM ECCR captures change data from a recoverable ESDS data set and you use multiple exit programs at the CICS XFCLDEL global exit point, ensure that the EDMKLDnn exit program is last. The EDMKLDnn exit program must be the last exit program called at this exit point by the CICS services.

• When multiple exit programs are defined at the XFCLDEL exit point for a registered ESDS data set, the EDMKLDnn program sets the return code to UERCLDEL instead of percolating any return codes from prior exit programs.

CICS Task-Related User Exit

CICS/VSAM CDC uses a single CICS task-related user exit (TRUE) to capture relevant syncpoints and UOW information for each task that updates a registered data set. The ECCR uses this information to coordinate syncpoint processing with the PowerExchange Logger for MVS and to handle ECCR requirements related to CICS shutdown processing.

You can use the CICS/VSAM ECCR command EDMC XPGM or EDMC EXITPGMS to display the task-related user exit points and global user exit points that are used by the PowerExchange exit programs and any other exit programs that are enabled at these same exit points. For more information, see the PowerExchange Command Reference.

RELATED TOPICS:
• “Using the EDMC Transaction and Keywords to Manage the CICS/VSAM ECCR” on page 173

CICS/VSAM ECCR Relationships with Other PowerExchange Components

The CICS/VSAM ECCR works with other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent to propagate change data.

Consider the following operational issues:
• The CICS/VSAM ECCR sends changes to a single PowerExchange Logger.
• The CICS/VSAM ECCR must run on the same z/OS system as the PowerExchange Logger and PowerExchange Agent.

However, if you use the Post-Log Merge option of the PowerExchange Logger, you can capture changes that originate from different z/OS systems. In this case, run a PowerExchange Logger on each z/OS system from which CICS transactions write changes to the VSAM source data sets.

• Operational issues in the PowerExchange Logger, such as waits for tape mounts, can cause CICS transactions to enter a wait state and hold up all CICS tasks. After you resolve the PowerExchange Logger issues and CICS transactions resume processing, PowerExchange resumes capturing and recording change data without any data loss.

To ensure that change data capture proceeds without interruption, monitor PowerExchange Logger processing.

RELATED TOPICS:
• “Monitoring the PowerExchange Logger for MVS” on page 72
• “Using Post-Log Merge” on page 90
Configuring CICS for CDC

To capture changes that CICS transactions make to VSAM data sets, edit the JCL and startup procedures for the CICS region. Also define the CICS/VSAM ECCR programs and transaction to CICS.

1. Edit the CICS JCL.
   a. Specify the PowerExchange LOAD library in the DFHRPL DD and STEPLIB DD statements.
      
      Note: If you included the LOAD library in the MVS LNKLST concatenation, add the library to the DFHRPL DD only.
   b. Add the EDMPARMS DD statement. Include a DSN option that points to the PowerExchange Logger USERLIB library, for example:

      //EDMPARMS DD DISP=SHR,DSN=hlq.logger_name.USERLIB

      The hlq variable is the high-level qualifier that you specified at installation.

2. To initialize the CICS/VSAM ECCR during the third stage of CICS initialization, add the EDMKOPER module name to the second part of CICS PLTP1 list, after the DFHDELIM entry.

   Note: Informatica recommends that you add EDMKOPER to the CICS initialization list to reduce the possibility of missing changes in a high-volume production environment. However, you can use the EDMC transaction with the INIT keyword to manually initialize the ECCR. For more information, see the PowerExchange Command Reference.

3. Add the //EDMKOVRD DD statement to the CICS startup procedure to override default capture settings or to activate change data capture for ESDS data sets in an online CICS TS environment.

   In the //EDMKOVRD DD statement or in the data set to which the DD statement points, specify options that selectively enable or disable CDC by VSAM data set type or for individual VSAM data sets. You can disable CDC even for data sets that are registered for change capture. Also, you can specify options for handling backouts for recoverable ESDS data sets.

   Options are:

   
   **CAPTURE_ESDS={ON|OFF}**

   Enables or disables change data capture for ESDS data sets. You must explicitly enter ON to enable CDC for ESDS data sets. Default is OFF.

   **CAPTURE_KSDS={ON|OFF}**

   Enables or disables change data capture for KSDS data sets. Enter OFF if you need to disable CDC for KSDS data sets. Default is ON.

   **CAPTURE_RRDS={ON|OFF}**

   Enables or disables change data capture for RRDS and VRDS data sets. Enter OFF if you need to disable CDC for RRDS and VRDS data sets. Default is ON.

   **CAPTURE_CMDT={ON|OFF}**

   Enables or disables change data capture for CICS-maintained data tables. Enter OFF if you need to disable CDC for CICS-maintained data tables. Default is ON.

   **BACKOUTRC={OVERRIDE|NOOVERRIDE}**

   For recoverable ESDS data sets, controls whether to override the return codes from any other active exit programs that are invoked at the XFCLDEL global exit point prior to the PowerExchange EDMKLDnn exit program for processing backouts as logical deletions. Options are:

   - **OVERRIDE**. Override the return codes from any prior exit programs at the XFCLDEL global exit point with the UERCLDEL return code from the EDMKLDnn program.
• **NOOVERRIDE.** Percolate the return codes from any prior exit programs at the XFCLDEL global exit point. In this case, the return code of a prior exit program might supercede the return code from the EDMKLDnn program. With this option, the ESDSFFAIL option is ignored.

\[
ESDSFAIL=\{YES|NO\}
\]

For recoverable ESDS data sets from which change data is captured, controls whether backouts are allowed to fail after a transaction abend or synpoint rollback. By default, the PowerExchange exit programs that you define at the XFCBOUT and XFCLDEL global exit points handle backouts as logical deletions with before and after images so that the change can be processed during CDC. If you capture change data from recoverable ESDS data sets, set this option to NO. If you enter ESDSFFAIL=YES, backouts will fail with many error messages.

If you specified BACKOUTRC=NOOVERRIDE, this option is ignored.

**DSN=dataset_name[,option]...**

To enter overrides for a specific VSAM source data set, specify the fully qualified data set named followed by one or more of the following optional options:

- **\{CAPTURE|NOCAPTURE\}.** Enter CAPTURE to enable change data capture for the specified data set, or enter NOCAPTURE to disable change data capture for the data set. If you specify NOCAPTURE, the BACKOUTOVERRIDE and BACKOUTFAIL options are ignored.

- **\{BACKOUTOVERRIDE|NOBACKOUTOVERRIDE\}.** For a recoverable ESDS data set, controls whether to override the return codes from any other active exit programs that are invoked at the XFCLDEL global exit point prior to the PowerExchange EDMKLDnn exit program. Enter BACKOUTOVERRIDE to override the return codes from any prior exit programs with the UERCLDEL return code from the EDMKLDnn exit program. Enter NOBACKOUTOVERRIDE to percolate the return codes from prior exit programs. If you specify NOBACKOUTOVERRIDE, do not specify NOBACKOUTFAIL.

- **BACKOUTFAIL|NOBACKOUTFAIL.** For a recoverable ESDS data set, controls whether backouts are allowed to fail after a transaction abend or synpoint rollback. Enter BACKOUTFAIL to allow backouts to fail, or enter NOBACKOUTFAIL to allow the PowerExchange exit programs that you define at the XFCBOUT and XFCLDEL global exit points to handle backouts as logical deletions with before and after images and continue CDC processing.

If you enter multiple options, separate them from one another with a comma. Do not use a space instead. For example:

```
DSN=EM.VSAM.ESDSS4,CAPTURE,BACKOUTOVERRIDE,NOBACKOUTFAIL
```

**Note:** You can use the options in the DSN statement to override the CAPTURE\_vsam\_source\_type, BACKOUTRC, and ESDSFFAIL settings for a specific data set only. To activate the override options, issue the EDMC REFRESH command.

4. Verify that you use a unique CICS/VSAM ECCR name for each CICS region that connects to PowerExchange and that each ECCR name is also unique within a PowerExchange Logger group.

PowerExchange uses an ECCR name for the following purposes:

- As the member name to join the XCF group that the PowerExchange Logger uses
- As part of the ECCR-UOW control information for each change record that the ECCR sends to PowerExchange Logger log files

The default ECCR name is the CICS SYSID value that is specified in the SYSIDNT parameter in the CICS system initialization table (SIT).
To override the default name, enter the INITPARM statement for the EDMKOPER module in the SIT or a SIT override:

```
INITPARM='EDMKOPER='option')
```

The following table lists the valid options and their substitution values:

<table>
<thead>
<tr>
<th>Option</th>
<th>Substitution Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>*SYSID</td>
<td>The CICS SYSID value</td>
</tr>
<tr>
<td>*JOBNAME</td>
<td>The CICS job or started task name</td>
</tr>
<tr>
<td>*APPLID</td>
<td>The VTAM application control block (ACB) name</td>
</tr>
<tr>
<td>1 through 8 alphanumerical characters</td>
<td>No substitution</td>
</tr>
</tbody>
</table>

When EDMKOPER processes the `option` value, it translates lowercase characters to uppercase characters.

**Tip:** Informatica recommends that you use the CICS job or started task name as the ECCR name. This practice makes it easier to identify the ECCR in PowerExchange Logger messages and output.

5. Define the CICS/VSAM ECCR programs and transaction to CICS.

To perform this step, use the sample members in the PowerExchange SAMPLIB library for the supported CICS Transaction Server (TS) versions.

The following table identifies these sample members:

<table>
<thead>
<tr>
<th>CICS/TS Version</th>
<th>Member Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1 and 3.2</td>
<td>#CICSV62</td>
</tr>
<tr>
<td>4.1</td>
<td>#CICSV66</td>
</tr>
<tr>
<td>4.2</td>
<td>#CICSV67</td>
</tr>
<tr>
<td>5.1</td>
<td>#CICSV68</td>
</tr>
<tr>
<td>5.2</td>
<td>#CICSV69</td>
</tr>
</tbody>
</table>

Copy the sample member for your CICS/TS version, and then edit the copy.

Use the CICS DFHCSDUP utility program or RDO commands to add the CICS/VSAM ECCR programs and transaction to the CICS system definition.

By default, the transaction name is EDMC. However, you can use another name.

If you need to change the ECCR name, you must either cold start the CICS region or enter NEWSIT=YES in the EXEC PARM option of the JCL or in the SIT or SIT override.
Starting the CICS/VSAM ECCR

The CICS/VSAM ECCR can start automatically during the third stage of CICS initialization, or you can manually start it.

Before you start ECCR, verify the following items:

- You have a CICS/TS version and maintenance level that PowerExchange supports for CDC.
- The CICS file control table (FCT) entries for VSAM source data sets specify RECOVERY(BACKOUTONLY) or RECOVERY(ALL). If the FCT entries specify RECOVERY(NONE), verify that you specified CCERR=CONTINUE in the EDMSDIR options module and AllowRecoveryNone in the INITPARM statement for the EDMKOPER module, or you specified CCERR=ABEND in the EDMSDIR options module.
- You started the PowerExchange Agent and then the PowerExchange Logger. Both are currently active.

To start the CICS/VSAM ECCR:

- Use one of the following methods:
  - If you added the EDMKOPER module name to the PLT initialization list (PLTPI), as recommended, start the CICS region.
    The ECCR starts during the third stage of CICS initialization.
  - From a CICS terminal or operator console, manually enter the CICS transaction name for the ECCR followed by the INIT keyword. If you use the default transaction name of "EDMC," the syntax is:

    EDMC INIT

After the ECCR starts, PowerExchange writes a report that indicates key options that are in effect. For example:

    LOG START
    PWXEDM1728521 Options in effect:
    Load Library containing EDMSDIR. . . . . . : PWX.PWXL.USERLIB
    EDMSDIR assembly date/time . . . . . . . . . . : 20100831 18.19
    Product distribution date . . . . . . . . . . . : 20100831
    Product distribution level . . . . . . . . . . . : 2.4.05
    Agent Id . . . . . . . . . . . . . . . . . . . . : PWX
    Logger Id . . . . . . . . . . . . . . . . . . . . : PWXL
    SYSOUT class . . . . . . . . . . . . . . . . . . . : *
    Action if ECCR error encountered . . . . . : Continue

To determine if startup was successful, look for messages PWXEDM176405I, PWXEDM176436I, and PWXEDM176404I in the data set that is specified in the MSGUSR DD of the JCL. For example:

    PWXEDM176405I CICS/VSAM ECCR Initialization in progress
    PWXEDM176436I Options in effect: ECCR Name ID=cccr_name, CCERR=CONTINUE AllowRecoveryNone
    PWXEDM176404I CICS/VSAM ECCR Initialization completed successfully

Managing CICS/VSAM CDC

After the CICS/VSAM CDC system is running, you might need to perform occasional management and maintenance tasks to keep CDC running smoothly.

The following list identifies some common tasks:

- Review CICS/VSAM ECCR output.
- Display the VSAM files that are participating in change data capture.
- Stop the CICS/VSAM ECCR.
• Manually reinitialize the CICS/VSAM ECCR after stopping it.
• Stop change capture for all or selected VSAM source data sets.
• Refresh capture registrations for the ECCR after you add or change them.
• Manage VSAM schema changes.

Output from the CICS/VSAM ECCR

The EDMMSG SYSOUT data set contains messages that report some startup options, ECCR processing status, and the number of captured change records by change type at ECCR termination.

The following information is a sample of this report:

PWXEDM172852I 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
PWXEDM172830I CICGLB loaded at 0F2873A8
PWXEDM1728111 XCF is in local mode only
PWXEDM172818I Joined XCF group 'EDML' as member 'VSM3'
PWXEDM1728411 EDM ECCR VSM3 connected to DETAIL Logger EDML, Log RBA='00000001D5E'
PWXEDM172808I Change Capture active for VSAM file CCV.EDM.VC1
Edition=0000000000000000 EDMNAME=SOURCE.EDMNAME.VC1
PWXEDM1728411 EDM ECCR VSM3 disconnected from DETAIL logger EDML,
Log RBA='0000000000000000' EDMNAME=SOURCE.EDMNAME.VC1
PWXEDM172818I Left XCF group 'EDML' as member 'VSM3'
PWXEDM172829I EDM ECCR sent 11 records to logger EDML (5 change records)
PWXEDM172809I Change Capture counts for CCV.EDM.VC1: Insert=5, Update=0, Delete=0

Note: This report can also include message PWXEDM172886I, which identifies any load module replacements that were applied.

Using the EDMC Transaction and Keywords to Manage the CICS/VSAM ECCR

Use the CICS transaction for the CICS/VSAM ECCR with the appropriate keyword to manage the ECCR. By default, the transaction name is "EDMC."

EDMC Syntax

To enter the EDMC transaction from a CICS terminal or operator console, use the following syntax:

EDMC keyword
EDMC Keyword Descriptions

The following table describes the EDMC keywords that you can specify and their functions:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY or DISP</td>
<td>Displays the names of the VSAM data sets that are registered for change data capture and that have been opened since the CICS/VSAM ECCR initialized. You can issue the EDMC transaction with this keyword only from a CICS terminal. This information is then displayed at the terminal.</td>
</tr>
<tr>
<td>EXITPGMS or XPGM</td>
<td>Lists all of the exit programs that are defined at the CICS task-related exit point and global user exit points that PowerExchange uses for CICS/VSAM CDC.</td>
</tr>
<tr>
<td>HELP</td>
<td>Displays a help panel that describes the valid EDMC keywords for the CICS/VSAM ECCR. You can issue the EDMC transaction with the HELP keyword only from a CICS terminal. This information is then displayed at the terminal.</td>
</tr>
<tr>
<td>INITIALIZE or INIT</td>
<td>Initializes CICS/VSAM ECCR in the CICS region. Also dynamically adds the PowerExchange exit programs that run at the CICS task-related user exit point and global user exit points that PowerExchange uses for CICS/VSAM CDC. Usually, the ECCR is started by adding the EDMKOPER module name to the PLT initialization list. The ECCR then starts automatically in the third stage of CICS initialization. <strong>Warning:</strong> If you have exit programs that run at the same CICS global exit points as the CICS/VSAM ECCR exit programs, do not use the INIT keyword. Otherwise, the CICS/VSAM ECCR exit programs might get control in the improper order, causing change capture problems.</td>
</tr>
<tr>
<td>OPTIONS or OPTS</td>
<td>Displays the CICS/VSAM CDC override options that are currently specified in the EDMKOVRD DD statement in the CICS region startup JCL or in the data set to which this DD statement points.</td>
</tr>
<tr>
<td>REFRESH or REFR</td>
<td>Refreshes the display of the CICS/VSAM CDC override options that are currently specified in the EDMKOVRD DD statement in the CICS region startup JCL or in the data set to which this DD statement points. Also validates these options and identifies any syntax errors. Use this keyword after you change the override options to identify any syntax errors.</td>
</tr>
<tr>
<td>RESTART or REST</td>
<td>Re-initializes the CICS/VSAM ECCR in the CICS region by issuing the EDMC TERM command followed by the EDMC INIT command. Use this keyword after changing any of the CDC override options in the EDMKOVRID DD statement or data set for your changes to take effect.</td>
</tr>
<tr>
<td>TERMINATE or TERM</td>
<td>Immediately stops the CICS/VSAM ECCR that is running in the CICS region, thereby stopping change data capture for all of the open VSAM source data sets. Also dynamically removes the PowerExchange exit programs that run at the CICS task-related user exit point and global use exit points that PowerExchange uses for CICS/VSAM CDC. <strong>Tip:</strong> If you need to stop capturing changes only for a single VSAM file, deactivate or delete the corresponding capture registration. Then close and reopen the VSAM file in CICS.</td>
</tr>
</tbody>
</table>

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

**RELATED TOPICS:**
- “CICS/VSAM CDC Use of CICS Global and Task-Related Exit Points” on page 165

Displaying the VSAM Data Sets from Which Changes Are Captured

Use the EDMC transaction with the DISPLAY or DISP keyword to display the VSAM data sets from which changes are being captured.

From a CICS terminal, enter the transaction:

```
EDMC DISP
```
The following sample output indicates that no VSAM files are participating in change data capture:

```
EDMC DISP  PWXEDM CICS/VSAM Change Capture  Init Date: 09/15/14
ID: CT41  Participating Files Display  Time: 01:14:56
(During PLTI)
```

<table>
<thead>
<tr>
<th>File Name</th>
<th>Dataset Name</th>
<th>Type</th>
<th>Warn/Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMDFILE</td>
<td>PWX.V961.FILE1</td>
<td>KSDS</td>
<td></td>
</tr>
<tr>
<td>EMDFILE2</td>
<td>PWX.V961.FILE2</td>
<td>RRDS</td>
<td></td>
</tr>
<tr>
<td>EMDFILE3</td>
<td>PWX.V961.FILE3</td>
<td>Rcv (None)</td>
<td></td>
</tr>
<tr>
<td>EMDFILE4</td>
<td>PWX.V961.FILE4</td>
<td>ESDS</td>
<td></td>
</tr>
<tr>
<td>EMDFILE0</td>
<td>PWX.V961.FILE0</td>
<td>KSDS</td>
<td></td>
</tr>
<tr>
<td>EDMPATH1</td>
<td>PWX.V961.FILE1</td>
<td>Path</td>
<td></td>
</tr>
<tr>
<td>EDMPATH0</td>
<td>PWX.V961.FILE0</td>
<td>P/AX</td>
<td></td>
</tr>
<tr>
<td>EDMPATH3</td>
<td>PWX.V961.FILE3</td>
<td>Path</td>
<td>Rcv (None)</td>
</tr>
</tbody>
</table>

No files are currently participating in CICS/VSAM Change Capture

The following sample output indicates that multiple VSAM files that are participating in change data capture:

```
EDMC DISP  PWXEDM CICS/VSAM Change Capture  Init Date: 09/15/14
ID: CT41  Participating Files Display  Time: 01:14:56
(During PLTI)
```

<table>
<thead>
<tr>
<th>File Name</th>
<th>Dataset Name</th>
<th>Type</th>
<th>Warn/Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>EMDFILE1</td>
<td>PWX.V961.FILE1</td>
<td>KSDS</td>
<td></td>
</tr>
<tr>
<td>EMDFILE2</td>
<td>PWX.V961.FILE2</td>
<td>RRDS</td>
<td></td>
</tr>
<tr>
<td>EMDFILE3</td>
<td>PWX.V961.FILE3</td>
<td>Rcv (None)</td>
<td></td>
</tr>
<tr>
<td>EMDFILE4</td>
<td>PWX.V961.FILE4</td>
<td>ESDS</td>
<td></td>
</tr>
<tr>
<td>EMDFILE0</td>
<td>PWX.V961.FILE0</td>
<td>KSDS</td>
<td></td>
</tr>
<tr>
<td>EDMPATH1</td>
<td>PWX.V961.FILE1</td>
<td>Path</td>
<td></td>
</tr>
<tr>
<td>EDMPATH0</td>
<td>PWX.V961.FILE0</td>
<td>P/AX</td>
<td></td>
</tr>
<tr>
<td>EDMPATH3</td>
<td>PWX.V961.FILE3</td>
<td>Path</td>
<td>Rcv (None)</td>
</tr>
</tbody>
</table>

This PWXEDM CICS/VSAM Change Capture report includes the following fields:

- **Init Date.** The date, in mm/dd/yy format, on which the ECCR initialized.
- **ID.** The ECCR name.
- **Time.** The time, in hh:mm:ss format, at which the ECCR initialized.
- **File Name.** The names of the VSAM files that participate in change data capture.
- **Dataset Name.** The fully-qualified data set names of the VSAM source data sets that participate in change data capture.
- **Type.** The type of VSAM data set. Valid values are:
  - KSDS. A key-sequenced data set.
  - ESDS. An entry-sequenced data set.
  - RRDS. A relative record data set (RRDS) or variable-length relative record data set (VRRDS).
  - Path. An alias path to a VSAM data set.
  - P/AX. A CICS alternate index (AIX) path to a VSAM data set.
- **Warn/Error.** A warning or error flag. Valid values are:
  - (During PLTI). Indicates the ECCR automatically initialized during the third stage of CICS initialization because you specified the EDMKOPER module name in the PLT initialization list.
  - Rcv (None). Indicates that the VSAM file was defined as RECOVERY(NONE). This RECOVERY option is allowable if you specify CCERR=CONTINUE in the EDMSDIR options module and AllowRecoveryNone in the INITPARM for EDMKOPER, or if you specify CCERR=ABEND in the EDMSDIR options module.
  - Reg Err. Indicates that the change capture status of the VSAM data set cannot be determined. The capture registration for the VSAM data set might have an error.

### Changing CDC Override Options for VSAM Data Sets

You can change the CDC override options in the //EDMKOVRD DD statement in the CICS startup JCL or in the data set to which the DD statement points while the CICS/VSAM ECCR is active.

For example, you might want to enable or disable change data for a VSAM data set type or a specific data set.
For KSDS and RRDS data sets and CICS-maintained tables, change data capture is enabled by default. You can disable change data capture for any of these source types, even if data sets of the specified type are registered for CDC, by specifying OFF for the following options:

- CAPTURE_KSDS
- CAPTURE_RRDS
- CAPTURE_CMDT

For ESDS data sets, change data capture is disabled by default. To enable change data capture for ESDS data sets, you must specify CAPTURE_ESDS=ON. You can also customize ECCR handling of backouts for ESDS data sets by specifying the optional BACKOUTRC and ESDFAIL override options.

For a specific VSAM data set, you can use the DSN option to enable or disable change data capture or override the default backout handling.

For more information about all of these options, see "Configuring CICS for CDC" on page 169.

After you change CDC override options, issue the EDMC REFRESH command to validate the syntax. If syntax errors are reported, correct them.

Then issue the EDMC RESTART command to re-initialize the CICS/VSAM ECCR so that the ECCR can start using the updated CDC override options.

**Note:** After you issue the RESTART command, the CICS/VSAM ECCR will be momentarily inactive between ECCR termination and re-initialization. If you issue the command during a period of high file I/O activity, the ECCR might miss some change data, which can damage data integrity.

### Migrating an ESDS Data Set from VSAM Batch CDC to CICS/VSAM CDC

If you previously used PowerExchange batch VSAM CDC to capture change data from VSAM ESDS data sets, you can migrate these data sets to CICS/VSAM CDC to capture change data records in an online CICS Transaction Server environment.

Perform the following migration steps:

1. In the PowerExchange Navigator, create a capture registration for the source ESDS data set.
2. Verify that the RDO definitions for the CICS fields that are associated with the registered ESDS data set are defined with RECOVERABLE(BACKOUTONLY) or RECOVERABLE(ALL) option.
3. Define CICS/VSAM ECCR override options, as needed, in the //EDMKOVRD DD statement of the CICS startup procedure or in the data set to which this DD statement points. To capture data from ESDS data sets, you must at least define the CAPTURE_ESDS=ON option.
4. Restart the CICS/VSAM ECCR in one of the following ways:
   - Restart the CICS region.
   - Issue the EDMC RESTART command.
   - Issue the EDMC REFRESH command.

### Stopping the CICS/VSAM ECCR

Use the EDMC transaction with the TERM keyword to stop the CICS/VSAM ECCR.

To enter the transaction from a CICS terminal or operator console, use the following syntax:

```
EDMC TERM
```
The ECCR stops immediately and no longer captures changes for all VSAM source data sets in the CICS environment.

The EDMMSG SYSOUT data set displays messages that report the number of records sent to the PowerExchange Logger and the number of change operations by type that were captured since the last time the VSAM data set was opened.

**Stopping Change Capture for a Specific VSAM Data Set**

To stop change data capture for a specific VSAM source data set, deactivate or delete its capture registration and close the VSAM data set.

1. Stop updates to the VSAM source data set. If you do not stop update activity, you might lose in-flight changes when you close the VSAM data set.
2. Close the VSAM data set in the CICS region.
3. Delete the capture registration and associated extraction map, or set the **Status** option in the registration to **History**.
   
   **Note:** You cannot reactivate the registration for change capture later.

The CICS/VSAM ECCR no longer captures changes for the data set.

**Refreshing Capture Registrations for the ECCR**

If you add or change capture registrations for VSAM sources, you must refresh the registrations for the CICS/VSAM ECCR to use them.

1. Close and reopen the VSAM file.
   
   **Note:** If you defined the file to open when it is referenced, the registrations in the CICS/VSAM ECCR are refreshed the next time the VSAM file is referenced.
2. If you configured the capture registrations to use PowerExchange Condense, start or restart the PowerExchange Condense task.

**Managing VSAM Schema Changes**

If you need to change the record layout of a VSAM source data set, use this procedure to retain access to previously captured data while capturing data of the new format.

1. Stop updates to the VSAM source file.
2. Close the VSAM file in the CICS region.
3. If you use PowerExchange Condense, verify that PowerExchange Condense has extracted all data from the PowerExchange Logger log files.
4. Complete extraction processing of all captured changes based on the existing extraction map.
5. In the VSAM capture registration, set the **Status** option to **History**.
6. Change the VSAM file structure.
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 be made the VSAM data set again.
This chapter includes the following topics:

- **Datacom Table-Based CDC Overview, 178**
- **Architectural Overview, 179**
- **Configuring Datacom for CDC, 180**
- **Configuring the Datacom Table-Based ECCR, 181**
- **Managing Datacom Table-Based CDC, 193**

## Datacom Table-Based CDC Overview

PowerExchange table-based change data capture (CDC) for Datacom captures changes asynchronously from Datacom CDC tables.

PowerExchange works with the Datacom Change Data Capture feature. 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 change data to the PowerExchange Logger for MVS.

### 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.
Implementing Datacom Table-Based CDC

Complete the following tasks to implement Datacom table-based CDC:

2. "Configuring the Datacom Table-Based ECCR" on page 181.
3. "Starting the Datacom Table-Based ECCR" on page 193.

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 Multi-User Facility (MUF) in which the transactions occur.
- Target MUF that contains the Datacom CDC tables, if different from the source MUF.
- Datacom 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 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 the following types of MUFs:

- 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.

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 for MVS.
- Removes records from the CDC tables that have been committed to the PowerExchange Logger for MVS.

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.

**Datacom Table-Based ECCR Parameters**

Configure parameters for the Datacom table-based ECCR in the RUNLIB(ECCRDCMP) member to which the DTLCAFCFG DD statement in the ECCR JCL points.

Based on your input during installation, the z/OS Installation Assistant adds values for some parameters to the ECCRDCMP member. You can change these values if necessary.

The ECCRDCMP member can contain the following parameters:

```
MUF=muf_name
    [REG_MUF=registered_muf_name]
    [NO_DATA_WAIT=seconds]
    [NO_DATA_WAIT2=seconds]
    ECCRNAME=ecrr_name
    DB_TYPE=DCM
    [COLDSTART={Y|N}]
    [CLEANUP={Y|N}]
    [CLEANUP_INTERVAL=seconds]
    [CDC_BASE=dbid]
    [CDC_ID=A]
    [CAPT_STATS={Y|N}]
    [CAPT_STATS_INTERVAL={Y|N}]
    [CAPT_STATSTERSE={Y|N}]
    [CLEANUP_STATISTICS={Y|N}]
    [LOCAL_TIME={Y|N}]
    [MONITOR={Y|N}]
    [ON_SUSPENSION_ERROR_CONTINUE={N|Y}]
    [REFRESH_ALLOWED={Y|N}]
    [RESTART_ADVANCE_ACTIVE=number_of_records]
```
The following table summarizes the Datacom table-based ECCR parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUF</td>
<td>Required</td>
<td>The name of the Datacom MUF for which change data is captured. This parameter is customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>REG_MUF</td>
<td>Optional</td>
<td>The Datacom MUF name that is defined in the registration group for the Datacom source. Use this parameter if you want to use the capture registrations defined for a MUF other than the one specified in the MUF parameter. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>Optional</td>
<td>The number of seconds that the ECCR waits after reading the Datacom CDC tables and finding no new change records before starting the next read operation. If the ECCR completes the next read operation without having read new changes, the NO_DATA_WAIT2 parameter takes effect. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>NO_DATA_WAIT2</td>
<td>Optional</td>
<td>After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the Datacom table-based ECCR waits after reading the Datacom CDC tables and finding no new change records before trying the read again. The NO_DATA_WAIT2 wait and retry cycle remains in effect as long as no changes are received. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>ECCRNAME</td>
<td>Required</td>
<td>The ECCR name.</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Required</td>
<td>The database type, which must be DCM for Datacom.</td>
</tr>
<tr>
<td>COLDSTART</td>
<td>Optional</td>
<td>Controls whether the ECCR cold starts or warm starts.</td>
</tr>
<tr>
<td>CLEANUP</td>
<td>Optional</td>
<td>Controls whether the PowerExchange cleanup subtask periodically removes changes that were committed to the PowerExchange Logger from the Datacom CDC tables. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>CLEANUP_INTERVAL</td>
<td>Optional</td>
<td>The number of seconds that the cleanup subtask waits before removing committed changes from the Datacom CDC tables. This parameter can be customized by the MVS Installation Assistant.</td>
</tr>
<tr>
<td>CDC_BASE</td>
<td>Optional</td>
<td>The database identifier (DBID) for the source database. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>CDC_ID</td>
<td>Optional</td>
<td>The version identifier of the Datacom CDC tables.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CAPT_STATS</td>
<td>Optional</td>
<td>Controls whether PowerExchange writes ECCR statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the Datacom table-based ECCR reaches the end of the change stream in the Datacom CDC tables.</td>
</tr>
<tr>
<td>CAPT_STATS_INTVL</td>
<td>Optional</td>
<td>The interval, in minutes, for which the Datacom table-based ECCR collects and reports the number of inserts, deletes, updates, and commits that were captured from the change stream. The ECCR also reports the current point in the change stream.</td>
</tr>
<tr>
<td>CAPT_STATS_TERSE</td>
<td>Optional</td>
<td>Controls whether the Datacom table-based ECCR prints PWX-06153 statistics messages only for registered sources for which the ECCR captured changes.</td>
</tr>
<tr>
<td>CLEANUP_STATISTICS</td>
<td>Optional</td>
<td>Controls whether the PowerExchange cleanup subtask issues detailed messages with statistics that help you determine the progress of the cleanup process relative to the main CDC reader process.</td>
</tr>
<tr>
<td>LOCAL_TIME</td>
<td>Optional</td>
<td>Controls whether the time stamps that the ECCR assigns to change records use the local time instead of the Coordinated Universal Time (UTC) time that Datacom uses.</td>
</tr>
<tr>
<td>MONITOR</td>
<td>Optional</td>
<td>Controls whether the ECCR starts another process to monitor and detect a hang in the CA Datacom API in the main ECCR reader process or in the ECCR cleanup process. Also, if cleanup is active, the monitor process detects any hang that might occur in the ECCR cleanup wait routine.</td>
</tr>
<tr>
<td>MONITOR_INTERVAL</td>
<td>Optional</td>
<td>If you set MONITOR to Y, the number of seconds between each monitoring check.</td>
</tr>
<tr>
<td>ON_SUSPENSION_ERROR_CONTINUE</td>
<td>Optional</td>
<td>If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether the ECCR ends or continues when a UOW that contains change records to be discarded or captured started at an invalid point in the change stream relative to the suspension window.</td>
</tr>
<tr>
<td>REFRESH_ALLOWED</td>
<td>Optional</td>
<td>Controls whether you can use the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command refreshes the list of registered Datacom records that the ECCR uses for change capture processing.</td>
</tr>
<tr>
<td>RESTART_ADVANCE_ACTIVE</td>
<td>Optional</td>
<td>The number of change records that an active Datacom ECCR processes after a special restart UOW before writing another updated special UOW to the PowerExchange Logger.</td>
</tr>
</tbody>
</table>

**Note:** If a parameter has a default value, it is marked as optional. A default value is the value that PowerExchange uses if the parameter is not defined. For some parameters, the z/OS Installation Assistant provides recommended values, which you can accept or change.

More detailed parameter descriptions follow.
**CAPT_STATS Parameter**

Controls whether PowerExchange writes ECCR statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the Datacom table-based ECCR reaches the end of the change stream in the Datacom CDC tables.

The ECCR issues PWX-06153 messages that report the number of inserts, deletes, and updates that were captured for each registration, grouped change stream read. The WTO messages indicate that the end of the change stream was reached and provide the capture counts.

**Related Parameters:** CAPT_STATS_INTVL, CAPT_STATS_TERSE

**Syntax:**
```
CAPT_STATS={N|Y}
```

**Valid Values:**
- **N.** Do not write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing the change stream.
- **Y.** Write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing the change stream.

Default is **N.**

**Usage Notes:**
- If you do not set the global CAPT_STATS parameter to Y, you can issue to STATISTICS ON command after the ECCR is started to enable statistics reporting for each ECCR change stream read of the Datacom CDC tables.
- If you also specify the CAPT_STATS_INTVL parameter or run the STATISTICS minutes, the ECCR also reports the total number of inserts, deletes, updates, and commits for the each interval.

For more information about the STATISTICS command and its parameters, see the *PowerExchange Command Reference.*

**CAPT_STATS_INTVL Parameter**

The interval, in minutes, for which the Datacom table-based ECCR collects and reports change capture statistics.

If you specify an interval, the ECCR prints a PWX-06181 message each time the interval elapses. The message reports the total number of inserts, deletes, updates, and commits that the ECCR processed during the interval.

You can use this ECCR parameter to print statistics messages at a specific frequency, for example, every 60 minutes.

For the ECCR to print capture statistics, you must set the CAPT_STATS parameter to Y in the RUNLIB(ECCRDCMP) member or run the ECCR STATISTICS ON command.

**Related Parameters:** CAPT_STATS, CAPT_STATS_TERSE

**Syntax:**
```
CAPT_STATS_INTVL=minutes
```

**Value:** For the minutes variable, enter a number from 1 through 1440. No default is provided.

**Usage Notes:**
- If you set the CAPT_STATS_INTVL parameter to 0, PowerExchange issues the error message PWX-00967.
• After you start the ECCR, message PWX-07805 identifies the collection interval that is defined.
• If you issue the `STATISTICS minutes` command, the number of minutes that is specified in the command overrides the `CAPT_STATS_INTVL` value for the duration of the ECCR run.

**CAPT_STATS_TERSE Parameter**

Controls whether the Datacom table-based ECCR prints PWX-06153 messages only for registered sources for which the ECCR captured changes. If no inserts, updates, or deletes occurred on a registered source, the ECCR does not report capture counts for it.

A PWX-06153 message reports the number of inserts, deletes, and updates that were captured for a registered source. The message is printed when the ECCR reaches the end of the change stream in the Datacom CDC tables and at the end of the ECCR run.

For the ECCR to print statistics, you must set the `CAPT_STATS=Y` parameter in the `RUNLIB(ECCRDCMP)` member or run the ECCR `STATISTICS ON` command.

**Related Parameters:** `CAPT_STATS, CAPT_STATS_INTVL`

**Syntax:**

```
CAPT_STATS_TERSE={N|Y}
```

**Valid Values:**

- **N.** Print statistics for all registered sources, including sources without any change activity.
- **Y.** Print statistics only for the registered sources for which the ECCR captured changes.

Default is N.

**Usage Notes:**

- If you set the `CAPT_STATS_TERSE` parameter to N and then issue the `STATISTICS SINCE TERSE` command, the TERSE option in the command overrides the `CAPT_STATS_TERSE` setting for the SINCE period. PWX-06153 messages are then printed only for registered sources for which changes were captured.

**CDC_BASE Parameter**

The database identifier for the Datacom database that contains the change data to capture.

**Syntax:**

```
CDC_BASE={2009|dbid}
```

**Value:** For the `dbid` variable, enter a Datacom database identifier. This value must match the value that you specify in the CDCL_DBID startup option.

Default is 2009. This is the DBID that Datacom uses by convention. If you use a DBID other than 2009 at your site, use the Datacom MUF CDCL_DBID startup option to assign the DBID to the CDC database.

**CDC_ID Parameter**

The version identifier for the Datacom CDC tables.

**Syntax:**

```
CDC_ID={A|version_id}
```

**Value:** For the `version_id` variable, enter the version identifier of the Datacom CDC tables. This value must match the value that you specify in the Datacom MUF CDCL startup option.
Default is A.

**Usage Notes:** If the format of the Datacom CDC tables changes in a later Datacom release, you must assign the new version identifier.

### CLEANUP Parameter

Controls whether the PowerExchange cleanup subtask starts at a specified interval to remove changes from the Datacom CDC MNT and TSN tables that were committed to the PowerExchange Logger logs.

**Related Parameters:** CLEANUP_INTERVAL

**Syntax:**

\[
\text{CLEANUP} = \{N|Y}\n\]

**Valid Values:**

- **Y.** Starts the cleanup subtask after the interval that is specified in the CLEANUP_INTERVAL parameter.
- **N.** The cleanup subtask does not start.

Default is **Y**.

**Usage Notes:** Use this parameter to prevent the Datacom CDC tables from becoming full.

### CLEANUP_INTERVAL Parameter

The number of seconds that the cleanup subtask waits before removing change data from the Datacom CDC tables that has been committed to the PowerExchange Logger logs.

You must also define `CLEANUP=Y` for the cleanup subtask to connect to the Datacom MUF and remove change data from the Datacom CDC tables that has been committed to the PowerExchange Logger logs. The cleanup subtask then waits for the CLEANUP_INTERVAL interval again before doing another cleanup run.

**Related Parameters:** CLEANUP

**Syntax:**

\[
\text{CLEANUP}\_\text{INTERVAL} = \{300|\text{seconds}\}
\]

**Value:** For the seconds variable, enter a number of seconds greater than 0.

Default is 300.

### CLEANUP_STATISTICS Parameter

Controls whether the PowerExchange cleanup subtask issues detailed messages with statistics that help you determine the progress of the cleanup process relative to the main CDC reader process.

**Related Parameters:** MONITOR, MONITOR_INTERVAL

**Syntax:**

\[
\text{CLEANUP}\_\text{STATISTICS} = \{Y|N\}
\]

**Valid Values:**

- **Y.** Issue detailed messages about the progress of the cleanup task.
- **N.** Do not issue detailed messages about the progress of the cleanup task.

Default is **N**.
Usage Notes: This parameter helps you determine the progress of the cleanup process relative to the main CDC reader process, that is, how far behind the cleanup process it. Use this parameter if you have concerns with the performance of the ECCR cleanup task or if the ECCR appears to hang.

COLDSTART Parameter
Controls whether the Datacom table-based ECCR cold starts or warm starts.

Syntax:
```
COLDSTART={N | Y}
```

Valid Values:
- **N.** The ECCR warm starts. Change capture starts from where it last left off without loss of data.
- **Y.** The ECCR cold starts. Change capture starts from the oldest record in the Datacom CDC tables.

Default is N.

DB_TYPE Parameter
Required. The database type.

Syntax:
```
DB_TYPE=DCM
```

Value: The value must be "DCM" for the Datacom table-based ECCR.

ECCRNAME Parameter
Required. A name for the Datacom table-based ECCR.

Syntax:
```
ECCRNAME=ecr_name
```

Value: For the ecr_name variable, enter a 1- to 8-character alphanumeric string.

No default. However, the z/OS Installation Assistant generates an ECCR name that begins with the PowerExchange Agent / Logger Prefix value followed by DCMEC, for example, PWXDCMEC.

Usage Notes:
- The ECCR uses this parameter value for the following purposes:
  - To connect to the PowerExchange Logger to write change data
  - As 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 that is written to PowerExchange Logger log files
- If you change the ECCRNAME value, the ECCR cannot warm start from where it last left off.
- The ECCR name must be unique within a PowerExchange Logger group.
- Informatica recommends that you use the same value for the ECCRNAME parameter and the Datacom ECCR started task or job name. This practice lets you to easily identify the Datacom ECCR when reviewing messages and data from the PowerExchange Logger.
LOCAL_TIME Parameter

Optional. Controls whether the time stamps that the Datacom table-based ECCR uses to indicate when database changes occurred use the local time or Coordinated Universal Time (UTC) time.

ECR time stamps indicate when changes were made in the database. They do not indicate when the ECCR captured the changes.

Syntax:

\[
\text{LOCAL\_TIME} = \{N|Y\}
\]

Valid Values:

- **N**. The ECCR time stamps use UTC time values based on the Datacom UTC time stamps in change records.
- **Y**. The ECCR time stamps use local time values based on the Datacom SQL time stamps in the change records.

Default is N.

MONITOR Parameter

The MONITOR parameter controls whether the ECCR starts another process to monitor and detect a hang in CA Datacom or in the ECCR cleanup wait routine.

Related Parameters: CLEANUP_STATISTICS, MONITOR_STATISTICS

Syntax:

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

Valid Values:

- **Y**. Starts the monitoring process.
- **N**. The monitoring process is not started.

Default is N.

Usage Notes: The monitoring process monitors and detects the following critical events:

- A hang in CA Datacom when the Datacom API is invoked from the cleanup process
- A hang in CA Datacom when the Datacom API is invoked from the main CDC reader process
- A hang in the ECCR cleanup wait routine

MONITOR_INTERVAL Parameter

If monitoring is enabled, MONITOR_INTERVAL specifies the number of seconds between each monitor check.

If you define MONITOR=Y, the MONITOR_INTERVAL parameter specifies the number of seconds between each monitor check.

Related Parameters: CLEANUP_STATISTICS, MONITOR

Syntax:

\[
\text{MONITOR\_INTERVAL} = \{600|seconds\}
\]

Value: For the seconds variable, enter a number of seconds greater than 0.

Default is 600, which is twice the default value of CLEANUP_INTERVAL. If you specify a value for MONITOR_INTERVAL that is less than twice the value of CLEANUP_INTERVAL, the ECCR assigns
MONITOR_INTERVAL the value of twice the CLEANUP_INTERVAL value. This precaution prevents the monitoring process from falsely detecting a hang situation.

**MUF Parameter**

Required. The name of the Datacom MUF for which change data is captured.

**Related Parameters:** REG_MUF

**Syntax:**

\[
\text{MUF=\text{muf}\_\text{name}}
\]

**Value:** For the \text{muf}\_\text{name} variable, enter the name of the Datacom MUF from which the ECCR captures change data.

This name must match the internal MUF name that is recorded as part of the key data in the Datacom CDC TSN table. This value also must match the MUF name in the registration group that you defined in the PowerExchange Navigator, unless the REG_MUF parameter specifies a different MUF value.

No default.

**NO_DATA_WAIT Parameter**

The number of seconds that the Datacom table-based ECCR waits after it reads the Datacom CDC tables and finds no new changes before starting the next read operation.

During the next read operation, if the ECCR still finds no changes, the NO_DATA_WAIT2 interval takes effect.

**Related Parameters:** NO_DATA_WAIT2

**Syntax:**

\[
\text{NO\_DATA\_WAIT=\{60|seconds\}}
\]

**Value:** For the \text{seconds} variable, enter a number of seconds greater than 0.

Default is 60.

**Usage Notes:** During the wait interval, the ECCR waits simultaneously for console input.

**NO_DATA_WAIT2 Parameter**

After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the Datacom table-based ECCR waits after reading the Datacom CDC tables and finding no new change records before doing another read operation.

During a subsequent read operation, if the ECCR finds changes, the NO_DATA_WAIT interval takes effect again. If the ECCR does not find changes, it waits for the NO_DATA_WAIT2 interval and then tries the read again. The ECCR continues to wait for the NO_DATA_WAIT2 interval and retry the read on an ongoing basis, as long as no changes are available.

**Related Parameters:** NO_DATA_WAIT

**Syntax:**

\[
\text{NO\_DATA\_WAIT2=\{600|seconds\}}
\]

The z/OS Installation Assistant enters 999 for this parameter in the ECCR configuration member unless you specify another value. If this parameter is not defined, the default of 600 is used.

**Value:** For the \text{seconds} variable, enter a number of seconds greater than 0.
Default is 600.

**Usage Notes:** During the wait interval, the ECCR waits simultaneously for input from the console.

**ON_SUSPENSION_ERROR_CONTINUE Parameter**

Optional. If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether the Datacom table-based ECCR ends or continues when a UOW that contains change records to be discarded or captured started at an invalid point in the change stream relative to the suspension window.

**Syntax:**

```
ON_SUSPENSION_ERROR_CONTINUE=\{N|Y\}
```

**Valid Values:**

- **N.** The ECCR issues an error message and ends.
- **Y.** The ECCR issues a warning and continues processing.

Default is N.

**Usage Notes:** If you use the PWXUCREG utility, this parameter controls whether the ECCR ends or continues in the following situations:

- When discarding change records for a suspended registrations, the ECCR determines that the associated UOW started before the beginning of the suspension window.
- When capturing change records for an activated registration, the ECCR determines that the associated UOW started before the end of the suspension window.

The suspension window is the time period between the suspension timestamp and reactivation timestamp. For more information about the PWXUCREG utility, see the *PowerExchange Utilities Guide*.

**REG_MUF Parameter**

The Datacom MUF name that you specified in the registration group for the Datacom source.

This parameter value can be the same as or different from the MUF parameter value. The ECCR uses the REG_MUF parameter to read capture registrations, and uses the MUF parameter to read change data from the Datacom CDC tables.

**Related Parameters:** MUF

**Syntax:**

```
REG_MUF=registered_muf_name
```

**Value:** For the `registered_muf_name` variable, enter the name of the Datacom MUF that you entered in the registration group from the PowerExchange Navigator.

Default is the MUF parameter value.

**Usage Notes:** Define the REG_MUF parameter if you want to use capture registrations that were created for one MUF to capture changes from a different MUF. For example, if you have test and production MUFs that have capture active for the same tables, you can use the same registrations for those tables.
JCL for the Datacom Log-Based ECCR

REFRESH_ALLOWED Parameter

Controls whether PowerExchange users can issue the ECCR REFRESH command. This command refreshes the list of Datacom records with active capture registrations that the Datacom table-based ECCR uses to capture change data.

When this parameter is set to Y, users can issue the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command updates the list of registered sources that the ECCR uses, without shutting down and restarting the ECCR.

Syntax:

    REFRESH_ALLOWED={N|Y}

Valid Values:

- N. Do not allow users to issue the REFRESH command. This option is intended for users of PowerExchange versions earlier than 9.5.0, when the REFRESH command was not available. This option maintains the previous behavior, which requires a restart of the ECCR after registration changes.
- Y. Allow users to issue the REFRESH command.

Default is N.

RESTART_ADVANCE_ACTIVE Parameter

The number of change records that an active Datacom table-based ECCR processes after a special restart UOW, before it writes another updated special UOW to the PowerExchange Logger.

This value can affect how far back the PowerExchange Logger searches for the restart point when the ECCR is restarted.

Syntax:

    RESTART_ADVANCE_ACTIVE=number

Valid Values: Enter a number from 1 to 10000. Default is 10000.

Usage Notes: When the ECCR is inactive and waiting for work, PowerExchange updates the special UOW before each NO_DATA_WAIT2 cycle.

JCL for the Datacom Table-Based ECCR

To configure the JCL for the Datacom log-based ECCR, edit the ECCRDCM member in the PowerExchange RUNLIB data set.

The JCL in the ECCRDCM member contains the following statements:

```
//******************************************************************************
//*/
//*/  RUN DETAIL DATA COM TABLE BASED ECCR
//*/
//******************************************************************************
//EC R A D1  EXEC  PGM=DTLCCDCR,REGION=50M
//STEPLIB  DD  DISP=SHR,DSN=PHILQ.LOADLIB
//  DD  DISP=SHR,DSN=PHILQ.LOAD
//  DD  DSN=DCOMCAI,
//  DISP=(SHR)
//  DD  DSN=DCOMSPL,
//  DISP=(SHR)
//  DD  DSN=DCOMLOAD,
//  DISP=(SHR)
//  DD  DSN=DCOMCUST,
//  DISP=(SHR)
//  DD  DSN=DCOMICPC,
```
// DISP=(SHR)
//CXZ DD DSN=DCOMCXX,
// DISP=(SHR)
//EDMPARMS DD DISP=SHR,DSN=HLQ.EDM..&LOGER&SUFX..USERLIB
//DTLCFG DD DISP=SHR,DSN=RUNLIB(EDMSDIR)
//DTLCACFG DD DISP=SHR,DSN=RUNLIB(LICENSE)
//DTLAMCPR DD DISP=SHR,DSN=HLQVS..CCT
//DTLMSG DD DISP=SHR,DSN=HLQ..DTLMSG
//* IF USING MESSAGE OVERRIDE THEN CUSTOMIZE BELOW
//*DTLMSGO DD DISP=SHR,DSN=RUNLIB(DTLMSGO)
//*
//DTLLOG DD SYSOUT=* 
//DTLLOGO1 DD SYSOUT=* 
//SYSUDUMP DD SYSOUT=* 
//SYSOUT DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//EDMSSG DD SYSOUT=* 
//CEEDUMP DD SYSOUT=* 

Statement Descriptions:

**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.

**DTLKEY**

Specifies the PowerExchange license key file, which contains the license key for the PowerExchange options that you use.

**DTLCACFG**

Points to the Datacom ECCR configuration member ECCRCMDP.

**DTLAMCPR**

Points to the data set that contains the capture registrations.

**DTLMSG**

Specifies the output data set for PowerExchange messages.

**DTLLOG**

Specifies the PowerExchange log file for messages. This SYSOUT file contains various messages that report the status and events for the Datacom table-based ECCR.
Managing Datacom Table-Based CDC

PowerExchange provides commands to start and stop the Datacom table-based ECCR.

Starting the Datacom Table-Based ECCR

You can run the Datacom table-based ECCR as a started task or batch job.

Before starting the ECCR, complete the following prerequisites:

2. Configure the Datacom table-based ECCR.
3. To run the ECCR as a started task, convert the ECCRDCM JCL to a PROC and copy the PROC to the system PROCLIB library for started tasks.
4. Configure the Datacom source MUF startup options.
5. Start the Datacom source and target MUFs.

To start the ECCR, use one of the following methods:

- To start the ECCR as a started task, use the MVS START (S) command, for example:
  
  START DTLCCDCR

- To start the ECCR as a batch job, submit the ECCRDCM JCL that you configured.

Tip: If you need to run the ECCR continuously over a long period, run it as a started task.

Stopping the Datacom Table-Based ECCR

To stop the Datacom table-based ECCR, use the MVS STOP (P) command.

Enter the command followed by the name of the started task or batch job, for example:

STOP DTLCCDCR

Adding a Datacom Capture Registration

You might need to add a capture registration for a new or existing Datacom record from which you want to start capturing change data. In this case, you can use the REFRESH command to refresh the list of registered Datacom records for the Datacom table-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(ECCRDCMP) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. If you need to begin capturing changes for the new registration from a specific point, stop any change activity on the source record.
2. In the PowerExchange Navigator, open the capture registration and set the Status field to Active.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:

   F eccr_task_name,REFRESH

   The newly registered source is added to the list of registered sources for the ECCR.
5. Enable change activity on the source to resume.
6. If you use PowerExchange Condense, restart it.
Deleting a Datacom Capture Registration

You might need to delete a capture registration that has been used for change capture processing. In this case, you can use the REFRESH command to refresh the list of registered Datacom records for the Datacom table-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(ECRDCMP) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. Stop applications and other activities that update the source record that is associated with the registration that you are deleting.
2. Ensure that the ECCR has captured all of the change data from the Datacom CDC tables for the source that is associated with the registration that you are deleting. Also ensure that the source data has been extracted and applied to the target. Then stop all workflows that extract change data for the source.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. In the PowerExchange Navigator, open the capture registration and set the Status field to History. Then delete the registration.
5. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:

   F eccr_task_name,REFRESH

6. Enable change activity on the source to resume.
7. If you use PowerExchange Condense, restart it.
8. Restart extraction processing.

Suspending Change Capture for Registered Datacom Sources Temporarily

Use this task flow to suspend change capture processing for registered Datacom table-based CDC sources temporarily.

You perform some tasks with the PWXUCREG utility and other tasks outside of the utility on the z/OS system.

Before you begin, ensure that the REFRESH_ALLOWED=Y parameter is specified in the RUNLIB(ECRDCMP) member to which the DTLCACFG DD statement in the ECCR JCL points. You must have the authority to issue a REFRESH command after each registration status change.

1. Stop database activity for the registered source or sources for which you want to suspend capture registrations.
2. To suspend the capture registrations, use the PWXUCREG utility to issue the SUSPEND_REGISTRATION command.

   The suspension window opens. The utility sets the suspension timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLOG log to report the registration status change.

   For each suspended registration, the PowerExchange Navigator Resource Inspector displays Suspended in the Status field and the suspension timestamp in the Suspend Time field. The Suspend Time value is not adjusted for the local time.
3. Enter the ECCR REFRESH command with the MVS MODIFY (F) command:

   F eccr_task_name,REFRESH

   The ECCR becomes aware of the registration status change and suspension timestamp. When the ECCR encounters the first change record to discard, it issues message PWX-07752. The ECCR discards change records that have a timestamp later than the suspension timestamp.
4. Run the jobs or processes that generate the changes that you do not want to capture for the source or sources that are associated with the suspended registrations.

5. After the jobs or processes complete, use the PWXUCREG utility to issue the ACTIVATE_REGISTRATION command to reactivate the capture registrations.
   The suspension window closes. The utility sets the activation timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLLLOG log to report the registration status change.
   For each reactivated registration, the PowerExchange Navigator Resource Inspector displays **Active** in the **Status** field and the activation timestamp in the **Active Time** field. The **Active Time** value is not adjusted for the local time.

6. Enter the ECCR REFRESH command with the MVS MODIFY (F) command again.
   The ECCR becomes aware of the registration status change and activation timestamp.

7. Enable database activity to resume on the registered source or sources.
   The ECCR starts capturing change records that have timestamps later than the activation timestamp.
   The ECCR issues message PWX-07753 when it encounters the first change record in the change stream after the end of the suspension window.

   **Note:** You can automate this processing if appropriate for your environment.

**Changing a Datacom Source Table Definition**

If you change the structure of a Datacom table that is registered for change data capture, use this procedure to retain access to previously captured change data and capture data under the new structure definition.


2. Ensure that PowerExchange processes all changes that occurred under the old schema.

3. Change the Datacom table schema, and restart Datacom.

4. Create a new PowerExchange capture registration that reflects the schema changes.

5. Restart the Datacom table-based ECCR.

6. Allow update activity to the Datacom database to resume.
CHAPTER 10

DB2 for z/OS Change Data Capture

This chapter includes the following topics:

- **DB2 for z/OS CDC Overview**, 196
- **DB2 CDC Considerations**, 197
- **Configuring DB2 for CDC**, 206
- **Configuring the DB2 ECCR**, 207
- **Defining Restart Tokens for a DB2 Target Table Materialized from an Image Copy**, 218
- **Starting the DB2 ECCR**, 218
- **Managing DB2 CDC**, 219
- **Managing DB2 Schema Changes**, 231

DB2 for z/OS CDC Overview

PowerExchange change data capture (CDC) for DB2 for z/OS captures changes that are written to DB2 for z/OS tables.

The DB2 ECCR captures the change data and sends it to the PowerExchange Logger for MVS for logging. On a single DB2 subsystem or z/OS image, you can run multiple DB2 ECCRs, each connecting to a different DB2 subsystem. A single DB2 ECCR can connect to only one DB2 subsystem and communicate with only a single PowerExchange Logger instance.

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.
**DB2 CDC Considerations**

Review these considerations before implementing DB2 for z/OS CDC.

**DB2 for z/OS Datatypes Supported for CDC**

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

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

<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>No¹</td>
</tr>
<tr>
<td>CHAR</td>
<td>Yes</td>
</tr>
<tr>
<td>CLOB</td>
<td>No¹</td>
</tr>
<tr>
<td>DATE</td>
<td>Yes</td>
</tr>
<tr>
<td>DBCLOB</td>
<td>No</td>
</tr>
<tr>
<td>DECIMAL</td>
<td>Yes</td>
</tr>
<tr>
<td>DISTINCT (user-defined)</td>
<td>No¹</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 VARGPHIC</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>No¹</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>Yes</td>
</tr>
<tr>
<td>TIME</td>
<td>Yes</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>Yes, including extended-precision TIMESTAMP columns in DB2 10 for z/OS and later, which support fractional seconds of up to 12 digits.</td>
</tr>
<tr>
<td>DB2 Datatype</td>
<td>Supported for CDC?</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>TIMESTAMP WITH TIME ZONE</td>
<td>No¹</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>No¹</td>
</tr>
</tbody>
</table>

¹ In the PowerExchange Navigator, you cannot select columns that have unsupported datatypes for registration. The PowerExchange Navigator ignores these columns.

DB2 CDC Operational Considerations

Review the following operational considerations for DB2 for z/OS CDC:

- The DB2 ECCR captures changes only for the columns that you select in the capture registration. If you want to capture changes for all columns, you can select Select All Columns or Select all and notify changes when you create the registration.
- The DB2 ECCR captures only the changes that are recorded in the DB2 log as SQL inserts, deletes, updates, or truncates for a registered source table.
- The DB2 ECCR does not support change data capture for DB2 views and aliases.
- The DB2 ECCR does not support change data capture for the TRUNCATE SQL statement with the IMMEDIATE option.
- The DB2 ECCR does not capture changes that result from a DROP TABLE DDL statement or from a DB2 REORG utility operation that uses the DISCARD option.
- The DB2 ECCR captures changes from the DB2 LOAD utility only if you specify the RESUME YES and SHRLEVEL CHANGE options for the utility. The DB2 ECCR does not capture changes from other DB2 utilities, even if you specify the LOG=YES option.
- The DB2 ECCR does not capture changes from a single UOW that contains both DML and DDL changes for the same table, such as CREATE or ALTER TABLE statements and SQL inserts, deletes, and updates.
- For DB2 10 new-function mode and later, PowerExchange CDC supports hash tables, history tables, extended-precision TIMESTAMP columns, and ALTER TABLESPACE statements that change DSSIZE and PGSIZE values for a table space.
- The DB2 ECCR ends abnormally if you run the DB2 COPY utility with the SHRLEVEL REFERENCE option to create a full image copy of the table space that contains the ECCR TCAPWORK table. This table temporarily stores changes that were made to the DB2 system catalog tables. If you need to create a full image copy of the table space with this table, run the COPY utility with the SHRLEVEL CHANGE option.
- If you use DB2 Version 9.1 new-function mode, ensure that you installed 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 that contain 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.

### Handling Compressed DB2 Table Spaces

If you capture changes for tables in table spaces that are defined with the COMPRESS YES option, review the following considerations:

- The compression dictionary must be available when the DB2 ECCR requests the DB2 log data. Do not stop the DB2 databases or table spaces that contain the source tables, unless you are certain that the DB2 ECCR has processed all of the DB2 log data for the tables.

- If a source table is in a compressed table space, the compression dictionary must be compatible with the compressed DB2 log records from which the DB2 ECCR reads change data for the table. Otherwise, DB2 cannot decompress the log records for the ECCR, and if the ECCR tries to read compressed log records, it ends abnormally.

- DB2 maintains the current compression dictionary on disk and a backup of the previous compression dictionary in memory for as long as DB2 runs. If you restart DB2, the compression dictionary in memory is lost. If DB2 needs this dictionary to decompress log records for a DB2 ECCR request, the ECCR ends abnormally when it tries to read the log records. If you want the ECCR to skip these log records and continue capture processing, you can set the ROWNOTDECOMPRESSED parameter to NOFAIL in the DB2 ECCR REPL2OPT DD data set. For more information about the ROWNOTDECOMPRESSED parameter, see "DB2 ECCR Configuration Statements in the REPL2OPT DD Data Set" on page 210.

- If you run one of the following DB2 utilities to process data in a compressed table space, use the KEEPDICTIONARY option to retain the current compression dictionary:
  - DB2 REORG TABLESPACE utility
  - DB2 LOAD utility with the REPLACE or RESUME NO options

If you run one of these utilities without the KEEPDICTIONARY option, the utility rebuilds or recovers the compression dictionary. DB2 cannot use the rebuilt compression dictionary to decompress DB2 log records that were written prior to the REORG or LOAD operation. In this case, the ECCR ends abnormally when it tries to read the compressed log records.

When you restart the DB2 ECCR, use either the START WARM or START STARTLOC statement to start from a specific point in the DB2 logs for which the compression dictionary is available. If the DB2 log records require an earlier compression dictionary, the DB2 ECCR ends abnormally.

**Note:** If you run the DB2 REORG or LOAD utility with the KEEPDICTIONARY option to convert a table space from Basic Row Format to Reordered Row Format, the KEEPDICTIONARY option might be ignored, causing the DB2 ECCR to end in a similar manner. The DB2 subsystem parameter HONOR_KEEPDICTIONARY controls whether KEEPDICTIONARY is ignored. If you use the default parameter value of NO, the utility...
ignores the KEEPDICTIONARY option. For more information about the HONOR_KEEPDICTIONARY parameter, see IBM APAR PK41156.

- In a DB2 data sharing environment, the DB2 subsystem to which the DB2 ECCR connects needs access to the compression dictionaries. Ensure that the DB2 subsystem can access the table spaces and buffer pools for the compressed tables that are registered for CDC. If the DB2 subsystem cannot access these table spaces and buffer pools, DB2 passes compressed change records to the DB2 ECCR. When the DB2 ECCR receives the compressed change records, it issues the following message to the EDMMMSG log and then abends with user abend code 3680 and reason code 02710009:

```
PWXEDM17746Z  Table 'creator.table_name' ROW NOT DECOMPRESSED AT rba_or_lrsn.data_sharing_member_id.sequence_number RC=return_code, RS=reason_code, DG=diagnostic_code
```

If this error occurs, remove the compressed tables from change capture processing by deleting or deactivating their registrations. Then, warm start the DB2 ECCR.

**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, complete the following tasks:
  - 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.

**How the DB2 ECCR Interacts with Other PowerExchange Components**

The DB2 ECCR works with other PowerExchange components, such as the PowerExchange Logger for MVS and the PowerExchange Agent.

Consider the following interactions:

- A DB2 ECCR must log all changes to a single PowerExchange Logger that runs on the same z/OS system.
- The PowerExchange Logger and PowerExchange Agent must run on the same z/OS system as the DB2 ECCR.
- A single DB2 ECCR that attaches to a single member of a DB2 data sharing group can process changes for all members of 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 can prevent further capture and recording of change data until the Logger issues are resolved. After you resolve the Logger issues, the DB2 ECCR can continues the change capture processing without loss of data.

**Tip:** Make sure that you carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.
Related Topics:

• "Monitoring the PowerExchange Logger for MVS" on page 72

DB2 ECCR Capture Directory Tables

The DB2 ECCR uses a set of DB2 tables, called the capture directory tables, to track information about DB2 tables registered for CDC.

The capture directory tables are created during PowerExchange installation. They must reside in their own database and table space on the DB2 subsystem to which the DB2 ECCR connects for change capture.

The following table describes the purpose of each DB2 ECCR capture directory table:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCAPCOLUMNS</td>
<td>Stores catalog and status information for all of the columns in the tables that are 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 that are registered for change data capture.</td>
</tr>
<tr>
<td>TCAPTABLES</td>
<td>Stores catalog and status information for the tables that are registered for change data capture.</td>
</tr>
<tr>
<td>TCAPTABLESPACE</td>
<td>Stores catalog and status information for all table spaces in the DB2 catalog, including table spaces that do not contain registered tables.</td>
</tr>
<tr>
<td>TCAPUPDATE</td>
<td>Stores information that the DB2 ECCR uses to coordinate the 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 these changes is committed. <strong>Note:</strong> If you need to create a full image copy of the table space that contains the TCAPWORK table, run the DB2 COPY utility with the SHRLEVEL CHANGE option. If you use the SHRLEVEL REFERENCE option instead, the DB2 ECCR ends abnormally.</td>
</tr>
</tbody>
</table>

In the z/OS 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 table 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.
DB2SGENB

Creates the capture directory tables for a DB2 Version 11 and later database. The XIDDB220 job uses this member if you selected the **DB2 V11+** option at installation.

**Note:** If you have a DB2 Version 9.1 or 10 subsystem and selected the **DB2 V11+** option at installation, as recommended, the XIDDB220 job still uses the DB2SGENB member. The resulting capture directory tables support DB2 9.1 and 10 and will work with DB2 11 if you migrate to DB2 11 later.

DB2SGEN8

Creates the capture directory tables for a DB2 Version 9.1 or 10 database. The XIDDB220 job uses this member if you did not select the **DB2 V11+** option at installation.

DB2IGEN

Creates the unique index for each of the tables.

**Buffer Pool Requirements for Capture Directory Tables**

The DB2 ECCR requires a minimum buffer pool size of 4 KB for all capture directory tables except TCAPWORK. For the TCAPWORK table, the ECCR requires a minimum buffer pool size of 16 KB.

You can assign buffer pool sizes larger than these minimums if needed.

**Capture Directory Table Sizing**

The DB2 ECCR capture directory tables are each created in their own unique DB2 table space. The tables have certain sizing requirements that must be met for the ECCR to function properly.

The default space allocations from PowerExchange installation are usually sufficient for most DB2 subsystems, although some of the table spaces might create secondary extents. If you have more than 5,000 tables in the DB2 subsystem, or if you have a large number of tables registered for change capture or a large number of columns in these tables, you might need to adjust the PRIQTY primary space and SECQTY secondary space values from installation. Monitor the table spaces to determine if they need to be extended. If the DB2 ECCR cannot extend the table space when needed to support the table sizing requirements, the ECCR abends.

The following table shows the default space allocations from PowerExchange installation and the table-sizing requirements for the capture directory tables:

<table>
<thead>
<tr>
<th>Default Table Space / 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 for each column, across all tables from which changes are captured</td>
</tr>
<tr>
<td>PWXPFLDS / TCAPFIELDS</td>
<td>3 KB</td>
<td>1</td>
<td>One row for each column that has a FIELDPROC, across all tables from which changes are captured</td>
</tr>
<tr>
<td>PWXPSTAT / TCAPSTATUS</td>
<td>3 KB</td>
<td>1 KB</td>
<td>One row for each table from which changes are captured</td>
</tr>
<tr>
<td>PWXPTABL / TCAPTABLES</td>
<td>180 KB</td>
<td>20 KB</td>
<td>Up to three rows for each table from changes are captured</td>
</tr>
<tr>
<td>PWXPTBSP / TCAPTABLESPACE</td>
<td>180 KB</td>
<td>20 KB</td>
<td>Up to three rows for each table space in the DB2 catalog, including table spaces that contain tables that are not registered for change capture</td>
</tr>
</tbody>
</table>
Running Multiple DB2 ECCRs

You might need to run multiple DB2 ECCRs on the same z/OS image in certain scenarios. Typically, you run multiple ECCRs in the following scenarios:

- You need to capture changes from multiple DB2 subsystems on the same z/OS image. The subsystems are not data sharing subsystems or are not part of the same data sharing group. For example, the subsystems might be test and production subsystems on the same z/OS image.
- You need to capture changes from a single DB2 subsystem and use separate capture environments for separate sets of tables. For example, the DB2 subsystem might contain both test and production tables, and you want to use separate capture environments for the test and production tables.

For each of these scenarios, certain considerations apply.

### Change Capture from Multiple DB2 Subsystems on Single z/OS Image

If you need to capture changes from multiple non-data-sharing DB2 subsystems that run on the same z/OS image, review the following considerations:

- A unique DB2 ECCR instance is required for each subsystem. The DB2 ECCR connects only to a single subsystem and captures changes only from that subsystem, assuming you do not use data sharing.
- The capture name that you specify in the CA statement in the DB2 ECCR REPL2CTL control file must be unique for each ECCR and across the z/OS image and sysplex.
- Each DB2 ECCR can have its own unique set of PowerExchange Listener, Agent, and Logger tasks, although configuring a separate environment for each ECCR is not required. For example, you might want to configure separate environments for test and production subsystems, but use the same environment for two test systems.

### Change Capture from a Single DB2 Subsystem with Multiple Capture Environments

If you need to capture changes from separate capture environments on the same DB2 subsystem, each environment with its unique DB2 ECCR, review the following considerations:

- Each DB2 ECCR execution must have its own unique parameter files. These files are specified in the REPL2CTL and REPL2OPT DD statements in the ECCR JCL.
- 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 ECCR name that you specify in the CA NAME statement in the REPL2CTL DD data set must be unique for each DB2 ECCR and across the z/OS image and a sysplex.
- DB2 registrations contain either the DB2 subsystem ID (SSID) or group attachment name. To allow registrations to be divided among multiple capture environments, each DB2 ECCR must have its own PowerExchange Listener, Agent, and Logger tasks.

### Table Sizing Requirements

<table>
<thead>
<tr>
<th>Default Table Space / Table Name</th>
<th>Install PRIQTY</th>
<th>Install SECQTY</th>
<th>Table Sizing Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>PWXPUPDT / TCAPUPDATE</td>
<td>3 KB</td>
<td>1 KB</td>
<td>One row for each DB2 ECCR</td>
</tr>
<tr>
<td>PWXPWORK / TCAPWORK</td>
<td>720 KB</td>
<td>48 KB</td>
<td>One row for each in-flight catalog change</td>
</tr>
</tbody>
</table>
DB2 Data-Sharing Considerations

A DB2 data sharing environment is composed of a collection of DB2 subsystems, called a *data sharing group*, that operate from the same DB2 catalog.

Subsystem members of a data sharing group can directly access any table in the DB2 catalog and change the same data while maintaining data integrity. DB2 controls access through grants, plans, and other usual methods.

Before implementing the DB2 ECCR in a data sharing environment, review the following configuration considerations:

- In the DB2 bind JCL for the DB2 ECCR in the RUNLIB(XIDDB225) member, 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 that are registered under the name that is specified in the RN parameter in the PLAN statement of the REPL2OPT DD data set. The RN parameter can specify the SSID of a data sharing group member or the group attachment name. All tables must be registered under a single DB2 SSID or group attachment name. Informatica recommends that you use a group attachment name for greater flexibility.

- The DB2 ECCR uses the CN parameter in the PLAN statement in the REPL2OPT DD data set to attach to DB2. You can specify either a SSID or group attachment name for the CN parameter. The CN parameter is optional unless you need to attach to a specific DB2 subsystem. If the CN parameter is not specified, the ECCR uses the RN parameter value to attach to the subsystem.

To have the flexibility to move the ECCR to another z/OS system that has active members in the same DB2 data sharing group without changing parameters, use the DB2 group attachment name in the CN parameter or default to the RN parameter value. The ECCR must still have access to the PowerExchange Agent and Logger.

- In a DB2 for z/OS Version 9.1 new-function mode or later 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.

- If you configure DB2 to write archive logs to tape, check the MAXRTU subsystem parameter on each data sharing instance where you plan to run the DB2 ECCR. The MAXRTU value must be greater than or equal to the maximum number of concurrently active members in the data sharing group. The MAXRTU parameter is specified on the DSNZPARM DSNLOGP macro. If the MAXRTU value is less than the maximum number of concurrently active members, the DB2 ECCR might hang.

If You Migrate from DB2 for z/OS Version 9.1 to Version 10

If you plan to migrate to DB2 10 for z/OS from DB2 9.1, review the following CDC considerations:

- You do not need to upgrade the DB2 ECCR capture directory tables, assuming that you previously upgraded them for support of DB2 9.1 new-function mode or DB2 8.1 new-function mode.

- You can migrate tables that have TIMESTAMP columns from DB2 9.1 to DB2 10 without changing their capture registrations. In DB2 9.1, TIMESTAMP columns always have a length of 10 and scale of 0. When you migrate to any DB2 10 migration mode, the TIMESTAMP columns retain these length and scale values. However, if you add or alter TIMESTAMP columns in these tables in any DB2 10 migration mode, you must re-create the capture registrations and extraction maps for the new or altered tables.

- You must rebind the DB2 plan and packages. The DB2BIND member in the RUNLIB library contains the BIND statements that the XIDDB225 installation job uses to perform the binds for DB2 Version 10. You must have SYSCTRL authority to run this job.
• The first time you start the DB2 ECCR after migrating to DB2 10 compatibility mode, perform a cold start. Also perform an ECCR cold start after you migrate from compatibility mode to new-function mode. If you do not cold start the ECCR, the ECCR might not capture DDL changes that were made to tables of CDC interest. In this event, the PowerExchange capture catalog tables become invalid, which can cause capture processing of the tables with the DDL changes to fail.

If You Migrate to DB2 for z/OS Version 11

If you migrate to DB2 Version 11 from an earlier DB2 version, you must upgrade the DB2 ECCR capture directory tables prior to migration.

The format of the capture directory tables for DB2 11 must support the extended 10-byte format of RBA and LRSN values in DB2 log records, which was introduced in DB2 11. Also, DB2 11 introduces some changes to the DB2 catalog tables that the DB2 ECCR uses.

You must upgrade the capture directory tables before migrating to DB2 11 conversion mode. If you use DB2 data sharing, Informatica recommends that you upgrade the capture directory tables before you migrate the first subsystem member of the data sharing group to DB2 11.

The DB2SGENB member in the RUNLIB library contains DDL for upgrading the capture directory tables for DB2 Version 11 support. How you upgrade the capture directory tables depends on whether you previously used the DB2 ECCR for the DB2 subsystem.

The first time you start the DB2 ECCR after migrating to DB2 11, you must perform a cold start. In a data sharing environment, cold start the ECCR after you migrate the first member of the data sharing group to DB2 11. You do not need to cold start the ECCR after migrating any of the other members in the data sharing group.

Also, you must rebind the DB2 plan and packages. The DB2BINDB member in the RUNLIB library contains the BIND statements that were customized at installation. The XIDDB225 installation job performs the binds for DB2 11.

Note: If you upgrade the capture directory tables to support DB2 Version 11 and then perform subsequent PowerExchange upgrades, you do not have to upgrade the capture directory tables again for DB2 11 support.

New Users of the DB2 ECCR

If you upgraded to a DB2 Version 11 subsystem and want to begin using the DB2 ECCR to capture changes from the new subsystem, run the z/OS Installation Assistant again. Perform a full installation.

In the Installation Assistant, select the DB2 CDC option on the Data Sources page. On the Select DB2 CDC Parameters page, verify that DB2 Change Data Capture is selected, select the DB2 V11+ option, and enter the name of the DB 11 database where you want to store the capture directory tables in the Capture Database field.

Then run the SETUPDB2 job, which submits the XIIDBB220 job. The XIIDBB220 job uses the DDL in the DB2SGENB member of the RUNLIB library to create capture directory tables that support DB2 11.

You should not need to perform any other tasks to upgrade the capture directory tables.

The first time you start the DB2 ECCR, perform a cold start.
If you migrate a DB2 subsystem from which you previously captured changes to DB2 Version 11, you must perform some tasks before you can start capturing changes from the migrated subsystem.

Immediately before you upgrade to DB2 11 conversion mode, make sure that the DB2 ECCR has captured all of the available changes from the earlier DB2 subsystem. After you upgrade to DB2 11 conversion mode and run the DB2 DSNTIJTC job to migrate the DB2 catalog, cold start the DB2 ECCR the first time you start it.

Before you migrate from DB2 11 conversion mode to new-function mode, make sure that the DB2 ECCR has captured all of the available changes captured from the DB2 11 subsystem while it was running in conversion mode. Before you run the DB2 DSNTIJEN job to enable new-function mode, check the DB2 DSNZPARM Existing Users of the DB2 ECCR

DSN6SPRM macro. If the macro specifies RESTRICT_ALT_COL_FOR_DDC=YES, set DATA CAPTURE NONE on statements without causing negative effects.

Also, rebind the DB2 plan and packages that PowerExchange uses, if you have not done so previously. Run the XIDDB225 job from installation, which uses the DB2BINDB member in the RUNLIB library.

Note: If you rebind the DB2 packages before upgrading the capture directory tables, the binds result in a condition code of 4, which is acceptable.

The first time you start the DB2 ECCR after migrating to DB2 11 new-function mode, you must cold start the ECCR.

Configuring DB2 for CDC

Before you can capture changes for DB2 for z/OS tables, you must complete some DB2 configuration tasks. These tasks are:

- Verify that PowerExchange supports your DB2 for z/OS version and that you applied the recommended IBM maintenance to the z/OS system.
- Start the DB2 subsystem, if it is not running, on the system where you plan to run the DB2 ECCR.
- Activate change data capture for DB2 catalog tables.
• Enable dual logging in DB2.

**Activating Change Data Capture for DB2 Catalog Tables**

PowerExchange requires that DATA CAPTURE CHANGES be defined for the following DB2 catalog tables:

* SYSTABLES
* SYSCOLUMNS
* SYSTABLESPACE
* SYSFIELDS
* SYSCOPY

**Warning:** If DATA CAPTURE CHANGES is not enabled for one or more of these catalog tables, the DB2 ECCR fails without processing any data.

The DB2 ECCR fails with the following message:

```
PWXEDM17754E Capture program of DB2 Replication ending - DB2 Catalog tables not Data Capture Changes
```

**RELATED TOPICS:**

* "Altering DB2 System Tables for DATA CAPTURE CHANGES" on page 224

**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 Usage Guidelines

The DB2 ECCR has the following usage guidelines:

- You must define DB2 source tables with the DATA CAPTURE CHANGES option. For more information about this option, see the IBM DB2 documentation.

- The first time you start the DB2 ECCR, use the START COLD parameter. Thereafter, use the START WARM parameter except when a cold start or special start is required for recovery purposes.

- You need at least one active capture registration to start the ECCR successfully. If no active registrations exist, the ECCR abends with a U3680 code, and PowerExchange issues message PWXEDM177S09E to indicate that no active registrations exist.

- The DB2 ECCR issues IFCID 306 READS requests to read DB2 log data. To issue the READS request, the ECCR requires that MONITOR TRACE 1 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 that includes the changes 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:
  
  PWXEDM177009I  -START TRACE (MONITOR) PLAN(plan) LOCATION(caname) CLASS{1} HAS BEEN EXECUTED

  If MONITOR TRACE 1 is started, the DB2 ECCR does not issue the START TRACE command. If MONITOR TRACE 1 was not started or has been stopped, the DB2 ECCR starts it.

- Ensure that the DB2 ECCR has read access to the physical data sets that underlie the DB2 table space with the CDC source tables when all of the following conditions apply:
  - You use reordered row format for the table space.
  - One or more of the tables in the table space contain variable-length columns.
  - You subsequently issue an ALTER TABLE command to add a fixed-length column to at least one of the tables that contain variable length columns.

  When these conditions are true, the physical column layout of the tables in the table space can change. The DB2 ECCR requires read access to the physical data set that contains the table space to get the information that it needs to interpret the row format changes and decode the data in the DB2 logs.

  In this situation, you must also run the DB2 ECCR under the control of a user ID that has read access to the physical data sets.

- 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 after 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.

- At DB2 ECCR startup, the ECCR issues SQL PREPARE and DESCRIBE statements to verify that the TCAP tables are of the correct format for the DB2 version from which the ECCR is capturing change data. If you use DB2 Version 10 or later, ensure that the user ID under which the ECCR runs has the EXPLAIN system privilege. This privilege is required for the ECCR to issue the PREPARE and DESCRIBE statements.
DB2 ECCR Access to DB2 Catalog Tables

The DB2 ECCR requires read access to the following DB2 for z/OS catalog tables:

- SYSIBM.SYSCOLUMNS
- SYSIBM.SYSFIELDS
- SYSIBM.SYSTABLEPART
- SYSIBM.SYSTABLES
- SYSIBM.SYSTABLESPACE

DB2 ECCR Control Statements in the REPL2CTL DD Data Set

To specify the DB2 for z/OS ECCR name, enter the CA NAME statement in the data set or RUNLIB member that is allocated by the REPL2CTL DD statement in the ECCR JCL. You can include the optional STOPAFT statement to indicate when to stop the ECCR.

At the completion of z/OS installation, PowerExchange creates the RUNLIB(REPDB2CT) member that contains the ECCR control statements. The REPL2CTL DD in the ECCR JCL points to this REPDB2CT member. You can edit the control statements in the REPDB2CT member, or you can copy the member under another name and then update the JCL to point to the new member name.

When editing the control statements, use the following syntax:

```
CA NAME=eccr_name
[STOPAFT {LOGLOC=rba|LOGTS=timestamp}]
[UOWPREFIX=xx]
```

**Statement descriptions:**

**CA NAME=eccr_name**

Required. A name for the DB2 ECCR. This name must be unique within a sysplex.

**Warning:** If you change the CA NAME value, the ECCR cannot warm start from the position where it last stopped.

The DB2 ECCR uses this name for the following purposes:

- The ECCR name that connects to the PowerExchange Logger for MVS 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

This name can be 1 to 8 alphanumeric characters in length. Default is PWXDB201. You can enter another name in the z/OS Installation Assistant during PowerExchange installation.

**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 for MVS.

**STOPAFT (LOGLOC=rba|LOGTS=timestamp)**

Optional. An RBA or timestamp that determines when the DB2 ECCR will stop. The ECCR uses this parameter regardless of how you started it.
You can specify only one STOPAFT statement. In the statement, you must specify one of the following parameters:

**LOGLOC=**\texttt{rba}  
A 20-digit hexadecimal RBA value that indicates where the DB2 ECCR will stop in the DB2 log. If the ECCR is connected to a DB2 data-sharing group member, this value is a log record sequence number (LRSN).

This RBA or LRSN value must be larger than the RBA or LRSN value at which the ECCR started. Otherwise, the ECCR stops as soon as it gets the first record from the DB2 log.

**LOGTS=**\texttt{timestamp}  
A date and time that determines where the DB2 ECCR will stop in the DB2 log. When the ECCR encounters a log record that has a timestamp equal to or later than the LOGTS timestamp, the ECCR stops.

The \texttt{timestamp} value has format YYYY-MM-DD-hh.mm.ss.nnnnnn. The date must be a valid date. For example, 2012-02-31-17.15.59.000000 is not valid because February 31st is not a valid date.

Depending on how you start the ECCR, use the following criteria to set this parameter:

- For a warm start, enter a timestamp value that is equal to or later than the timestamp of the last log record processed.
- For a special start, enter a log record timestamp that is equal to or later than the timestamp of the log record that is specified in the STARTLOC keyword of the START statement.
- For a cold start, enter a timestamp value that is equal to or later than the current time.

If STOPAFT statement is not specified, the ECCR runs until you explicitly stop it.

**UOWPREFIX=**\texttt{xx}  
A two-character prefix that is used as the first 2 bytes of the UOW ID that the DB2 ECCR creates and sends to the PowerExchange Logger for MVS when a DB2 unit-of-recovery contains data to be captured. By default, the last two characters of the CA NAME value are used. If you use multiple DB2 ECCRs with CA NAME values that end with the same last two characters, you can use this parameter to define a unique prefix for each ECCR to include in its UOW IDs.

### Sample Statement in the REPL2CTL DD Data Set

The following example control statement is in the RUNLIB(REPDB2CT) member to which the REPL2CTL DD points:

\begin{verbatim}
CA  NAME=PWXDB201
\end{verbatim}

### DB2 ECCR Configuration Statements in the REPL2OPT DD Data Set

To configure DB2 for z/OS ECCR processing, you can specify statements in the data set or RUNLIB member that is allocated by the REPL2OPT DD statement in the ECCR JCL.

At installation completion, PowerExchange creates the RUNLIB(REPDB2OP) member that contains these ECCR statements, as customized based on your installation input. The REPL2OPT DD in the ECCR JCL points to this REPDB2OP member. You can edit the REPDB2OP member, or you can copy it under another name and then update the JCL.

If you edit the statements after starting the ECCR, you must refresh or restart the ECCR, depending on which statements you change.
Use the following syntax:

```sql
DB2 PLAN=plan_name (RN=reg_ssid|CN=conn_ssid) RN=reg_ssid CN=conn_ssid
START [COLD|WARM|STARTLOC=rba] [USEdir],[USESTAT]

[CHKSCHM {NO|YES|WARN}]
[COMMITINT {MB=milliseconds}]
[EC {PERMIT=number_of_errors|@}]
[IF {OPT=[N|Y]}] [4KPages={nnn|SO}]
[MODE {RR|CM}]
[ROWNOTDECOMPRESSED {FAIL|NOFAIL}]
[STAT LEV={ST|S}] [SEC=seconds]
[TRACE {option}]
```

All of the statements must begin in column 1.

**Statement Descriptions**

**DB2 PLAN=plan_name (RN=reg_ssid|CN=conn_ssid) RN=reg_ssid CN=conn_ssid**

Required. Specifies the plan and subsystem name or group name for the DB2 system to which the DB2 ECCR attaches.

You can specify RN, CN, or both RN and CN. At least one of these keywords is required. If you specify RN or CN only, the specified keyword is used for the non-specified keyword.

**Tip:** When implementing the DB2 ECCR in a data-sharing environment, Informatica recommends entering the group attachment name for the RN keyword and for 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. By using the group attachment name, you make the 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 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=reg_ssid**

Specifies the DB2 subsystem identifier that appears in the capture registrations.

This value must match the value that is specified in the Database Instance field in the registration group in the PowerExchange Navigator. If not specified, the CN value is used by default.

A valid value is:

- A DB2 subsystem ID (SSID) or DB2 group attachment name
- An uppercase value of one to four characters in length that begins in column 19

**CN=connect_ssid**

Specifies the DB2 subsystem identifier to which the DB2 ECCR connects. If not specified, the RN value is used by default.
A valid value is:

- A DB2 subsystem ID (SSID) or DB2 group attachment name
- An uppercase value of one to four characters in length that begins in column 27

The following examples show combinations of RN and CN keywords:

- If you have DB2 subsystem SS01 in non-data-sharing environment, use the following DB2 statement:
  ```sql
  DB2 PLAN=plan_name RN=SS01
  ```

- If you migrate subsystem SS01 to a data-sharing environment called GRP1, use the following DB2 statement:
  ```sql
  DB2 PLAN=plan_name RN=SS01 CN=GRP1
  ```

- If you add a 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.
  ```sql
  DB2 PLAN=plan_name RN=GRP1
  ```

 Restart the DB2 ECCR to activate a change for this statement.

**START (COLD|WARM|STARTLOC=rb<stdin|stdin>,[USEDIR],[USESTAT])**

Required. Controls the method by which the DB2 ECCR is started.

Options are:

**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=rb<stdin|stdin>,[USEDIR],[USESTAT]**

Restarts change-capture process from a specific point in the DB2 log.

The `rb` value specifies the 20-digit hexadecimal RBA value or the log record sequence number (LRSN) at which the DB2 ECCR should start in the DB2 log.

The following options are optional:

- **USEDIR.** The DB2 ECCR uses the source table information from the data-resource information that was registered in the PowerExchange when the STARTLOC option was specified.
- **USESTAT.** The DB2 ECCR uses a status of 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.
**CHKSCHM (NO|YES|WARN)**

Optional. Specifies whether the DB2 ECCR verifies schema registrations at ECCR startup. Also determines how errors, if found, are handled. This schema verification processing is in addition to the verification processing that is performed when the ECCR receives the first change record for a registered schema.

Enter one of the following options:

- **NO**. Does not verify registered schema at ECCR startup. When the ECCR receives the first change record for a schema, the ECCR verifies each registered schema against the information in the DB2 catalog.
- **YES**. Verifies all registered schema information against the information in the DB2 catalog at ECCR startup and when you refresh the ECCR. If the verification process encounters errors, the ECCR ends.
- **WARN**. Verifies all registered schema information against the information in the DB2 catalog at ECCR startup and when you refresh the ECCR. If the verification process encounters errors, the ECCR issues a warning message and continues processing.

Default is NO.

Refresh or restart the DB2 ECCR to activate a change for this statement.

**COMMITINT [MS=milliseconds]**

Specifies the time interval, in milliseconds, after which the DB2 ECCR issues an SQL COMMIT to free resources that are held on its behalf because of IFI306 activity.

Valid values are 0 through 999999. Default is 60000, or 60 seconds.

A value of 0 disables time-based SQL COMMITs. The DB2 ECCR issues SQL COMMITs only after the following types of events:

- ECCR startup
- Processing of a DB2 ECCR REFRESH command
- Processing of a UR that contains DDL

**EC PERMIL=number_errors**

Optional. Specifies the maximum number of acceptable errors per thousand updates.

Default value is 0.

Refresh or restart the DB2 ECCR to activate a new value for this statement.

**IFI306 [OPT={N|Y}] [4KPAGES={nnn}|50])**

Controls the DB2 ECCR interaction with the DB2 instrumentation facility interface (IFI). Specify the OPT keyword, the 4KPAGES keyword, or both. At least one of these keywords is required.

OPT

Set this keyword to Y to reduce the number of log records that the DB2 IFI passes to the ECCR in each transmission.

The OPT keyword must be in all uppercase and begin in column 8.

Default is N.

**Note**: When the OPT keyword is set to Y, the ECCR does not capture DB2 QUIESCE operations from the SYSCOPY table.
4KPAGES

Enter the number of 4-KB pages of KEY-7 CSA storage to use for the IFI 306 buffer that stores the data to pass to the ECCR. When entering this value, the following rules apply:

- The 4KPAGES keyword must be in all uppercase and begin in column 14.
- You can enter up to three digits.
- You must pad a value less than three digits with spaces, ending in column 24.

Default is 50.

**Important:** Do not change the default value unless Informatica Global Customer Support directs you to do so.

If you add, remove, or change the IFI306 statement, you must restart the DB2 ECCR for your change to take effect.

**MODE (RB|CM)**

Optional. Specifies whether the DB2 ECCR runs in rollback mode or compensation mode.

Enter one of the following options:

- **RB.** 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.

Default is RB.

Restart the DB2 ECCR to activate a change for this statement.

**ROWNOTDECOMPRESSED (FAIL|NOFAIL)**

Optional. Indicates whether the DB2 for z/OS ECCR continues or fails when it encounters row data that has not been decompressed for a table with an active capture registration. This situation can occur, for example, if a REORG operation causes the DB2 compression dictionary to become invalid.

Enter one of the following options:

- **FAIL.** If the ECCR encounters rows with compressed data, it ends abnormally with abend code U3680, reason code 02710009. PowerExchange issues error message PWXEDM177462E to the EDMMSG data set and as a WTO message.
- **NOFAIL.** If the ECCR encounters rows with compressed data, it skips them and continues reading the DB2 log. PowerExchange issues informational messages PWXEDM177462I and PWXEDM177596I to the EDMMSG data set and as WTO messages.

Default is FAIL.

**Tip:** You can use the WTO messages to automate alert notifications to the appropriate system users.

**STAT LEV=(ST|SQ) [SEC=seconds]**

Optional. Specifies the interval 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 that PowerExchange prints to the EDMMSG data set for tables of interest to the DB2 ECCR.

- **ST.** Prints summary statistics for the tables of interest.
- **SQ.** Prints statistics on SQL INSERT, UPDATE, and DELETE operations for each table of interest.

Default is ST.

**Tip:** 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=seconds**

Specifies the number of seconds in the reporting period. Default is 3600 seconds, or 1 hour.

Refresh or restart the DB2 ECCR to activate a change for this statement.

**TRACE [option]**

Optional. Enables tracing for troubleshooting behavior and performance problems of the DB2 ECCR.

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

To activate more than one trace, you must enter the TRACE statement multiple times. If you specify TRACE without a keyword, a minimal trace is activated. A minimal trace is the same level of tracing that the MINI keyword sets.

The TRACE statement must start in column 1. The trace option, if specified, must start in column 7.

The following table describes the trace options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALL</td>
<td>Activates all tracing within the DB2 ECCR.</td>
</tr>
<tr>
<td>CAPDIR</td>
<td>Traces DB2 ECCR capture directory access.</td>
</tr>
<tr>
<td>CB</td>
<td>Traces DB2 ECCR internal control block activity.</td>
</tr>
<tr>
<td>COMMIT</td>
<td>Traces DB2 ECCR commit and rollback activity.</td>
</tr>
<tr>
<td>DB2CAT</td>
<td>Traces DB2 catalog access.</td>
</tr>
<tr>
<td>DECOMPRESSION</td>
<td>Traces record decompression for captured records.</td>
</tr>
<tr>
<td>ECCRINFO</td>
<td>Provides diagnostic information about the DB2 ECCR environment.</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>
<tr>
<td>LOGIFI</td>
<td>Traces reading the DB2 log through IFI.</td>
</tr>
<tr>
<td>LOGREC</td>
<td>Traces reading a DB2 log record.</td>
</tr>
</tbody>
</table>
### Configuring the DB2 ECCR JCL

The following sample JCL for the DB2 ECCR is provided in the RUNLIB library member ECCRDB2:

```plaintext
//PWXDB2EC 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=PWX29200,TIME=NOLIMIT
//STEPLIB DD DISP=SHR,DSN=<HLQ..LOADLIB
//  DD DISP=SHR,DSN=<HLQ..LOAD
//  DD DISP=SHR,DSN=<db2exit>
```

---

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINI</td>
<td>Activates a minimal trace. This option is the default.</td>
</tr>
<tr>
<td>RECCDC</td>
<td>Traces log record processing for captured DB2 change data.</td>
</tr>
<tr>
<td>RECDL</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>ROLLBACK</td>
<td>Provides the same function as COMMIT.</td>
</tr>
<tr>
<td>SERVICES</td>
<td>Traces DB2 ECCR services.</td>
</tr>
</tbody>
</table>

Note: When you use the TRACE statement and its keywords, the REPL2TRA DD statement must be present in the ECCR JCL.

Refresh or restart the DB2 ECCR to activate a new value for this statement.

### Sample Statements in the REPL2OPT DD Data Set

The following example statements are in the RUNLIB(REPDB2OP) member to which the REPL2OPT DD points:

```plaintext
******************************************************************************
* 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=0000000000000000 USEDIR,USESTAT
  * DB2 PLAN=DTLCPV80 RN=DSN1
  DB2 PLAN=<plan_name> RN=<ssid>
  * DB2 PLAN=DTLCPV52 CN=DSN1
EC PERMIL=000
STAT LEV=ST SEC=3600
CHKSCHM NO
MODE RB
```
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 (REPD2CT in RUNLIB) associated with the ECCR.</td>
</tr>
<tr>
<td>REPL2OPT DD</td>
<td>Specify the REPL2OPT file (REPD2OP in RUNLIB) associated with the ECCR.</td>
</tr>
<tr>
<td>REPL2TRA DD</td>
<td>Specify the output data set for the DB2 ECCR TRACE output. The default and recommended specification is SYSOUT=*. The DB2 ECCR writes data to this DD statement in error situations and if the TRACE statement is included in the REPL2OPT file.</td>
</tr>
</tbody>
</table>
Defining Restart Tokens for a DB2 Target Table Materialized from an Image Copy

If you materialized a DB2 for z/OS target table from a full image copy, set up restart tokens for change data extraction based on the event markers from the QUIESCE TABLESPACE step in the materialization job.

The QUIESCE TABLESPACE step generates the following message in the PowerExchange Logger output:

`PWXEDM172774I Event Mark generated by ECCR RCRDB201 for:
  DB2 QUIESCE of TABLESPACE RROACCT .RROACCT2 at DB2 RBA/LRSN 006CFA4A958D
  EDP Logger RBA ................ : D9D906D3404000000EDD83CE00000000
  Sequence number ................ : 00000EDD83CE0000000
  Edition number ................ : C637E64841AB5000
  Source EDNAME(s) ............... : DB2DSNWaccount21`

Copy and paste the **EDP Logger RBA** and **Sequence number** values into the RESTART1 and RESTART2 statements in the restart token file before you start the CDC session.

- In the RESTART1 statement, paste the sequence number twice and then enter eight zeros. For example:
  ```
  RESTART1=00000EDD83CE0000000000000EDD83CE00000000000000
  ```
- In the RESTART2 statement, paste the EDP Logger RBA once. For example:
  ```
  RESTART2=D9D906D3404000000EDD83CE00000000
  ```

After you start the CDC session, PowerExchange uses the restart tokens to determine the point in the change stream from which to begin extracting changes.

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 203
- “DB2 Data-Sharing Considerations” on page 204

### Managing DB2 CDC

This section describes how to start and stop the DB2 ECCR. It also describes how to control DB2 ECCR statistics and output.

#### Stopping the DB2 ECCR

You can stop the DB2 ECCR by issuing the ECCR QUIESCE command or the MVS STOP command.

If you use the QUIESCE command, the DB2 ECCR waits until it reaches a point in the DB2 log where no in-flight UOWs exist before shutting down. Informatica recommends that you use the QUIESCE command, unless you need to stop the ECCR immediately, for faster restart processing later. Also, you must use the QUIESCE command before you upgrade PowerExchange or DB2. On a busy DB2 subsystem, quiesce processing might take a long time. To issue the QUIESCE command, use the following syntax:

```
F ecr_task_name,QUIESCE
```

Use the MVS STOP command to stop the DB2 ECCR immediately, even though in-flight UOWs might exist. To issue the STOP command, use the following syntax:

```
{STOP|P} ecr_task_name
```

After you stop the ECCR, PowerExchange issues messages that report the ECCR starting RBA, the number of records sent to the PowerExchange Logger for MVS, the RBA or LRSN of the last DB2 log record that the DB2 ECCR read, and the URID of the oldest open UOW at that location in the DB2 log.

#### Example QUIESCE Command Output

The following messages are issued when you use the QUIESCE command to stop the DB2 ECCR:

```
PWXEDM177048I CAPTURE PROGRAM ACKNOWLEDGES A QUIESCE COMMAND
PWXEDM177276I DB2 CAPTURE ENDING DUE TO CAPTURE QUIESCE COMMAND
```
Example MVS STOP Command Output

The following messages are issued when you use the MVS STOP command to stop the DB2 ECCR:

PWXEDM1770461 CAPTURE PROGRAM ACKNOWLEDGES A MVS STOP COMMAND
PWXEDM1772761 DB2 CAPTURE ENDING DUE TO MVS STOP COMMAND
PWXEDM1772821 BEGIN DB2 CAPTURE TERMINATION
PWXEDM1770081 STOP TRACE(MONITOR) CLASS(1) HAS BEEN EXECUTED
PWXEDM1772681 LAST READ DB2 LOG
LOCATION=rba_or_lrsn.data_sharing_member_id.sequence_number
PWXEDM1728091 Change Capture counts for DEBB/ROADAKG.SOURCE: Insert=0, Update=0, Delete=0
PWXEDM1728091 Change Capture counts for DEBB/ROADAGK.DGKSRC01: Insert=0, Update=0, Delete=1
PWXEDM1728411 EDM ECCR DEBB0001 disconnected from EDM Logger PWXL, Log
RBA=X'0000014A5400000'
PWXEDM1728218I Left XCF group 'PWXL' as member 'PWXDB2CC'
PWXEDM1728291 EDM ECCR sent 1 records to Logger PWXL (1 change records)
PWXEDM1770121 ECCR STATUS: LAST DB2 READ LOC
rba_or_lrsn.data_sharing_member_id.sequence_number
OLDEST OPEN UOW urid.data_sharing_member_id

Commands for Controlling DB2 ECCR Processing

You can control the DB2 ECCR either by issuing MVS MODIFY commands to the ECCR. You can also change configuration statements in the REPL2OPT DD file and then issue the REFRESH command or restart the ECCR.

The following table summarizes the MVS MODIFY commands that you can use to control the DB2 ECCR:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DISPLAY</td>
<td>Prints statistics reports on ECCR activity.</td>
</tr>
<tr>
<td>QUIESCE</td>
<td>Stops the DB2 ECCR after all in-flight UOWs for the ECCR complete and the ECCR sends the change records to the PowerExchange Logger.</td>
</tr>
</tbody>
</table>
DB2 ECCR Reports

The DB2 ECCR prints reports at startup, at completion of capture processing, at a specified statistics reporting interval, and when a DISPLAY command is issued.

At startup, the ECCR generates a report that shows the default ECCR options that are in effect and initialization processing. At shutdown, the ECCR reports the number of captured changes. If you applied any zaps or load module replacements to PowerExchange, the ECCR also reports which ones you applied. The ECCR prints these reports to the output queue or to the location that is specified in the ECCR procedure JCL.

The ECCR also prints summary and detail-level statistics in messages PWXEDM177084I and PWXEDM177085I based on the reporting interval that is specified in the STAT statement of the REPL2OPT DD configuration data set or when you issue a DISPLAY command. The ECCR prints these statistics to the EDTMSG data set. If you set the STAT LEV parameter to the default value of ST, or if you issue the DISPLAY,ST command, the ECCR prints totals by table in the detail-level report. If you set the STAT LEV to SQ, or if you issue the DISPLAY,SQ command, the ECCR prints the counts of the inserts, updates, and deletes captured by table in the detail-level report. The DISPLAY, DISPLAY,ST, and DISPLAY,SQ commands also print the summary statistics report to the JES job log and MVS hardcopy log.

Example DB2 ECCR Startup Report

During DB2 ECR startup, the following report identifies the DB2 ECCR options that are in effect and initialization processing:

```
PWXEDM172852I 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 . . . . . . . . . . . . . . . : PWXA
  Logger Id. . . . . . . . . . . . . . . . : PWXL
  SYSOUT class . . . . . . . . . . . . . . : *
  Action if ECCR error encountered . . . : Continue
PWXEDM172818I Joined XCF group 'PWXL' as member 'PWXDB2CC'
PWXEDM172841I EDM ECCR PWXDB2CC connected to EDM Logger PWXL,
```
Example DB2 ECCR Statistics Reports

Examples of the summary and detail-level statistics reports are provided, with a description of each field in the reports.

The reports are printed to the EDMMSG data set at the reporting interval set in the STAT statement of the REPL2OPT DD data set and in response to a DISPLAY command. Summary statistics are printed in message PWXEDM177084I, and detail-level statistics are printed in message PWXEDM177085I. The level of detail in the PWXEDM177085I message depends on how you set the LEV parameter in the STAT parameter and on whether you specified the ST or SQ parameter in the DISPLAY command.

Note: The DISPLAY, DISPLAY,ST, and DISPLAY,SQ commands also print summary statistics to the JES log and MVS hardcopy log. In this case, only the summary statistics in message PWXEDM177084I are printed.

The following example statistics report is written to the EDMMSG data set in response to a DISPLAY,ST command or STAT LEV=ST statement in the REPL2OPT DD data set:

```
PWXEDM177084I A96D16GC capture statistics at 2014-01-10 20.52.51
DB2 Log Location 00CCB3333DA8F0000000.0000001
DB2 Log Timestamp 2014-01-10 20.41.44
Current Delay= 5.64 sec  Average Delay= 1.60 sec
DB2 Log records REC_TOT REC_INTV REC_PSEC
11,485 6,628 1
EDM Messages MSG_TOT MSG_INTV MSG_PSEC
0 0 0
PWXEDM177085I DETAIL LEVEL STATISTICS FOLLOW
MSG_TOT MSG_INTV MSG_PSEC TABLE_NAME
0 0 0 CWXX11.KANJI6
0 0 0 CWXX11.KANJI6
0 0 0 CWXX11.KANJI2
PWXEDM1774361 No UOWs found
```

Note: In this report, the detail-level statistics in message PWXEDM177085I show total captured changes for each table across different time periods.

The following example statistics report is written to the EDMMSG data set in response to a DISPLAY,SQ command or the STAT LEV=SQ statement in the REPL2OPT DD data set:

```
PWXEDM177084I KHAB201 capture statistics at 2013-10-23 16.39.13
DB2 Log Location 00000000000000000000000000000000
DB2 Log Timestamp 2013-10-23 16.30.01
Current Delay= sec Average Delay= sec
DB2 Log records REC_TOT REC_INTV REC_PSEC
5,475 0 0
EDM Messages MSG_TOT MSG_INTV MSG_PSEC
2 0 0
PWXEDM177085I DETAIL LEVEL STATISTICS FOLLOW
TABLE: KHALL1.TENCHAR
2 INSERTS,
0 UPDATES,
0 DELETES
```

Note: The PWXEDM177085I message shows detailed counts of inserts, updates, and deletes by table.

The following example summary statistics report is written to the EDMMSG data set in response to a DISPLAY command or to the JES job log and MVS hardcopy log in response to a DISPLAY,ST or DISPLAY,SQ command:

```
PWXEDM177084I KHAB201 capture statistics at 2013-10-23 16.39.13 031
DB2 Log Location 00000000000000000000000000000000
DB2 Log Timestamp 2013-10-23 16.30.01
```
The following table describes all of the fields in the summary and detail-level statistics reports:

<table>
<thead>
<tr>
<th>Report Field</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>Current Delay</td>
<td>Displays the delay, in seconds, for the last change record. The delay is the difference between the time when a change record was written to the DB2 log and the time when the ECCR read the record.</td>
</tr>
<tr>
<td>Average Delay</td>
<td>Displays the average delay, in seconds, for processing a change record during the statistical reporting period. The delay is the difference between the time when a change record was written to the DB2 log and the time when the ECCR read the record.</td>
</tr>
<tr>
<td>REC_TOT</td>
<td>In the DB2 Log records section of the summary report, displays the total number of DB2 log records that were read by the ECCR since the ECCR started.</td>
</tr>
<tr>
<td>REC_INTV</td>
<td>In the DB2 Log records section of the summary report, displays the number of DB2 log records that were read by the ECCR since the last statistics reporting interval.</td>
</tr>
<tr>
<td>REC_PSEC</td>
<td>In the DB2 Log records section of the summary report, displays the average number of DB2 log records that the ECCR read per second during the current statistics reporting interval.</td>
</tr>
<tr>
<td>MSG_TOT</td>
<td>In the EDM Messages section of the summary report, displays the total number of changes that the DB2 ECCR captured since the ECCR started. This count includes backout records. The PWXEDM177084I message shows a grand total across all tables, whereas the PWXEDM177085I message from a DISPLAY,ST command shows the total for each table.</td>
</tr>
<tr>
<td>MSG_INTV</td>
<td>In the EDM Messages section of the summary report, displays the total number of changes that the DB2 ECCR captured since the last statistics reporting interval. This count includes backout records. The PWXEDM177084I message shows a grand total across all tables, whereas the PWXEDM177085I message from a DISPLAY,ST command shows the total for each table.</td>
</tr>
<tr>
<td>MSG_PSEC</td>
<td>In the EDM Messages section of the summary report, displays the average number of changes that the ECCR captured per second during the current statistics reporting interval. This average includes backout records. The PWXEDM177084I message shows the average across all tables, whereas the PWXEDM177085I message from a DISPLAY,ST command shows the average for each table.</td>
</tr>
<tr>
<td>TABLE_NAME</td>
<td>In the EDM Messages section of the summary 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 detailed SQL operation statistics report, displays the name of the table for which the INSERTS, UPDATES, and DELETES statistics are reported.</td>
</tr>
<tr>
<td>Report Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>INSERTS</td>
<td>In the detailed SQL operation statistics report, displays the total number of inserts on the table since the ECCR started.</td>
</tr>
<tr>
<td>UPDATES</td>
<td>In the detailed SQL operation statistics report, displays the total number of updates on the table since the ECCR started.</td>
</tr>
<tr>
<td>DELETES</td>
<td>In the detailed SQL operation statistics report, displays the total number of deletes on the table since the ECCR started.</td>
</tr>
</tbody>
</table>

Recovering the DB2 ECCR

You can recover the DB2 ECCR if it fails or if the PowerExchange Logger for MVS stops or fails while attached to the ECCR.

When the PowerExchange Logger stops or abends while attached to the ECCR, the ECCR also abends when it receives the first change record after the PowerExchange Logger failure.

1. Determine the cause of the DB2 ECCR failure.
   - The EC PERMIL statement in the REPL2OPT DD data set indicates the maximum number of errors that the ECCR tolerates before ending.
2. Correct the error.
   - If the DB2 ECCR failed because the PowerExchange Logger ended, restart the PowerExchange Logger.
3. Restart the DB2 ECCR from the point at which it abended.
   - Specify the STARTUP WARM statement in the REPDB2OPT DD data set. Use the same REPL2CTL file that you used prior to the abend.

When you restart the DB2 ECCR or the PowerExchange Logger, the PowerExchange Logger determines the point at which to resume capturing changes for the corresponding CA name.

**RELATED TOPICS:**
- “DB2 ECCR Configuration Statements in the REPL2OPT DD Data Set” on page 210

Altering DB2 System Tables for DATA CAPTURE CHANGES

The DB2 ECCR requires that certain DB2 system tables specify the DATA CAPTURE CHANGES option.

1. Use the QUIESCE command to shut down the DB2 ECCR:
   ```sql
   MODIFY ecr_task_name,QUIESCE
   ```
2. In the EDMMSG data set, find message PWXEDM177012I and record the last read RBA or LRSN that is in this message.
   ```sql
   PWXEDM177012I ECCR STATUS: LAST_DB2_READ.LOC
   rba_or_lrsn.data_sharing_member_id.sequence_number
   OLDEST OPEN UOW uid.data_sharing_member_id
   ```
   You will need this value when you perform a special start of the DB2 ECCR.
3. If any of the following DB2 catalog tables use the DATA CAPTURE NONE option, alter them to use DATA CAPTURE CHANGES:
   - SYSTABLES
You can use the following SQL command:

```
ALTER TABLE SYSEIBM.table_name DATA CAPTURE CHANGES
```

4. Special start the DB2 ECCR.

You must enter the STARTLOC parameter in the START statement in the REPDB2OP member of the
RUNLIB library, or in whichever member the REPL2OPT DD in the ECCR JCL points. For the STARTLOC
value, enter the last read RBA or LRSN from message PWXEDM177012I. For example:

```
START STARTLOC=000000004C372 USEDIR,USESTAT
```

5. After the DB2 ECCR special start completes, verify that no PWXEDM177540W messages were issued.

PWXEDM177540W Some DB2 catalog tables not defined with Data Capture Changes

6. Edit the REPDB2OP member of the RUNLIB library again to specify the WARM parameter in the START
statement.

The next time the DB2 ECCR is restarted, it resumes change capture from last read RBA or LRSN so that
no data loss occurs.

### DB2 ECCR Capture Directory Table Upgrades

You must upgrade the DB2 ECCR capture directory tables under the following DB2 for z/OS and
PowerExchange upgrade scenarios:

- You migrate to DB2 Version 11 conversion mode from DB2 9.1 or 10.
- You migrate to DB2 9.1 or 10 from a version earlier than DB2 8 new-function mode.

### SAMPLIB Members for Upgrading the Capture Directory Tables

PowerExchange provides the following the SAMPLIB members, each with detailed comments, for upgrading
the capture directory tables:

- **BNDECCRB**
  Serves as a template of all of the DB2 BIND statements that are required to bind the DB2 ECCR plan for
  DB2 11 support. The BIND statements are equivalent to those in DB2BIND8. If you use the BNDECCRB
  member to create another bind member, change the PACKAGE, OWNER, and QUALIFIER keywords to
  match those that your DB2 ECCR uses. You must rebind the DB2 ECCR plan after upgrading the capture
directory tables.

- **EXPNDCCP4**
  Creates copies of the capture directory tables to be upgraded for DB2 11 support.

- **EXPNC5L2**
  Upgrades the capture directory tables to support DB2 11 and later, as well as DB2 9.1 and 10, in a DB2
data sharing environment. Also performs the same function as EXPNDCP4. If you ran the SQL in
  EXPNDCP4 previously, you can still run the SQL in EXPNC5L2 without generating errors.

- **EXPNC5R2**
  Upgrades the capture directory tables to support DB2 11 and later, as well as DB2 9.1 and 10, in a DB2
  environment that does not use data sharing. Also performs the same function as EXPNDCP4. If you ran
  the SQL in EXPNDCP4 previously, you can still run the SQL in EXPNC5L2 without generating errors.
**EXPNDCP3**

Expands the TCAPWORK capture directory table to increase the size of the RBA column to properly support longer LRSN values that can occur in DB2 9.1 data sharing environments. Use this member only if you upgraded to PowerExchange 9.6.0 from PowerExchange 9.0.1 or 8.6.1 HotFix 12 or earlier and did not previously apply patch PS23210. The upgraded capture directory tables will not support DB2 11.

**Tip:** Informatica recommends that you use the EXPNDC51 member with the EXPNC5L2 or EXPNC5R2 member instead. The upgraded tables then support DB2 9.1, 10, and 11, and you will not have to upgrade these tables again when you eventually migrate to DB2 11.

**EXPNDCP4**

Increases the length of the SCHEMA_VERSIONS column in the TCAPTABLES table to prevent the DB2 ECCR from ending abnormally when gathering schema version information. Use of this member is optional. This function is also included in EXPNC5L2 and EXPNC5R2.

---

**Upgrading the DB2 ECCR Capture Directory Tables**

You must upgrade the DB2 ECCR capture directory tables before migrating a subsystem to DB2 Version 11 conversion mode or later. If you previously used the ECCR to capture changes from the DB2 subsystem, use this procedure to safely upgrade the tables without losing changes.

If you plan to continue to use DB2 9.1 or 10, you do not have to upgrade the capture directory tables. You can upgrade the tables anytime before migrating to DB2 11.

**Important:** Do not change the schemas of the DB2 tables that are registered for change data capture until after you upgrade the capture directory tables and restart the DB2 ECCR.

1. If the DB2 ECCR is running, use the QUIESCE command to stop it.
2. Customize the SQL statements in the sample EXPNDC51 member in the SAMPLIB library. This SQL creates copies of the current capture directory tables prior to upgrading them.
3. Use SPUFI or a batch SQL utility to execute the SQL statements in the modified EXPNDC51 member.
4. Customize the SQL statements in the sample EXPNC5L2 or EXPNC5R2 member in the SAMPLIB library. Use the EXPNC5L2 member in a data sharing environment, or use the EXPNC5R2 member in a non-data-sharing environment. These members drop the old capture directory tables and create new capture directory tables that support DB2 11 as well as DB2 9.1 and 10. For more information, see the comments in these members.
5. Use SPUFI or a batch SQL utility to execute the SQL statements in the modified EXPNC5L2 or EXPNC5R2 member.
6. Rebind the DB2 plan and packages for the DB2 ECCR.

If you selected **Upgrade by Using New Data Set Names** in the z/OS Installation Assistant when you performed the PowerExchange upgrade, a customized DB2BINDB member is available in the RUNLIB library and contains the latest bind package statements for DB2 11 support. To perform the binds, just run the XIDDB225 installation job again.

If you selected **Upgrade by Using Existing Data Set Names** in the z/OS Installation Assistant, you must add the bind package statements to the previously used DB2BIND member in the RUNLIB library. Use the BNDDECCRB member in the SAMPLIB library as a template of all of the bind statements that are required for the DB2 ECCR for DB2 11 support. If you copy bind statements from this member, edit the PACKAGE, OWNER, and QUALIFIER keywords to match those that the ECCR currently uses. Then rebind the DB2 plan and packages.

**Note:** If you add the new bind statements to the DB2BIND member and rebind the DB2 packages before upgrading the capture directory tables, the bind operation results in a condition code of 4, which is acceptable.
7. Warm start the DB2 ECCR.
   The ECCR resumes change data capture with the upgraded catalog capture directory tables.
   You can now migrate your subsystem to DB2 11.

Reducing the Amount of Data Sent to the DB2 ECCR by Using the IFI306 OPT Statement

By default, DB2 sends all log records to the DB2 ECCR. The ECCR inspects the log records to find change data for registered tables of interest. This ECCR activity might cause high levels of CPU usage and I/O.

If DATA CAPTURE CHANGES is defined on many or all of the tables in the DB2 subsystem, you cannot substantially reduce the amount of data that DB2 sends to the DB2 ECCR.

If DATA CAPTURE CHANGES is defined on only a few tables, you can specify the IFI306 statement with the OPT keyword in the REPL2OPT DD data set to reduce the amount of data that DB2 sends to the ECCR. However, when the IFI306 OPT statement is specified, the DB2 ECCR does not detect DB2 QUIESCE operations on tables that are registered for change data capture. This limitation can lead to change data loss unless you manually create an event marker in the PowerExchange Logger for MVS logs to indicate the restart point. You must balance the benefit of reducing the volume of change data sent to the ECCR against the potential for change data loss from undetected DB2 QUIESCE operations.

Warning: Because the IFI306 OPT statement can result in change data loss, Informatica recommends that you do not use it.

Implementing the IFI306 OPT Statement for the DB2 ECCR

Implement the IFI306 OPT statement in the REPL2OPT DD configuration member to reduce the amount of data that the DB2 ECCR retrieves and improve capture performance. First verify that the IFI306 OPT restriction related to DB2 QUIESCE operations is tolerable in your environment.

Warning: When the IFI306 OPT statement is specified, the DB2 ECCR does not detect DB2 QUIESCE operations. If ignored, this restriction can result in change data loss. For this reason, Informatica recommends that you do not use the IFI306 OPT statement.

You can also use this procedure to remove the IFI306 OPT statement if it was previously implemented.

1. If you run the DB2 ECCR in a DB2 data-sharing environment, use the following QUIESCE command to shut down the DB2 ECCR:
   ```sql
   MODIFY ecr_task_name,QUIESCE
   ```
   If you shut down the DB2 ECCR with this QUIESCE command or if you do not run the DB2 ECCR in a DB2 data-sharing environment, skip to Step 3.
   If you cannot use the QUIESCE command to shut down the DB2 ECCR and you run the ECCR in a DB2 data-sharing environment, continue with this procedure to implement the IFI306 OPT statement. However, use of the IFI306 OPT statement in this situation can cause data loss.

2. In the EDMMSG data set, find message PWXEDM177012I and record the LAST DB2 READ LOC, which is an RBA or LRSN value, from this message.
   ```sql
   PWXEDM177012I ECCR STATUS: LAST DB2 READ LOC
   rba_or_lrsn.data_sharing_member_id.sequence_number
   OLDEST OPEN UOW urid.data_sharing_member_id
   ```
   You will need this value to perform a special start of the DB2 ECCR.

3. Define the IFI306 OPT statement in the REPDB2OP member of the RUNLIB library, or in whichever member to which the REPL2OPT DD statement in the DB2 ECCR JCL points.
4. If you run the DB2 ECCR in a DB2 data-sharing environment and used the ECCR QUIESCE command to shut down the ECCR, or if you do not run the DB2 ECCR in a DB2 data-sharing environment, warm start the DB2 ECCR.

At this point, the implementation of the IFI306 OPT statement is complete.

5. If you run the DB2 ECCR in a DB2 data-sharing environment but did not use the QUIESCE command to shut down the ECCR, perform the following substeps:

   a. Special start the ECCR.
      You must enter the STARTLOC parameter in the START statement in the REPDB2OP member of the RUNLIB library, or if the REPL2OPT DD in the ECCR JCL points to another member, enter the STARTLOC parameter in that member. For the STARTLOC value, enter the last read RBA or LRSN from message PWXEDM177012I. For example:

      ```
      START STARTLOC=000C9041C372 USEDIR,USESTAT
      ```

      **Note:** You do not need to issue a DB2 ECCR REFRESH command to activate use of the IFI306 statement.

   b. Edit the REPDB2OP member of the RUNLIB library again to specify the WARM parameter in the START statement.

      The next time the DB2 ECCR is restarted, it resumes change capture from last read RBA or LRSN so that no data loss occurs.

**Related Topics:**

- "DB2 ECCR Configuration Statements in the REPL2OPT DD Data Set" on page 210

**Manually Creating an Event Marker for the DB2 QUIESCE Utility When Using the IFI306 OPT Statement**

When the DB2 ECCR detects a DB2 table space quiesce, it usually creates an event marker in the PowerExchange Logger for MVS logs that contains restart information. However, if you use the IFI306 OPT statement in the REPL2OPT DD data set, the ECCR does not create an event marker because it cannot read the DB2 log records for the QUIESCE utility.

In this case, you must manually generate an event marker. To generate an event marker, use either the PowerExchange EDMXLUTL Event Marker utility or the DTLUAPPL utility.

**Replacing a Table with Another Table That Has the Same Name**

If you need to replace a table from which changes are captured with another table that has the same name, use this procedure.

1. Stop SQL changes from being written to the table.
   You can set read-only access on the table.

2. Verify that the DB2 ECCR captured all changes up to the point at which change activity was stopped.

3. Rename the table from which changes were captured to a different name.

4. Delete or deactivate the capture registration for the table.

5. Issue the DB2 ECCR REFRESH command to drop the table from the DB2 ECCR.

6. Rename the new table to the name of the dropped table that was previously registered for change capture.

7. Create and activate a capture registration for the newly renamed table.
8. Issue the DB2 ECCR REFRESH command again to add the newly renamed table to the DB2 ECCR.
9. Allow change activity to resume on both tables.

Migrating to a DB2 Data Sharing Environment

If you migrate to a DB2 data sharing environment from a non-data-sharing environment, consider the following points:

- The DB2 ECCR connects to the subsystem with the SSID that is specified in the CN statement of the REPL2OPT DD data set, or in the RN statement if the CN statement is not specified. This single DB2 ECCR performs the following processing:
  - Gets the log records of all DB2 subsystems that are members of the data sharing group.
    - 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 is not lost as long as the DB2 logs are still available, but access to the data might be delayed.
  - Processes all updates for the DB2 subsystems that are members of the DB2 data sharing group.
- If you create a single data sharing group that includes existing DB2 subsystems and you want to continue to use their existing capture registrations, you must run multiple DB2 ECCRs. Run one ECCR for each subsystem from which you capture changes. Each subsystem SSID must be specified in a RN statement in the REPL2OPT DD data set.

After successfully migrating to a data sharing environment, you can minimize the number of ECCRs by combining those that are on members in the data sharing group. However, you must then register all of the DB2 tables from which the ECCRs capture changes under a common SSID or group attachment name. You also might need to change your extraction mappings and processes.

Migrating from a DB2 Data Sharing Environment

If you need to migrate a DB2 environment from data sharing to non-data-sharing mode during DB2 CDC, complete this procedure.

**Note:** Before migrating to non-data-sharing mode, wait until the DB2 ECCR processes all of the change records that were produced in data sharing mode. Otherwise, change data might be lost, which can cause data inconsistencies and require target table rematerialization.

1. Verify that the DB2 ECCR successfully captured all of the log records for source table changes that were written in data sharing mode.
2. Configure read-only (RO) access for the database and each table space. Use the following commands:
   For a database:
   ```
   START DATABASE(database_name) ACCESS(RO)
   ```
   For a table space:
   ```
   START DATABASE (database_name) SPACENAME(table_space_name) ACCESS(RO)
   ```
3. To verify that the DB2 ECCR processed all of the log records that were written prior to setting RO access on the table spaces, issue the following command:
   ```
   MODIFY job_name,DISPLAY
   ```
   This command returns the DB2 log timestamp for when the last-read log record was created. This timestamp must be later than the recorded time at which the last table space with source tables was set to RO access.
4. Stop the DB2 ECCR by issuing the following command:

   STOP job_name

5. After you complete the migration from the DB2 data sharing environment, start the DB2 subsystem in non-data-sharing mode. Then start the DB2 ECCR on one of the following ways:
   - Cold start the ECCR. Then set read-write (RW) access on the table spaces that contain the source tables and allow updates on the source tables again. PowerExchange begins capturing changes that are written in non-data-sharing mode to the source tables.
   - Special start the DB2 ECCR before doing any DDL operations on the source tables. You can perform the special start before or after allowing updates on the source tables again.

6. For the special start, determine the STARTLOC keyword value in the START statement in the REPL2OPT DD data set:
   - Run the DB2 DSNJU004 utility.
   - From the DSNJU004 print output, get the MIN RBA FOR TORBA value.
   - Use the MIN RBA FOR TORBA value as the STARTLOC value.
   - If you specified the group attachment name in the CN statement of the REPL2OPT DD data set, or used the RN statement value by default, specify a DB2 subsystem ID.

After you migrate to a DB2 non-data-sharing environment, DB2 does not support read operations on the log records that were written in data sharing mode.

If you run multiple ECCRs and 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 registered table that is not in the DB2 catalog, the following message is issued:

   P9XEDM177371W TABLE 'creator.table_name' does not exist in DB2 catalog

This warning message 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 DB2 Change Data Capture

You can stop capture processing at various levels depending on your situation.

The following table identifies the methods of stopping change capture by level:

<table>
<thead>
<tr>
<th>Level at Which to Stop Change Capture</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 tables</td>
<td>Alter the DB2 table to specify DATA CAPTURE NONE. Use the following DDL statement: ALTER owner.table_name DATA CAPTURE NONE. <strong>Warning:</strong> When you change the structure of a DB2 table to DATA CAPTURE NONE, changes are no longer written to the DB2 log in the expanded format that is required for change data capture. Consequently, the changes cannot be retrieved later.</td>
</tr>
<tr>
<td>DB2 environment</td>
<td>Stop the ECCR. Use the QUIESCE command or the MVS STOP command.</td>
</tr>
<tr>
<td>Registered DB2 tables</td>
<td>In the PowerExchange Navigator, deactivate or delete the capture registration. Then refresh the DB2 ECCR, or stop and restart the ECCR. <strong>Warning:</strong> Keep at least one active DB2 data-resource registration in the PowerExchange repository (CCT file). If you deactivate or delete all of the DB2 registrations, the DB2 ECCR ends abnormally when you refresh or restart it. For proper restart and recovery, do not delete registrations.</td>
</tr>
</tbody>
</table>
Managing DB2 Schema Changes

To capture changes without interruption to DB2 tables registered for capture, you must manage changes to
DB2 tables and tablespaces.

Schema Verification

When the DB2 ECCR captures the first change record for a DB2 table, the ECCR verifies that the table schema
in the DB2 catalog matches the schema in the corresponding PowerExchange capture registration.

The schema verification routine does not 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 in the catalog matches the schema in the activated registration, capture
processing continues.
• If the DB2 table schema in the catalog does not match the activated schema registration, the verification
routine prints a report and the DB2 ECCR ABENDs.

You can request that the DB2 ECCR also run this schema verification routine at startup by specifying the
CHKSCHM statement in the RUNLIB member to which the REPL2OPT DD statement in the ECCR JCL points.

Sample Schema Verification Report and Abend Messages

The sample schema verification report shows the messages and information that are displayed when
schema verification fails.

In this example, schema verification fails because the schema in the capture registration contains a column
that is not defined in the DB2 catalog. This situation can occur if a column was removed from the table after
the table was registered.

The following example report shows the output and abend messages that are printed:

PKXEDM177502I The DB2 schema for table 'DTUSER.DEFINFO' does not match the active profile schema. DB2 log time = 2004-06-21-17.10.13.296528.
PKXEDM17751I Schema verification failed for table 'DTUSER.DEFINFO'.
PKXEDM172807E ABEND issued by schema verification, Abend code=3680, Reason code=10040001.

Field Descriptions

The following table describes the fields in the example schema verification report:

<table>
<thead>
<tr>
<th>Field</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>
</tbody>
</table>
### Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>Sequential number of the column in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>NL</td>
<td>Length of the column name in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>Column Name</td>
<td>Name of the column in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>Datatype</td>
<td>Datatype of the column in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>Len</td>
<td>Length of the column in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>Pr</td>
<td>Precision of the column in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>Sc</td>
<td>Scale of the column in the DB2 table and associated schema registration.</td>
</tr>
<tr>
<td>N</td>
<td>Whether the column in the DB2 table and associated schema registration can have null values.</td>
</tr>
</tbody>
</table>

---

### Changing the Schema of DB2 Source Tables

If you need to change the schema of a DB2 source table that is registered for change data capture, complete this procedure.

1. Stop SQL changes from being written to the table. You can set read-only access on the table.
2. Verify that the DB2 ECCR captured all changes up to the point at which change activity was stopped.
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 of the captured changes to the target.
5. Change the schema for the DB2 table. If necessary, reorganize the table space that contains the table.
6. Delete the capture registration and associated extraction map.
7. Create another capture registration that uses the new schema.
8. Issue the DB2 ECCR REFRESH command so that the ECCR can use the new registration based on the changed schema.
9. Allow change activity to resume on the table.
10. Restart any extraction processes and, if applicable, PowerExchange Condense.

---

### Recovering from Unplanned Schema Changes to DB2 Source Tables

If schema changes were improperly made to DB2 source tables in a CDC environment, the DB2 ECCR might abend when it reads the first change record for the table after the schema change is made.

When the DB2 ECCR abends, it writes the following messages to the EDMMSG data set:

```plaintext
PNXEDM17751E Schema verification failed for table 'creator.table_name'
PNXEDM172807E ABEND issued by schema verification, Abend code=3685, Reason code=10040001. report_text.
```
Tip: To prevent this problem, make schema changes by following the procedure in "Changing the Schema of DB2 Source Tables" on page 232.

1. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then, shut down PowerExchange Condense.
2. Extract all of the changes to the target.
3. Delete the capture registration and extraction map.
4. Create a new capture registration that uses the new schema.
5. Issue the DB2 ECCR REFRESH command so that the ECCR can use the new registration based on the changed schema.
6. Warm start the DB2 ECCR.
7. Restart any extraction processes and, if applicable, PowerExchange Condense.

Related Topics:
- "Changing the Schema of DB2 Source Tables" on page 232

Altering Columns in DB2 Source Tables

You can alter columns in tables from which the DB2 ECCR captures changes.

You can alter columns in the following ways:

- Increase the length of a VARCHAR or VARGRAPHIC column.
- Rename a column
- Alter the datatype of a column.
- Set the column default or drop the column default.

Important: Depending on how you set the DB2 subsystem parameter RESTRICT_ALT_COL_FOR_DCC, which is defined by the DSNTIJUZ job, you might need to disable DATA CAPTURE CHANGES before altering the column datatype or default:

- If you accept the default value of NO, DB2 allows altering column datatypes or defaults in tables that are defined with the DATA CAPTURE CHANGES option.
- If you set this parameter to YES, you must disable DATA CAPTURE CHANGES before altering a column datatype or default. Otherwise, DB2 issues the SQLCODE -148 error code. Also, after you alter the column datatype or default, you must reorganize the table space that contains the table. While DATA CAPTURE CHANGES is disabled, prevent change activity on the source table. Otherwise, the ECCR does not capture the changes, and the target table must be rematerialized.

Note: Some DB2 releases and maintenance levels also require reorganization of the table space or partition if you increase the length of VARCHAR or VARGRAPHIC columns.

When you are done altering columns, refresh the capture registrations for the tables that contain the altered columns. For more information, see "Changing the Schema of DB2 Source Tables" on page 232.
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 that contains tables registered for change data capture.

However, you must take action if all of the following conditions exist:

- The table space contains multiple tables.
- You altered at least two tables that contain a minimum of one variable-length column to add fixed-length columns.
- The altered tables are not registered for change data capture.

**Tip:** Reorganize the table space before you make any changes to the second altered table. Otherwise, the DB2 ECCR fails because it cannot process DB2 log records for the second table.

For the ECCR to capture changes correctly in this situation, complete the following actions:

- Register one of the altered tables for change data capture, and then refresh or warm start the DB2 ECCR.
- Register the other altered table for change data capture, and then refresh or warm start the DB2 ECCR.
- Change the qualifier of the table space that contains the tables. Use the ALTER TABLESPACE statement with the USING VCAT or USING STOGROUP clause.

**Note:** The DB2 ECCR can capture changes for the altered tables only if you change the qualifier for the table space that contains the tables after you register both altered tables and refresh or warm start the DB2 ECCR.
CHAPTER 11

IDMS Log-Based Change Data Capture

This chapter includes the following topics:

- IDMS Log-Based CDC Overview, 235
- PowerExchange Log Catalog for IDMS Log-Based CDC, 238
- Configuring and Starting the IDMS Log-Based ECCR, 241
- Managing IDMS Log-Based CDC, 251

IDMS Log-Based CDC Overview

PowerExchange IDMS log-based change data capture (CDC) captures changes to registered IDMS source records from the IDMS log files. PowerExchange logs those changes to PowerExchange Logger for MVS log data sets.

To implement a CDC environment, you must complete the following tasks:

- Create capture registrations for the sources in the PowerExchange Navigator.
- Create a PowerExchange Logger Catalog (PWXLOGCAT).
- Populate the PWXLOGCAT with information about the IDMS logs.
- Configure and start the IDMS ECCR.
- Configure restart tokens.
- Enable data access.
PowerExchange IDMS Log-Based CDC Components

The PowerExchange IDMS log-based CDC uses various components on the z/OS and Windows systems.

The following figure shows the PowerExchange IDMS log-based CDC architecture:

In this figure, the components through which the data flows appear as shaded, rectangular shapes with numeric labels. The components that control the data flow appear as elliptical shapes with alphabetic labels.

A user application updates the IDMS source database. IDMS writes the changes to its log files. The PowerExchange IDMS ECCR captures changes from the IDMS logs and sends it to the PowerExchange Logger. The PowerExchange Logger stores the changes in its log files. If you use PowerExchange Condense, PowerExchange Condense performs full or partial condense processing on the change data and stores the data in condense files. When a CDC session runs, the change data is pulled from the PowerExchange Logger log files or PowerExchange Condense condense files.

The following list summarizes the PowerExchange IDMS log-based CDC components:

**PowerExchange Agent**

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.

**PowerExchange Condense**

Optional. Extracts changes from the PowerExchange Logger log data set, performs full or partial condense processing on the data, and then stores the data in condense files.

**PowerExchange IDMS ECCR**

Captures change data from the IDMS logs that are recorded in the PowerExchange Log Catalog and makes that data available to the PowerExchange Logger. The ECCR can run as a batch job or started task.

**PowerExchange Logger for MVS**

Records the change data that the ECCR captured in log data set. When CDC sessions run, PWXPC in conjunction with PowerExchange extracts change data from the PowerExchange Logger log files through the PowerExchange Listener.
PWXLOGCAT or PowerExchange Log Catalog

Contains information about all of the IDMS logs from which to capture change data. You use the PowerExchange Log Catalog utilities, DTLULCAT and DTLULO GC, to build and maintain this catalog.

Warning: Multiple schemas can be registered in 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 test 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:

• "Configuring IDMS Log Catalog Procedures" on page 238
• "Running DTLULCAT" on page 239
• "Running DTLULO GC" on page 240

IDMS Log-Based ECCR Operational Considerations

The IDMS log-based ECCR works with other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent to capture change data.

Before implementing the ECCR, review the following information about ECCR relationships and operational issues:

• The ECCR must log all changes to a single PowerExchange Logger.
• The ECCR must run on the same z/OS system as the PowerExchange Logger and PowerExchange Agent.
• Operational issues in the PowerExchange Logger can cause the ECCR to enter a wait state. While in a wait state, the ECCR cannot capture and record additional changes. After you resolve the PowerExchange Logger issues, the ECCR can resume the capture and recording of change data without any data loss.

Tip: Carefully monitor the PowerExchange Logger to ensure that change data capture proceeds without interruption.

• PowerExchange can capture IDMS change data that has been compressed with the CA-IDMS Presspack package or IDMSCOMP database procedure. If you use IDMSCOMP compression and IDMSDCOM decompression, you do not need to take any special configuration action for CDC.

If you use Presspack compression, you must modify the IDMS ECCR JCL so that the ECCR can use Presspack decompression during change capture. Complete the following configuration steps:

- If the ECCR does not run APF-authorized, specify the standard IDMS runtime libraries, including the data characteristic tables (DCTs) and a valid DMCL, in the STEPLIB concatenation of the ECCR JCL.
- If the ECCR runs APF-authorized, create copies of the IDMS runtime libraries and APF-authorize the copies. Then specify these copies in the STEPLIB concatenation. If you add or modify DCTs for records that are registered for change capture later, you must manually copy the tables to these duplicate libraries.

Warning: Do not APF-authorize the original IDMS libraries. If you do, some IDMS utilities might fail with a SOC4.

- Unless you use the default DMCL named "IDMSDMCL," add a SYSIDMS DD input card and specify the name of the DMCL that you use in the input stream.
The PowerExchange Log Catalog for IDMS log-based CDC contains information about the IDMS logs from which change data is captured.

At PowerExchange installation, the Log Catalog is created as a VSAM file. This file has the default name &HLQ..LOGSCAT and contains a dummy record.

To add information to the Log Catalog, use the DTLULCAT and DTLULOGC utilities. DTLULCAT formats input to DTLULOGC, and DTLULOGC populates the Log Catalog.

You can run the DTLULCAT and DTLULOGC utilities consecutively by using the JCL in the RUNLIB(DTLULCAU) member. Schedule a job that contains this JCL to run as soon as the latest IDMS log is spooled off. For timely CDC processing, it is important that you correctly schedule the addition of logs to the Log Catalog.

Occasionally, you might need to run DTLULOGC separately. In this case, you must manually code the input file.

Ensure that Log Catalog information is updated in a timely manner and is secure and available. IDMS logs that are not recorded in the Log Catalog are unknown to PowerExchange for CDC processing.

**Configuring IDMS Log Catalog Procedures**

Develop procedures for running the DTLULCAT and DTLULOGC utilities in a manner that adds IDMS logs to the Log Catalog in the correct sequence.

The preferred method of operation is to include DTLULCAT and DTLULOGC JCL in an archive log job. Use the DTLULCAU JCL to run DTLULCAT followed by DTLULOGC. You can submit the job by using a WTOEXIT that intercepts a WTO message.

Include the following steps in the job:

1. Offload the active journal to an archive log.
2. Write the archive log to a data set by using a utility such as IEBGENER.
   - You can use a unique data set name and a GDG data set structure.
   - Retain this copy of the log until PowerExchange has captured all changes.
3. Add the data set 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.

To add logs to the Log Catalog, use the following rules:

- A local mode journal must not be added to the Log Catalog if the last available timestamp in the journal is later than the timestamp of the previously added CV mode journal.
If Central Versions (CVs) are varied offline to run in Local Mode, ensure that Local Mode logs are added to the Log Catalog before any CV logs. If a database is varied offline and then back online again and the Local Mode log is not added immediately, a subsequent attempt to add the Local Mode log might fail.

Add logs in the correct sequence.

If you add logs in the incorrect sequence, PowerExchange issues messages such as the following message:

```
51007 162240 MVS 1 PWX-19862 IDMS CATLG FILE: Add Entry Failure - Timestamp not greater than previous for key
XYLOGSID00000000000000000000101DL15CDBAIMSE150DTLUSR.IDMS.DI5SP0.0FF.J4
```

**Running DTLULCAT**

Use the DTLULCAT utility to take a supplied journal name and use it to prepare the input for the DTLULOGC utility.

PowerExchange provides the DTLULCAT utility as an executable on Windows and as the RUNLIB(DTLULCAT) member on z/OS.

The DTLULCAT utility writes information to DDCARD SYSPUNCH. This file is the input to the DTLULOGC utility.

The following example shows sample utility statements:

```plaintext
IDMS_VERSION=15
FILE_TYPE=C
MEDIA_TYPE=D
MEDIA_CONTENT=BI
SERVICE=IDMSE150
INSTANCE_IDENTIFIER=XYLOGSID.
```

The following table describes the utility statements:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDMS_VERSION</td>
<td>A supported IDMS version.</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 media.</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>The LOGSID identifier.</td>
</tr>
</tbody>
</table>
Running DTLULOGC

Use the DTLULOGC utility to populate the PowerExchange Log Catalog with information about the IDMS logs to process.

The following example DTLULCAU JCL runs DTLULCAT followed by DTLULOGC:

```// **************************************************************
// /          SAMPLE JCL TO:   
// /          CAPTURE IDMS JOURNAL FILE INFORMATION AND INPUT STREAM 
// /          INTO FOR DTLULOGC LOG FILE CATALOG ROUTINE.          
// /          NORMALLY THE SYSIN INPUT STREAM WOULD BE A PDS MEMBER. 
// /          THIS NEEDS TO BE INTEGRATED INTO THE END USERS JOURNAL 
// /          ARCHIVING PROCEDURE, WHICH MAY BE Different FROM SITE TO SITE. 
// /          A MECHANISM WILL NEED TO BE ESTABLISHED TO REPLACE THE DATASET 
// /          SPECIFIED VIA THE LOGFILE DD STATEMENT WITH THE LOGFILE 
// /          WHICH IS CURRENTLY THE OBJECT OF THE USERS ARCHIVING PROCEDURE 
// /          AND OUR CATALOG OPERATION 
// /          ******************************************************
//INC1 INCLUDE MEMBER=GENBULK
//DTLULCAT EXEC PGM=DTLULCAT
//STEPLIB DD DISP=SHR,DSN=DTLSR.V800B14.LOADLIB
//DTLFCFG DD DISP=SHR,DSN=DTLSR.V800B14.RUNLIB(DBMOVER)
//DTLKEY DD DISP=SHR,DSN=DTLSR.V800B14.RUNLIB/LICENSE
//DTLMSG DD DISP=SHR,DSN=**HLQ..DTLMSG,FREE=CLOSE
//DTLLOG DD SYSOUT=* 
//LOGFILE DD DISP=SHR,DSN=DTLSR.IDMS.E15SP0.OFF.LOADED.JOURNAL1 
//SYSPRINT DD SYSOUT=* 
//SYSPUNCH DD DSN=**LOGDATA, 
// Disp=(,PASS), 
// Space=(CYL,(2,1),RLSE), 
// DCB=(RECFM=FB,LRECL=80,BLKSIZ=3120) 
//SYsin DD *
//IDMS_VERSION=15
//FILE_TYPE=C
//MEDI_A_TYPE=D
//MEDIACONTENT=BI
//SERVICE=IDMSE150
//INSTANCE_IDENTIFIER=XYLOGSID
//*/
//DTLULOCG EXEC PGM=DTLULOCG
//STEPLIB DD DISP=SHR,DSN=DTLSR.V800B14.LOADLIB
//DTLFCFG DD DISP=SHR,DSN=DTLSR.V800B14.RUNLIB(DBMOVER)
//DTLKEY DD DISP=SHR,DSN=DTLSR.V800B14.RUNLIB/LICENSE
//DTLSCN DD DISP=SHR,DSN=DTLSR.V800B14.RUNLIB(SIGNON)
//DTLMSG DD DISP=SHR,DSN=**HLQ..DTLMSG
//LOGSCAT DD DISP=SHR,DSN=DTLSR.V800B14.V1.LOGSCAT
//DTLLOG DD SYSOUT=* 
//SYSDUMP DD SYSOUT=* 
//SYSPRINT DD SYSOUT=* 
//REPORT DD SYSOUT=* 
//EXPORT DD SYSOUT=* 
//SYsin DD DISP=SHR,DSN=**LOGDATA```

This JCL adds the log DTLUSR.IDMS.E15SP0.OFF.LOADED.JOURNAL1 for an IDMS Version 15 environment with the CV Name IDMSE150. The log resides on disk storage and is accessed with a LOGSID value of XYLOGSID. The SYSin data is specified as instream data for clarity. However, when running against a CV (DTLIDLL for Local Job mode), the JCL points to the DTLIDLC member.
Configuring and Starting the IDMS Log-Based ECCR

To use IDMS log-based CDC, you must complete a few configuration tasks.

The configuration tasks are:
• Configure the IDMS log-based ECCR options in the RUNLIB(ECCRIDLP) member.
• Populate the PowerExchange Log Catalog.
• Run the DTLUCSR2 utility to enable the ECCR to find SR2 and SR3 records.
• Customize the JCL in the RUNLIB(ECCRIDL) member.

After completing these tasks, you can start the ECCR to begin change data capture.

Configuring IDMS Log-Based ECCR Parameters

Configure IDMS log-based ECCR parameters in the RUNLIB(ECCRIDLP) member to which the DTLCACFG DD statement in the ECCR JCL points.

Based on your input during installation, the z/OS Installation Assistant adds values for some parameters to the ECCRIDLP member. You can accept or change these values.

The ECCRIDLP member can contain the following parameters:

```
LOGSID=logsid
[NO_DATA_WAIT=minutes]
[NO_DATA_WAIT2=seconds]
ECCRNAME=PWXIDLEC
DB_TYPE=IDL
[ABRT_TERMINATION_BLOCK_COUNT=(number|10000)]
[CAPT_STATS=(Y|N)]
[CAPT_STATSIntlVl=minutes]
[CAPT_STATS_TERSE=(Y|N)]
[COLDSTART=(Y|N)]
[ON_SUSPENSION_ERROR_CONTINUE=(N|Y)]
[REFRESH_ALLOWED=(Y|N)]
[RESTART_ADVANCE_ACTIVE=number_of_records]
```

The following table summarizes the ECCR parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOGSID</td>
<td>Required</td>
<td>The LOGSID value that is specified in the DBMOVER configuration file. This parameter is customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>Optional</td>
<td>The number of minutes that the ECCR waits after an end-of-log condition before starting the next log read. If the next log read returns no changes, the NO_DATA_WAIT2 interval takes effect. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NO_DATA_WAIT2</td>
<td>Optional</td>
<td>After the NO_DATA_WAIT interval is no longer in effect, the ECCR waits after an end-of-log condition before trying another log read. The NO_DATA_WAIT2 wait and retry cycle remains in effect as long as no changes are received. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>ECCRNAME</td>
<td>Required</td>
<td>The ECCR name.</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Required</td>
<td>The database type, which must be IDL for IDMS.</td>
</tr>
<tr>
<td>ABRT_TERMINATION_BLOCK_COUNT</td>
<td>Optional</td>
<td>After the IDMS log-based ECCR encounters ABRT records in the IDMS journal that result from an IDMS ROLLBACK or ROLLBACK CONTINUE command, the number of subsequent IDMS journal blocks that the ECCR processes before it passes the job-level ABRT record to the PowerExchange Logger for MVS. By processing these additional blocks, the ECCR can catch any additional updates from the job before the job-level ABRT record is logged. If the ECCR encounters additional updates, the job-level ABRT operation is canceled. If this block count is too high, the ECCR might not resolve outstanding UOWs that contain ABRT records in timely manner, which prevents the journals from being freed. If you use small journals, you can decrease this parameter value to resolve these outstanding UOWs more quickly. Valid values are 100 through 10000. Default is 10000.</td>
</tr>
<tr>
<td>CAPT_STATS</td>
<td>Optional</td>
<td>Controls whether PowerExchange writes ECCR statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the IDMS log-based ECCR finishes processing an IDMS log.</td>
</tr>
<tr>
<td>CAPT_STATS_INTVL</td>
<td>Optional</td>
<td>The interval, in minutes, for which the IDMS log-based ECCR collects and reports the number of inserts, deletes, updates, and commits that were captured from the change stream. The ECCR also reports the current point in the change stream.</td>
</tr>
<tr>
<td>CAPT_STATS_TERSE</td>
<td>Optional</td>
<td>Controls whether the IDMS log-based ECCR prints PWX-06153 statistics messages only for registered sources for which the ECCR captured changes.</td>
</tr>
<tr>
<td>COLDSTART</td>
<td>Optional</td>
<td>Controls whether the IDMS log-based ECCR cold starts or warm starts.</td>
</tr>
<tr>
<td>ON_SUSPENSION_ERROR_CONTINUE</td>
<td>Optional</td>
<td>If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether the ECCR ends or continues when a UOW that contains change records to be discarded or captured started at an invalid point in the change stream relative to the suspension window.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Required or Optional</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>REFRESH_ALLOWED</td>
<td>Optional</td>
<td>Controls whether you can use the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command refreshes the list of registered IDMS records that the ECCR uses for change capture processing.</td>
</tr>
<tr>
<td>RESTART_ADVANCE_ACTIVE</td>
<td>Optional</td>
<td>The number of change records that an active IDMS ECCR processes after a special restart UOW before writing another updated special UOW to the PowerExchange Logger.</td>
</tr>
</tbody>
</table>

**Note**: If a parameter has a default value or is not required, it is marked as optional. A default value is the value that PowerExchange uses if the parameter is not defined. For some parameters, the z/OS Installation Assistant provides recommended values, which you can accept or change.

More detailed parameter descriptions follow.

**CAPT_STATS Parameter**

Controls whether PowerExchange writes ECCR statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the IDMS log-based ECCR finishes processing an IDMS log.

The ECCR issues PWX-06153 messages that report the number of inserts, deletes, and updates that were captured for each registration, grouped by IDMS log. The WTO messages also notify the system operator that a log was closed and provide capture counts.

Regardless of the CAPT_STATS setting, the ECCR always reports the total number of inserts, deletes, updates, and commits across all of the IDMS logs at the end of the ECCR run.

**Related Parameters**: CAPT_STATS_INIVL, CAPT_STATS_TERSE

**Syntax**:

```
CAPT_STATS={N|Y}
```

**Valid Values**:

- **N**: Do not write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing each log.
- **Y**: Write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing each log.

Default is **N**.

**Usage Notes**:

- If you do not set the global CAPT_STATS parameter to **Y**, you can issue the STATISTICS ON command after the ECCR is started to enable statistics reporting for each IDMS log.
- If you also specify the CAPT_STATS_INIVL parameter or run the STATISTICS minutes, the ECCR also reports the total number of inserts, deletes, updates, and commits for each interval.

For more information about the STATISTICS command and its parameters, see the *PowerExchange Command Reference*. 
**CAPT_STATS_INTVL Parameter**

The interval, in minutes, for which the IDMS log-based ECCR collects and reports change capture statistics.

If you specify an interval, the ECCR prints a PWX-06181 message each time the interval elapses. The message reports the total number of inserts, deletes, updates, and commits that the ECCR processed during the interval and the last log position.

You can use this ECCR parameter to print statistics messages at a specific frequency, for example, every 60 minutes.

For the ECCR to print capture statistics, you must set the CAPT_STATS parameter to Y in the RUNLIB(ECCRIDLP) member or run the ECCR STATISTICS ON command.

**Related Parameters**: CAPT_STATS, CAPT_STATS_TERSE

**Syntax**:

```
CAPT_STATS_INTVL=minutes
```

**Value**: For the `minutes` variable, enter a number from 1 through 1440. No default is provided.

**Usage Notes**:

- If you set the CAPT_STATS_INTVL parameter to 0, PowerExchange issues the error message PWX-00967.
- After you start the ECCR, message PWX-07805 identifies the collection interval that is defined.
- If you issue the `STATISTICS minutes` command, the number of minutes that is specified in the command overrides the CAPT_STATS_INTVL value for the duration of the ECCR run.

**CAPT_STATS_TERSE Parameter**

Controls whether the IDMS log-based ECCR prints PWX-06153 messages only for registered sources for which the ECCR captured changes. If no inserts, updates, or deletes occurred on a registered source, the ECCR does not report capture counts for it.

A PWX-06153 message reports the number of inserts, deletes, and updates that were captured for a registered source. The message is printed when the ECCR finishes processing an IDMS log and at the end of the ECCR run.

For the ECCR to print statistics, you must set the CAPT_STATS=Y parameter in the RUNLIB(ECCRIDLP) member or run the ECCR STATISTICS ON command.

**Related Parameters**: CAPT_STATS, CAPT_STATS_INTVL

**Syntax**:

```
CAPT_STATS_TERSE={Y|N}
```

**Valid Values**:

- `Y`. Print statistics only for registered sources with change activity.
- `Y`. Print statistics only for the registered sources for which the ECCR captured changes.

Default is N.

**Usage Notes**:

- If you set the CAPT_STATS_TERSE parameter to N and then issue the STATISTICS SINCE TERSE command, the TERSE option in the command overrides the CAPT_STATS_TERSE setting for the SINCE period. PWX-06153 messages are then printed only for registered sources for which changes were captured.
COLDSTART Parameter

Controls whether the IDMS log-based ECCR cold starts or warm starts.

When the ECCR cold starts, it begins reading change records from the IDMS logs that are recorded at the beginning of the PowerExchange Log Catalog (LOGSCAT). When the ECCR warm starts, it resumes reading change records from where it last left off.

Syntax:

```
COLDSTART={N|Y}
```

Valid Values:

- **N**: Warm starts the ECCR.
- **Y**: Cold starts the ECCR.

Default is N.

Usage Notes: If you use a PowerExchange Logger to which the ECCR has not previously connected, or if you change the ECCRNAME value in the RUNLIB(ECCRIDLP) options member, the ECCR automatically cold starts, regardless of the COLDSTART setting. In these situations, you cannot warm start.

If you clear the LOGSCAT, you must set COLDSTART to Y to clear the restart information and cold start.

DB_TYPE Parameter

The database type.

Syntax:

```
DB_TYPE=IDL
```

Value: This value must be "IDL" for the IDMS log-based ECCR.

ECCRNAME Parameter

A name for the IDMS log-based ECCR.

Syntax:

```
ECCRNAME=ecrname
```

Value: For the *ecrname* variable, enter an alphanumeric string from 1 to 8 characters long.

No default. However, the z/OS Installation Assistant generates an ECCR name that begins with the PowerExchange Agent / Logger Prefix value followed by IDLEC, for example, PWXIDLEC.

Usage Notes:

- The IDMS log-based ECCR uses the ECCRNAME value for the following purposes:
  - The ECCR name for connecting 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
- This name must be unique within a PowerExchange Logger group.
- If you change the ECCRNAME value, the ECCR cannot warm start from its last position in the change stream. You must cold start the ECCR. Also, in-flight UOWs might occur in the PowerExchange Logger log files. To clean up in-flight UOWs, use the PowerExchange Logger RESOLVE_INDOUBT command.
Informatica recommends that you use the same value for both the ECCRNAME parameter and the IDMS log-based ECCR started task name or job name. This practice allows you to easily identify the IDMS log-based ECCR when reviewing messages and data from the PowerExchange Logger.

**LOGSID Parameter**

The LOGSID value that is specified in the DBMOVER configuration file.

**Syntax:**

```
LOGSID=logsid
```

**Value:** For the `logsid` variable, enter the LOGSID value that is specified in the DBMOVER configuration file. This value indicates the location of the IDMS logs and the PowerExchange Log Catalog.

**NO_DATA_WAIT Parameter**

The number of minutes that the IDMS log-based ECCR waits after an end-of-log condition before it starts the next log read operation.

During the next log read, if the ECCR reaches another end-of-log condition without finding new changes, the NO_DATA_WAIT2 interval takes effect.

**Syntax:**

```
NO_DATA_WAIT={number|60}
```

**Valid Values:**

- 0. The ECCR shuts down when no more logs are available to process.
- A number greater than 0. The ECCR waits the specified number of minutes for more logs or changes before shutting down.

Default is 60.

**NO_DATA_WAIT2 Parameter**

After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the IDMS log-based ECCR waits after an end-of-log condition before starting another log read.

During a read operation, if the ECCR captures changes, the NO_DATA_WAIT interval takes effect again. If the ECCR does not capture changes, it waits for the NO_DATA_WAIT2 interval and then tries the read again. The ECCR continues to wait for the NO_DATA_WAIT2 interval and retry the read on an ongoing basis, as long as no changes are available.

To determine if new log data sets have been registered, the ECCR reads the Log Catalog.

**Syntax:**

```
NO_DATA_WAIT2={number|600}
```

**Value:** For the `number` variable, enter a number greater than 0.

The z/OS Installation Assistant enters 999 for this parameter in the ECCR configuration member unless you specify another value. If this parameter is not defined, the default of 600 is used.
ON_Suspension_Error_Continue Parameter

Optional. If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether
the IDMS log-based ECCR ends or continues when a UOW that contains change records to be discarded or
captured started at an invalid point in the change stream relative to the suspension window.

Syntax:

ON_Suspension_Error_Continue={N|Y}

Valid Values:

• **N.** The ECCR issues an error message and ends.
• **Y.** The ECCR issues a warning and continues processing.

Default is **N.**

Usage Notes: If you use the PWXUCREG utility, this parameter controls whether the ECCR ends or continues in
the following situations:

• When discarding change records for a suspended registrations, the ECCR determines that the associated
  UOW started before the beginning of the suspension window.
• When capturing change records for an activated registration, the ECCR determines that the associated
  UOW started before the end of the suspension window.

The suspension window is the time period between the suspension timestamp and reactivation timestamp.
For more information about the PWXUCREG utility, see the *PowerExchange Utilities Guide.*

REFRESH_ALLOWED Parameter

Controls whether PowerExchange users can issue the ECCR REFRESH command. This command refreshes
the list of IDMS records with active capture registrations that the IDMS log-based ECCR uses to capture change data.

When this parameter is set to **Y,** users can issue the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command updates the list of registered sources that the ECCR uses, without shutting down and restarting the ECCR.

Syntax:

REFRESH_ALLOWED={N|Y}

Valid Values:

• **N.** Do not allow users to issue the REFRESH command. This option is intended for users of
  PowerExchange versions earlier than 9.5.0, when the REFRESH command was not available. This option
  maintains the previous behavior, which requires a restart of the ECCR after registration changes.
• **Y.** Allow users to issue the REFRESH command.

Default is **N.**

RESTART_ADVANCE_ACTIVE Parameter

The number of change records that an active IDMS log-based ECCR processes after a special restart UOW,
before it writes another updated special UOW to the PowerExchange Logger.

This value can affect how far back the PowerExchange Logger searches for the restart point when the ECCR is restarted.
Configure the JCL for the IDMS log-based ECCR before each NO_DATA_WAIT2 cycle.

**Valid Values:** Enter a number from 1 to 10000. Default is 10000.

**Usage Notes:** When the ECCR is inactive and waiting for work, PowerExchange updates the special UOW before each NO_DATA_WAIT2 cycle.

### Configuring the IDMS Log-Based ECCR JCL

Configure the JCL for the IDMS log-based ECCR.

Use the following sample JCL in the RUNLIB(ECCRIDL) member:

```plaintext
//*********************************************************************************/
/*  RUN DETAIL IDMS LOG BASED ECCR                                           */
/*                                                                                */
//*********************************************************************************/
//ECCRAD1  EXEC  PGM=DTLCCIDL,REGION=50M
//STEPLIB  DD  DISP=SHR,DSN=&HLQ..LOADLIB
//      DD  DISP=SHR,DSN=&HLQ..LOAD
//EDMPARMS DD  DISP=SHR,DSN=&HLQ..&LOGGER..USERLIB
//DTCFG    DD  DISP=SHR,DSN=&RUNLIB(DBMOVER)
//DCCFG    DD  DISP=SHR,DSN=&RUNLIB(ECCRIDLP)
//SYSDM    DD  DISP=SHR,DSN=&RUNLIB(MYDML)
//DTLMPR   DD  DISP=SHR,DSN=&HLQ..CCT
//DTLMSG   DD  DISP=SHR,DSN=&HLQ..DTLMSG
/* IF USING MESSAGE OVERRIDE THEN CUSTOMIZE BELOW */
/* DTLMSGO DD DISP=SHR,DSN=&RUNLIB(DTLMSGO) */
/*                                                                                */
//*********************************************************************************/
/* FOLLOWING FILE CONCATENATION POINTS TO THE RESULTS FILES CREATED */
/* BY THE DTLUCSR2 UTILITY AND ALLOWS BUILDING OF SR2 RUNTIME TABLE. */
/*                                                                                */
//SR2INPUT DD  DISP=SHR,DSN=&HLQ..SR2TOTAL
//      DD  DISP=SHR,DSN=&HLQ..SR2OUT
//DTLOG    DD  SYSOUT=* 
//DTLOG1   DD  SYSOUT=* 
//DPRINT   DD  SYSOUT=* 
//DDRUCK   DD  SYSOUT=* 
//SYSSDUMP DD  SYSOUT=* 
//SYSOUT   DD  SYSOUT=* 
//SYSPRINT DD  SYSOUT=* 
//EDMSG    DD  SYSOUT=* 
//CEDUMP   DD  SYSOUT=* 
```

**Note:** PowerExchange inserts values for the &HLQ and &LOGGER variables based on information that you provide in the z/OS Installation Assistant.
The following table describes the PowerExchange-related JCL statements for the ECCR startup PROC:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXEC</td>
<td>Specifies the ECCR program, DTLCCIDL.</td>
</tr>
</tbody>
</table>
| STEPLIB DD  | Specifies the PowerExchange LOADLIB and LOAD libraries. If you capture change data from IDMS records that use Presspack compression, you must also specify the IDMS runtime libraries so the ECCR can use Presspack decompression during change capture. Specify the IDMS libraries as follows:  
  - If the ECCR runs APF-authorized, copy the standard IDMS runtime libraries, including the libraries that contain the DCTs and a valid DMCL, and APF-authorize the copies. Then specify these copies in the STEPLIB concatenation.  
  - If the ECCR does not run APF-authorized, specify the standard IDMS runtime libraries in the STEPLIB concatenation. You do not need to copy the libraries. |
| EDMPARMS DD | Specifies the name of the user library that contains the EDMSDIR default options module that is associated with the PowerExchange Logger. 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 get the configuration options. |
| DTLCFG DD   | Specifies the DBMOVER configuration file for PowerExchange, which contains some parameters applicable to the IDMS log-based ECCR. |
| DTLKEY DD   | Specifies the PowerExchange license key file.                                                                                                    |
| DTLCACFG DD | Points to the RUNLIB(ECRIDL0) member that contains IDMS log-based ECCR options.                                                                    |
| SYSIDMS DD  | Include this statement only if you capture change data from IDMS records that use Presspack compression and you do not use the default DMCL named "IDMSDMCL." This statement either points to a data set that contains your DMCL statement or specifies the DMCL inline. Use the following syntax to specify the DMCL inline:  
  /SYSIDMS DD *  
  DMCL=name  
  /*  
  Where name is a DMCL name up to eight characters in length. |
| DTLAMCPR DD | Specifies the data set that contains the capture registrations.                                                                                   |
| DTLMMSG DD  | Specifies the output data set for PowerExchange messages.                                                                                         |
| SR2INPUT DD | Specifies the DTLUCSR2 utility result files. These files contain information that is used to generate the SR2-SR3 internal table.                 |
| DTLLOG DD and DTLLOG01 DD | Specifies the output data sets for ECCR capture statistics.                                                                                     |
| EDMMSG DD   | Specifies the output data set for IDMS log-based ECCR messages.                                                                                   |

**Detecting Matching SR2 and SR3 Records for ECCR Capture of Relocated Records**

For the IDMS log-based ECCR to capture changes made to a relocated SR3 record, it must find the matching SR2 that contains the original record ID. The ECCR uses this record ID to determine if the changes are of CDC interest. To enable the ECCR to find the SR2 record, run the DTLUCSR2 utility. The utility records the pairs of
matching SR2 and SR3 records in an internal table. The ECCR can then perform a lookup on the table with an SR3 database key to find the matching SR2 record that contains the original record ID.

Run the DTLUCSR2 utility before you start the ECCR for the first time and after events that tend to relocate records. For example, run the utility after the following events:

- An IDMS REORG operation
- An IDMS dictionary migration utility (RHDCMIG1 and RHDCMIG2) run
- An alter table operation that adds one or more columns, or any other schema change that can increase the record size
- The following PowerExchange program logic errors, which are issued for an after image (AFTR) or before image (BFOR):

  PWX-00999 Program logic error. Prog="program". Line=line_number. Pl=UOW - SR3 AFTR  
  hex_SR3_database_key, not found in hash table". P2=1
  PWX-00999 Program logic error. Prog="program". Line=line_number. Pl=UOW - SR3 BFOR  
  hex_SR3_database_key, not found in hash table". P2=1

After you run the utility, restart the ECCR so that it can detect the SR2 and SR3 pairs that the utility recorded.

Running the DTLUCSR2 Utility

Run the DTLUCSR2 utility before you run the IDMS log-based ECCR the first time and after any event that tends to create SR2 and SR3 records.

Before you start the utility, ensure that you added the SR2INPUT DD statement to the IDMS log-based ECCR JCL. This DD statement points to the utility result files that contain information for building the SR2-SR3 internal table. For more information, see the PowerExchange CDC Guide for z/OS.

1. Edit the DTLICSRI member in the RUNLIB library.

   For each database with source tables to be registered for change capture, customize the following sample statements:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DD_NAME</td>
<td>The DDNAME to be added to the DTLUCSR2 JCL. This name does not have to match a DD name from an IDMS region, but it must exactly match the DD name in the 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 zero, you must specify the PAGE_GROUP number.</td>
</tr>
<tr>
<td>RADIX</td>
<td>If you want to use a RADIX value other than the default of 8, enter a value from 2 to 12.</td>
</tr>
</tbody>
</table>

Note: DTLUCSR2 writes control information to the SR2TOTAL file and SR2/SR3 link information to the SR2OUT file. These files are created with default information at installation time. You might need to change the file sizes, depending on the number of SR3 records.

2. Add DD cards to the DTLUCSR2 JCL that match the DD names in the DTLICSRI parameter file. The DD cards point to the relevant IDMS data set names.

3. Run the JCL in RUNLIB member DTLUCSR2.
Starting the IDMS Log-Based ECCR

You can run the IDMS log-based ECCR as a started task or batch job. Usually, the ECCR runs as a started task. Whenever you create or change IDMS capture registrations, you must restart the ECCR to activate those registrations.

Before you start the ECCR, verify that you completed the following tasks:

- Configure the IDMS log-based ECCR options.
- Configure the IDMS log-based ECCR JCL.
- Activate the PowerExchange Agent, Listener, and Logger.
- Create capture registrations for the IDMS sources and activate the registrations.
- Create and populate the PowerExchange Log Catalog for IDMS logs.

1. To start the ECCR as a started task, use the MVS START command:
   
   ```
   S eccr_task_name
   ```
   
   If you set the COLDSTART option to Y in the ECCRIDLP options member, the ECCR cold starts.
   
   If you set the COLDSTART option to N and previously ran the ECCR, the ECCR warm starts from where it left off.

2. Verify that all of the IDMS logs of interest for CDC processing have been added to the PowerExchange Log Catalog.

   When the IDMS log-based ECCR is running, it regularly checks whether logs have been added to the PowerExchange Log Catalog for capture processing. If logs have been added, the ECCR captures the change data from the logs and sends the data to the PowerExchange Logger.

Managing IDMS Log-Based CDC

Occasionally, you might need to alter the Log Catalog or recover change capture processing after IDMS ECCR failures or IDMS restore operations.

Adding an IDMS Capture Registration

You might need to add a capture registration for a new or existing IDMS record from which you want to start capturing change data. In this case, you can use the REFRESH command to refresh the list of registered IDMS records for the IDMS log-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(ECCRIDLP) member to which the DTLCACFG DD statement in the ECCR JCL points.

Enter the context of your task here (optional).

1. If you need to begin capturing changes for the new registration from a specific point, stop any change activity on the source record.

2. In the PowerExchange Navigator, open the capture registration and set the Status field to Active.

3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.

4. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:

   ```
   F eccr_task_name,REFRESH
   ```

   The newly registered source is added to the list of registered sources for the ECCR.
5. Enable change activity on the source to resume.
6. If you use PowerExchange Condense, restart it.

Deleting an IDMS Capture Registration

You might need to delete a capture registration that has been used for change capture processing. In this case, you can use the REFRESH command to refresh the list of registered IDMS records for the IDMS log-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(ECCRIDLP) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. Stop applications and other activities that update the source record that is associated with the registration that you are deleting.
2. Ensure that the ECCR has processed all of the IDMS logs that contain changes for the source that is associated with the registration that you are deleting. Also ensure that the source data has been extracted and applied to the target. Then stop all workflows that extract change data for the source.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. In the PowerExchange Navigator, open the capture registration and set the Status field to History. Then delete the registration.
5. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:
   
   F eccr_task_name,REFRESH

6. Enable change activity on the source to resume.
7. If you use PowerExchange Condense, restart it.
8. Restart extraction processing.

Suspending Change Capture for Registered IDMS Sources Temporarily

Use this task flow to suspend change capture processing for registered IDMS log-based CDC sources temporarily.

You perform some tasks with the PWXUCREG utility and other tasks outside of the utility on the z/OS system.

Before you begin, ensure that the REFRESH_ALLOWED=Y parameter is specified in the RUNLIB(ECCRIDLP) member to which the DTLCACFG DD statement in the ECCR JCL points. You must have the authority to issue a REFRESH command after each registration status change.

1. Stop database activity for the registered source or sources for which you want to suspend capture registrations.
2. To suspend the capture registrations, use the PWXUCREG utility to issue the SUSPEND_REGISTRATION command.

   The suspension window opens. The utility sets the suspension timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLOG log to report the registration status change.

   For each suspended registration, the PowerExchange Navigator Resource Inspector displays Suspended in the Status field and the suspension timestamp in the Suspend Time field. The Suspend Time value is not adjusted for the local time.
3. Enter the ECCR REFRESH command with the MVS MODIFY (F) command:
   
   F eccr_task_name,REFRESH
   
   The ECCR becomes aware of the registration status change and suspension timestamp. When the ECCR
   encounters the first change record to discard, it issues message PWX-07752. The ECCR discards change
   records that have a timestamp later than the suspension timestamp.

4. Run the jobs or processes that generate the changes that you do not want to capture for the source or
   sources that are associated with the suspended registrations.

5. After the jobs or processes complete, use the PWXUCREG utility to issue the ACTIVATE_REGISTRATION
   command to reactivate the capture registrations.

   The suspension window closes. The utility sets the activation timestamp to the current system time
   without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLLOG
   log to report the registration status change.

   For each reactivated registration, the PowerExchange Navigator Resource Inspector displays Active in
   the Status field and the activation timestamp in the Active Time field. The Active Time value is not
   adjusted for the local time.

6. Enter the ECCR REFRESH command with the MVS MODIFY (F) command again.

    The ECCR becomes aware of the registration status change and activation timestamp.

7. Enable database activity to resume on the registered source or sources.

    The ECCR starts capturing change records that have timestamps later than the activation timestamp.
    The ECCR issues message PWX-07753 when it encounters the first change record in the change stream
    after the end of the suspension window.

    Note: You can automate this processing if appropriate for your environment.

### Changing an IDMS Source Schema

If you make schema changes to an IDMS source, use this procedure to retain access to historically captured
data while capturing data in the new format.

1. Stop all update activity against the IDMS database.
2. Ensure that PowerExchange processed all changes that occurred under the old schema.
3. Change the IDMS schema.
4. Create a new PowerExchange capture registration that reflects the schema changes.
5. Ensure that the updated schemas are in place in the PowerExchange copies of the IDMS libraries.
6. Allow update activity to the IDMS database to resume.

### Manipulating the Log Catalog

During normal IDMS log processing, PowerExchange updates the Log Catalog, LOGSCAT, to add the next
available log by using the PowerExchange DTLULCAT and DTLULOGC log catalog utilities.

If you need to add, change, or remove log entries in the Log Catalog, run the DTLULOGC utility standalone
with hand-coded input. Use the sample DTLULOGC JCL in the RUNLIB library.

Customize the DTLULOGC JCL to perform any of the following tasks:

- Add a LOGSID instance.
- Add a log entry to the Log Catalog.
• Update a log entry.
• Delete a log entry.
• Export a log entry to another data set for offloading.

Then specify the DTLULOGC file as input in the SYSIN DD card.
The following table describes the keywords and parameters that you can code in the 80-byte DTLULOGC JCL file:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD_INSTANCE</td>
<td>Add a LOGSID instance to the catalog. Each LOGSID requires an instance to be added to the log catalog.</td>
<td>- INSTANCE_IDENTIFIER. A LOGSID value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- VERSION. A version number of the entry.</td>
</tr>
<tr>
<td>ADD_ENTRY</td>
<td>Adds a specific log to the log catalog.</td>
<td>- BLOCK_SIZE. The block size of the log. Required if the logs are to be shipped to another platform.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ENTRY_NUMBER. A sequential number, which should be incremented by 1 for each new log added to the log catalog.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FILE_TYPE. One of the following values:</td>
</tr>
<tr>
<td></td>
<td></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></td>
<td>- FIRST_RECORD_SEQUENCE_NUMBER. The sequence number of the first record in the block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- FIRST_RECORD_TIME_STAMP. The time stamp of the first record in the block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- IDMS_VERSION. The version number of IDMS. Specified as an integer.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- INSTANCE_IDENTIFIER. A LOGSID value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LAST_RECORD_IDENTIFIER. The record ID of the last record in the block or zeros if a non-data record.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LAST_RECORD_OFFSET. The offset of last valid offset in the block.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOG_DATA_TYPE. &quot;IDL&quot; for MVS IDMS log data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- LOG_FILE_NAME. The name of IDMS log file.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- MEDIA_CONTENT. One of the following values:</td>
</tr>
<tr>
<td></td>
<td></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></td>
<td>- MEDIA_TYPE. One of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- D. Disk.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- T. Tape.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- NUMBER_OF_BLOCKS. The number of blocks in the log.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- SERVICE. The CV name or Local Mode job name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- STATUS. One of the following values:</td>
</tr>
<tr>
<td></td>
<td></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></td>
<td>- ENTRY_TYPE. One of the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1. File entry.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2. Reserved for future use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- VERSION. The version number of the entry.</td>
</tr>
<tr>
<td>UPDATE_ENTRY</td>
<td>Updates a log entry.</td>
<td>Valid parameters are those listed for ADD_ENTRY.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Use the following parameters to identify the entry:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- INSTANCE_IDENTIFIER</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- ENTRY_NUMBER</td>
</tr>
<tr>
<td>Keyword</td>
<td>Description</td>
<td>Parameters</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>DELETE_ENTRY</td>
<td>Deletes the last log for the specified INSTANCE_IDENTIFIER.</td>
<td>INSTANCE_IDENTIFIER</td>
</tr>
<tr>
<td>REPORT_INSTANCE</td>
<td>Lists catalog entries for the specified INSTANCE_IDENTIFIER.</td>
<td>INSTANCE_IDENTIFIER</td>
</tr>
<tr>
<td>EXPORT_INSTANCE</td>
<td>Exports all information for a specified INSTANCE_IDENTIFIER to a file.</td>
<td>INSTANCE_IDENTIFIER</td>
</tr>
</tbody>
</table>

**Note:** Keywords are separated by a semicolon (;). Parameters are separated by a comma (,).

The following sample input adds two instances (LOGSIID), adds log entries, deletes a log entry, reports instance LOGSIIDA, exports instance LOGSIIDA to a file (dtluge.txt), and deletes instance LOGSIIDA:

```
ADD INSTANCE INSTANCE_IDENTIFIER=LOGSIIDA, VERSION=224;
ADD_ENTRY INSTANCE_IDENTIFIER=LOGSIIDA, INSTANCE_NUMBER=777, VERSION=0,
ENTRY_TYPE=1, STATUS=A, LOG_DATA_TYPE=IOD, IDMS_VERSION=15,
FILE_TYPE=C, MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE=IDMSE150,
LOG_FILE_NAME=XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX, 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=LOGSIIDA, INSTANCE_NUMBER=778, VERSION=0,
ENTRY_TYPE=1, STATUS=A, LOG_DATA_TYPE=IOD, IDMS_VERSION=15, FILE_TYPE=C,
MEDIA_TYPE=D, MEDIA_CONTENT=B1, SERVICE=IDMSE150,
LOG_FILE_NAME=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Starting the ECCR After Clearing the Log Catalog

If you clear or reinitialize the PowerExchange Log Catalog, hlq.LOGSCAT, you must cold start the IDMS log-based ECCR for the ECCR to correctly process IDMS logs or journals that are subsequently added to the Log Catalog.

The cold start causes the ECCR to ignore obsolete log position information from the PowerExchange Logger, which includes Log Catalog information that was cleared.

Recovering from Failures

Use the following guidelines to recover from certain events that disrupt change data capture.

These events include:

- IDMS log-based ECCR abnormal ends that occur because of an ECCR failure or a PowerExchange Logger stoppage
- Restores of the IDMS database

Recovering the IDMS Log-Based ECCR

You must recover the IDMS log-based ECCR when the IDMS log-based ECCR fails or 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, complete the following steps:

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 warm starts if warm start data is available from the Agent or Logger. It restarts at the correct point.

If warm start data is not available, the ECCR issues a prompt for a cold start. Use the same ECCRNAME parameter in your ECCRIDLP parameter file that you used for the ECCR that abended.

Recovering after IDMS Restores or Reruns

Use the PowerExchange Logger to log all PowerExchange activity.

You do not normally restore the PowerExchange Logger.

When you restore the source database because of application failures, you typically reset the application extraction start points to the relevant point.

To identify the correct point to start, use the Event Marker utility, EDMXLUTL, to put markers into the Logger on a regular basis. When you add these markers, they appear in the PowerExchange log.
IMS Log-Based Change Data Capture

This chapter includes the following topics:

- **IMS CDC Overview, 258**
- **Configuring IMS for Log-Based CDC, 261**
- **Configuring the IMS Log-Based ECCR, 262**
- **Managing IMS Log-Based CDC, 274**

IMS CDC Overview

PowerExchange change data capture (CDC) for IMS captures changes made to IMS databases and logs those changes to PowerExchange Logger for MVS log files.

You can use PowerCenter CDC sessions to extract the captured change data from the PowerExchange Logger log files or from PowerExchange Condense condense files and apply that data to one or more target databases.

PowerExchange provides the following alternative methods for performing IMS CDC:

- **Synchronous IMS CDC.** Captures 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.** Asynchronously captures changes by reading them from the IMS archive logs and logging them to the PowerExchange Logger. The IMS log-based ECCR runs in a separate address space as a started task or a batch job.

The following table compares IMS synchronous CDC and the IMS log-based CDC methods:

<table>
<thead>
<tr>
<th>Feature</th>
<th>IMS Synchronous CDC</th>
<th>IMS Log-Based CDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does real-time capture of change data.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reads IMS archive logs to capture IMS change data asynchronously.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>PowerExchange IMS interface modules install into the IMS RESLIB.</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>
The IMS log-based ECCR asynchronously captures change data from closed IMS archive logs (SLDSs) for a registered IMS database.

The ECCR passes the changes to the PowerExchange Logger for MVS. After the PowerExchange Logger logs the changes to its log files, the changes are available for extraction processing. Based on specific parameters, the ECCR periodically inspects the IMS RECON data sets for new archive logs to process.

The IMS log-based ECCR runs in a separate address space either continuously or in batch mode. Because the ECCR runs within a multitasking environment, data capture, processing, and delivery can proceed in parallel.

During initialization, the ECCR reads capture registration information from the CCT data set to determine the segments in an IMS database that are registered for change capture. For each source database, you must complete the following tasks in the IMS environment:

- Change the DBD to include the EXIT statement.
- Register the database in DBRC, if it is not already registered.

Because the IMS log-based ECCR reads closed IMS archive logs, a delay occurs between the time a change is made and the time the change is captured. The length of this delay depends on the following factors:

- How quickly IMS archives the active logs after a change is made.
- How frequently the IMS log-based ECCR checks for new archive logs.
PowerExchange transforms the captured changes into a PowerExchange internal format, which is essentially the same for all data sources.

If the IMS log-based change capture stops for any reason and updates to the IMS database continue, PowerExchange can resume change data capture from where it left off after you correct the problem and start change capture again. No changes are lost.

**Note:** The IMS log-based ECCR can capture change data from complex tables. A complex table includes records for multiple segments in the IMS database hierarchy. If you need to capture change data from a complex table, do not use IMS field (FLD) calls to make changes to a low-level segment in the complex table. In this case, IBM IMS cannot provide the data for the parent segments. Conversely, if you need to allow FLD calls against the source database, Informatica recommends that you do not define complex tables as sources. If you must use complex tables with FLD calls, contact Informatica Global Customer Support to determine the best strategy for getting change data from the parent segments.

**IMS Log-Based ECCR Processing Phases**

On startup, the IMS log-based ECCR goes through the following processing phases:

- Initialization
- Reading and processing blocks of change data
- Waiting for data

**Initialization**

During initialization, the IMS log-based ECCR performs the following tasks:

- Checks and loads the capture registrations.
- Determines which of the RECON data sets that are specified in the ECCR input parameters is the current data set.
- Uses the current RECON data set to determine which log data sets to process and the order in which to process them.
- 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**

After the ECCR processes the IMS archive logs up to the point that was the current point when the ECCR run began, it receives a "no more log data" return code. The IMS log-based ECCR then waits for more change data to process.

The ECCR waits until one of the following events occur:

- The wait interval that is specified in the NO_DATA_WAIT2 parameter elapses.
- An interrupt from an event.
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.

RELATED TOPICS:
- "Monitoring the PowerExchange Logger for MVS" on page 72

Configuring IMS for Log-Based CDC

Before you can use IMS log-based CDC, verify that the following conditions are true 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 Statement

For IMS to write log records from which the PowerExchange IMS log-based ECCR can capture data, you must specify the IMS Data Capture EXIT parameter in the DBD statement that the DBDGEN utility uses.

The EXIT parameter causes IMS to create log record type x'99' for data that IMS logs for a segment. The IMS log-based ECCR reads the x'99' records to capture change data.

The following example DBD statement includes the EXIT parameter:

```
DBD NAME=DBFSAMD3,ACCESS=DED,B, RMNAME=DBFHDC40,
EXIT=(*,KEY,PATH,(CASCADE,KEY,PATH),LOG)
```

Use of the EXIT parameter increases the number of x'99' log records for the IMS online and batch regions. If you need to reduce the number of x'99' records, edit the EXIT parameter to change PATH to NOPATH. PATH causes IMS to log the entire hierarchical path for the segment, whereas NOPATH causes IMS to log only the segment. However, you can use NOPATH only if the PowerExchange capture registrations each represent a single segment.
IMS 12 introduced some additional EXIT options for Fast Path databases, which might interfere with ECCR change data capture. If your source is an IMS 12 or later Fast Path database, Informatica recommends that you do not specify the following options in the EXIT parameter:

- **NOBEFORE**. No before data is included in X’99’ log records for REPL calls. As a result, the ECCR cannot capture the IMS REPL operations.
- **NODLET**. No X’99’ log records are written for DLET calls. As a result, the ECCR cannot capture the delete operations.
- **NODLET with CASCADE**. If children of a segment are registered for change capture and you delete the parent segment, the ECCR does not capture the delete for the segment but does capture the deletes for the children.

For more information about the EXIT parameter, see the IBM IMS reference information for the system utility DBDGEN.

### Configuring the IMS Log-Based ECCR

Before you start the IMS log-based ECCR, complete the following configuration tasks:

- Configure the ECCR parameters.
- APF-authorize the libraries in the STEPLIB concatenation.
- Configure the IMS log-based ECCR JCL.
- Create at least one capture registration for the IMS source. For more information, see the *PowerExchange Navigator User Guide*.

### IMS Log-Based ECCR Programs

PowerExchange provides a separate IMS log-based ECCR program for each supported IMS version. It also provides an ECCR program that works the DBRC API and can be used with IMS 10 or later.

You specify the ECCR program in the ECCRIMS EXEC statement of the ECCR JCL. The z/OS Installation Assistant uncomments the correct ECCRIMS EXEC statement based on your input when it generates the JCL. To change the ECCR program, edit the JCL.

The following table describes the ECCR programs that are available for each supported IMS version:

<table>
<thead>
<tr>
<th>IMS Version</th>
<th>ECCR Program</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>DTLCCIM8</td>
<td>DTLCCIM8 works with IMS 8.1 only.</td>
</tr>
<tr>
<td>9.1</td>
<td>DTLCCIM9</td>
<td>DTLCCIM9 works with IMS 9.1 only.</td>
</tr>
<tr>
<td>10</td>
<td>DTLCCIMA or DTLCCIMX</td>
<td>DTLCCIMA works with IMS Version 10 only. DTLCCIMX uses the DBRC API and works with IMS 10 or later.</td>
</tr>
<tr>
<td>11</td>
<td>DTLCCIMB or DTLCCIMX</td>
<td>DTLCCIMB works with IMS Version 11 only. DTLCCIMX uses the DBRC API and works with IMS 10 or later.</td>
</tr>
<tr>
<td>12</td>
<td>DTLCCIMC or DTLCCIMX</td>
<td>DTLCCIMC works with IMS Version 12 only. DTLCCIMX uses the DBRC API and works with IMS 10 or later.</td>
</tr>
</tbody>
</table>
Configuring the IMS Log-Based ECCR Parameters

Configure the IMS log-based ECCR parameters in the RUNLIB(CAPTIMS) member to which the DTLCACFG DD refers.

Based on your input during installation, the z/OS Installation Assistant adds values for some parameters to the CAPTIMS member. You can change these values if necessary.

The CAPTIMS member can contain the following parameters:

- **DBID=**imsregn
- **DB_TYPE=**IMS
- **ECCRNAME=**IMSEC
- **[RECID=AO]**
- **IMSID=(subsystem,dbd,**
- **RECON=(imsrecon1,**
- **imsrecon2,**
- **imsrecon3))**
- **[NO_DATA_WAIT=seconds]**
- **[NO_DATA_WAIT2=seconds]**
- **[BYPASS_VERSION_CHECKING={Y|N}]**
- **[CAPT_STATS={Y|N}]**
- **[CAPT_STATS_INTVL=minutes]**
- **[CAPT_STATS_TERSE={Y|N}]**
- **[COLDSTART={Y|N}]**
- **[ERROR_LOG={ABEND|SKIP|WAIT|WTOR|No response}]**
- **[MSG_LVL=I|1]**
- **[ON_SUSPENSION_ERROR_CONTINUE={N|Y}]**
- **[REFRESH_ALLOWED={N|Y}]**
- **[STARTTIME="Y/MM/DD hh:mm:ss[.nnnnnn]"]**
- **[WRITE_RESTART_SECS=seconds]**

The following table summarizes the IMS log-based ECCR parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DBID</strong></td>
<td>Required</td>
<td>The RECON identifier that is specified in the registration group for the IMS source from which the ECCR captures changes. This parameter is customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td><strong>DB_TYPE</strong></td>
<td>Required</td>
<td>The database type, which must be IMS.</td>
</tr>
<tr>
<td><strong>ECCRNAME</strong></td>
<td>Required</td>
<td>The ECCR name.</td>
</tr>
</tbody>
</table>

**Note:** With the DTLCCIMX program, you do not have to switch to another ECCR program when you upgrade IMS. Also, you can print ECCR capture statistics by using the STATISTICS command and CAPT_STATS=Y parameter. For information about implementing DTLCCIMX, contact Informatica Global Customer Support.

DTLCCIMT uses the DBRC API and works with IMS 10 or later. DTLCCIMX works with IMS Version 13 only. If you use DTLCCIMX with IMS Version 10, you must apply IBM APAR PK50752.
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECID</td>
<td>Optional</td>
<td>A hexadecimal value that corresponds to the record type of user-defined records that the DTLCUIML utility writes to the IMS SLDS. You can use these record IDs to define a start marker for the IMS log-based ECCR in the IMS SLDS.</td>
</tr>
<tr>
<td>IMSID</td>
<td>Required</td>
<td>The IMS subsystem ID, the DBDLIB data set, and the RECON data sets. This parameter is customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>NO_DATA_WAIT</td>
<td>Optional</td>
<td>The number of seconds that the ECCR waits after an end-of-log condition before starting the next log read. During the next read operation, if the ECCR receives another end-of-log condition without having processed new changes, the NO_DATA_WAIT_2 parameter takes effect. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>NO_DATA_WAIT2</td>
<td>Optional</td>
<td>After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the ECCR waits after an end-of-log condition before trying another log read. The NO_DATA_WAIT2 wait and retry cycle remains in effect as long as no changes are received. This parameter can be customized by the z/OS Installation Assistant.</td>
</tr>
<tr>
<td>BYPASS_VERSION_CHECKING</td>
<td>Optional</td>
<td>Controls whether the ECCR checks that the IMS version matches the IMS version of the DBRC RECON data sets.</td>
</tr>
<tr>
<td>CAPT_STATS</td>
<td>Optional</td>
<td>Controls whether PowerExchange writes ECCR statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the IMS log-based ECCR finishes processing a SLDS.</td>
</tr>
<tr>
<td>CAPT_STATS_INTVL</td>
<td>Optional</td>
<td>The interval, in minutes, for which the IMS log-based ECCR collects and reports the number of inserts, deletes, updates, and commits that were captured. The ECCR also reports the log position up to which changes were processed.</td>
</tr>
<tr>
<td>CAPT_STATS_TERSE</td>
<td>Optional</td>
<td>Controls whether the IMS log-based ECCR prints PWX-06153 statistics messages only for registered sources for which the ECCR captured changes.</td>
</tr>
<tr>
<td>COLDSTART</td>
<td>Optional</td>
<td>Controls whether the ECCR cold starts or warm starts.</td>
</tr>
<tr>
<td>ERROR_LOG</td>
<td>Optional</td>
<td>Controls how the ECCR behaves when it encounters an IMS log in the RECON data set that is marked as in error or is otherwise unavailable.</td>
</tr>
<tr>
<td>MSGLEVEL</td>
<td>Optional</td>
<td>Controls whether PowerExchange issues detailed messages that indicate the status of ECCR processing of IMS logs that are recorded in the RECON data sets and that contain CDC statistics.</td>
</tr>
</tbody>
</table>
### Parameter Table

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Required or Optional</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON_SUSPENSION_ERROR_CONTINUE</td>
<td>Optional</td>
<td>If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether the ECCR ends or continues when a UOW that contains change records to be discarded or captured started at an invalid point in the change stream relative to the suspension window.</td>
</tr>
<tr>
<td>REFRESH_ALLOWED</td>
<td>Optional</td>
<td>Controls whether you can use the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command refreshes the list of registered IMS segments that the ECCR uses for change capture processing.</td>
</tr>
<tr>
<td>STARTTIME</td>
<td>Optional</td>
<td>The date and time when the ECCR starts processing change records from IMS logs after a cold start.</td>
</tr>
<tr>
<td>WRITE_RESTART_SECS</td>
<td>Optional</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.</td>
</tr>
</tbody>
</table>

**Note:** If a parameter has a default value or is not required, it is marked as optional. A default value is the value that PowerExchange uses if the parameter is not defined. For some parameters, the z/OS Installation Assistant provides recommended values, which you can accept or change.

More detailed parameter descriptions follow.

#### BYPASS_VERSION_CHECKING Parameter

Controls whether the IMS log-based ECCR checks that the IMS version matches the IMS version of the DBRC RECON data sets.

**Syntax:**

```
BYPASS_VERSION_CHECKING={N|Y}
```

**Valid Values:**

- **N.** The ECCR checks that the IMS version matches the IMS version of the DBRC RECON data sets.
- **Y.** The ECCR bypasses version checking. Enter this value if you plan to upgrade RECON data sets to a later IMS release in preparation for upgrading IMS.

Default is N.

#### CAPT_STATS Parameter

Controls whether PowerExchange writes ECCR statistics messages to the DTLLOG and DTLOUT data sets and WTO messages to the system operator console when the IMS log-based ECCR finishes processing a SLDS.

IMS log-based ECCR statistics reporting is supported only for the ECCR DTLCCIMX program that works with the DBRC API. PowerExchange supplies the DTLCCIMX program for IMS 10 and later.

The ECCR issues PWX-06153 messages that report the number of inserts, deletes, and updates that were captured for each registration, grouped by SLDS. The WTO messages also notify the system operator that a SLDS was closed and provide capture counts.
Regardless of the CAPT_STATS setting, the ECCR always reports the total number of inserts, deletes, updates, and commits across all of the SLDSs at the end of the ECCR run.

Related Parameters: CAPT_STATS_INTVL, CAPT_STATS_TERSE

Syntax:

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

Valid Values:

- **N.** Do not write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing each SLDS.
- **Y.** Write the ECCR capture statistics messages to the DTLLOG and DTLOUT data sets and WTO capture count messages when the ECCR finishes processing each SLDS.

Default is N.

Usage Notes:

- If you do not set the global CAPT_STATS parameter to Y, you can issue to STATISTICS ON command after the ECCR is started to enable statistics reporting for each SLDS.
- If you also specify the CAPT_STATS_INTVL parameter or run the STATISTICS minutes, the ECCR also reports the total number of inserts, deletes, updates, and commits for each interval.

For more information about the STATISTICS command and its parameters, see the PowerExchange Command Reference.

CAPT_STATS_INTVL Parameter

The interval, in minutes, for which the IMS log-based ECCR collects and reports change capture statistics. If you specify an interval, the ECCR prints a PWX-06181 message each time the interval elapses. The message reports the total number of inserts, deletes, updates, and commits that the ECCR processed during the interval and the last log position.

You can use this ECCR parameter to print statistics messages at a specific frequency, for example, every 60 minutes.

For the ECCR to print capture statistics at a specific interval, you must also set the CAPT_STATS parameter to Y in the RUNLIB(CAPTIMS) member or run the ECCR STATISTICS ON command.

Related Parameters: CAPT_STATS, CAPT_STATS_TERSE

Syntax:

\[ \text{CAPT_STATS_INTVL}=\text{minutes} \]

Value: For the minutes variable, enter a number from 1 through 1440. No default is provided.

Usage Notes:

- If you set the CAPT_STATS_INTVL parameter to 0, PowerExchange issues the error message PWX-00967.
- After you start the ECCR, message PWX-07805 identifies the collection interval that is defined.
- If you issue the STATISTICS minutes command, the number of minutes that is specified in the command overrides the CAPT_STATS_INTVL value for the duration of the ECCR run.
**CAPT_STATS_TERSE Parameter**

Controls whether the IMS log-based ECCR prints PWX-06153 messages only for registered sources for which the ECCR captured changes. If no inserts, updates, or deletes occurred on a registered source, the ECCR does not report capture counts for it.

A PWX-06153 message reports the number of inserts, deletes, and updates that were captured for a registered source. The message is printed when the ECCR finishes processing a SLDS and at the end of the ECCR run.

For the ECCR to print statistics, you must set the CAPT_STATS=Y parameter in the RUNLIB(CAPTIMS) member or run the ECCR STATISTICS ON command.

**Related Parameters:** CAPT_STATS, CAPT_STATS_INTVL

**Syntax:**

```
CAPT_STATS_TERSE={N|Y}
```

**Valid Values:**

- **N.** Print statistics for all registered sources, including sources without change activity.
- **Y.** Print statistics only for the registered sources for which the ECCR captured changes.

Default is **N.**

**Usage Notes:**

- If you set the CAPT_STATS_TERSE parameter to **N** and then issue the STATISTICS SINCE TERSE command, the command overrides the CAPT_STATS_TERSE setting for the SINCE period. PWX-06153 messages are then printed only for registered sources for which changes were captured.

**COLDSTART Parameter**

Controls whether the IMS log-based ECCR cold starts or warm starts.

A cold start causes the ECCR to start processing changes with the next IMS log file that is created. A warm start causes the ECCR to resume change processing where it last left off.

**Syntax:**

```
COLDSTART={N|Y}
```

**Valid Values:**

- **N.** Warm starts the ECCR.
- **Y.** Cold starts the ECCR. If you specify **Y**, you can also specify the STARTTIME parameter.

Default is **N.**

**Usage Notes:** The following actions cause the IMS log-based ECCR to cold start, regardless of the COLDSTART setting:

- When you start the ECCR with a PowerExchange Logger to which the ECCR has not previously connected.
- When you change the ECCRNAME value in the **hlq.RUNLIB(CAPTIMS)** member.

**DB_TYPE Parameter**

Required. The database type.

**Syntax:**

```
DB_TYPE=IMS
```
**Value:** The value must be "IMS" for the IMS log-based ECCR.

**DBID Parameter**

Required. A value that matches the RECON identifier in the registration group for the IMS source database from which the IMS log-based ECCR captures changes.

**Syntax:**

```
DBID=recon_id
```

**Value:** For the `recon_id` variable, enter a value that matches both the RECON Identifier value that is specified in the registration group and the first positional parameter in the IMSID statement of the DBMOVER configuration file.

**ECCRNAME Parameter**

Required. A name for the IMS log-based ECCR.

**Syntax:**

```
ECCRNAME=eccr_name
```

**Value:** For the `eccr_name` variable, enter an alphanumeric string from 1 to 8 characters long.

No default. However, the z/OS Installation Assistant generates an ECCR name that begins with the PowerExchange Agent / Logger Prefix value followed by IMSEC, for example, PWXIMSEC.

**Usage Notes:**

- The ECCR uses the ECCRNAME value for the following purposes:
  - To connect to the PowerExchange Logger to write change data
  - As 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 ECCRNAME value must be unique within a PowerExchange Logger group.
- 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.
- If you change the ECCRNAME value, the ECCR cannot warm start from where it last left off.

**ERROR_LOG Parameter**

Controls how the IMS log-based ECCR behaves when it encounters an IMS log that is marked as in error in the RECON data set or is otherwise unavailable.

**Syntax:**

```
ERROR_LOG={ABEND|SKIP|WAIT|WTOR|No_response}
```

**Valid Values:**

- **ABEND**
  
  When the IMS ECCR encounters a log that is marked as in error, it ends and issues a WTO message to the system operator console. The ECCR also issues messages that report the start and stop times of the log in error. The ECCR ends in a manner that enables you to restart it after resolving the log in error.
** SKIP **

The IMS ECCR skips any log that is marked as in error. The ECCR issues messages that indicate which logs have been skipped, including their names and start and stop times.

Use this option carefully. The ECCR might miss changes, which can invalidate the target data.

** WAIT **

When the IMS ECCR encounters a IMS log that is marked as in error, it issues informational messages that indicate the status of the log. Then, the ECCR sleeps. Periodically, the ECCR becomes active, based on the NO_DATA_WAIT2 value, to check the log status. After you resolve the log in error, the ECCR continues processing. Optionally, you can change the status of the log in IMS or remove the log from the RECON data set. If you do so, ensure that no changes are lost.

** WTOR **

Stops the IMS ECCR from continuing and issues a WTOR that asks for which option to use.

** No response **

The IMS ECCR waits on continuing basis. The ECCR issues messages that identify the log that is in error and the reason for the error.

Default is ABEND.

** Usage Notes:**

- Typically, a IMS log is marked as in error when some type of media error occurs, such as an x37 abend, while data is being written to the log.
- After a log has been ignored or skipped, you cannot try to process it again. You must rematerialize the target data.

** IMSID Parameter **

Required. The IMS subsystem ID, the DBDLIB data set name, and the RECON data set names. Defines the IMS subsystem to the IMS log-based ECCR.

** Syntax:**

```plaintext
IMSID=(ims_ssid, dbdlib, RECON=(recon1, recon2, recon3))
```

** Values:** Enter values for all of the positional parameters and options that are represented by the following variables:

- **ims_ssid**
  An IMS subsystem ID. This value can be from one to eight characters in length. Enter a value that matches the IMS SSID value in the IMS data map that you used to register the IMS source.

- **dbdlib**
  The name of the IMS DBDLIB data set that contains the DBD load modules. This value is an alphanumeric string from one to eight characters in length.

- **recon1, recon2, recon3**
  The names of the IMS RECON data sets that the ECCR uses. Enter values for all three parameters. Use a comma to separate the data set names.

The z/OS Installation Assistant can enter these RECON data set names based on your input on the IMS CDC Parameters page of the installation wizard.
MSGLVL Parameter

Controls whether the IMS log-based ECCR issues detailed messages that indicate the status of ECCR processing of IMS logs that are recorded in the RECON data sets and that contain CDC statistics.

Syntax:

\[ \text{MSGLVL} = \{0 | 1\} \]

Valid Values:

- 0. The ECCR does not issue detailed messages.
- 1. The ECCR issues detailed messages.

Default is 0. Recommended value is 1.

NO_DATA_WAIT Parameter

The number of seconds that the IMS log-based ECCR waits after an end-of-log condition before it starts the next log read operation.

During the next read operation, if the ECCR receives another end-of-log condition without having processed new changes, the NO_DATA_WAIT_2 parameter takes effect.

Related Parameters: NO_DATA_WAIT

Syntax:

\[ \text{NO_DATA_WAIT} = \{60 | \text{seconds}\} \]

Value: For the \text{seconds} variable, enter a number from 1 to 99999999.

Default is 60.

NO_DATA_WAIT2 Parameter

After the NO_DATA_WAIT interval is no longer in effect, the number of seconds that the IMS log-based ECCR waits after an end-of-log condition before starting another log read operation.

During a read operation, if the ECCR captures changes, the NO_DATA_WAIT interval takes effect again. If the ECCR does not capture changes, it waits for the NO_DATA_WAIT2 interval and then tries the read again. The ECCR continues to wait for the NO_DATA_WAIT2 interval and retry the read on an ongoing basis, as long as no changes are available.

The ECCR checks the RECON data sets to determine whether a new LOG data set was registered.

Related Parameters: NO_DATA_WAIT

Syntax:

\[ \text{NO_DATA_WAIT2} = \{600 | \text{seconds}\} \]

Value: For the \text{seconds} variable, enter a number from 1 to 99999999.

Default is 600.

ON_SUSPENSION_ERROR_CONTINUE Parameter

Optional. If you use the PWXUCREG utility to suspend and reactivate capture registrations, controls whether the IMS log-based ECCR ends or continues when a UOW that contains change records to be discarded or captured started at an invalid point in the change stream relative to the suspension window.

Enter the syntax information of your reference here (optional).
**Valid Values:**

- **N.** The ECCR issues an error message and ends.
- **Y.** The ECCR issues a warning and continues processing.

Default is **N**.

**Usage Notes:** If you use the PWXUCREG utility, this parameter controls whether the ECCR ends or continues in the following situations:

- When discarding change records for a suspended registrations, the ECCR determines that the associated UOW started before the beginning of the suspension window.
- When capturing change records for an activated registration, the ECCR determines that the associated UOW started before the end of the suspension window.

The suspension window is the time period between the suspension timestamp and reactivation timestamp. For more information about the PWXUCREG utility, see the *PowerExchange Utilities Guide*.

**RECID Parameter**

A hexadecimal value that corresponds to the record type of user-defined records that the DTLCUIML utility writes to the IMS SLDS. You can use these record IDs to define a start marker for the IMS log-based ECCR in the IMS SLDS.

The ECCR looks for the markers when reading IMS SLDS. When the ECCR encounters a marker, it triggers a message in the PowerExchangeLogger that provides restart and sequence tokens for registration tags.

**Syntax:**

```plaintext
RECID=\{nn|A0\}
```

**Value:** For the **nn** variable, enter a hexadecimal value from A0 to FF that is unique in your PowerExchange environment.

Default is **A0**.

**REFRESH_ALLOWED Parameter**

Controls whether PowerExchange users can issue the ECCR REFRESH command. This command refreshes the list of IMS segments with active capture registrations that the IMS log-based ECCR uses to capture change data.

When this parameter is set to **Y**, users can issue the REFRESH command after adding or deleting capture registrations or after suspending or reactivating capture registrations with the PWXUCREG utility. The REFRESH command updates the list of registered sources that the ECCR uses, without shutting down and restarting the ECCR.

**Syntax:**

```plaintext
REFRESH_ALLOWED=\{N|Y\}
```

**Valid Values:**

- **N.** Do not allow users to issue the REFRESH command. This option is intended for users of PowerExchange versions earlier than 9.5.0, when the REFRESH command was not available. This option maintains the previous behavior, which requires a restart of the ECCR after registration changes.
- **Y.** Allow users to issue the REFRESH command.

Default is **N**.
STARTTIME Parameter

The date and time in the IMS logs where the IMS log-based ECCR starts processing change records after a cold start.

For the ECCR to use this parameter, you must also set the COLDSTART parameter to Y.

Syntax:

```
STARTTIME="YY/MM/DD hh:mm:ss.nnnnnn"
```

Value: In the syntax, the following variables are:

- **YY**: A two-digit year value from 00 to 99.
- **MM**: A two-digit month value from 01 to 12.
- **DD**: A two-digit day value from 01 to 31.
- **hh**: A two-digit hour value from 01 to 23.
- **mm**: A two-digit minutes value from 00 to 59.
- **ss**: A two-digit seconds value from 00 to 59.
- **nnnnnn**: Optional. A subsecond value of up to six digits.

Examples:

```
STARTTIME="10/12/31 23:59:59"
STARTTIME="10/12/31 23:59:59.123456"
```

WRITE_RESTART_SECS Parameter

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 for the restart point when the ECCR is restarted.

Syntax:

```
WRITE_RESTART_SECS=(seconds|600)
```

Value: For the `seconds` variable, enter a number greater than 0.

Default is 600.

APF-Authorize the Libraries in the STEPLIB Concatenation

All libraries that are specified in the STEPLIB concatenation of the IMS log-based ECCR JCL must be APF-authorized.

1. Verify that the PowerExchange LOAD and LOADLIB libraries are APF-authorized. You should have completed this step during installation.
2. If you use the DTLCCIMX ECCR program that works with the DBRC API, APF-authorize the IMS RESLIB library, which must be included in the STEPLIB concatenation.

To determine the library names, see the ECCR JCL member, called `xxxIMSEC`, that PowerExchange generates in the PROCLIB library based on your entries in the z/OS Installation Assistant.

Configuring the IMS Log-Based ECCR JCL

Configure the IMS log-based ECCR JCL that PowerExchange installation generates in the `xxxIMSEC` member of the PROCLIB library, where `xxx` is the **PowerExchange Agent / Logger Prefix** value that you specified in the z/OS Installation Assistant.
The generated JCL is customized based on your input in the z/OS Installation Assistant. The ECCRMS EXEC statement that is compatible with your IMS version is uncommented. You can customize the JCL as needed. For example, to run the ECCR as a started task, configure a PROC instead of a JOB card.

The following sample JCL contains statements that were generated at installation:

```plaintext
//IMSEC PROC HLQ=PWX.PROD,LOGGER=PWXL,
//   HSEQDM=PWX.PROD,
//   RUNLIB=PWX.PROD.RUNLIB,
//   HLQVS=PWX.PROD.VSM
//   IMSRES=IMS1110.SDFSRESL
//ECCCRMS EXEC PGM=DTLCCIM8,TIME=NOLIMIT,REGION=0M (V8)
//ECCCRMS EXEC PGM=DTLCCIM9,TIME=NOLIMIT,REGION=0M (V9)
//ECCCRMS EXEC PGM=DTLCCIMA,TIME=NOLIMIT,REGION=0M (V10)
//ECCCRMS EXEC PGM=DTLCCIMB,TIME=NOLIMIT,REGION=0M (V11)
//ECCRMS EXEC PGM=DTLCCIMC,TIME=NOLIMIT,REGION=0M (V12)
//ECCRMS EXEC PGM=DTLCCIMX,TIME=NOLIMIT,REGION=0M  IMS DBRC API
//STEPLIB DD DISP=SHR,DSN=PWX.PROD.LOADLIB
//   DD DISP=SHR,DSN=PWX.PROD.RUNLIB
//   DD DISP=SHR,DSN=PWX.PROD.EDPARMS
//   DD DISP=SHR,DSN=PWX.PROD.FWKL.USERLIB
//------------------------------
//DTLCFG DD DSN=PWX.PROD.RUNLIB(DBMOVER),
//   DISP=SHR
//DTLKEY DD DSN=PWX.PROD.RUNLIB(LICENSE),
//   DISP=SHR
//DTLMSG DD DSN=PWX.PROD.DTLMSG,
//   DISP=SHR
//* IF USING MESSAGE OVERRIDE THEN CUSTOMIZE BELOW
//  *DTLMSGO DD DISP=SHR,DSN=PWX.PROD.RUNLIB(DTLMSGO)
//  DTLOG DD SYSOUT=*  
//  DTLOLOGO1 DD SYSOUT=*  
//  * 
//DATAMAP DD DSN=PWX.PROD.VSM.DATAMAP,
//   DISP=SHR
//SYSUDUMP DD SYSOUT=*  
//SYSOUT DD SYSOUT=*  
//SYSPRINT DD SYSOUT=*  
//CEEDUMP DD SYSOUT=*  
//DTLCACHG DD DUMMY
//DTLCA CDC DD DSN=PWX.PROD.VSM.CDCT,
//   DISP=SHR
//DTLCAF G DD DSN=PWX.PROD.RUNLIB(CAPTIMS),
//   DISP=SHR
//DTLAM CFR DD DSN=PWX.PROD.VSM.CCT,
//   DISP=SHR
//* For DBRC API
//DTLDRC DD SYSOUT=*  
```

**Note:** To configure the JCL for use of the DTLCCIMX ECCR, contact Informatica Global Customer Support.

### IMS Log-Based ECCR DD Statements

You must specify certain DD statements in the IMS log-based JCL. The DDs identify data sets that the ECCR uses.

Some of the data sets are allocated at installation, while others are created dynamically by the IMS log-based ECCR.
The following table describes the DD statements:

<table>
<thead>
<tr>
<th>DD Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATAMAP</td>
<td>Identifies the data set that contains the data maps that the PowerExchange Listener uses for nonrelational access to data.</td>
</tr>
<tr>
<td>DTLAMCPR</td>
<td>Identifies the data set that contains capture registration information. This data set is used by both the PowerExchange Listener and 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>DTLCACFG</td>
<td>Identifies the data set that contains the IMS log-based ECCR configuration parameters.</td>
</tr>
<tr>
<td>DTLCFG</td>
<td>Identifies the main PowerExchange configuration member, which is usually named DBMOVER. Some of these parameters also apply to the IMS log-based ECCR.</td>
</tr>
<tr>
<td>DTLDBRC</td>
<td>To use the DTLCCIMX ECCR program, you must specify this DD with SYSOUT=* . If you use another ECCR program, this DD is not required, but you can leave it uncommented in the JCL without causing errors.</td>
</tr>
<tr>
<td>DTLKEY</td>
<td>Identifies the PowerExchange License key file that contains the license key for PowerExchange, including the PowerExchange CDC options you have.</td>
</tr>
<tr>
<td>DTLLOG and DTLLOG01</td>
<td>Identifies the PowerExchange message log files. These SYSOUT files contain messages that report the IMS log-based ECCR status and events.</td>
</tr>
<tr>
<td>DTLMSG</td>
<td>Identifies the 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

After you configure the IMS log-based ECCR, you can start it.

1. Verify that the PowerExchange Listener, PowerExchange Agent, and PowerExchange Logger for MVS are running.
2. Use one of the following methods to start the ECCR:
   - To run the ECCR as a started task, use the MVS START command.
   - To run the ECCR as a batch job, submit the job.
Stopping IMS Log-based Change Data Capture

You can stop IMS log-based change data capture at various capture levels.

The following table summarizes methods of stopping change data capture by level:

<table>
<thead>
<tr>
<th>Capture Level</th>
<th>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. Then restart the IMS log-based ECCR.</td>
</tr>
</tbody>
</table>

**RELATED TOPICS:**

- "Configuring the IMS Log-Based ECCR JCL" on page 272

Stopping the IMS Log-Based ECCR

To stop the IMS log-based ECCR, use the MVS STOP command.

By stopping the IMS log-based ECCR, you stop the capture of change data until you restart the ECCR.

The IMS log-based ECCR disconnects from the PowerExchange Logger and displays a set of messages. The messages include the number and type of changes that the ECCR captured since the last time the data set was opened. For example, the ECCR might display the following messages:

14:07:37,56 PKXEEM0728091 Change Capture counts for INLIMS1IMSVXCP0100000: Insert=3, Update=0, Delete=0
RBA=0"00000004E8A50000"'
14:07:39,12 PKXEEM0728181 Left XCF group 'LOG' as member 'Log1ECCR'
14:07:39,12 PKXEEM0728931 EDM ECCR sent 3 records to Logger DLCL \3 change records.

For more information about the STOP command for the IMS log-based ECCR, see the **PowerExchange Command Reference**.

Deactivating or Deleting Registrations

To delete or deactivate capture registrations, use the PowerExchange Navigator.

To activate capture registration changes, you must restart the IMS log-based ECCR.

Using the DTLCUIML Utility to Create Restart Points

Use the DTLCUIML utility 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 must 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.

Adding an IMS Capture Registration

You might need to add a capture registration for a new or existing IMS segment from which you want to start capturing change data. In this case, you can use the REFRESH command to refresh the list of registered IMS segments for the IMS log-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(CAPTIMS) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. If you need to begin capturing changes for the new registration from a specific point, stop any change activity on the source database.
2. In the PowerExchange Navigator, open the capture registration and set the Status field to Active.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:
   F eCCR_task_name,REFRESH
   The newly registered source is added to the list of registered sources for the ECCR.
5. Enable change activity on the source to resume.
6. If you use PowerExchange Condense, restart it.

Deleting an IMS Capture Registration

You might need to delete a capture registration that has been used for change capture processing. In this case, you can use the REFRESH command to refresh the list of registered IMS segments for the IMS log-based ECCR, without restarting the ECCR.

Before you begin, ensure that REFRESH_ALLOWED=Y is specified in the RUNLIB(CAPTIMS) member to which the DTLCACFG DD statement in the ECCR JCL points.

1. Stop applications and other activities that update the source database that is associated with the registration that you are deleting.
2. Ensure that the ECCR has processed all of the IMS SLDSs that contain changes for the source that is associated with the registration that you are deleting. Also ensure that the source data has been extracted and applied to the target. Then stop all workflows that extract change data for the source.
3. If you use PowerExchange Condense, ensure that PowerExchange Condense has processed all of the captured changes. Then shut down PowerExchange Condense.
4. In the PowerExchange Navigator, open the capture registration and set the Status field to History. Then delete the registration.
5. Enter the ECCR REFRESH command using the MVS MODIFY (F) command:
   F eCCR_task_name,REFRESH
6. Enable change activity on the source to resume.
7. If you use PowerExchange Condense, restart it.
8. Restart extraction processing.

**Suspending Change Capture for Registered IMS Sources Temporarily**

Use this task flow to suspend change capture processing for registered IMS log-based CDC sources temporarily.

You perform some tasks with the PWXUCREG utility and other tasks outside of the utility on the z/OS system.

Before you begin, ensure that the REFRESH_ALLOWED=Y parameter is specified in the RUNLIB(CAPTIMS) member to which the DTLCACFG DD statement in the ECCR JCL points. You must be able to issue a REFRESH command after each registration status change.

1. Stop database activity for the registered source or sources for which you want to suspend capture registrations.

2. To suspend the capture registrations, use the PWXUCREG utility to issue the SUSPEND_REGISTRATION command.

   The suspension window opens. The utility sets the suspension timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLLOG log to report the registration status change.

   For each suspended registration, the PowerExchange Navigator Resource Inspector displays **Suspended** in the **Status** field and the suspension timestamp in the **Suspend Time** field. The **Suspend Time** value is not adjusted for the local time.

3. Enter the ECCR REFRESH command with the MVS MODIFY (F) command:

   ```
   F eccr_task_name,REFRESH
   ```

   The ECCR becomes aware of the registration status change and suspension timestamp. When the ECCR encounters the first change record to discard, it issues message PWX-07752. The ECCR discards change records that have a timestamp later than the suspension timestamp.

4. Run the jobs or processes that generate the changes that you do not want to capture for the source or sources that are associated with the suspended registrations.

5. After the jobs or processes complete, use the PWXUCREG utility to issue the ACTIVATE_REGISTRATION command to reactivate the capture registrations.

   The suspension window closes, and the utility sets the activation timestamp to the current system time without any adjustment for the local time. Also, the utility issues message PWX-03716 to the DTLLOG log to report the registration status change.

   For each reactivated registration, the PowerExchange Navigator Resource Inspector displays **Active** in the **Status** field and the activation timestamp in the **Active Time** field. The **Active Time** value is not adjusted for the local time.

6. Enter the ECCR REFRESH command with the MVS MODIFY (F) command again.

   The ECCR becomes aware of the registration status change and activation timestamp.

7. Enable database activity to resume on the registered source or sources.

   The ECCR starts capturing change records that have timestamps later than the activation timestamp. The ECCR issues message PWX-07753 when it encounters the first change record in the change stream after the end of the suspension window.

**Note:** You can automate this processing if appropriate for your environment.
Changing IMS Source Schema

If you need to change the structure an IMS database that is registered for change capture, use this procedure to retain access to historically captured data while capturing data of the new format.

1. Stop all update activity against the IMS database.
2. Verify that the IMS log-based ECCR has captured all changes under the current schema.
3. Stop the IMS log-based ECCR.
4. Complete extraction processing of all captured changes for the IMS database.
5. Make the schema changes.
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 to resume.
Chapter 13

IMS Synchronous Change Data Capture

This chapter includes the following topics:

- IMS Change Data Capture Overview, 279
- Configuring the IMS Synchronous ECCR, 283
- Activating the IMS Synchronous ECCR, 290
- Managing IMS Synchronous CDC, 292

IMS Change Data Capture Overview

PowerExchange change data capture (CDC) for IMS captures changes made to IMS databases and logs those changes to PowerExchange Logger for MVS log files.

You can use PowerCenter CDC sessions to extract the captured change data from the PowerExchange Logger log files or from PowerExchange Condense condense files and apply that data to one or more target databases.

PowerExchange provides the following alternative methods for performing IMS CDC:

- **Synchronous IMS CDC.** Captures 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.** Asynchronously captures changes by reading them from the IMS archive logs and logging them to the PowerExchange Logger. The IMS log-based ECCR runs in a separate address space as a started task or a batch job.

The following table compares IMS synchronous CDC and the IMS log-based CDC methods:

<table>
<thead>
<tr>
<th>Feature</th>
<th>IMS Synchronous CDC</th>
<th>IMS Log-Based CDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does real-time capture of change data.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Reads IMS archive logs to capture IMS change data asynchronously.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>PowerExchange IMS interface modules install into the IMS RESLIB.</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.

<table>
<thead>
<tr>
<th>Feature</th>
<th>IMS Synchronous CDC</th>
<th>IMS Log-Based CDC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses the IMS external subsystem to communicate with IMS ECCR.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>PowerExchange libraries must be added to the IMS region JCL.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>An EXIT statement must be added to the DBD for each database from which you capture changes.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>All databases from which you capture changes must be registered in DBRC.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>ECCR uses the current RECON data set to determine which IMS archive logs to process.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Captures change data within an IMSplex.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Captures multiple segments with a single capture registration.</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Captures non-keyed and non-uniquely keyed segments.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Captures changes from compressed databases.</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Adds additional data to the IMS log data sets.</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>
• 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 following restrictions pertain to the IMS synchronous ECCR:

- The IMS synchronous ECCR requires components from the BMC Software CHANGE RECORDING FACILITY, DATABASE INTEGRITY PLUS, and Fast Path Online Restructure/EP products. If you do not have any of these BMC Software products, you can use the `hlq.CRG.LOAD` library that PowerExchange supplies. The CRG software is based on version 4.9.00 Level 1401B of the BMC Software products.
- 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
  - When `PROCOPT=L` is specified in the program specification block (PSB)
  - When user data is in secondary indexes
  - When an update request does not change 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 283

**IMS Synchronous Change Data Capture Considerations**

The following considerations pertain to IMS synchronous change data capture:

- 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 that contains the relative byte address (RBA) of the segment. This RBA value is passed to the PowerExchange Logger where it is logged along with the change data.
  
  To use this RBA value to create a unique key field for the segment, you must create a user-defined field in the data map for the segment. Use the GetIMSRBAByLevel function in an expression to populate this field with the captured RBA value. The GetIMSRBAByLevel function enables you to get the RBA of an unkeyed or non-uniquely keyed parent segment. Then use the altered data map to create the extraction map.

  Reorganizing the IMS source database changes the RBA values of its 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 if the source is reorganized.

- If you need to capture changes for paired logical children, use the following guidelines:
  - 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. However, IMS supports the SETU and ROLS functions for applications that accept the SC and RC status codes. If your application accepts the SC and RC status codes, the IMS synchronous ECCR can capture change data from the SETU and ROLS functions.

- If IMS synchronous change data capture stops for any reason and updates to the data in the IMS source database are made while capture is down, PowerExchange cannot capture those changes after you activate capture again. Change capture might stop because a /STOP SUBSYS or /SSR xEDP-ABORT command was issued or because a PowerExchange component involved in change capture, such as the PowerExchange Logger or PowerExchange Agent, ended abnormally.
  
  If PowerExchange cannot capture changes and you specified ABEND for the Change Capture Error parameter at installation or for the corresponding CCERR parameter in the EDMSDIR options module, or if you issued the /SSR xEDP-ABORT command, IMS online transactions and BMP batch jobs that update an IMS database might also abend if PowerExchange determines that the database is of interest or of possible interest to change capture. If PowerExchange determines that the database is not of interest to change capture, the transactions and BMP batch jobs do not abend. If a DL/I batch job attempts to start while change capture is down, the job abends. If a DL/I batch job is active when change capture stops, the job abends when it attempts to access the database of interest or when it attempts to access any other database for the first time.

**ECCR Relationships with Other PowerExchange Components**

The IMS synchronous ECCR coordinates with other PowerExchange components such as the PowerExchange Logger and the PowerExchange Agent during change data capture.

Consider the following relationships among these components:

- 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 an IMS synchronous ECCR, PowerExchange Logger, and PowerExchange Agent on each MVS image. On all MVS images, configure the PowerExchange Logger to use Post-Log Merge.

Operational issues in the PowerExchange Logger can cause IMS transactions to wait. While transactions wait, PowerExchange cannot capture changes. After you resolve the PowerExchange Logger issues, the transactions can resume and PowerExchange can capture and log change data without any data loss. Monitor the PowerExchange Logger closely to ensure that change data capture proceeds without interruption.

**RELATED TOPICS:**

- "Monitoring the PowerExchange Logger for MVS" on page 72
- "Using Post-Log Merge" on page 90

---

**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

**RELATED TOPICS:**

- "Compatibility with BMC Software Products" on page 283
- "Configuring IMS DBRC" on page 284
- "Configuring the IMS Region JCL" on page 285
- "MVS LNKLST Concatenation" on page 289

---

**Compatibility with BMC Software Products**

The IMS synchronous ECCR requires components from the BMC Software CHANGE RECORDING FACILITY, DATABASE INTEGRITY PLUS, and Fast Path Online Restructure/EP products.

**Note:** These components are also part of some other BMC Software products such as CONCURRENT REORG and BMC MAXM Reorg/Online for IMS.

If you have one of these BMC Software products, use it instead of the PowerExchange hlq.CRG.LOAD library.
If you have BMC Software CHANGE RECORDING FACILITY, DATABASE INTEGRITY PLUS, or Fast Path Online Restructure/EP, verify that it meets the minimum version requirements for IMS synchronous CDC. The following table shows the minimum version by product and IMS version:

<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, which supports IMS Version 9.1.</td>
</tr>
<tr>
<td></td>
<td>- 4.6.00, which supports IMS Version 10.1.</td>
</tr>
<tr>
<td></td>
<td>- 4.7.00 Level 1002A, which supports IMS Version 11. You must also install the BMC fix BPQ5203.</td>
</tr>
<tr>
<td></td>
<td>- 4.8.00 Level 1301B, which supports IMS Version 12.</td>
</tr>
<tr>
<td></td>
<td>- 4.9.00 Level 1302B, which supports IMS Version 13</td>
</tr>
<tr>
<td>DATABASE INTEGRITY PLUS</td>
<td>- 4.5.04, which supports IMS Version 9.1.</td>
</tr>
<tr>
<td></td>
<td>- 4.6.00, which supports IMS Version 10.1.</td>
</tr>
<tr>
<td></td>
<td>- 4.7.00 Level 1002A with BMC fix BPQ5203, which supports IMS Version 11.</td>
</tr>
<tr>
<td></td>
<td>- 4.8.00 Level 1301B, which supports IMS Version 12.</td>
</tr>
<tr>
<td></td>
<td>- 4.9.00 Level 1302B, which supports IMS Version 13</td>
</tr>
<tr>
<td>Fast Path Online Restructure/EP</td>
<td>- 3.1.00 with any PUT level for IMS 12</td>
</tr>
<tr>
<td></td>
<td>- 3.11.00 Level 1302B for IMS Version 13</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Not applicable for IMS versions prior to version 12.</td>
</tr>
</tbody>
</table>

If you do not know the version of the CHANGE RECORDING FACILITY, DATABASE INTEGRITY PLUS, or Fast Path Online Restructure/EP product that is installed, browse the BMC load library and select the CRGLEVEL and DBILEVEL load modules. The version information is on the last line, after the date. If you need assistance, call BMC Software Technical Support.

Based on whether your BMC Software product meets the minimum version requirement, complete one of the following actions:

- If the BMC Software product meets the minimum version requirement, use its BMC Software load libraries instead of the PowerExchange hlq.CRG.LOAD library.
- If the BMC Software product version is earlier than the minimum required version, upgrade the product before you configure IMS synchronous ECCR JCL and activate the ECCR.

### 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 DSPCRTC0. 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 DSPCRTC0 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 DBICRY\text{vr} load module:

<table>
<thead>
<tr>
<th>IMS version.release</th>
<th>CRG.LOAD Module Name</th>
<th>DBICRY\text{vr} Module Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.1</td>
<td>DBICRX91</td>
<td>DBICRY91</td>
</tr>
<tr>
<td>10.1</td>
<td>DBICRXA1</td>
<td>DBICRYA1</td>
</tr>
<tr>
<td>11.1</td>
<td>DBICRXB1</td>
<td>DBICRYB1</td>
</tr>
<tr>
<td>12</td>
<td>DBICRXC1</td>
<td>DBICRYC1</td>
</tr>
<tr>
<td>13</td>
<td>DBICRXD1</td>
<td>DBICRYD1</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 \text{hlq}.SAMPLIB. This sample job contains two SMP/E USERMODs:

- USERMOD MODDBI1 includes DBICRX\text{vr} from \text{hlq}.CRG.LOAD and DSPCRTR0 from the IMS RESLIB to create the DBICRY\text{vr} load module in the IMS RESLIB.
- USERMOD MODDBI2 includes DBICRT00 from \text{hlq}.CRG.LOAD to replace DSPCRTR0 in the IMS RESLIB.

\textbf{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 \text{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 \text{hlq}.SAMPLIB, which can be used instead of the SMP/E USERMODs. This sample JCL manually link-edits the DBICRX\text{vr} and the DBICRT00 modules to create the necessary combination load modules. The job places the resulting load modules in \text{hlq}.CRG.LOAD.

\textbf{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.

**Configuring the IMS Region JCL**

You must modify the IMS region JCL before activating the IMS synchronous ECCR.

Complete the following tasks:

1. If you have the BMC Software CHANGE RECORDING FACILITY for IMS, DATABASE INTEGRITY PLUS, or Fast Path Online Restructure/EP product, verify that the installed versions meet the minimum version requirements for the IMS synchronous ECCR. See "\textit{Compatibility with BMC Software Products}" on page 283.
2. If you do not have the CHANGE RECORDING FACILITY for IMS product, add the CRG.LOAD library to the IMS region JCL.
3. If you do not have the DATABASE INTEGRITY PLUS or Fast Path Online Restructure/EP product, 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 286
- “Adding the CRG.LOAD Library to the DBRC JCL” on page 286
- “Adding the CRG.LOAD Library to the IMS Region JCL” on page 286
- “Configuring the IMS External Subsystem” on page 287
- “Adding Remaining PowerExchange Libraries” on page 287

**Verifying Installed Version of BMC Products**

PowerExchange has minimum version requirements for the BMC Software CHANGE RECORDING FACILITY, DATABASE INTEGRITY PLUS, and Fast Path Online Restructure/EP products. If you have these products, verify that the installed version is the same as or later than the minimum 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.

**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**

If you do not have the BMC Software DATABASE INTEGRITY PLUS or Fast Path Online Restructure/EP product installed, complete this task.

Verify that the PowerExchange modifications have been applied to DBRC. Then add the `hlq.CRG.LOAD` library to DBRC in one of the following ways:

- 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
  ```

- Customize and execute the DBICOPY member in the `hlq.SAMPLIB` library. DBICOPY copies the required DATABASE INTEGRITY PLUS or Fast Path Online Restructure/EP load modules from the `hlq.CRG.LOAD` library to the IMS RESLIB.
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:

```plaintext
//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:

```plaintext
//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.

  - 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. Separate the parameters with a comma (,).
The following table describes these parameters:

<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>REO</td>
<td>Yes</td>
<td>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.</td>
</tr>
<tr>
<td>CRC</td>
<td>No</td>
<td>One character command recognition character that allows external subsystem commands from IMS terminals or automated operator interface (AOI) applications. Any EBCDIC value except &quot;/&quot; is permitted. The &quot;/&quot; 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. <strong>Note:</strong> 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.</td>
</tr>
</tbody>
</table>

The following example shows the fields that define the external subsystem for the IMS synchronous ECCR using the positional format:

```
PWX, 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=PWXA, ESMT=EDMCESMT, 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

All libraries in the DFSESL concatenation must be APF-authorized.

You do not need to add the hlq.logger_name.USERLIB or hlq.CRG.LOAD to the DFSESL concatenation. However, the hlq.CRG.LOAD library must also 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 search order, 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 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:

```plaintext
//DFSESL DD DSN=IMS.SDFSRESL,DISP=SHR
```

The EDMSDIR module 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:

```plaintext
//DFSESL DD DSN=IMS.SDFSRESL,DISP=SHR
   DD 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.
Activating the IMS Synchronous ECCR

You must activate the IMS synchronous ECCR to capture changes.

Before you activate the IMS synchronous ECCR, complete configuration of the IMS DBRC and IMS region JCL and activate the capture registrations.

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 then reopen it by using the IMS START command to capture changes.

The IMS synchronous ECCR is activated in IMS regions that contain the PowerExchange modules in the STEPLIB concatenation. You can prevent the ECCR from capturing changes for a specific job or region by adding the following DD statement to the JCL:

```
//EDMNOCAP DD DUMMY
```

1. Start the PowerExchange Listener, PowerExchange Agent, and the PowerExchange Logger for MVS tasks. These tasks must be active before the IMS subsystem is started. Otherwise, no change data is captured.

2. Start the IMS subsystem. The IMS synchronous ECCR starts during initialization of the IMS subsystem and generates a report that begins with message PWXEDM172852I in the EDMMSG 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 is activated. The following messages are issued when using the PowerExchange hlq.CRG.LOAD library. The messages differ slightly if you use the BMC Software DATABASE INTEGRITY PLUS, CHANGE RECORDING FACILITY, or Fast Path Online Restructure/EP product instead of the CRG code.

- In the DBRC region, verify that the job log (JESMGLG) contains the following messages:

  ```
  BMC27001I NO VALID DBI PASSWORD FOUND
  BMC44001I DI+ INITIALIZATION COMPLETE
  BMC44008I DI+ LABEL PROCESSING SUSPENDED
  DFS3613I - DRC TCB INITIALIZATION COMPLETE imsid
  ```

  The variable imsid is the IMS subsystem name.

  Message BMC44001I indicates that the DBRC modification that the IMS synchronous ECCR requires is installed.

- In the IMS control region, verify that the job log (JESMGLG) contains the following messages:

  ```
  *DFS08001I Awaiting notification from subsys nsid  imsid
  BMC250011I CRP V4600 12/21/07 INITIALIZATION COMPLETED, RC=0, RSN=00000000
  ```

- 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
```
The variable ssid is the IMS external subsystem and imsid is the IMS subsystem name.

Message BMC250011I indicates that the CHANGE RECORDING FACILITY (CRF) product, which the IMS synchronous ECCR requires, has initialized. PowerExchange generates the MVS MODIFY command following CRF activation to notify IMS that the external subsystem is active and ready to connect. Message BMC90488W can be ignored.

The following messages in the EDMMSG SYSOUT data set indicate that the IMS synchronous ECCR connected successfully to the PowerExchange Logger and completed initialization:

PWXEDM172818I Joined XCF group 'logger_id' as member 'imsid'
PWXEDM1728411 EDM ECCR imsid connected to EDM Logger logger_id, Log RBA=X'000000011680000'
PWXEDM1728521 DFSESLE DD allocation options:
DSNs to allocate to DFSESLE DD. . . . . . : user.data.set1
: IMS910.SDFSRESL
: DSN810.SDSNLOAD
: user.data.set2

PWXEDM172820I Change Capture initialized for IMS Online on V9.1.0 - imsid

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 . . . . . . . . : D.05
Agent Id . . . . . . . . . . . . . . . . . . : I24A
Logger Id. . . . . . . . . . . . . . . . . : I24L
SYSOUT class . . . . . . . . . . . . . . . : *
Action if ECCR error encountered . . . . : Abend

PWXEDM172818I Joined XCF group 'I24L' as member 'EDMA'
PWXEDM1728411 EDM ECCR EDMA connected to EDM Logger I24L, Log RBA=X'000000011680000'
PWXEDM1728521 DFSESLE DD allocation options:
DSNs to allocate to DFSESLE DD. . . . . . : EDM.EDM.EDMA91.SDFSRESL
: IMS910.SDFSRESL
: DSN810.SDSNLOAD
: EDM.PROD_LOAD

PWXEDM1728201I Change Capture active for IMS DDB/DSN DDBLOG50P/EDM.EDM.DBLOG5
Segment=DB#AASEG SegCode=1 Edition=0000000000000000 EDNAME=IMS.DDBLOG50P.DB#AASEG
Segment=DB#BASEG SegCode=2 Edition=0000000000000000 EDNAME=IMS.DDBLOG50P.DB#BASEG
Segment=DB#CASEG SegCode=3 Edition=0000000000000000 EDNAME=IMS.DDBLOG50P.DB#BASEG
Segment=DB#BBSEG SegCode=4 Edition=0000000000000000 EDNAME=IMS.DDBLOG50P.DB#BBSEG

PWXEDM1728531 Change Capture counts for IMS DDB DDBLOG50P
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#BBSEG ISRT=0 REPL=0 DLET=0
PWXEDM1728411 EDM ECCR EDMA disconnected from EDM Logger I24L, Log RBA=X'00000000013F80000'
PWXEDM172818I Left XCF group 'I24L' as member 'EDMA'
PWXEDM1728201I 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.                                                                          |

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:

<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.                                                               |
| /SSR xEDP-STAT   | Produces an IMS synchronous ECCR snapshot report in the EDMMMSG SYSOUT data set. The report contains the number of record inserts, replacements, and deletes that the IMS ECCR has captured. The records are grouped by database area and segment. |
| /SSR xEDP-STATWTO| Produces an IMS synchronous ECCR snapshot report in the job log of the IMS region. The report contains the number of record inserts, replacements, and deletes that the IMS ECCR has captured. The records are grouped by database area and segment. |

In the commands, substitute the variable x with the command recognition character (CRC) that you specified when defining the external IMS subsystem.

**Note:** The IMS external commands are available only if you have modified the member DFSPBxxx and the SSM member in the IMS PROCLIB to include a matching command recognition character (CRC).

**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
IEE600I REPLY TO 89 IS;/DISPLAY SUBSYS I24A
DFSD001I SUBSYS CRC REGID PROGRAM LTERM STATUS EDM
DFSD001I I24A # CONN EDM
DFSD001I *07304/211738* EDM
```

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.
DFSD059I SSR COMMAND COMPLETED EDM
PWXEDM172889I 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.
DFS058I SSR COMMAND COMPLETED   EDMA
PWXEDM172889I Action if ECCR error encountered has been set to CONT.
```

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.
DFS058I SSR COMMAND COMPLETED   EDMA
PWXEDM172890W 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:

```
PWXEDM172853I Change Capture counts for IMS DBD DBLOG50F
  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#BASEG 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 IMS Synchronous Change Data Capture

You can stop IMS synchronous change data capture at various capture levels.

The following table summarizes the methods of stopping change capture by level:

<table>
<thead>
<tr>
<th>Capture Level</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMS database</td>
<td>Close the database or data set. Alternatively, stop the IMS synchronous ECCR that captures data from the database.</td>
</tr>
<tr>
<td>All IMS synchronous ECCR capture</td>
<td>Stop the IMS synchronous ECCR.</td>
</tr>
<tr>
<td>Any registered data object</td>
<td>Deactivate or delete the corresponding capture registration. Then, close or stop the database or data set, as appropriate, and refresh the ECCR.</td>
</tr>
</tbody>
</table>

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.

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.
• You cannot use the IMS Batch Backout utility to back out any farther than the last IMS checkpoint on the batch log.
• If IMS Dynamic Backout runs due to an abend, you do not need to run the Batch Backout utility.

Managing IMS Schema Changes

If you change the structure of an IMS database that is registered for change data capture, use this procedure to retain access to historically captured data while capturing data under the new structure.

1. Stop all update activity against the IMS database.
2. Ensure that PowerExchange has processed all changes that occurred under the old schema.
3. Make the structural changes to the IMS database.
4. Create a new PowerExchange capture registration that reflects the schema changes.
5. Restart PowerExchange processing.
6. Allow update activity to the IMS database to resume.
CHAPTER 14

Remote Logging of Data

This chapter includes the following topics:

- Remote Logging Overview, 297
- Requirements for Capture Registrations, 300
- Security Settings for Data from z/OS Sources, 300
- Configuration Tasks for Remote Logging, 301
- Example of Remote Logging from a z/OS Data Source, 306

Remote Logging Overview

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

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

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

RELATED TOPICS:

- "Requirements for Capture Registrations" on page 300
- "Configuration Tasks for Remote Logging" on page 301
- "Customizing the dbmove Configuration File on the System to Which Data Is Logged" on page 303
- "Customizing the PowerExchange Logger Configuration File for Remote Logging" on page 301
- "Customizing the dbmove Configuration File on the PowerCenter Integration Service System" on page 305

Remote Logging of Data from i5/OS or z/OS Sources

You can use the PowerExchange Logger for Linux, UNIX, and Windows to extract change data for data sources on i5/OS and z/OS and relog that data to a less costly Linux, UNIX, or Windows system. Multiple
PowerCenter CDC sessions can then retrieve the change data from the local PowerExchange Logger for Linux, UNIX, and Windows log files.

For i5/OS and z/OS sources, the remote logging of data to a Linux, UNIX, or Windows system has the following benefits:

- Moves resource-intensive, column-level processing and UOW Cleanser processing off of the i5/OS or z/OS system onto the Linux, UNIX, or Windows system where the PowerExchange Logger for Linux, UNIX, and Windows runs.
- Extracts change data from the DB2 for i5/OS journal receivers or PowerExchange Logger for MVS log files on z/OS in a single pass and transmits that data over the network to the PowerExchange Logger for Linux, UNIX, and Windows. The data is then available locally for PowerCenter CDC sessions to process. This single-pass processing reduces network traffic and avoids the overhead of multiple data extraction reads.
- Reduces costly CPU usage, disk space, and CDC processing time on the i5/OS or z/OS source system.

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

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

For example, you can configure the PowerExchange Logger for Linux, UNIX, and Windows to extract DB2 for z/OS change data from PowerExchange Logger for MVS logs files on a z/OS system and then relog that data...
to PowerExchange Logger for Linux, UNIX, and Windows log files on the PowerCenter ISP machine. The following image shows this remote logging configuration:

In this scenario, set the PowerExchange Logger CAPTURE_NODE statement to point to the node name of the PowerExchange Listener on the z/OS system with the DB2 logs. Set the PowerCenter Location connection attribute to the node name of the PowerExchange Listener on the PowerCenter ISP machine where the PowerExchange Logger for Linux, UNIX, and Windows runs. Set the Map Location connection attribute to point to the node name of the PowerExchange Listener on the z/OS system.

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

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

- To use the PowerExchange Logger for Linux, UNIX, and Windows, you must configure capture registrations for partial condense processing. In the PowerExchange Navigator, select Part in the Condense list for each registration. If you have remote i5/OS or z/OS data sources with capture registrations that specify Full for the Condense option, the PowerExchange Logger for Linux, UNIX, and Windows ignores these registrations. The PowerExchange Logger also ignores any capture registration that specify None for the Condense option.

- A PowerExchange Logger for Linux, UNIX, and Windows process must be able to read all of the capture registrations that it uses from a single CCT file on the source system.

- For the remote data sources, you cannot use capture registrations that were created from data maps that use any of the following features:
  - User access methods
  - User-defined fields that invoke programs by using the CALLPROG function
  - Record-level exits

Security Settings for Data from z/OS Sources

For the highest level of security for data from z/OS data sources, set the SECURITY option to 2 in the z/OS DBMOVER configuration member where the extraction maps are located. With this setting, PowerCenter CDC sessions are permitted to extract z/OS data from PowerExchange Logger for Linux, UNIX, and Windows log files only if their user credentials pass z/OS security checking.

When defining a PWXPC connection for the CDC sessions that extract data from the PowerExchange Logger log files, enter a valid z/OS user ID and password in the Map Location User and Map Location Password connection attributes. If the location of the log files is not local, enter the z/OS user ID and password in the User Name and Password connection attributes for use by the PowerExchange Listener on the Linux, UNIX, or Windows system where the log files reside.

For data extraction, these z/OS user credentials must have the following permissions:

- READ access to the PowerExchange data set that is defined in the DTLCAMAP DD statement of the PowerExchange Listener JCL
- READ access to CAPX.CND.* resource profiles in the FACILITY class, which are managed by your z/OS security product.

For more information about security, see the PowerExchange Reference Manual.
Configuration Tasks for Remote Logging

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

1. Install PowerExchange on the system where the PowerExchange Logger log files will be located.
2. Customize the pwxccl.cfg configuration file on the system with the PowerExchange Logger log files.
3. Customize the dbmover configuration file on the system with the PowerExchange Logger log files.
   
   Copy the source-specific CAPI_CONNECTION statements from the source system to the dbmover file on the system with the PowerExchange Logger log files.
   
   **Note:** Each PowerExchange Logger must have a unique pwxccl.cfg configuration file and a unique dbmover configuration file.
4. Configure a dbmover configuration file for the PowerExchange Listener on the system with the PowerExchange Logger log files.
   
   You can use the same dbmover file for the PowerExchange Logger and the PowerExchange Listener. If you use different dbmover files, both files must specify the same CAPT_PATH value.
   
   If the PowerExchange Logger log files are on the PowerCenter Integration Service machine, you can use a local connection instead of the PowerExchange Listener for change data extractions.
5. If you are not using a "local" connection, start the PowerExchange Listener on the system with the PowerExchange Logger log files.
6. Start the PowerExchange Logger on the system with the PowerExchange Logger log files.
7. Customize the dbmover configuration file on the PowerCenter Integration Service machine.
8. Configure capture registrations for PowerExchange Logger use.
9. Configure PWX CDC Real Time connection attributes for the CDC session to extract change data from the PowerExchange Logger log files.

Customizing the PowerExchange Logger Configuration File for Remote Logging

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

PowerExchange provides a sample configuration file, named pwxccl, in the PowerExchange installation directory on the Linux, UNIX, or Windows system. You can copy this file and customize the copy.

For a complete list of PowerExchange Logger configuration parameters, see the PowerExchange Logger for Linux, UNIX, and Windows chapter in the PowerExchange CDC Guide for Linux, UNIX, and Windows.
The following table describes the parameters that are used for remote logging from a z/OS source:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPTURE_NODE</td>
<td>Required for remote logging. The node name that the PowerExchange Logger uses to retrieve capture registrations and change data from the z/OS source system. This name must be defined in a NODE statement in the dbmover configuration file on the system where the PowerExchange Logger runs. The PowerExchange Logger uses this node name to connect to the PowerExchange Listener on the source system. This name should correspond to the node name in the LISTENER statement on the source system.</td>
</tr>
<tr>
<td>CAPTURE_NODE_EPWD or CAPTURE_NODE_PWD</td>
<td>Optional. An encrypted password (EPWD) or clear text password (PWD) that is associated with the user ID specified in the CAPTURE_NODE_UID parameter. If you specify CAPTURE_NODE_UID, you must specify either CAPTURE_NODE_EPWD or CAPTURE_NODE_PWD. However, do not specify both CAPTURE_NODE_EPWD and CAPTURE_NODE_PWD.</td>
</tr>
<tr>
<td>CAPTURE_NODE_UID</td>
<td>A user ID that controls PowerExchange Logger read access to capture registrations and change data on the remote node that is specified in the CAPTURE_NODE parameter. Whether this parameter is required depends on the operating system of the remote node and the SECURITY setting in the DBMOVER configuration file for the PowerExchange Listener on that node. If CAPTURE_NODE specifies a z/OS node that has a SECURITY setting of 0, do not specify this parameter. PowerExchange uses the user ID under which the PowerExchange Listener job runs to control access to capture registrations and change data. If CAPTURE_NODE specifies a z/OS node that has a SECURITY setting of 1, you must enter a valid operating system user ID for this parameter. Otherwise, error message PWX-00231 is issued, indicating a signon failure. However, PowerExchange uses the user ID under which the PowerExchange Listener job runs to control access to capture registrations and change data. If CAPTURE_NODE specifies a z/OS node that has a SECURITY setting of 2, you must enter a valid operating system user ID for this parameter. Otherwise, error message PWX-00231 is issued, indicating a signon failure. PowerExchange uses this user ID to control access to capture registrations and change data. If the specified user ID does not have the authority that is required to read capture registrations or change data, access fails.</td>
</tr>
<tr>
<td>CONDENSENAME</td>
<td>Optional. A name for the command-handling service for a PowerExchange Logger for Linux, UNIX, and Windows process to which pwxcmd commands are issued. This service name must match the service name in the associated SVCNODE statement in the dbmover configuration file.</td>
</tr>
<tr>
<td>CONN_OVR</td>
<td>Recommended. The name of the override CAPI_CONNECTION statement to use for the PowerExchange Logger. If you do not enter CONN_OVR, the PowerExchange Logger uses the default CAPI_CONNECTION in the dbmover configuration file, if specified. For z/OS data sources, enter the name of the UOW Cleanser (UOWC) CAPI_CONNECTION statement. Informatica recommends that you specify CONN_OVR because it is the only type of override that the PowerExchange Logger can use.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DB_TYPE</td>
<td>Required. The source database type. For z/OS sources, options are:</td>
</tr>
<tr>
<td></td>
<td>- ADA. For Adabas sources.</td>
</tr>
<tr>
<td></td>
<td>- DB2. For DB2 for z/OS sources.</td>
</tr>
<tr>
<td></td>
<td>- DCM. For Datacom sources.</td>
</tr>
<tr>
<td></td>
<td>- IDL. For IDMS log-based CDC sources.</td>
</tr>
<tr>
<td></td>
<td>- IMS. For IMS sources.</td>
</tr>
<tr>
<td></td>
<td>- VSM. For VSAM sources.</td>
</tr>
<tr>
<td>DBID</td>
<td>Required. A source identifier, sometimes called the instance name, that is</td>
</tr>
<tr>
<td></td>
<td>defined in capture registrations. When used with DB_TYPE, it defines selection</td>
</tr>
<tr>
<td></td>
<td>criteria for capture registrations in the CCT file.</td>
</tr>
<tr>
<td></td>
<td>This value must match the instance or database name that is displayed in the</td>
</tr>
<tr>
<td></td>
<td>Resource Inspector of the PowerExchange Navigator for the registration group</td>
</tr>
<tr>
<td></td>
<td>that contains the capture registrations.</td>
</tr>
<tr>
<td></td>
<td>Enter one of the following values, depending on the source type:</td>
</tr>
<tr>
<td></td>
<td>- For Adabas, enter the Instance name that is displayed for the registration</td>
</tr>
<tr>
<td></td>
<td>group.</td>
</tr>
<tr>
<td></td>
<td>- For Datacom, enter the MUF Name value that is displayed for the registration</td>
</tr>
<tr>
<td></td>
<td>group. Alternatively, enter the value of REG_MUF parameter in the</td>
</tr>
<tr>
<td></td>
<td>ECCRDCMP member of the RUNLIB library.</td>
</tr>
<tr>
<td></td>
<td>- For DB2 for z/OS, enter the Instance name that is displayed for the</td>
</tr>
<tr>
<td></td>
<td>registration group. This name should match the RN parameter value in the</td>
</tr>
<tr>
<td></td>
<td>DB2 statement in the RUNLIB(REPDB2OP) member.</td>
</tr>
<tr>
<td></td>
<td>- For IDMS Log-based CDC, enter the Logsid value that is displayed for the</td>
</tr>
<tr>
<td></td>
<td>registration group. This value should match the LOGSID parameter value in</td>
</tr>
<tr>
<td></td>
<td>the RUNLIB(ECCRIDLP) member.</td>
</tr>
<tr>
<td></td>
<td>- For IMS, enter the IMSID value that is displayed for the registration</td>
</tr>
<tr>
<td></td>
<td>group. For IMS log-based CDC, this value should match the first parameter</td>
</tr>
<tr>
<td></td>
<td>value in the IMSID statement in the RUNLIB(CAPTIMS) member.</td>
</tr>
<tr>
<td></td>
<td>- For VSAM, enter the Instance name that is displayed for the registration</td>
</tr>
<tr>
<td></td>
<td>group.</td>
</tr>
<tr>
<td>EXT_CAPT_MASK</td>
<td>Required. An existing directory path and a unique prefix to be used for</td>
</tr>
<tr>
<td></td>
<td>generating the PowerExchange Logger log files.</td>
</tr>
</tbody>
</table>

**Customizing the dbmover Configuration File on the System to Which Data IsLogged**

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

PowerExchange provides a sample dbmover file in the PowerExchange installation directory on the Linux, UNIX, or Windows system. You can copy this file and customize the copy. For a complete list of all dbmover configuration statements, see the *PowerExchange Reference Manual*. 
The following table describes the dbmover statements that are used for remote logging:

<table>
<thead>
<tr>
<th>Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPT_PATH</td>
<td>Required. The path to the directory on the Linux, UNIX, or Windows system where the PowerExchange Logger CDCT file resides. The PowerExchange Logger stores information about its log files in the CDCT file. Each PowerExchange Logger that captures change data requires its own CDCT file.</td>
</tr>
<tr>
<td>CAPX CAPI_CONNECTION</td>
<td>Required. Parameters that the Consumer API (CAPI) uses for continuous extraction of change data from PowerExchange Logger for Linux, UNIX, and Windows log files. The DFLTINST parameter value in this statement must match the DBID value in the PowerExchange Logger configuration file, pwxccl.</td>
</tr>
<tr>
<td>LOGPATH</td>
<td>Optional. A unique path and directory for PowerExchange message log files on the Linux, UNIX, or Windows system where the PowerExchange Logger logs data in its log files.</td>
</tr>
<tr>
<td>NODE</td>
<td>Required. Information that PowerExchange uses to connect to the PowerExchange Listener on the system from which change data is captured. This information includes a unique user-defined node name, the TCP/IP host name, and the port number. The node name that you enter in this statement must match the CAPTURE_NODE parameter value in the PowerExchange Logger configuration file.</td>
</tr>
<tr>
<td>Source-specific CAPI_CONNECTION</td>
<td>Required. A named set of parameters that the CAPI uses to connect to the change stream for a source type and control CDC processing. Copy the source-specific CAPI_CONNECTION statements from the DBMOVER configuration file on the source system. For z/OS sources, copy the LRAP and UOWC CAPI_CONNECTION statements. Remove the z/OS-specific parameters from the UOWC statement.</td>
</tr>
<tr>
<td>SVCNODE</td>
<td>Optional. The TCP/IP port on which a command-handling service for a PowerExchange process, such as a PowerExchange Logger for Linux, UNIX, and Windows process, listens for pwxcmd commands.</td>
</tr>
</tbody>
</table>
Customizing the dbmover Configuration File on the PowerCenter Integration Service System

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

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

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

Configuring Capture Registrations for the PowerExchange Logger

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

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

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

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

Configuring PowerCenter Connection Attributes for Extracting Data from the Log Files

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

The following table describes these connection attributes:

<table>
<thead>
<tr>
<th>Connection Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Enter the node name for the PowerExchange Listener that runs on the system where the PowerExchange Logger log files reside. If the log files are on the PowerCenter Integration Service machine, you can enter &quot;local.&quot;</td>
</tr>
<tr>
<td>Map Location</td>
<td>Enter the node name for the location where the PowerExchange Listener on the source system stores the extraction maps. Usually, this node is the source system node.</td>
</tr>
</tbody>
</table>
Enter a user ID and password that can access the extraction maps.
If the PowerExchange Listener runs on a source system with PowerExchange security enabled, the user ID and password depends on the SECURITY statement settings in the DBMOVER configuration file.
- READ access to the PowerExchange data set that is defined in the DTLCAMAP DD statement of the PowerExchange Listener JCL
- READ access to CAPX.CND.* resource profiles in the FACILITY class, which are managed by your z/OS security product

Enter the name of the CAPX CAPI_CONNECTION statement that is used by the PowerExchange Listener on the system where the PowerExchange Logger for Linux, UNIX, and Windows log files reside.

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

Example of Remote Logging from a z/OS Data Source

In this example, you use a PowerExchange Logger for Linux, UNIX, and Windows instance on a UNIX system to relog change data for DB2 for z/OS data sources. The system where the PowerExchange Logger runs is separate from the PowerCenter Integration Service system where you run the CDC sessions.

The PowerExchange Logger for MVS captures change data for registered DB2 for z/OS tables and logs that data to its log files on the z/OS system. The PowerExchange Logger for Linux, UNIX, and Windows reads data from the PowerExchange Logger for MVS log files and relogs that data on the UNIX system. PowerCenter CDC sessions then extract change data from the PowerExchange Logger for Linux, UNIX, and Windows log files rather than from the log files on the z/OS source system.

You need the PowerExchange Logger for Linux, UNIX, and Windows to read change data for registered tables in the DB2 instance DSN9 and then relog that data to its log files on the remote UNIX system. To do so, you must customize a PowerExchange Logger for Linux, UNIX, and Windows configuration file on the UNIX system and dbmover configuration files on both the z/OS and UNIX systems. Also, for the PowerCenter CDC sessions to extract change data from the PowerExchange Logger log files on UNIX, you must add NODE statements for the source and PowerExchange Logger systems to the dbmover configuration file on the Integration Service system and configure some PWXPC connection attributes.

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

1. On the z/OS source system, ensure that the DBMOVER member in the RUNLIB library includes the following CAPI_CONNECTION statements:

   LISTENER=(MVS02,TCP1P,2480)
   /* UOW Cleanser
2. On the UNIX system with the PowerExchange Logger for Linux, UNIX, and Windows log files, ensure that the dbmover configuration file includes the following statements:

```bash
/*
*/
/* dbmover */
LISTENER={unix1,TCP1P,2480)
NODE=(MVS02,TCP1P,prodmvs2,2480)
...
LOGPATH=/px/logs/mvscond
CAPT_XTRA=/px/capture/mvscond/camaps
CAPT_PATH=/px/capture/mvscond
/*
*/
/* Source-specific CAPI Connection */
CAPI_CONNECTION={NAME=MV2UOWC,TYPE=(UOWC,CAPINAME=MV2_LRAP, RSTRADV=600,MEMCACHE=20480, DATACLAS=UOWC})
/*
*/
/* CAPI Connection for continuous extraction */
CAPI_CONNECTION={NAME=DPXDSN9,TYPE=(CAPX, DFLTINST=DSN9, FILEWAIT=60, RSTRADV=600)
```

Note: In the CAPI CONNECTION, the DFLTINST value is the name that is displayed in the Instance field for the registration group in the PowerExchange Navigator.

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

```bash
/*
*/
/* pwxcl */
/*
*/
DBID=DSN9
DB_TYPE=DB2
CONN_UVR=MV2UOWC
CAPTURE_NODE=MVS02
PROMPT=Y
EXT_CAPT_MASK=/px/capture/mvscond/condense
COND_CDCRT_P=50
LOGGER_DELETES_EXPIRED_CDCT_RECORDS=Y
COLL_END_LOG=0
NO_DATA_WAIT=0
NO_DATA_WAIT2=10
FILE_SWITCH_VAL=20000
FILE_SWITCH_CRTIT=R
CAPT_IMAGE=BA
```

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

4. Start the PowerExchange Listener and PowerExchange Logger for Linux, UNIX, and Windows on the UNIX system. Verify that the PowerExchange Listener is also running on the z/OS system.

5. On the PowerCenter Integration Service system, add the following NODE statements to the dbmover file:

- A NODE statement that points to the PowerExchange Listener on the source system
- A NODE statement that points to the PowerExchange Listener on the UNIX system with the PowerExchange Logger log files, if you are not using a "local" connection

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

```bash
NODE=(unix1,TCP1P,unix1,2480)
NODE=(MVS02,TCP1P,prodmvs2,2480)
```

6. Create a PowerCenter mapping, session, and workflow.
7. Configure a PWX DB2zOS CDC Real Time application connection for the CDC sessions that extract change data from the PowerExchange Logger log files on the UNIX system.

   For this example, set the following connection attributes:
   
   - For the Location attribute, enter unix2 to point to the node where the PowerExchange Logger for Linux, UNIX, and Windows log files reside. CDC sessions will read data from this location.
   - For the Map Location attribute, enter MVS02 to point to the location of the extraction maps, which the z/OS source system node.
   - For the Map Location User attribute, enter a valid user ID for the map location.
   - For the Map Location Password attribute, enter the password for the map location user.
   - For the CAPI Connection Name attribute, enter CAPXDSN9 to indicate the CAPX CAPI_CONNECTION statement to use.

8. Cold start the CDC session.

   The session begins extracting change data from the PowerExchange Logger log files on the UNIX system.
Part IV: Change Data Extraction

This part contains the following chapters:

- Introduction to Change Data Extraction, 310
- Extracting Change Data, 322
- Managing Change Data Extractions, 349
This chapter includes the following topics:

- Change Data Extraction Overview, 310
- Extraction Modes, 311
- PowerExchange-Generated Columns in Extraction Maps, 311
- Uses of BI and CI Fields in Extraction Maps, 316
- Restart Tokens and the Restart Token File, 317
- Multiple-Source Processing in CDC Sessions, 318
- Commit Processing with PWXPC, 320
- Tuning Options, 321

**Change Data Extraction Overview**

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

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

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

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

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

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

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

**Real-time extraction mode**

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

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

**Batch extraction mode**

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

To implement this mode, configure the following items:

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

**Continuous extraction mode.**

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

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

To implement this mode, configure the following items:

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

PowerExchange-Generated Columns in Extraction Maps

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

These PowerExchange-generated columns contain CDC-related information, such as the type of SQL change and time stamp.
When you import an extraction map in Designer, PWXPC includes the PowerExchange-generated columns in the source definition.

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

**Note:** By default, all columns except the DTL__columnname_CNT and DTL__columnname_IND columns are selected in an extraction map. To select these columns, you must edit the extraction map.

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__CAPXROWID          | For PowerExchange Oracle CDC with LogMiner and Express CDC for Oracle, provides the physical rowid value. PowerExchange can include rowid values in change records for Oracle tables only if the tables do not have row movement enabled.  
  To enable the capture of rowid values, you must configure one of the following parameters:  
  - For PowerExchange Oracle CDC with LogMiner, set the ROWID parameter in the ORCL CAPI_CONNECTION statement to Y or ALLOW.  
  - For PowerExchange Express CDC for Oracle, include the OPTIONS ROWID=Y statement in the Express CDC configuration file.  
  The rowid is useful for processing rows in unkeyed tables during CDC extraction sessions. | CHAR     | 18     |
<p>| DTL__CAPXRRN            | For DB2 on i5/OS only, the relative record number.                                                                                                                                                            | DECIMAL  | 10     |</p>
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Datatype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTL__CAPXUOW</td>
<td>A binary value that represents the position in the change stream of the start of the UOW for the change record.</td>
<td>VARBIN</td>
<td>255</td>
</tr>
</tbody>
</table>
| DTL__CAPXUSER| The user ID of the user who made the change to the data source, with the following exceptions:  
- For Adabas 8.3 CDC sources, this value is the Security User-id (SECUID) of the user if the Adabas File Definition includes the system field SY=SECUID.  
- For Datacom table-based CDC sources, this value is the MUF name.  
- For DB2 for i5/OS sources, this value depends on the LIBASUSER parameter in the AS4J CAPI_CONNECTION statement. If LIBASUSER=Y, this value is the library name and file name of the file where the change was made. If LIBASUSER=M, this value is the library name, file name, and data member name of the file where the change was made. If LIBASUSER=N, this value is the user ID of the user who made the change.  
- For DB2 for z/OS sources, this value depends on the UIDFMT parameter in the LRAP CAPI_CONNECTION. Depending on the parameter setting, this value can be a DB2 connection identifier, correlation identifier, connection type, plan name, user ID, or all of these values in the format **UID:PLAN:CORR:CONN:CTYPE**. If you do not specify the UIDFMT parameter, this value is the user ID of the user who made the change.  
- For IDMS sources, this value is the value that the user program puts in the program name field of the application subschema control block. Usually, this value is the user program name.  
- For Microsoft SQL Server sources, this value depends on the UIDFMT parameter in the MSQL CAPI_CONNECTION statement. If UIDFMT=DBNAME, this value is the SQL Server publication database name. If UIDFMT=NONE, this value is a null.  
- For Oracle sources, this value is a user ID that PowerExchange gets from Oracle, if available. Otherwise, this value is null. This information applies to both PowerExchange Oracle CDC with LogMiner and PowerExchange Express CDC for Oracle.                                                                                     | VARCHAR  | 255    |
### Column Description

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Datatype</th>
<th>Length</th>
</tr>
</thead>
</table>
| DTL__CAPXTIMESTAMP       | The time stamp that the source DBMS records for a change on the source database. This value can be either the time stamp that the source DBMS writes to the change record in the database logs or the time stamp of the transaction commit on the source database. The time stamp type depends on the source type and certain parameters:  
  - For DB2 for Linux, UNIX, and Windows sources, the transaction commit time stamp.  
  - For Microsoft SQL Server sources, the time at which the change was written to the distribution database.  
  - For PowerExchange Express CDC for Oracle sources, the time stamp type is controlled by the TIME_STAMP_MODE parameter in the OPTIONS statement of the Express CDC configuration file.  
  - For all sources that require a UOWC CAPI_CONNECTION statement, the time stamp type is controlled by the TIMESTAMP parameter in the UOWC CAPI_CONNECTION statement in the DBMOVER file.  
  For more detailed information about time stamps for each source type, see Appendix B, "DTL__CAPXTIMESTAMP Time Stamps" on page 387.  
  The time stamp format is:  
  YYYYMMDDhhmssnnnnnnnnn  
  Where:  
  - YYYY is the four-digit year.  
  - MM is the month.  
  - DD is the day.  
  - hhmssnnnnnnn is hours, minutes, seconds, and microseconds.  
  **Note:** DB2 for Linux, UNIX, and Windows and Oracle do not support microseconds in the time stamp.                                                                                     | CHAR     | 20     |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Datatype</th>
<th>Length</th>
</tr>
</thead>
</table>
| DTL__CAPXACTION      | A single character that indicates the type of change record that PowerExchange passes to the target during extraction processing. A DTL__CAPXACTION value corresponds to the type of change operation on the source database. Valid values are:  
- I. Insert.  
- D. Delete.  
- U. After image of an update.  
- T. Before image of an update. (ODBC connections only)  
If you specify an Image Type of BA on the connection for a CDC session, PowerExchange generates a delete record followed by an insert record for a source update. In the delete record, the DTL__CAPXACTION column contains the value D. In the insert record, the DTL__CAPXACTION column contains the value I.  
If you specify an Image Type of AI on the connection for a CDC session, PowerExchange generates one record for an update. In this record, the DTL__CAPXACTION column contains the U value.  
If you use an ODBC connection to write change data to a staging table and either set the ODBC driver CAPXIMAGETYPE parameter to TU or enter the SQL escape sequence DTLIMTYPE=TU in PowerCenter, this column can contain a value of T or U. For each source update, PowerExchange delivers two records to the staging table: one for the before image and another for the after image. In the before image record, the DTL__CAPXACTION column contains the T value. In the after image record, the DTL__CAPXACTION column contains the U value. | CHAR     | 1      |
| DTL__CAPXCASDELIND   | 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. | CHAR     | 1      |
| DTL__BI_columnname   | For UPDATE operations, the value of the before image of the selected column in the change record.                                                                                                           | Datatype of the source column | Length of the source column |
| DTL__CI_columnname   | 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. | CHAR     | 1      |
<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Datatype</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<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>

**Uses of BI and CI Fields in Extraction Maps**

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

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

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

**Case 1. Filtering Change Data for Extraction and Apply Processing**

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

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

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

To filter change data for extraction and apply processing:

1. In the PowerExchange Navigator, edit the extraction map that you plan to import as the source definition for the CDC session. For each column that you want to filter on, add a CI field. PowerExchange generates CI fields that have names in the format DTL__CI_column_name.

   For more information about adding CI fields to extraction maps, see the PowerExchange Navigator User Guide.
2. In PowerCenter, define WHERE clause filters in the **Filter Override** attribute of the CDC session properties.
   For the filters, enter DTL__CI_column_name conditions. For example, enter `DTL__CI_ACCOUNT='Y'`, where
   'Y' indicates an Update occurred.
   For more information about filter overrides on CDC sessions, see *PowerExchange Interfaces for PowerCenter*.

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

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

**Case 2. Updating Primary Key Fields on the Target**

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

To prevent this problem, you can select the **BA** option for the **Image Type** attribute on PWX CDC application
connections. This option causes PWXPC to generate two transactions for each source UPDATE: a DELETE
followed by an INSERT. The DELETE deletes the old row based on the before image. The INSERT inserts a
row based on the after image.

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

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

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

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

**Restart Tokens and the Restart Token File**

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

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

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

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

**Sequence token**

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

**Restart token**

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

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

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

### Multiple-Source Processing in CDC Sessions

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

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

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

**Restriction:** For nonrelational sources, you must import extraction maps.

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

- The source definition contains the extraction map name. You do not need to provide this name when you configure the session.
- The source definition contains the PowerExchange-generated CDC columns, such as the DTL__CAPX columns. You do not need to add these columns to the source definition.
During change data extraction, PowerExchange processes all source definitions in the mapping that have the same source type. Do not include multiple data source types in the mapping. Otherwise, the CDC session fails with message PWXPC_10080.

For example, you cannot run a CDC session that contains a mapping with both VSAM and IMS source definitions, even if changes for those sources are in the same change stream. Instead, 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 the VSAM sources, and once for the session with the IMS sources.

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

![Mapping Designer](image)

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

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

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

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

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

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

The following table describes these connection attributes:

<table>
<thead>
<tr>
<th>Connection Attribute</th>
<th>PWX 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 limit is met. PWXPC does not wait for a UOW boundary to commit the change data. Default is 0, which causes PWXPC to not use this maximum rows limit.</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, PowerExchange skips commit records and passes only the change records to PWXPC. Default is 0, which causes PowerExchange to not use this minimum rows limit.</td>
</tr>
<tr>
<td>Real-time Flush Latency in milliseconds</td>
<td>Real Time</td>
<td>Number of milliseconds that must elapse before PWXPC flushes the data buffer to commit change data to the targets. When this latency period expires, PWXPC continues to read the changes in the current UOW until it reaches the end of the UOW. Then, PWXPC flushes the data buffer to commit the change data to the targets. Default is 0, which causes PWXPC to use 2,000 milliseconds.</td>
</tr>
<tr>
<td>UOW Count</td>
<td>Both</td>
<td>Number of UOWs that PWXPC must process before flushing the data buffer to commit the change data to the targets. Default is 1.</td>
</tr>
</tbody>
</table>

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

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

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

```
PWXPC_12075 [INFO] [CDCRestart] Session complete. Next session will restart at: Restart 1 [restart1_token] : Restart 2 [restart2_token]
```

**RELATED TOPICS:**

- "Commitment Control Attributes" on page 333
- "Examples of Controlling Commit Processing" on page 335

## Tuning Options

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

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

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

- **Offload processing.** Use offload processing to transfer column-level extraction processing from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine. Also, if the data source type requires use of the UOW Cleanser (UOWC), offloading transfers UOWC processing to the Integration Service machine. Use offloading to help increase throughput when resources available for the PowerExchange Listener are constrained on the source system.

- **Remote logging of change data.** Configure a PowerExchange Logger for Linux, UNIX, and Windows instance on a system other than the source system. The PowerExchange Logger reads change data from the source and writes the data to its local log files. CDC sessions extract the change data from the PowerExchange Logger log files. This configuration moves resource-intensive, column-level processing from the source system to the PowerExchange Logger system. Use remote logging to help improve throughput for CDC sessions when resources on the source system are constrained.

- **Multithreading.** Enable the use of multiple worker threads for resource-intensive, column-level extraction processing. You can use multithreading on the source system to process data from Linux, UNIX, or Windows data sources, or on another system where the extraction processing runs. Enable multithreading only if extractions appear to be CPU bound. You can use multithreading with the offloading feature or remote logging.
Overview of Extracting Change Data

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

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

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

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

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

Optionally, you can configure event table processing to stop a CDC session that uses real-time extraction mode based on user-defined events.
Also, you can use the following tuning options to help take maximum advantage of the available system resources and maximize throughput for CDC sessions:

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

### Task Flow for Extracting Change Data

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

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

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

- “Creating Restart Tokens for Extractions” on page 343
- “Configuring the Restart Token File” on page 344
- “Starting PowerCenter CDC Sessions” on page 349
- “Testing an Extraction Map” on page 324

---

**Testing an Extraction Map**

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

A database row test enables you to:

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

1. In the PowerExchange Navigator, open the extraction group and the extraction map.
2. Select the extraction map and click **File > Database Row Test**.
3. In the **Database Row Test** dialog box, enter information in the following fields:
   - **DB Type**: An option that indicates the extraction mode:
     - CAPXRT. Real-time extraction mode or continuous extraction mode.
     - CAPX. Batch extraction mode.
   - **Location**: Node name for the location of the system on which the captured change data resides. This name must be defined in a NODE statement in the dbmover.cfg configuration file on the Windows machine from which you run the database row test.
   - **UserID and Password**: Optional. A user ID and password that provides access to the source data.
   - **Fetch**: To preview data, select **Data**.
   - **Application**: An application name. For a row test, an application name is not required. However, you must enter at least one character in this field. PowerExchange does not retain this value.
   - **SQL Statement**: A SQL SELECT statement that PowerExchange generates for the fields in the extraction map. You can edit this statement, if necessary.
     In the statement, a table is identified as follows:
     
     `Schema.RegName_TableName`
Where:

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

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

4. Click **Advanced**.

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

6. Click **OK**.

7. Click **Go**.

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

---

**Configuring PowerCenter CDC Sessions**

After you import metadata for CDC data sources and targets into PowerCenter, you can create a mapping, connection, and a CDC session for extracting change data. You must configure many session and connection attributes.
### Changing Default Values for Session and Connection Attributes

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

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

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Location</th>
<th>Recommended Value for CDC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commit Type</td>
<td>Properties Tab for the session</td>
<td>Source</td>
<td>Default value is <strong>Target</strong>. If you accept the default, the PowerCenter Integration Service automatically overrides the default to use source-based commit processing. However, you should change this attribute to <strong>Source</strong> so that you can disable the <strong>Commit On End Of File</strong> attribute.</td>
</tr>
<tr>
<td>Commit On End Of File</td>
<td>Properties Tab for the session</td>
<td>Disabled</td>
<td>By default, this attribute is enabled. If you accept the default, the PowerCenter Integration Service commits the change data in the buffer to the targets when the session ends. The final commit occurs after the PWXPC CDC reader has committed all complete UOWs in the buffer, along with their restart tokens, to the targets. This timing can cause the restart tokens and target data to be out of sync. The final restart tokens might represent a point in the change stream that is earlier than final change data that the PowerCenter Integration Service commits to the targets. As a result, duplicate data might occur when the CDC session restarts. To prevent potential duplicate data, disable this attribute.</td>
</tr>
<tr>
<td>Recovery Strategy</td>
<td>Properties Tab for the session</td>
<td>Resume from last checkpoint</td>
<td>Default value is <strong>Fail task and continue workflow</strong>. To properly restart CDC session, PowerExchange CDC and PWXPC require that this option is set to <strong>Resume from last checkpoint</strong>.</td>
</tr>
</tbody>
</table>
| Stop on errors          | Config Object Tab for the session   | 1                         | Default value is 0. By default, the PowerCenter Integration Service does not consider errors when writing to targets as fatal. The following types of error are non-fatal:  
  - Key constraint violations
  - Loading nulls into a not null field
  - Database trigger responses
  If write errors occur, change data loss might occur because PWXPC has advanced the restart tokens values. To maintain target data and restart token integrity, set this option to 1. |
| Application Name        | Application Connection              | Enter a unique name for each CDC session. | Default is the first 20 characters of the workflow name. **Attention**: Because the default might not result in a unique name, enter a unique name. |
| RestartToken File Folder| Application Connection              | Default value              | The default is $PMRootDir/Restart. This default is acceptable for CDC. |
## Configuring Application Connection Attributes

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

### RELATED TOPICS:
- "Image Type" on page 327
- "Event Table Processing" on page 330
- "CAPI Connection Name Override" on page 328
- "Idle Time" on page 328
- "Restart Control Attributes" on page 330
- "Flush Latency" on page 331
- "Target Latency “ on page 332

### Image Type

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

Enter one of the following options for this attribute:

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

Default is **BA**.

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

### Attributes Table

<table>
<thead>
<tr>
<th>Attribute Name</th>
<th>Attribute Location</th>
<th>Recommended Value for CDC</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RestartToken File Name</td>
<td>Application</td>
<td>Enter a unique name for each CDC session. If you enter an Application Name value, the default is that application name. If you do not enter an Application Name value, the default is the workflow name. <strong>Attention</strong>: Because a default might not result in a unique name, enter a unique restart token file name.</td>
<td></td>
</tr>
<tr>
<td>Number of Runs to Keep RestartToken File</td>
<td>Application</td>
<td>1 or greater</td>
<td>Default is 0. PWXPC keeps only one backup copy of the restart token initialization and termination files. Enter a value greater than 0 to make history available for recovery purposes.</td>
</tr>
</tbody>
</table>
If you specify **AI**, you can still use before images of the data, if available, in extraction processing. PWXPC can embed before-image data and after-image data in the same Update row. To embed before-image data, you must complete the following configuration tasks:

- In the PowerExchange Navigator, add BI and CI fields to the extraction map that you plan to import for the source definition in PowerCenter.

- If you use batch or continuous extraction mode, enter BA for the CAPT_IMAGE parameter in the PowerExchange Condense or PowerExchange Logger for Linux, UNIX, and Windows configuration file. This setting causes both before and after images to be stored in the PowerExchange Logger log files or PowerExchange Condense condense files. When CDC sessions run, they extract data from these files.

Informatica recommends that you use the **AI** setting if you want to process before images of data. CDC sessions can process a single Update record more efficiently than separate Delete and Insert records to get the before image data.

For example, embed before-image data and after-image data in the same Update row to handle changes to primary keys. Relational databases that allow changes to primary keys, such as DB2 for z/OS, treat these Updates as equivalent to deleting the row and readding it with a new key value. To enable PowerExchange to detect primary key changes, include BI and CI fields for the primary key columns in the extraction map for the source definition. Then, in PowerCenter, define a Flexible Target Key Custom transformation to apply the changes to the target as a Delete followed by an Insert. Include the transformation in the mapping for the CDC session. If a target relational database does not allow changes to primary keys, updates to primary keys fail.

**Note:** To use a Flexible Target Key Custom transformation, you must set the **Image Type** attribute to **AI** and configure BI and CI fields in the PowerExchange extraction map for the source.

For more information about adding BI and CI columns, see the **PowerExchange Navigator User Guide**.

### CAPI Connection Name Override

If you define multiple CAPI_CONNECTION statements in the DBMOVER configuration file, you can use the **CAPI Connection Name Override** connection attribute to select one of the statements for a CDC session.

PowerExchange allows a maximum of eight CAPI_CONNECTION statements in the DBMOVER configuration file. You might want to use multiple CAPI_CONNECTION statements to extract changes for multiple source types with a single PowerExchange Listener on a single machine. For example, you can extract changes for Oracle and DB2 sources through a single PowerExchange Listener by specifying multiple CAPI_CONNECTION statements.

If you use CDC offload processing, you must define the CAPI_CONNECTION statements in the dbmover.cfg file on the PowerCenter Integration Service machine. If you do not use CDC offload processing, you must define the CAPI_CONNECTION statements on the system where the change data resides.

To specify the CAPI_CONNECTION statement to use for a specific CDC session, enter the name of the CAPI_CONNECTION statement in the **CAPI Connection Name Override** connection attribute. By using the override instead of a default CAPI_CONNECTION statement, you clearly indicate which statement to use for a session.

### Idle Time

Use the **Idle Time** connection attribute to indicate whether a CDC session that uses real-time or continuous extraction mode runs continuously or shuts down after it reaches the end-of-log (EOL).

You can specify that PowerExchange wait for a certain period without change activity before shutting down.
Enter one of the following values:

- `-1`. The CDC session runs continuously. PowerExchange returns an end-of-file (EOF) only when you manually stop the CDC session.
- `0`. After reaching the EOL, PowerExchange returns an EOF and the CDC session ends. If you want a CDC session to end periodically on an active system that is rarely idle, enter 0.
- `n`. After reaching the EOL, PowerExchange waits the specified number of seconds, `n`. If PowerExchange receives no change data of interest during this time interval, PowerExchange sends an EOF to the PowerCenter Integration Service and the CDC session ends successfully. If you enter a low value, such as 1, the CDC session might end before PowerExchange has read all available data in the change stream.

Default is `-1`.

PowerExchange determines the EOL by using the current end of the change stream at the point that PowerExchange started to read the change stream. PowerExchange uses the concept of EOL because the change stream is usually not static. The actual EOL is continually moving forward. After PowerExchange reaches the EOL, it writes message PWX-09967 in the PowerExchange message log.

Often, CDC sessions that run in real-time or continuous extraction mode use the default value of `-1`. You can manually stop a long-running CDC session by using the PowerCenter Workflow Monitor, `pmcmd` commands, or the PowerExchange STOPTASK command.

If you set the **Idle Time** attribute to 0, when PowerExchange reaches the EOL, it returns an EOF to PWXPC. PWXPC and the PowerCenter Integration Service then perform the following processing:

1. PWXPC flushes all buffered UOWs and ending restart tokens to the targets.
2. The CDC reader ends.
3. After the PowerCenter Integration Service finishes writing the flushed data to the targets, the writer ends.
4. After any post-session commands and tasks run, the CDC session ends.

If you set the **Idle Time** attribute to a positive number, the following processing occurs:

1. PowerExchange reads the change stream until it reaches EOL and then the **Idle Time** wait interval begins.
2. If more data is in the change stream after the EOL, PowerExchange continues to read the change stream, looking for change data of interest to the CDC session, as follows:
   - If the idle time expires before PowerExchange reads a change record of interest for the CDC session, PowerExchange stops reading the change stream.
   - If PowerExchange reads a change record of interest to the CDC session, PowerExchange restarts the timer, passes the change data to PWXPC, and continues to read the change stream. This processing continues until the idle time expires.
3. After the idle time expires, PowerExchange passes an EOF to PWXPC.
4. PWXPC and the PowerCenter Integration Service perform the same processing as when the **Idle Time** value is 0 and the CDC session ends.

When a CDC session ends because the idle time elapsed or a PowerExchange STOPTASK command was issued, PWXPC writes the following message in the session log:

```
[PWXPC_10072] [INFO] [CDCDispatcher] session ended after waiting for [idle_time] seconds. Idle Time limit is reached
```

If you stop a continual CDC session with the PowerExchange STOPTASK command, PWXPC substitutes 86400 for the `idle_time` variable in the PWXPC_10072 message.

**Note:** If you specify both the **Reader Time Limit** and **Idle Time** attributes, the PowerCenter Integration Service stops reading data from the source when one of these attribute conditions is met, whichever one is first.
Because the reader time limit does not result in normal termination of a CDC session, Informatica recommends that you use only the idle time limit.

**Restart Control Attributes**

Use PWXPC restart control attributes to identify restart information to use for a CDC session. The restart information determines the point from which PowerExchange starts reading change data for the session.

Specify restart control attributes in the following situations:

- When you create CDC session.
- When you add a source to an existing CDC session and need to specify restart information for that source.
- When you want to override some restart information that is in the state table or file for a CDC session.

The following table describes the restart control attributes that you can enter on a PWX CDC application connection:

<table>
<thead>
<tr>
<th>Connection Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Name</td>
<td>A unique application name for the CDC session. The application name is case sensitive and cannot exceed 20 characters. Default is the first 20 characters of the workflow name. Because the default might not result in a unique name, Informatica recommends that you enter a unique name.</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>The unique file name of the restart token file. This file is in the directory that is specified in the RestartToken File Folder attribute. PWXPC uses the contents of this file, if any, in conjunction with the state table or state file to determine the restart point for the CDC session. Default is the Application Name value, or if you do not specify the application name, default is the workflow name.</td>
</tr>
</tbody>
</table>

**Attention:** The values for the Application Name and RestartToken File Name attributes must be unique for each CDC session. If either one of these values is not unique, unpredictable results might occur, including session failure and potential data loss.

**Event Table Processing**

Use event table processing to stop the extraction of changes based on user-defined events, such as an end-of-day event.

For example, to stop an extraction process every night, after all changes for the day are processed, write a change to the event table at midnight. This change triggers PowerExchange to stop reading change data and shut down the extraction process after the current UOW completes.

Use the following rules and guidelines:

- You can use event table processing only with real-time or continuous extraction modes.
- You must create the event table and define the applications that can update the table.
- You must register the event table for change data capture from the PowerExchange Navigator.
A CDC session monitors a single event table. Each user-defined event requires its own event table and a separate extraction process.

The event table and all of the source tables in the CDC session must be of the same source type.

## Implementing Event Table Processing

Use this procedure to implement event table processing. With event table processing, you can stop change data extraction processing based on user-defined events.

1. Create an event table.
   - The event table must be of the same source type and on the same machine as the change data to be extracted. For example, if you extract DB2 change data on z/OS, the event table must be a DB2 table in the same DB2 subsystem as the DB2 source tables for the extraction.

2. In the PowerExchange Navigator, create a capture registration for the event table.
   - When you create the capture registration, the PowerExchange Navigator generates a corresponding extraction map.

3. In PowerCenter, create a CDC connection and session.
   - In the **Event Table** attribute on the PWX CDC Real Time application connection, enter the name of the extraction map associated with the capture registration that you created.

4. Define applications that write an update to the event table whenever the defined event occurs.
   - PowerExchange reads the update and places an end-of-file (EOF) in the change stream. PWXPC processes the EOF, passes it to the PowerCenter Integration Service, and then shuts down the PowerExchange reader. The PowerCenter Integration Service completes writing all of the data that is in the pipeline to the targets and then ends the CDC session.

## Flush Latency

PowerExchange reads change data into a buffer on the source system, or into a buffer on the PowerCenter Integration Service machine if you use offload processing. The PowerExchange Consumer API (CAPI) periodically flushes the buffer to transfer the change data to PWXPC on the PowerCenter Integration Service machine.

The CAPI flushes the buffer to PWXPC when the one of the following events occurs:

- The buffer becomes full.
- The CAPI timeout value that is specified by the **PowerExchange Latency in seconds** attribute on the PWX CDC Real Time connection expires.
- A commit point occurs.

To specify the flush latency for CDC sessions that run in real-time or continuous extraction mode, set the **PWX Latency in seconds** attribute on the PWX CDC Real Time application connection. This attribute specifies the maximum time that PowerExchange waits for more change data before flushing data to PWXPC. This attribute applies to PowerExchange on the source system, or to the PowerExchange client on the PowerCenter Integration Service machine if you use offload processing.

For CDC sessions that use batch extraction mode, PowerExchange always uses 2 seconds for the flush latency.

PowerExchange writes message PWX-09957 to the PowerExchange message log to identify the CAPI timeout value based on the **PWX Latency in seconds** attribute. If you select **Retrieve PWX Log Entries** on the application connection, PWXPC also writes this message in the session log.
After PowerExchange flushes the change data, PWXPC provides the data to the source qualifiers in the CDC session for further processing. Then the PowerCenter Integration Service commits the data to the targets.

**Note:** The PWX Latency in seconds value also affects how fast a CDC session responds to a stop command from Workflow Monitor or pmcmd program. Before PWXPC can process a stop request, it must wait for PowerExchange to return control to it. Use the default value of 2 seconds for the PWX Latency in seconds attribute to avoid unacceptable delays in stop command processing.

**Target Latency**

Target latency is the total time for applying change data to the targets.

This total includes the time that PWXPC takes to extract change data from the change stream and the time that PowerCenter Integration Service takes to apply that change data to the targets. If extraction and apply processing occurs quickly, target latency is low.

The values for the commitment control attributes affect target latency. When you set the commitment control attributes, balance target latency requirements with resource consumption on the PowerCenter Integration Service machine and the target databases.

Lower target latency values result in higher resource use. The increased resource use occurs because the PowerCenter Integration Service must flush the change data more frequently. Also, the target databases must process more commit requests.

The following table describes the default values for the commitment control attributes, which provide the lowest latency:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Default</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows Per commit</td>
<td>0, which disables this attribute</td>
</tr>
<tr>
<td>Minimum Rows Per commit</td>
<td>0, which disables this attribute</td>
</tr>
<tr>
<td>Real-time Flush Latency in milli-seconds</td>
<td>0, which is equivalent to 2000 milliseconds or 2 seconds</td>
</tr>
<tr>
<td>UOW Count</td>
<td>1</td>
</tr>
</tbody>
</table>

These values decrease target latency because PWXPC commits changes after each UOW or on UOW boundaries. However, these values can have the following drawbacks:

- Highest resource consumption on the source system, PowerCenter Integration Service machine, and target databases
- Decreased throughput for the CDC sessions because PWXPC flushes change data too frequently for the PowerCenter Integration Service or target databases to handle this processing

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

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

Then disable the other attributes.
Commitment Control Attributes

PWXPC, in conjunction with PowerExchange and the PowerCenter Integration Service, controls the timing of commit processing for CDC sessions based on commitment control attributes on PWX CDC connections.

Commit processing is not controlled by a single commitment control attribute. When setting these attributes, try to balance performance and resource consumption with latency requirements.

The Maximum Rows Per commit, Real-Time Flush Latency in milli-seconds, and UOW Count attributes control the timing of real-time flushes of change data to the targets. The Minimum Rows Per commit attribute controls if a commit can occur.

Set one or more of the following commitment control attributes on PWX CDC connections:

Maximum Rows Per commit

Maximum number of change records in a source UOW that PWXPC processes before flushing the data buffer to commit the change data to the targets.

Use this attribute to have PWXPC commit change data to the targets without waiting for the UOW boundary, or end-UOW, to be met. This type of commit is called a subpacket commit. By using subpacket commits for large UOWs, you can minimize use of storage on the PowerCenter Integration Service machine and locking contention on the target databases.

Attention: Because PWXPC can commit the change data to the targets between UOW boundaries, relational integrity (RI) might be compromised. Do not use this connection attribute if you have targets in the CDC session with RI constraints.

After the maximum rows limit is met, PWXPC flushes the change data from the buffer on the PowerCenter Integration Service machine and commits the data to the targets. PWXPC also writes message PWXPC_12128 to the session log. After commit processing completes, the RDBMS releases locks on the target databases and PowerCenter Integration Service can reuse the buffer space for additional change records.

The maximum rows limit is cumulative across all sources in the CDC session. PWXPC issues a real-time flush when the limit is met, regardless of the number of sources with changes.

PWXPC resets the maximum rows limit when a real-time flush occurs. The flush can occur because of the maximum rows limit, UOW count limit, or real-time flush latency timer.

If PWXPC reaches a UOW boundary and the maximum row limit has not been met, PWXPC continues to process change records across UOW boundaries.

Use a maximum rows limit if you have extremely large UOWs in the change stream that might cause the following problems:

- Locking issues on the target database
- Resource issues on the PowerCenter Integration Service node

For example, you have a large UOW with 10,000 updates for a single source, and you set the Maximum Rows per Commit attribute to 1000. In this case, PWXPC issues a subpacket commit after each 1,000 change records.

Or, you might have a UOW that contains updates for more than one source. For example, the UOW contains 900 updates for source 1, 100 updates for source 2, and then 500 more updates for source 1. If you set the Maximum Rows per Commit attribute to 1000, PWXPC issues a subpacket commit after reading 1,000 change records, or after processing the updates for source 2.

Default is 0, which causes PWXPC to not use this maximum rows limit. If you specify 0 or do not enter a value for the maximum rows limit, commits occur only on UOW boundaries.
If you specify a low maximum rows limit, the CDC session uses more system resources on the PowerCenter Integration Service machine and target systems. This increased resource use occurs because PWXPC flushes data to the targets more frequently.

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

**Minimum Rows Per commit**

Minimum number of change records that PowerExchange must pass to PWXPC before passing a commit record. Until the minimum rows limit is met, PowerExchange discards any commit records that it reads from the change stream and passes only change records to PWXPC. After the minimum rows limit is met, PowerExchange passes the next commit record it encounters to PWXPC and then resets the minimum rows counter.

If the change stream has many small UOWs, you can set the Minimum Rows Per commit attribute to create larger UOWs of a more uniform size. Online transactions that run in transaction control systems such as CICS and IMS often commit after only a few changes, which results in many, small UOWs in the change stream. PowerExchange and PWXPC process a few large UOWs more efficiently than many small UOWs. By using the minimum rows limit to increase the size of UOWs, you can improve CDC processing efficiency.

The minimum rows limit does not impact the relational integrity of the change data because PowerExchange does not create additional commit points in the change data. PowerExchange skips some of the original commit records in the change stream.

Default is 0, which causes PowerExchange to not use this minimum rows limit.

If you enter a minimum rows limit, PowerExchange changes the number of change records in a UOW to match or exceed this limit.

**Note:** PWXPC does not commit change data to the targets based on the minimum rows limit. PWXPC commits change data to the targets based on the Maximum Rows Per commit, Real-Time Flush Latency in milli-seconds, and UOW Count attributes.

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

For real-time or continuous extraction mode, the number of milliseconds that must elapse before PWXPC flushes the data buffer to commit change data to the targets. After the flush latency interval expires and PWXPC reaches a UOW boundary, PWXPC issues a real-time flush to commit change data and restart tokens to the targets. PWXPC also writes message PWXPC_10082 in the session log.

PWXPC resets the flush latency interval when a real-time flush occurs. The flush can occur because of the maximum rows limit, UOW count limit, or real-time flush latency timer.

Valid values for the real-time flush latency are:

- -1. Disables data flushes based on time.
- 0 through 2000. Sets the interval to 2000 milliseconds, or 2 seconds.
- 2000 through 86400. Sets the interval to the specified number of milliseconds.

Default is 0.

If you set the flush latency interval value is 0 or greater, PWXPC flushes the change data for all complete UOWs after the interval expires and the next UOW boundary occurs. The lower you set the flush latency interval, the faster PWXPC commits change data to the targets. If you require a low latency for applying changes to the targets, enter a low value for the flush latency interval.
However, if you specify a low flush latency interval, the CDC session might consume more system resources on the PowerCenter Integration Service and target systems. This increased consumption occurs because PWXPC commits change data to the targets more frequently.

UOW Count

Number of complete UOWs that PWXPC reads from the change stream before flushing change data to the targets. When PWXPC reads change data from PowerExchange and provides it to the source qualifier in the CDC session, the count of the UOWs begins.

After the UOW count limit is met, PWXPC issues a real-time flush to commit the change data and restart tokens to the targets. PWXPC also writes message PWXPC_10081 in the session log.

PWXPC resets the UOW count after a real-time flush occurs because of the UOW count limit or the real-time flush latency interval.

Valid values for UOW count are:

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

Default is 1.

The lower you set the UOW count value, the faster the PowerCenter Integration Service commits change data to the target. If you require the lowest possible latency, enter a UOW count of 1. However, a low latency might result in the session using more system resources on the PowerCenter Integration Service and the target systems.

Attention: In the session properties, verify that the Commit Type attribute specifies Source and that the Commit at End of File attribute is disabled. The Commit at End of File attribute is enabled by default. If you accept the default, the PowerCenter Integration Service writes additional data to the targets after the CDC reader has committed the restart tokens and shut down. When you restart the CDC session, the session might write duplicate data to the targets.

Examples of Controlling Commit Processing

Review the following examples to learn how to use the commitment control attributes to control commit processing with PWXPC.

Example 1. Subpacket Commit and UOW Count

This example uses the Maximum Rows Per commit and UOW Count attributes to control commit processing.

The change data is composed of UOWs of the same size. Each UOW contains 1,000 change records.

The following table describes the commitment control attribute values that this example uses:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows Per commit</td>
<td>300</td>
</tr>
<tr>
<td>Minimum Rows Per commit</td>
<td>0, which disables this attribute</td>
</tr>
<tr>
<td>Real-time Flush Latency in milli-seconds</td>
<td>0, which is equivalent to 2 seconds</td>
</tr>
<tr>
<td>UOW Count</td>
<td>1</td>
</tr>
</tbody>
</table>
Based on the maximum rows value, PWXPC flushes the data buffer after reading the first 300 records in a UOW. This action commits the change data to the targets. PWXPC continues to commit change data to the targets every 300 records.

PWXPC commits on UOW boundaries only for the UOW count and real-time flush latency interval. If the real-time flush latency interval expires before PWXPC reads 300 change records, PWXPC still commits based on the maximum rows value because that threshold is met before a UOW boundary occurs.

When the end of the UOW is read, PWXPC commits the change data because the **UOW Count** value is 1. PWXPC resets the UOW and maximum row counters and the real-time flush latency timer each time it commits. Because all of the UOWs have the same number of change records, PWXPC continues to read change data and to commit the data to the targets at the same points in each UOW.

In this example, PWXPC commits change data at the following points:
- 300 change records based on the maximum rows value
- 600 change records based on the maximum rows value
- 900 change records based on the maximum rows value
- 1,000 change records based on the UOW count value

**Example 2. UOW Count and Time-Based Commits**

This example uses the **UOW Count** and **Real-time Flush Latency in milli-seconds** attributes to control commit processing. The change data consists of UOWs of varying sizes.

The following table describes the commitment control attribute values that this example uses:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows Per commit</td>
<td>0, which disables this attribute</td>
</tr>
<tr>
<td>Minimum Rows Per commit</td>
<td>0, which disables this attribute</td>
</tr>
<tr>
<td>Real-time Flush Latency in milli-seconds</td>
<td>5000, which is equivalent to 5 seconds</td>
</tr>
<tr>
<td>UOW Count</td>
<td>1000</td>
</tr>
</tbody>
</table>

Initially, PWXPC reads 900 complete UOWs in 5 seconds. Because the real-time flush latency interval has expired, PWXPC flushes the data buffer to commit the change data to the targets. PWXPC then resets both the UOW counter and real-time flush latency timer. When PWXPC reaches UOW 1,000, PWXPC does not commit change data to the targets because the UOW counter was reset to 0 after the last commit.

PWXPC reads the next 1,000 UOWs in 4 seconds, which is less than the real-time flush latency timer. PWXPC commits this change data to the target because the UOW counter has been met. After this commit, PWXPC then resets the real-time flush latency timer and the UOW counter.

PWXPC continues to read change data and commit the data to the targets, based on the UOW count or the real-time flush latency flush time, whichever limit is met first.

In this example, PWXPC commits change data at the following points:
- After UOW 900 because the real-time latency flush latency timer matched first.
- After UOW 1,900 because the UOW count matched first during the second commit cycle.
Example 3. Minimum Rows and UOW Count

This example uses the **Minimum Rows Per commit** and **UOW Count** attributes to control commit processing. The change data consists of UOWs of the same size. Each UOW contains ten change records. The following table describes the commitment control attribute values that this example uses:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Rows Per commit</td>
<td>0, which disables this attribute</td>
</tr>
<tr>
<td>Minimum Rows Per commit</td>
<td>100</td>
</tr>
<tr>
<td>Real-time Flush Latency in milli-seconds</td>
<td>-1, which disables this attribute</td>
</tr>
<tr>
<td>UOW Count</td>
<td>10</td>
</tr>
</tbody>
</table>

PWXPC passes the minimum rows value to PowerExchange and requests change data from the change stream. Because the minimum rows value is 100, PowerExchange skips the commit records of the first nine UOWs. When PowerExchange reads the last change record in the tenth UOW, the minimum rows limit is met. So, PowerExchange passes the commit record for the tenth UOW to PWXPC and resets the minimum rows counter. PWXPC increases the UOW counter to one.

PowerExchange and PWXPC continue to read the change data until the UOW counter is 10. At this point, PWXPC flushes the data buffer to commit the change data to the targets and resets the UOW counter. PWXPC commits change data after 1,000 change records, or after every 10 UOWs, because each UOW contains 10 change records and the **UOW Count** is 10.

Recovery and Restart Processing for CDC Sessions

If you select the **Resume from last checkpoint** option for the **Recovery Strategy** attribute in a CDC session that extracts change data, PWXPC and PowerCenter provide recovery and restart processing for that session.

If a session fails, the PowerCenter Integration Service recovers the session state of operation, and PWXPC recovers the restart information.

PWXPC saves restart information for all sources that are in a CDC session. The restart information for CDC sessions, including the restart tokens, originates from PowerExchange on the system from which the change data is extracted. You can include both relational and nonrelational targets in a single CDC session. PWXPC uses one of the following locations to store and retrieve restart information, based on the target type:

- For relational targets, PWXPC uses recovery state tables in the target databases. PWXPC, in conjunction with the PowerCenter Integration Service, commits both the change data and the restart tokens for that data in the same commit operation. This commit ensures that the applied data and the restart tokens are in-sync.

- For nonrelational targets, PWXPC uses the recovery state file that is in the shared location on the PowerCenter Integration Service machine. PWXPC, in conjunction with the PowerCenter Integration Service, writes the change data to the target files and then writes the restart tokens to the recovery state file. As a result, duplicate data might be applied to the targets when you restart the failed CDC sessions.

The PowerCenter Integration Service saves the session state of operation and maintains target recovery tables. The PowerCenter Integration Service stores the session state of operation in the shared location that
is specified in $PMStorageDir. The PowerCenter Integration Service saves relational target recovery information in the target database.

When you run a CDC session that uses a resume recovery strategy, PWXPC writes the following message to the session log to indicate that recovery is in effect:

```
PWXPC_12094 [INFO] [CDCRestart] Advanced GMD recovery in effect. Recovery is automatic.
```

When you recover or restart a CDC session, PWXPC uses the saved restart information to resume reading the change data from the point of interruption. The PowerCenter Integration Service restores the session state of operation, including the state of each source, target, and transformation. PWXPC, in conjunction with the PowerCenter Integration Service, determines how much of the source data it needs to reprocess. PowerExchange and PWXPC use the restart information to determine the correct point in the change stream from which to restart extracting change data and then applying it to the targets.

If you run a session with resume recovery strategy and the session fails, do not change the mapping, the session, or the state information before you restart the session. PowerCenter and PWXPC cannot guarantee recovery if you make any of these changes.

**Restriction:** If any of the targets in the CDC session use the PowerCenter File Writer to write CDC data to flat files, do not use a resume recovery strategy. Restart tokens for all targets in the CDC session, including relational targets, will be compromised if a flat file target is in the same session. Data loss or duplication might occur.

### PowerCenter Recovery Tables for Relational Targets

When the PowerCenter Integration Service runs a CDC session that has a resume recovery strategy, it writes information to recovery tables on the target database system.

When the PowerCenter Integration Service recovers the session, it uses the information in the recovery tables to determine where to begin loading data to target tables. PWXPC also uses information in the recovery tables to determine where to begin reading the change stream.

If you want the PowerCenter Integration Service to create the recovery tables, grant table creation privileges to the database user name that is configured in the target database connection. Otherwise, you must create the recovery tables manually.

For relational targets, the PowerCenter Integration Service creates the following recovery tables in the target database:

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM_RECOVERY</td>
<td>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.</td>
</tr>
<tr>
<td>PM_TGT_RUN_ID</td>
<td>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.</td>
</tr>
<tr>
<td>PM_REC_STATE</td>
<td>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.</td>
</tr>
</tbody>
</table>
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 that PWXPC creates by appending the task instance ID of the CDC session to the value that you specify in the **Application Name** attribute in the source PWX CDC application connection. When this value matches an APPL_ID value for a row in the state table, the PowerCenter Integration Service, in conjunction with PWXPC, selects the row from the state table for the CDC session.

**STATE_DATA**

Contains the restart information for the session in a variable-length, 1,024-byte binary column. When the PowerCenter Integration Service commits change data to the targets tables, it also commits the restart information for that data in this column. PWXPC uses the restart information from this column to perform restart processing for the CDC session.

If the amount of restart information for a session exceeds 1,024 bytes, the PowerCenter Integration Service adds additional rows to accommodate the remainder of the restart information. For each row added, the PowerCenter Integration Service increases the value of the SEQ_NUM column by one, starting from zero.

**PowerCenter Recovery Files for Nonrelational Targets**

If you configure a resume recovery strategy for a CDC session, the PowerCenter Integration Service stores the session state of operation in the shared location, $PMStorageDir, on the PowerCenter Integration Service machine. For nonrelational targets, the PowerCenter Integration Service also stores the target recovery status in a recovery state file in the shared location on the PowerCenter Integration Service machine. PWXPC stores the restart information for nonrelational target files in this state file.

**Recovery State File**

For all nonrelational targets in a CDC session, the PowerCenter Integration Service uses a recovery state file on the PowerCenter Integration Service machine.

Nonrelational target files include MQ Series message queues, PowerExchange nonrelational targets, and other PowerCenter nonrelational targets.
CDC sessions with heterogeneous target tables have state table entries in each unique relational target
database and an entry in a state file on the PowerCenter Integration Service machine for each nonrelational
target.

The PowerCenter Integration Service creates the recovery state file in the shared location, $PMStorageDir.
The file name has the following prefix:

```
   pm_rec_state_appl_id
```

PWXPC creates the value for the `appl_id` variable in the file name by appending the task instance ID of the
CDC session to the value that you specify in the Application Name attribute in the source PWX CDC
application connection. The PowerCenter Integration Service uses various task and workflow repository
attributes to complete the file name. The message CMN_65003, which the PowerCenter Integration Service
writes to the session log, contains the complete file name.

**Application Names**

PWXPC, in conjunction with the PowerCenter Integration Service, uses the application name you specify as
part of the key when it stores and retrieves the restart information for a CDC session.

When you configure the PWX CDC application connection for a CDC session, enter a unique value for the
Application Name attribute. PWXPC appends the repository task instance ID for the CDC session to this value
to create the APPL_ID value in the recovery state table and the `appl_id` portion of the recovery state file name.

Because the value of the APPL_ID column and the state recovery file contains the task instance ID for the
CDC session, changes to the session can affect restart processing. If you add or remove sources or targets
for a CDC session, you must use the restart token file to provide restart tokens and then cold start the
session.

**Restart Processing for CDC Sessions by Start Type**

How you start a CDC session affects how PWXPC determines the restart points for sources in the session. Each source has its own restart point.

For each start type, PWXPC determines the restart point as follows:

- For a cold start, PWXPC uses the restart token file to acquire restart tokens for all data sources. PWXPC
does not read the state tables or state file and does not attempt to recover the session. The CDC session
continues to run until it is stopped or interrupted.

- For a warm start, PWXPC reconciles the restart tokens that are in the restart token file with the restart
tokens in the state tables and state file. If necessary, PWXPC performs recovery processing. The session
continues to run until it is stopped or interrupted.

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

Before you run a CDC session for the first time, create and populate the restart token file with restart token
pair for each source in the session. Each restart token pair should match a point in the change stream where
the source and target are in a consistent state.

For example, materialize a target table and stop update activity on the source. To define a start or restart
point, specify a special override statement that contains the CURRENT_RESTART option in the restart token
file. Use the restart token file that has the file name that matches the restart token file name in the PWX CDC
application connection. When you cold start the CDC session, PWXPC requests that PowerExchange use the
current end-of-log as the extraction start point. You can then resume update activity on the sources.

If you cold start a CDC session and a restart token file does not exist, the PowerCenter Integration Service
runs the session. PWXPC passes Null restart tokens for all sources to PowerExchange. PowerExchange
issues message PWXPC_12060 to indicate that the restart tokens for each source are Null and then assigns the default restart point to each source.

**Attention:** If you use Null restart tokens, the CDC session might have incorrect results. Provide valid restart tokens when you cold start CDC sessions.

### Default Restart Points for Null Restart Tokens

If PowerExchange receives null restart tokens for all sources in a CDC session, it uses the default restart points.

For all z/OS data sources, the default restart points vary by extraction mode, as follows:

- For batch extraction mode and continuous extraction mode, the default restart point is the oldest condense file that is recorded in the CDCT file.
- For real-time extraction mode, the default restart point is the best available restart point, as determined by the PowerExchange Logger for MVS. The best available restart point is one of the following:
  - Oldest restart point for which an archive log is available.
  - The current active log if no archive logs are available.

PowerExchange uses the default restart point only if all sources in a CDC session have null restart tokens. If some sources have non-null restart tokens, PWXPC assigns the oldest restart point from those tokens to any sources for which no restart tokens are specified.

For example, a new CDC session contains the sources A, B, and C. The restart token file contains restart tokens for sources A and B. The restart point for source A is older than that for source B. Source C does not have existing or supplied restart tokens. Because some sources in the CDC session have explicit restart points, PWXPC does not assign null restart tokens to source C. Instead, PWXPC assigns the restart point for source A to source C because this restart point is the oldest one supplied.

### Determining the Restart Tokens for Cold Start Processing

When you cold start a CDC session, PWXPC uses the restart token file to determine the restart tokens for all sources. PWXPC ignores any entries in the state tables or state file for the sources in the CDC session.

More specifically, PWXPC uses one of the following methods to determine the restart tokens:

- If the restart token file is empty or does not exist, PWXPC assigns null restart tokens to all sources in the CDC session.
- If the restart token file contains only explicit override statements, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the oldest supplied restart point to any sources for which an explicit override statement was not specified.
- If the restart token file contains only the special override statement, PWXPC assigns the restart tokens in the special override statement to all sources.
- If the restart token file contains a special override statement and explicit override statements, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the restart tokens in the special override statement to all remaining sources.
Determining the Restart Tokens for Warm Start Processing

When you warm start a CDC session, PWXPC uses the state tables and state file, in conjunction with restart token file, to determine the restart tokens for all sources.

More specifically, PWXPC uses one of the following methods to determine the restart tokens:

- If the restart token file is empty or does not exist and there is no matching entry in a state table or state file, PWXPC assigns null restart tokens to all sources in the session.

- If the restart token file is empty or does not exist and if some but not all sources have a matching entry in a state table or a state file, PWXPC performs the following processing:
  - Assigns any restart tokens found in a state table and state file to the appropriate sources.
  - Assigns the oldest available restart point to all sources that do not have restart tokens.

- If the restart token file is empty or does not exist and if all sources have an entry in a state table or state file, PWXPC uses the restart tokens from the state tables or state file.

- If the restart token file contains explicit override statements and no sources have a matching entry in a state table or no state file, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the oldest supplied restart point to all sources that do not have restart tokens.

- If the restart token file contains explicit override statements and if some but not all sources have a matching entry in a state table or a state file, PWXPC performs the following processing:
  - Assigns the restart tokens 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 from state tables or the state file to all remaining sources that do not have restart tokens supplied in the restart token file.

- If the restart token file contains only the special override statement, PWXPC assigns the restart tokens in the special override statement to all sources.

- If the restart token file contains a special override statement and explicit override statements, PWXPC performs the following processing:
  - Assigns the restart tokens in the explicit override statements to the specified sources.
  - Assigns the restart tokens in the special override statement to all remaining sources.
Creating Restart Tokens for Extractions

Before you begin extracting change data, you must create restart tokens to indicate the extraction start point.

When generating restart tokens, consider the following points:

- The optimal start point matches the point in the change stream at which you last synchronized the source and target. This point marks the end of the change stream, or current end-of-log (EOL), if you stop update activity on the source, as recommended, until after target materialization and restart token generation are complete.
- On z/OS, the sequence tokens have the same length for all source types.
- If you use PowerExchange Condense with Full condense processing, PowerExchange uses the sequence token to determine the point from which to start reading change data from condense files, and uses the restart token to verify that the source instance is correct for the starting change record. The sequence token represents the full condense file and the position of the change record in that file. The restart token contains the source instance name from the registration group.

To create restart tokens for the current EOL, use one of the following methods:

**PWXPc restart token file**

To generate current restart tokens for a CDC session that uses real-time or continuous extraction mode, specify the CURRENT_RESTART option on the RESTART1 and RESTART2 special override statements in the PWXPc restart token file. When the CDC session runs, PWXPc requests that PowerExchange provide restart tokens for the current EOL. PWXPc uses the restart information to locate the extraction start point.

**Database row test**

In the PowerExchange Navigator, perform a database row test with a SELECT CURRENT_RESTART SQL statement.

**DTLUAPPL utility**

Run the DTLUAPPL utility with the GENERATE RSTKKN option.

If you use the DTLUAPPL utility or PowerExchange Navigator to generate restart tokens, enter the token values in the PWXPc restart token file before you start the CDC session.

You can also construct restart tokens by using the RBA or LRSN of an event mark record in the PowerExchange Logger log files. You can use the EDMXLUTL utility to generate event marks. Also, the following PowerExchange ECCRs for z/OS data sources generate event marks in some situations:

- 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.
- The IMS log-based ECCR generates an event mark when it reads records that the DTLCUIML utility created in the IMS logs.
- The Adabas ECCR generates an event mark when it reads an Adabas PLOG data set.
Displaying Restart Tokens

You can display restart token values in the output from a database row test, extraction session, or DTLUAPPL PRINT function.

If you run a database row test on an extraction map from the PowerExchange Navigator, the output includes a restart token pair for each row of change data. The following columns show the token values:

- DTL__CAPXRESTART1 shows the sequence token value.
- DTL__CAPXRESTART2 shows the restart token value.

If you include the DTL__CAPXRESTART1 and DTL__CAPXRESTART2 columns in the PowerCenter source definition, PowerExchange provides the restart tokens for each row when you extract change data in a CDC session.

When a CDC session runs, PowerExchange and PWXPC display restart token values in the following messages:

- In messages PWX-04565 and PWX-09959, the sequence token is in the Sequence field and restart token is in the PowerExchange Logger field.
- In messages PWXPC_12060 and PWXPC_12068, the sequence token is in the Restart Token 1 field and the restart token is in the Restart Token 2 field.
- In messages PWXPC_10081, PWXPC_10082, and PWXPC_12128, the sequence token is the first token value and the restart token is the second token value.

If you use the DTLUAPPL utility to generate restart tokens, you can use the PRINT statement to display the generated values. In the PRINT output, DTLUAPPL displays the sequence token, without the usual trailing eight zeros, in the Sequence field and displays the restart token in the Restart field.

Configuring the Restart Token File

When you configure the CDC session in PowerCenter, specify the name and location of the restart token file.

To specify the restart token file, enter the following attributes on the PWX CDC application connection for the source:

**RestartToken File Folder**

Enter the name of the directory that contains the restart token file. If you use the default value of $PMRootDir/Restart and the Restart directory does not exist, PWXPC creates the directory. PWXPC does not create any restart token directory under another name.

**RestartToken File Name**

Enter a unique name for the restart token file. If you do not specify this value, PWXPC uses the value in the **Application Name** attribute, if present. Otherwise, PWXPC uses the workflow name. Because this name must be unique, Informatica recommends that you always specify a value for the **RestartToken File Name** attribute.

When you run a CDC session, PWXPC verifies that the restart token file exists. If one does not exist, PWXPC uses the name specified in this attribute to create an empty restart token file.

**Restriction:** The value of **RestartToken File Name** attribute in must be unique for every CDC session. Non-unique file names can cause unpredictable results, such as change data loss and session failures.
To find the restart token file name for a CDC session, use the following methods:

- For CDC sessions that have run, look for message PWXPC_12057 in the session log. This message indicates the restart token file directory and file name.
- In Workflow Manager, look for the restart token file folder and file name in the attributes on the PWX CDC application connection associated with the source in the CDC session. If the restart token file name is not present, PWXPC uses the application name, if specified. Otherwise, PWXPC uses the workflow name.

Before you run a CDC session the first time, configure the restart token file to indicate the point in the change stream from which to start extracting change data. Later, you might need to modify the restart token file to add sources to a CDC session or to indicate the point from which to restart change data extraction.

**Restart Token File Statements**

You can specify explicit override, special override, and comment statements in the restart token file.

Use these statements as follows:

- **Explicit override.** Use this statement type to specify a restart token pair for a specific source. You must provide the PowerExchange extraction map name.
- **Special override.** Use this statement type to 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.
- **Comment.** Use this statement type to enter any comments that you want to add to the file.

**Restart Token File Syntax by Statement Type**

In the restart token file, you can specify explicit override statements, special override statements, and comments.

For explicit override control statements, use the following syntax:

```
extraction_map_name={sequence_token|CURRENT_RESTART}
extraction_map_name={restart_token|CURRENT_RESTART}
```

For special override control statements, use the following syntax:

```
RESTART1={sequence_token|CURRENT_RESTART}
RESTART2={restart_token|CURRENT_RESTART}
```

For comments, use the following syntax:

```
<!-- comment_text
```

The following syntax 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:**

```
<!--
```
- **In each file, you can specify one or more sets of explicit override statements and one set of special override statements.**
- **Explicit override statements for a source take precedence over any set of special override statements.**
Explicit Override Statements

Use explicit override statements to specify a restart token pair for a specific source.

You can specify explicit override statements for one or more sources in a CDC session. Also, you can use explicit override statements in conjunction with special override statements to provide restart tokens for all sources in a CDC session.

When you warm start a CDC session, the explicit override statements for a source override the restart tokens in the state table or state file for that source.

When defining explicit override statements for a source, specify a pair of statements, each containing an extraction map name and a restart or sequence token value. Because a source can have multiple extraction maps with distinct names, you might have multiple pairs of explicit override statements for a source.

Explicit override statements use the following parameters:

- `extraction_map_name=(restart1_token|CURRENT_RESTART)`
- `extraction_map_name=(restart2_token|CURRENT_RESTART)`

Review the following parameter descriptions:

- **extraction_map_name**

  The name of an extraction map for the data source. To determine the extraction map name, use one of the following methods:

  - For CDC data map sources, see the Schema Name Override and Map Name Override attributes in the session properties. These attributes override the schema name and map name in the source extraction map. Or, in Designer, see the Schema Name and Map Name values in the source Metadata Extensions.
  
  - For relational sources, see the Extraction Map Name attribute in the session properties.

- **restart1_token**

  The sequence token portion of a restart token pair. This value varies based on data source type.

- **restart2_token**

  The restart token portion of a restart token pair. This value based on data source type.

- **CURRENT_RESTART**

  PowerExchange generates restart tokens for the current end of the change stream. The PWXPC CDC reader opens a separate connection to PowerExchange, requests the generation of current restart tokens, and then provides the token values to all applicable sources.

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

**Restriction:** Use CURRENT_RESTART only for CDC sessions that use real-time extraction mode or continuous extraction mode.

Special Override Statement

Use a special override statement to specify or generate restart tokens for one or more sources.

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

When you warm start a CDC session, the special override statement overrides the restart tokens in the state table or state file for all sources except those specified in explicit override statements.
A special override statement is composed of a pair of RESTART1 and RESTART2 statements, as shown in the following syntax:

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

Specify only one set of these special override statements in the restart token file.

In RESTART1 and RESTART2 statements, use the following parameters to specify either a pair of sequence and restart token values or the current end of the change stream:

- **restart1_token**
  - The sequence token portion of a restart token pair. This value varies based on data source type.

- **restart2_token**
  - The restart token portion of a restart token pair. This value varies based on data source type.

- **CURRENT_RESTART**
  - PowerExchange generates restart tokens for the current end of the change stream. The PWXPC CDC reader opens a separate connection to PowerExchange, requests the generation of current restart tokens, and then provides the token values to all applicable sources.

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

  **Restriction:** Use CURRENT_RESTART only for CDC sessions that use real-time extraction mode or continuous extraction mode.

### Comment Statements

You can insert a comment statement anywhere in the restart token file.

Comment statements must begin with:

```
<!--
```

### Example Restart Token File

This example restart token file is for a CDC session with seven source tables. The file includes explicit override statements that provide restart tokens for three source tables and a special override statement that provides restart tokens for the remaining tables.

The restart token file contains the following statements:

```
<!-- Restart Tokens for existing tables -->
Restart1=000000AD7756000000000000AD77560000000000000000
Restart2=C1E4E2D34040000000AD5F2C00000000
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_001 -->
dldsn9.rrtb0001_RRTB_SRC_001=0000006D10B20000000013006D10B200000000000000000
(dldsn9.rrtb0001_RRTB_SRC_001=C1E4E2D340400000013FF36200000000)
<-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_002 -->
dldsn9.rrtb0002_RRTB_SRC_002=000000A371950000000000A3719500000000000
(dldsn9.rrtb0002_RRTB_SRC_002=C1E4E2D3404000000968FC60000000)
<!-- Restart Tokens for the Table: rrtb0001_RRTB_SRC_003 -->
dldsn9.rrtb0003_RRTB_SRC_003=0000006D84E7800000000006D84E7800000000000
(dldsn9.rrtb0003_RRTB_SRC_003=C1E4E2D34040000006D1E61000000)
```

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:

```
<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>didsn9.rttb0001_RRTB_SRC_001</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file</td>
</tr>
<tr>
<td>didsn9.rttb0002_RRTB_SRC_002</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file</td>
</tr>
<tr>
<td>didsn9.rttb0003_RRTB_SRC_003</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file (special override)</td>
</tr>
<tr>
<td>didsn9.rttb0004_RRTB_SRC_004</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file</td>
</tr>
<tr>
<td>didsn9.rttb0005_RRTB_SRC_005</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file (special override)</td>
</tr>
<tr>
<td>didsn9.rttb0006_RRTB_SRC_006</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file (special override)</td>
</tr>
<tr>
<td>didsn9.rttb0007_RRTB_SRC_007</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>000000000000000000000000000000000000000000000000000000000</td>
<td>Restart file (special override)</td>
</tr>
</tbody>
</table>
```

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

For the sources to which PWXPC assigns the special override restart tokens, PWXPC writes "Restart file
(special override)" in the Source column.
Chapter 17

Managing Change Data Extractions

This chapter includes the following topics:

- Starting PowerCenter CDC Sessions, 349
- Stopping PowerCenter CDC Sessions, 351
- Changing PowerCenter CDC Sessions, 353
- Recovering PowerCenter CDC Sessions, 355

Starting PowerCenter CDC Sessions

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

Also, you can start the entire workflow, part of a workflow, or a task in the workflow.

Use one of the following methods to start a CDC session:

Cold start

To cold start a CDC session, use the Cold Start command in Workflow Manager or Workflow Monitor. You can also use the pmcmd starttask or startworkflow commands with the norecovery option. A CDC session that uses real-time or continuous extraction mode runs continuously until it is stopped or interrupted. A CDC session that uses batch extraction mode runs until it reaches the end of log (EOL) or it is stopped or interrupted.

When you cold start a CDC session, PWXPC uses the restart token file to acquire restart tokens for all sources. PWXPC does not read the state tables or file or makes any attempt to recover the session.

Warm start

To warm start a CDC session, use the Start or Restart commands in Workflow Manager or Workflow Monitor. You can also use the pmcmd starttask or startworkflow commands. A CDC session that uses real-time or extraction mode runs continuously until it is stopped or interrupted. A CDC session that uses batch extraction mode runs until it reaches EOL or it is stopped or interrupted.

When you warm start a CDC session, PWXPC reconciles any restart tokens provided in the restart token file with any restart tokens that exist in the state tables or file. If necessary, PWXPC performs recovery processing.
Recovery start

To start recovery for a CDC session, use the Recover command from Workflow Manager or Workflow Monitor. You can also use the pmcmd recoverworkflow command or the starttask or startworkflow commands with the recovery option. When recovery completes, the CDC session ends.

When you recover a CDC session, PWXPC reads the restart tokens from any applicable state tables or state file. If necessary, PWXPC performs recovery processing. PWXPC updates the restart token file with the restart tokens for each source in the CDC session. Then the session ends. To begin extracting change data again, either cold start or warm start the session.

Cold Start Processing

To cold start workflows and tasks, use the Cold Start command in Workflow Manager or Workflow Monitor. Alternatively, you can use the pmcmd starttask or startworkflow commands with the norecovery option.

After you request a cold start for a CDC session, the following processing occurs:

1. PWXPC writes the following message in the session log:
   PWXPC_12091 [INFO] [CDCRestart] Cold start requested
2. PWXPC reads the restart tokens from only the restart token file and associates a restart token with each source in the session.
3. PWXPC creates the initialization restart token file with the initial restart tokens.
4. PWXPC commits the restart tokens for each source to the appropriate state tables or state file and then writes message PWXPC_12104 to the session log.
5. PWXPC passes the restart tokens to PowerExchange. PowerExchange begins extracting change data and passing the data to PWXPC for processing.
6. PWXPC continues processing change data from PowerExchange and committing the data and restart tokens to the targets. This processing continues until the session ends or you stop it.

Warm Start Processing

To warm start workflows and tasks, use the Start or Restart command in Workflow Manager or Workflow Monitor. Alternatively, you can use the pmcmd starttask or startworkflow commands.

When you warm start a workflow or task, PWXPC automatically performs recovery. You do not need to recover failed workflows and tasks before you restart them.

After you request a warm start for a CDC session, the following processing occurs:

1. PWXPC writes the following message in the session log:
   PWXPC_12092 [INFO] [CDCRestart] Warm start requested. Targets will be resynchronized automatically if required
2. PWXPC queries the PowerCenter Integration Service about the commit levels of all targets. If all targets in the session have the same commit level, PWXPC skips recovery processing.
3. PWXPC reconciles the restart tokens from the restart token file and from the state tables or file.
   Restriction: If a CDC session requires recovery processing, PWXPC does not use the restart token file. Consequently, you cannot override restart tokens for sources.
4. PWXPC creates the initialization restart token file with the reconciled restart tokens.
5. If recovery is required, PWXPC re-reads the change data for the last unit-of-work (UOW) that was committed to the targets with the highest commit level and then flushes the data to the targets with lower commit levels. The PowerCenter Integration Service commits flushed change data and restart tokens to any relational targets and updates any nonrelational files.
6. If recovery is not required and the reconciled restart tokens differ from those in the state tables or state file, PWXPC commits the reconciled restart tokens and then writes message PWXPC_12104 to the session log.

7. PWXPC passes the restart tokens to PowerExchange. PowerExchange begins extracting change data and passing the data to PWXPC for processing.

8. PWXPC continues processing change data from PowerExchange and commits the data and restart tokens to the targets. This processing continues until the session ends or you stop it.

Recovery Processing

To recover workflows and tasks, use the Recover command in Workflow Manager or Workflow Monitor. Alternatively, you can use the pmcmd recoverworkflow command, or the starttask or startworkflow command with the recovery option.

Use the recovery start method to populate the restart token file with the restart tokens for all sources in a CDC session. You can then cold start the CDC session or verify that the targets and restart tokens are in a consistent state. However, you do not need to recover failed workflows and tasks before you restart them because PWXPC automatically performs recovery processing when you warm start a workflow or task.

After you request recovery for a CDC session, the following processing occurs:

1. PWXPC writes the following message in the session log:
   PWXPC_12093 [INFO] [CDCRestart] Recovery run requested. Targets will be resynchronized if required and processing will terminate

2. PWXPC queries the PowerCenter Integration Service about the commit levels of all targets. If all targets in the session have the same commit level, PWXPC skips recovery processing.

3. PWXPC reads the restart tokens from the recovery state tables or state file.

   **Restriction:** If a CDC session requires recovery processing, PWXPC does not use the restart token file. Consequently, you cannot override restart tokens for sources.

4. PWXPC creates the initialization restart token file with the reconciled restart tokens.

5. If recovery is required, PWXPC re-reads the change data for the last UOW that was committed to the targets with the highest commit level and then flushes the data to the targets with lower commit levels. The PowerCenter Integration Service commits any flushed change data and restart tokens to any relational targets, and updates any nonrelational files.

6. PWXPC updates the restart token file with the final restart tokens, creates the termination restart token file, and ends.

To process change data from the point of recovery, warm start or cold start the workflow or task.

Stopping PowerCenter CDC Sessions

You can stop CDC sessions from PowerCenter or PowerExchange.

In PowerCenter, issue the Stop or Abort command in Workflow Monitor. Alternatively, use the pmcmd stoptask, stopworkflow, aborttask, or abortworkflow commands.

- If you issue the Stop command in Workflow Monitor or use the pmcmd stoptask or stopworkflow command, the PWXPC CDC reader and PowerCenter Integration Service complete processing all of the data in the pipeline and shut down. Then, the CDC session ends.
If you issue the Abort command in Workflow Monitor or use the pmcmd aborttask or abortworkflow command, the PowerCenter Integration Service waits 60 seconds to allow the readers and writers to complete processing all of the data in the pipeline and shut down. If the PowerCenter Integration Service cannot finish processing and committing data within this period, it kills the DTM process and ends the CDC session.

For more information about these PowerCenter commands, see the Informatica Command Reference or PowerCenter Workflow Basics Guide.

In PowerExchange, issue the PowerExchange Listener STOPTASK command in one of the following ways:

- From the command line on the system where extraction processing occurs
- From the PowerExchange Navigator
- With the DTLUTSK utility
- With the pwxcmd program

When you issue the STOPTASK command, PowerExchange stops the extraction task in the PowerExchange Listener and passes an EOF to the PowerCenter Integration Service. Then the CDC session ends. For more information about the STOPTASK command, see the PowerExchange Command Reference.

Stop Command Processing

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

**Note:** To stop CDC sessions and workflows, you can use the Stop command in Workflow Monitor or the pmcmd stoptask or stopworkflow command. Alternatively, you can use the PowerExchange STOPTASK command.

1. If you use a PowerCenter stop command, the PowerCenter Integration Service requests PWXPC to stop. If you use the PowerExchange STOPTASK command, PowerExchange sends an EOF to PWXPC.
2. When PWXPC receives an EOF, it flushes any complete and uncommitted UOWs and the associated restart tokens to the targets. PWXPC then writes messages PWXPC_12101 and PWXPC_12068 to the session log.
3. The PowerCenter Integration Service processes all of data in the pipeline and writes it to the targets.
4. The PowerCenter Integration Service sends an acknowledgment to PWXPC indicating that the targets have been updated.
5. PWXPC writes the termination restart token file, and then writes the message PWXPC_12075 to the session log.
6. The PWXPC CDC reader shuts down.
7. The PowerCenter Integration Service performs any post-session tasks and ends the session.

Terminating Conditions

You can have CDC sessions stop based on user-defined events or at EOL if you configure certain terminating conditions.

When PWXPC encounters a terminating condition, it stops reading change data from sources, flushes change data to the targets, and passes an EOF to the PowerCenter Integration Service. The PowerCenter Integration Service commits the data to the targets and ends the CDC session.

Use the following connection attributes and features as terminating conditions:
Event table processing

Create an event table and a capture registration for the table. Then specify the extraction map for the table in the **Event Table** attribute of the PWX CDC Real Time application connection for the CDC session. After PowerExchange reads a change record for the event table, it passes an EOF to PWXPC to end the CDC session.

Idle time

Enter 0 for the **Idle Time** attribute on a PWX CDC Real Time application connection. Then, whenever PowerExchange reaches EOL, it passes an EOF to PWXPC to end the CDC session.

Batch extraction mode

If you use batch extraction mode, PowerExchange reads all closed PowerExchange Condense condense files or PowerExchange Logger for Linux, UNIX, and Windows log files. Then PowerExchange passes an EOF to PWXPC to end the CDC session.

Changing PowerCenter CDC Sessions

Use this procedure to change CDC sessions. You might need to add or remove sources and targets.

After you change a CDC session, you must cold start it. Because a cold start is required, you must also get the latest restart tokens for the original sources before restarting the session. To do so, you can perform a recovery.

To change a CDC session:

1. Stop the workflow.
2. After the workflow ends, recover the CDC session.
   
   When you recover tasks, PWXPC writes the ending restart tokens for all sources in a CDC session to the restart token file that you specified on the PWX CDC application connection.
3. Make changes to the session or workflow, if necessary.
4. Verify that the restart token file in the source CDC connection points to the same restart token file updated in the recovery.
5. If you add sources to the CDC session, add statements to the restart token file that provide restart tokens for the additional sources.
6. If you remove sources from the CDC session, update the restart token file to remove their restart tokens.
7. Cold start the CDC session.

Examples of Adding Sources and Creating Restart Tokens

The following examples show how to add sources to CDC sessions and create restart tokens for those sources.

The first example uses the CURRENT_RESTART option of the special override statement in the restart token file to generate current restart tokens. The second example uses DTLUAPPL to generate current restart tokens.
Example 1. Creating Current Restart Tokens with Special Override Statements

This example adds a source table, RRTB_SRC_004, to a CDC session that has three other sources. You edit the restart token file to generate restart tokens that represent the current end of the change stream for the additional source.

In the restart token file, you define special override statements with CURRENT_RESTART option for the RRTB_SRC_004 source.

For the other three sources, you retain the existing restart points.

To add a source with CURRENT_RESTART restart tokens:

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

PWXPC writes the following messages in the session log:

```
PWXC_12060 [INFO]  [CDCRestart]
```

PWXPC also writes the restart tokens in the restart token file that is identified the CDC application connection attributes.

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

4. Edit the restart token file to add RESTART1 and RESTART2 special override statements that specify the CURRENT_RESTART option for the RRTB_SRC_004 source.

The updated file appears as follows:

```
<!-- existing sources
<idsn9.rrrb0001 RRTB_SRC_001=000000AD220F00000000000000000AD220F0000000000000
 idsn9.rrrb0001 RRTB_SRC_001=C1E4E2D3404000000AD09C0000000
 idsn9.rrrb0002 RRTB_SRC_002=000000AD220F00000000000000000AD220F0000000000000
 idsn9.rrrb0002 RRTB_SRC_002=C1E4E2D3404000000AD09C0000000
 idsn9.rrrb0003 RRTB_SRC_003=000000AD220F00000000000000000AD220F0000000000000
 idsn9.rrrb0003 RRTB_SRC_003=C1E4E2D3404000000AD09C0000000
<!-- new source
RESTART1=CURRENT_RESTART
RESTART2=CURRENT_RESTART
```

5. Cold start the session.

PWXPC connects to PowerExchange and generates restart tokens that match the current end of the change stream for the RRTB_SRC_004 source. PWXPC passes the generated restart tokens to PowerExchange to begin change data extraction. Because the restart points for the other sources are earlier than the restart point for RRTB_SRC_004, PWXPC does not pass any change data for RRTB_SRC_004 until it reads the first change after the generated restart point.

Example 2. Creating Current Restart Tokens with the DTLUAPPL Utility

This example adds the source table, RRTB_SRC_004, to a CDC session that has three other sources. You use the DTLUAPPL utility to generate restart tokens that represent the current end of the change stream.

For the other three sources, you retain the existing restart points.

1. In Workflow Monitor, use Stop command to stop the workflow.
2. After the workflow stops, select the Recover Task command to run a recovery session.
PWXPC writes the following messages in the session log:

PWXC_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>d1dsn.rttb0002</td>
<td>000000AD22OF00000000000000AD22OF0000000000000000</td>
<td>C1E4E2D3404000000AD0D9C00000000</td>
<td>QND storage</td>
</tr>
<tr>
<td>d1dsn.rttb0001</td>
<td>000000AD22OF00000000000000AD22OF0000000000000000</td>
<td>C1E4E2D3404000000AD0D9C00000000</td>
<td>QND storage</td>
</tr>
<tr>
<td>d1dsn.rttb0003</td>
<td>000000AD22OF00000000000000AD22OF0000000000000000</td>
<td>C1E4E2D3404000000AD0D9C00000000</td>
<td>QND storage</td>
</tr>
</tbody>
</table>

PWXPC also writes the restart tokens in the restart token file that is identified in the CDC application connection attributes.

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

4. Run the DTLUAPPL utility with RSTTKN GENERATE parameter to generate restart tokens that represent the current end of the change stream for the additional source.

Use the following DTLUAPPL control cards:

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

The PRINT command produces the following output:

```plaintext
Registration name=<rttb004.i> tag=<DB2DSN7rttb0041>
Sequence=<0000000000D8F240A000000000000> Restart <-C1E4E2D3404000000DBF238200000000>
```

You can add eight zeros to the end of the Sequence value to create the sequence value for the restart token file.

5. Edit the restart token file to add the source and its restart tokens.

The updated file contains the following lines:

```plaintext
<!-- existing sources
    d1dsn9.rttb0001_RRTB_SRC_001=000000AD22OF00000000000000AD22OF0000000000000000
    d1dsn9.rttb0001_RRTB_SRC_001=C1E4E2D3404000000AD0D9C00000000
    d1dsn9.rttb0002_RRTB_SRC_002=000000AD22OF00000000000000AD22OF0000000000000000
    d1dsn9.rttb0001_RRTB_SRC_002=C1E4E2D3404000000AD0D9C00000000
    d1dsn9.rttb0003_RRTB_SRC_003=000000AD22OF00000000000000AD22OF0000000000000000
    d1dsn9.rttb0003_RRTB_SRC_003=C1E4E2D3404000000AD0D9C00000000
<!-- new source
    d1dsn9.rttb0004_RRTB_SRC_004=000000DBF240A000000000000DBF240A000000000000
    d1dsn9.rttb0004_RRTB_SRC_004=C1E4E2D3404000000DBF238200000000
```

6. Cold start the session.

PWXPC passes the restart tokens to PowerExchange to begin change data extraction. Because the restart points for the other sources are earlier than the restart point for RRTB_SRC_004, PWXPC does not pass any change data for RRTB_SRC_004 until it reads the first change after the generated restart point.

---

**Recovering PowerCenter CDC Sessions**

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

A CDC session can fail for the following reasons:

- Permanent errors, such as source or target data errors
- Transitory or environmental errors, such as infrastructure problems, server failures, and network availability issues
If you run a session with a resume recovery strategy and the session fails, do not edit the state information or the mapping for the session before you restart the session.

If a session fails because of transitory or environmental errors, restart the session after you have corrected the errors. When you warm start a CDC session, PWXPC automatically performs recovery, if required. Alternatively, you can recover a CDC session, and then restart the session.

If a CDC session fails because of permanent errors, such as SQL or other database errors, you must correct the errors before restarting the CDC session. With some failures, you can correct the error and then restart the CDC session. In other cases, you might need to rematerialize the target table from the source table before you start extracting and applying change data again. If you rematerialize the target table, provide restart tokens that match the materialization point in the change stream, and then cold start the CDC session.

**Restriction:** If a CDC session requires recovery processing, you cannot override the restart tokens because PWXPC does not read the restart token file.

### Example of Session Recovery

This example describes recovery processing for a CDC session with relational targets.

Assume that you aborted the CDC session from the Workflow Monitor and then issued the Restart Task command to restart the session.

PWXPC automatically performs a recovery processing when the session warm starts and writes the following message in the session log:

```
PWXPC_12092 [INFO] [CDCRestart] Warm start requested. Targets will be resynchronized automatically if required
```

PWXPC then reads the restart tokens from the state tables and writes message PWXPC_12060 in the session log. This message records the restart tokens for the session and its sources, for example:

```
PWXPC_12060 [INFO] [CDCRestart]

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

<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>didsn8.rttb0004_RTTH_SRC_004</td>
<td>000000CA658400000000000EDE04A00000000FPFFFFF</td>
<td>C1E42D3404000000D21B1A50000000</td>
<td>GM storage</td>
</tr>
<tr>
<td>didsn8.rttb0005_RTTH_SRC_005</td>
<td>000000CA658400000000000EDE04A00000000FPFFFFF</td>
<td>C1E42D3404000000D21B1A50000000</td>
<td>GM storage</td>
</tr>
<tr>
<td>didsn8.rttb0006_RTTH_SRC_006</td>
<td>000000CA658400000000000EDE04A00000000FPFFFFF</td>
<td>C1E42D3404000000D21B1A50000000</td>
<td>GM storage</td>
</tr>
<tr>
<td>didsn8.rttb0007_RTTH_SRC_007</td>
<td>000000CA658400000000000EDE04A00000000FPFFFFF</td>
<td>C1E42D3404000000D21B1A50000000</td>
<td>GM storage</td>
</tr>
</tbody>
</table>

If PWXPC detects that recovery is required, PWXPC writes message PWXPC_12069 in the session log. This message usually includes the restart tokens for both the begin-UOW and end-UOW for the oldest uncommitted UOW that PWXPC re-reads during recovery. PWXPC usually stores end-UOW restart tokens in the state table or state file. However, if you specify a **Maximum Rows Per commit** threshold on the connection, PWXPC can commit change data and restart tokens between UOW boundaries. As a result, the restart tokens might not represent an end-UOW.

The following example PWXPC_12069 message includes "from" restart tokens that are the same as those in the example PWXPC_12060 message:

```
PWXPC_12069 [INFO] [CDCRestart] Running in recovery mode. Reader will resend the oldest uncommitted UOW to resync targets:
  from: Restart 1 [000000CA658400000000000EDE04A00000000FPFFFFF] : Restart 2 [C1E42D3404000000D21B1A50000000]
  to: Restart 1 [000000CA658400000000000D30000000000000FPFFFFF] : Restart 2 [C1E42D3404000000D21B1A50000000].
```

Because this session specifies a maximum rows threshold, the restart token values in the Restart 2 fields in both the “from” and “to” restart tokens is the begin-UOW value. The sequence token values in the Restart 1 fields represent the start and end change records in the UOW that is displayed in the Restart 2 field.

During recovery processing, PWXPC reads the change data records between the restart points that are defined by the two restart token values in the PWXPC_12069 message. Then PWXPC issues a commit for the
change data and restart tokens. The PowerCenter Integration Service writes the change data to the target tables and writes the restart tokens to the state table. Then the session ends.
Part V: Monitoring and Tuning

This part contains the following chapters:

- Monitoring CDC Sessions, 359
- Tuning CDC Sessions, 371
- zILP Exploitation, 382
Monitoring CDC Sessions

This chapter includes the following topics:

- Monitoring Overview, 359
- Monitoring CDC Sessions in PowerExchange, 359
- Monitoring CDC Sessions in PowerCenter, 366

Monitoring Overview

PowerExchange, PWXPC, and PowerCenter issue messages that you can use to monitor the progress of CDC sessions.

PWXPC can also display progress and statistical information about CDC sessions in PowerCenter Workflow Monitor.

Related Topics:

- "Monitoring CDC Sessions in PowerExchange" on page 359
- "Monitoring CDC Sessions in PowerCenter" on page 366

Monitoring CDC Sessions in PowerExchange

You can use certain PowerExchange messages and commands to monitor the extraction of change data by CDC sessions.

Use the following types of PowerExchange messages and output for monitoring extractions:

- Read progress messages. You can request that PowerExchange write messages that indicate the number of change records read by a CDC session.
- Extraction statistics messages. When extraction sessions end, PowerExchange writes messages that include statistical information about the change records processed.
- Multithreaded processing statistics messages. You can request that PowerExchange write statistical information about CDC sessions that use multithreaded processing.
- DISPLAY ACTIVE or LISTTASK command. Use one of these PowerExchange Listener commands, based on your operating system and mode of command execution, to list active CDC sessions. For more information about these commands, see the PowerExchange Command Reference.
Read Progress Messages

You can request that PowerExchange write read progress messages to the PowerExchange message log file. These messages indicate the number of change records read for a CDC session.

If you select the Retrieve PWX log entries option on the PWX CDC application connection, PWXPC also writes these messages in the session log.

To have PowerExchange write read progress messages, include the following statements in the DBMOVER configuration file:

PRGIND=Y

Enter Y to have PowerExchange write PWX-04587 messages to the PowerExchange message log file. These messages indicate the number of records read for a CDC session. Default is N.

PRGINT=records

Enter the number of records that PowerExchange must read before writing PWX-04587 messages to the PowerExchange message log file. Default is 250 records.

For example, to have PowerExchange write read progress messages after reading 100 records, specify the following statements:

PRGIND=Y
PRGINT=100

PWX-04587 messages have the following format:

PWX-04587 int_server/workflow_name/session_name: Records read=records

Where:

- int_server is the name of the PowerCenter Integration Service.
- workflow_name is the name of the workflow that contains the CDC session.
- session_name is the name of the CDC session.
- records is the cumulative number of records read since the CDC session started.

For example, for a CDC session named s_cdc_DB2_SQL_stats runs, PowerExchange writes the following messages:

PWX-04587 intserv/wf_cdc_mon_stats/s_cdc_DB2_SQL_stats: Records read=100
PWX-04587 intserv/wf_cdc_mon_stats/s_cdc_DB2_SQL_stats: Records read=200
PWX-04587 intserv/wf_cdc_mon_stats/s_cdc_DB2_SQL_stats: Records read=300

PowerExchange continues to write PWX-04587 messages for this CDC session until the session ends. In the PowerExchange message log file, each of these messages has a date and time stamp. Use this information to determine the speed with which PowerExchange processes change data from the change stream.

Extraction Statistics Messages

When a CDC session ends, PowerExchange writes messages that contain statistical information about extraction processing for the session.

These messages are:

- PWX-04578. PowerExchange writes this message for each source in the CDC session. The message includes the number of Insert, Update, Delete, Commit, and total records read for the source.
- PWX-04588. PowerExchange writes this message for the entire CDC session. This message includes the total number of records read for the session.
Important: The statistical information in the PowerExchange messages represents the change data that PowerExchange read for a CDC session. This information might not reflect the data that was applied to the targets. For statistical information about the change data applied to a target, review the session log.

Multithreaded Processing Statistics

If you use multithreaded processing, you can configure PowerExchange to issue messages that contain statistics on multithreaded extraction processing.

To issue these messages, you must specify the SHOW_THREAD_PERF statement in the DBMOVER configuration file on the PowerCenter Integration Service machine:

```plaintext
SHOW_THREAD_PERF=number_of_records
```

This statement specifies the number of records PowerExchange must process before writing statistics messages about multithreaded extraction processing to the PowerExchange message log file. For more information about this statement, see the PowerExchange Reference Manual.

If you select the Retrieve PWX log entries attribute on the application connection for the CDC session, PWXPC writes these messages in the session log. Also, you must specify 1 or greater for the Worker Threads attribute on the application connection to implement multithreaded processing so that statistics can be generated.

PowerExchange writes the following messages during each statistics interval:

- **PWX-31255.** Cycle time, which is the total time that PowerExchange on the PowerCenter Integration Service machine spent processing the change data before passing it to PWXPC. This message includes the total percentage of time and average, minimum, and maximum times in microseconds.
- **PWX-31256.** I/O time, which is the time that PowerExchange on the PowerCenter Integration Service machine spent reading change data from the PowerExchange Listener on the source system. This message includes the I/O percentage of the total time and average, minimum, and maximum times in microseconds.
- **PWX-31257.** Parsing time, which is the time that PowerExchange on the PowerCenter Integration Service machine spent in column-level processing for the change records on all threads. This message includes the parsing percentage of the total time and average, minimum, and maximum times in microseconds.
- **PWX-31258.** External time, which is the time that PowerExchange on the PowerCenter Integration Service machine spent combining the change records from all threads back into a single UOW to pass to PWXPC and for PWXPC to flush the data to PowerCenter. This message includes the external percentage of the total time and average, minimum, and maximum times in microseconds.
- **PWX-31259.** Delay time, which is the time that the PowerExchange on the PowerCenter Integration Service machine waited to receive new change records to process from the PowerExchange Listener on the source system. This message includes the delay percentage of the total time and average, minimum, and maximum times in microseconds.

For example, if you specify SHOW_THREAD_PERF=10000, PowerExchange writes the following messages after reading 10,000 change records and reaching the next UOW boundary:

```plaintext
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: 7% (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)
```
**PowerExchange Listener DISPLAY ACTIVE or LISTTASK Command**

Run the PowerExchange Listener DISPLAY ACTIVE or LISTTASK command to display CDC sessions that are active in the PowerExchange Listener.

The specific command name and syntax depends on how you issue the command, as follows:

- Issue the DISPLAY ACTIVE command from the command line on the system where the PowerExchange Listener runs. For more information, see the PowerExchange Command Reference.
- Use the pwxcmd program to issue the listtask command to a PowerExchange Listener that runs on the local system or a remote system. For more information, see the PowerExchange Command Reference.
- In the PowerExchange Navigator, issue the LISTTASK command from the Database Row Test dialog box. For more information, see the Informatica Command Reference.
- If you run the PowerExchange Listener as an application service in the Informatica domain, run the infacmd pwx program to issue the ListTaskListener command. For more information, see the Informatica Command Reference.

In the command output, the **PwrCntrSess** field displays the PowerCenter session name in the following format:

```
integration_server_name/workflow_name/session_name
```

For example, when two CDC sessions are active, the DISPLAY ACTIVE or LISTTASK command produces the following output:

```
PWX-00711 Active tasks:
PWX-00712 TaskId=1, Partner=10.10.10.01, Port=2480, PwrCntrSess=intserv1/workflow1/cdc_sess1, Application=appl_name1, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=  
PWX-00712 TaskId=2, Partner=10.10.10.02, Port=2480, PwrCntrSess=intserv2/workflow2/cdc_sess2, Application=appl_name2, Status=Active, AM=CAPXRT, Mode=Read, Process=, SessId=  
PWX-00713 2 active tasks  
PWX-00709 0 Dormant TCBs
```

**PowerExchange Listener DISPLAYSTATS Command**

You can use the PowerExchange Listener DISPLAYSTATS or pwxcmd displayystats command to publish monitoring statistics for a PowerExchange Listener that runs on i5/OS, Linux, zLinux, UNIX, Windows, or z/OS.

Before you run the command, configure the following statements in the DBMOVER configuration file:

- Specify the MONITOR parameter in the STATS statement in the DBMOVER configuration file to enable PowerExchange to collect these statistics. You can include the `interval` subparameter to publish the statistics at a regular interval as well as on demand.
- For the proper display of monitoring output on z/OS, set the `LOG_LINE_LIMIT` statement to 132. Otherwise, the lines might wrap awkwardly, making the output hard to read.

You can issue the command in any of the following ways:

- From the command line on the Linux, UNIX, Windows, or zLinux system where the PowerExchange Listener runs.
- With the MVS MODIFY (F) command on the z/OS system where the PowerExchange Listener runs.

```
PWX-31257 Parse time: 79% (avg: 4549 min: 4108 max: 5425 usecs)  
PWX-31258 Extern time: 20% (avg: 1144 min: 616 max: 3242 usecs)  
PWX-31259 Delay time: 0% (avg: 7 min: 4 max: 115 usecs)
```

If the parsing and external processing times are greater than the I/O time, you can increase the number of threads for the CDC session to try to improve throughput.
With the pwxcmd program from a remote Linux, UNIX, and Windows system to a Listener on any supported operating system.

**Note:** You must use this method to publish monitoring statistics for a PowerExchange Listener on i5/OS on demand.

The command syntax depends on the operating system type and whether you use pwxcmd. For more information, see the *PowerExchange Command Reference*.

Depending on which command parameter you use, you can publish one of the following types of reports:

- **Listener.** Reports PowerExchange Listener summary statistics on memory usage, CPU processing time, and activity on behalf of client requests. These statistics include counts of client tasks, connections, number of messages sent and received, bytes of data sent and received, and netport jobs (z/OS only). These statistics include both bulk data movement and CDC tasks.

  **Note:** If you run a PowerExchange Listener Service in the Informatica Domain, you can use the infacmd pwx displayStatsListener command to publish these statistics. For more information, see the *Informatica Command Reference*.

- **Accessmethods.** Reports statistics on PowerExchange Listener message and data transfer activity by client task and access method. For each active task and access method combination, these statistics include the number of rows read and written, bytes of data read and written, the source or target file name or data map file name, and the CPU processing time. For CDC requests that use the CAPX or CAPXRT access method, the report also includes counts of the SQL inserts, updates, and deletes that the task processed.

- **Clients.** Reports information about the active client tasks that are running under the PowerExchange Listener. For each task, the statistics show some or all of the following information: the status, access method, read or write mode, process name and session ID if available, CPU processing time, and start date and time. The statistics also include the client port number and IP address. If the client is PowerCenter, the statistics include the PowerCenter session ID and the application name for CDC.

By default, the Listener report is published.

The reports for a PowerExchange Listener on z/OS are similar to those for a PowerExchange Listener on i5/OS, Linux, zLinux, UNIX, or Windows.

The following example Listener report is for a PowerExchange Listener on z/OS:

```
PWX-00723 Command <displayStats Listener> succeeded
PWX-37101 Listener <PWXLST> > ASID=375 {z'177'} UserID=AUSRID
PWX-37102 Memory
PWX-37103 Region below 16-MB line: In Use 108 KB Limit Value 9192 KB Free 9084 KB
PWX-37104 Region above 16-MB line: In Use 53912 KB Limit Value 1675264 KB Free 1621352 KB
PWX-37117 CPU Time
PWX-37118 TCB Time = 0 
PWX-37119 Listener = 0 hrs, 0 mins, 1 secs, 275762 nsecs
PWX-37106 Cumulative Requests
PWX-37107 Total Tasks= 11 Active Tasks = 3 HWM Tasks = 3 MaxTasks = 50
PWX-37108 Connections= 11 Accepted = 11 Active = 0
PWX-37109 Mgs Sent = 0 Mgs Received= 22
PWX-37110 Data Sent = 0 Data Received= 7304
PWX-37111 NetportJobs= 0
```

The **Memory, TCB Time, SRB Time, and NetportJobs** values are specific to the PowerExchange Listener on z/OS. For a PowerExchange Listener on i5/OS, Linux, UNIX, or Windows, the report displays the total memory usage.

You can use this report determine if the number of client tasks is reaching the limit that is set in the MAXTASKS statement of the DBMOVER configuration file. Compare the HWM Tasks value to the Maxtasks value. If the HWM Tasks value reaches the MAXTASKS limit, PowerExchange Listener processing might be delayed, which can cause reduced throughput and connection timeouts.
The following example accessmethods report is for a PowerExchange Listener on z/OS, but the same fields are displayed for a PowerExchange Listener on i5/OS, Linux, UNIX, Windows, or zLinux:

```
PWX-00723 Command <displaystats AccessMethods> succeeded
PWX-37201 Active Access Methods
PWX-37202 Task ID = 42412 AM = CAPXRT
PWX-37203 Rows read = 1029 Rows written = 0
PWX-37204 Bytes read = 116277 Bytes written = 0
PWX-37205 File = d2pwd0.d002root_ROOT
PWX-37206 Table = <Capture Extract Realtime>
PWX-37208 Inserts = 564 Updates = 0 Deletes = 465
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 299809 mcrs
PWX-37202 Task ID = 42414 AM = NRDB
PWX-37203 Rows read = 10 Rows written = 0
PWX-37204 Bytes read = 570 Bytes written = 0
PWX-37205 File = ABC.VEAM,MASTER_REC
PWX-37206 Table = <Non-relational source>
PWX-37202 Task ID = 42414 AM = KSDS
PWX-37203 Rows read = 10 Rows written = 0
PWX-37204 Bytes read = 800 Bytes written = 0
PWX-37205 File = XYQ.TEST.V1.KSDS
PWX-37206 Table = XYQ.TEST.V1.KSDS
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 76151 mcrs
```

For the CAPXRT and CAPX access methods, the report includes the number of SQL inserts, updates, and deletes that the task processed for a CDC request.

A client task can have multiple access methods, for example, one for reading source data and one for mapping nonrelational source data to a relational format. In the example output, task 42414 uses the NRDB access method with the data map file specified in the File field to map nonrelational data to a relational format. The same task uses the KSDS access method to retrieve data from the KSDS data set specified in the File field.

The following example clients report is for a PowerExchange Listener on Windows, but the same fields are displayed for a PowerExchange Listener on i5/OS, Linux, zLinux, UNIX, or z/OS:

```
PWX-00723 Command <displaystats Clients> succeeded
PWX-37112 Active Tasks
PWX-37113 Task ID = 41942 Status = Active
PWX-37114 Port = 2400 Partner = 1.27.0.0.1
PWX-37115 PwCtrlSess = N/A
PWX-37207 Application = N/A
PWX-37116 AM = DB2 Mode = Read Process = DTLST3 SessionId =
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 62400 mcrs
PWX-37122 Start time = 2014-05-01 14:21:37
PWX-37113 Task ID = 41943 Status = Active
PWX-37114 Port = 2400 Partner = 1.27.0.0.1
PWX-37115 PwCtrlSess = N/A
PWX-37207 Application = N/A
PWX-37116 AM = DB2 Mode = Read Process = DTLST3 SessionId =
PWX-37121 CPU time = 0 hrs, 0 mins, 0 secs, 124000 mcrs
PWX-37122 Start time = 2014-05-01 14:22:01
```

The Partner field displays the IP address of the client that issued the request that caused the PowerExchange Listener to create the task. This value begins with :ffff for an IPv6 address.

For more information about the fields in each of these reports, see the PowerExchange Command Reference.

PowerExchange Logger for Linux, UNIX, and Windows Monitoring Statistics

You can use the PowerExchange Logger DL and DG commands or the pwxcmd displaystats -tp (logger groups) command to publish monitoring statistics for a PowerExchange Logger process and its tasks or PowerExchange Logger group definitions.

Before you run one of these commands, you must configure the STATS=(MONITOR) parameter in the PowerExchange Logger pwxcl.cfg configuration file to enable collection of the statistics. In this parameter, you can include the optional interval subparameter to also publish the statistics at a regular interval.
Issue the commands in the following ways:

- Issue the DL and DG commands from the command line window on the Linux, UNIX, or Windows system where the PowerExchange Logger runs. The PowerExchange Logger must be running in the foreground.
- Issue the pwxcmd displaystats -tp logger or pwxcmd displaystats -tp groups command from a Linux, UNIX, or Windows system to the PowerExchange Logger on a remote system or the same system. You must use this method to issue the command to a PowerExchange Logger process that runs in background mode.

The command output is displayed on screen and printed to the PowerExchange message log.

**Logger Report**

The DL and pwxcmd displaystats -tp logger commands produce statistics for the PowerExchange Logger process and its tasks. The following example report shows these statistics:

```
PWX-26011 Command handler received command "DS"
PWX-00723 Command <display L stats> succeeded
PWX-37130  PWXCL pid = 7144  Writer status = Reading or waiting for source data
PWX-37134  CPU Time =  0:00:02.589616
PWX-37131  Memory  (Current/Total/Maximum)  
PWX-37132  Controller: (981/983/1849) KB  Command Handler: (0/0/34) KB  Writer: (5127/5147/5181) KB
PWX-37135  Status  7144  Totals I=000000024344 U=00000000000 D=000000024336
C=000000004040 Total=000000052684
C=000000004040 Total=000000052684
PWX-37137  Active Cycle : 2015-08-11 13:21:01 I=000000024344 U=00000000000 D=000000024336
C=000000004040 Total=000000052684
```

This report contains the following fields:

- PWXCL pid. The process ID of the PowerExchange Logger process.
- Writer status. The status of the PowerExchange Logger Writer subtask at the time the command was issued.
- CPU Time. The amount of CPU time used by the PowerExchange Logger since it started.
- Controller. The amount of memory in kilobytes that the PowerExchange Logger Controller has used.
- Command Handler. The amount of memory in kilobytes that the PowerExchange Logger Command Handler has used.
- Writer. The amount of memory in kilobytes that the PowerExchange Logger Writer subtask has used.
- Status. The process ID of the PowerExchange Logger process.
- I, U, D, C, and Total. Counts of inserts, updates, deletes, and commits that the PowerExchange processed, with the total for all of these operation types. These values are reported for the PowerExchange Logger process, the currently open log file, and the active logging cycle.
- CurrentFileOpened. The timestamp that indicates when the log file was opened.
- Active Cycle. The timestamp that indicates when the active logging cycle started.

**Logger Group Definition Reports**

The DG and pwxcmd displaystats -tp groups commands produce statistics for each PowerExchange Logger group definition that is definition. A group definition defines a set of PowerExchange Logger log files for a group of registered source tables. The following example report shows these statistics:

```
PWX-26011 Command handler received command "DG"
PWX-37138  Grp: dt1d04  Regs=1  UID=000000000000  C=000000000000 Unflushed=000000000000
PWX-37138  Grp: dt1d03  Regs=2  UID=0000000000470  C=000000000028 Unflushed=000000000000
PWX-37138  Grp: dt1d02  Regs=2  UID=000000003276  C=000000000196 Unflushed=000000000000
```

This report contains the following fields:

- Grp. The name of the group definition.
Monitoring CDC Sessions in PowerCenter

In PowerCenter, you can monitor the progress of CDC sessions.

Use the following information to monitor CDC sessions:

- Messages in the session log. PWXPC writes messages to the session log.
- Performance details in Workflow Monitor. If you configure a CDC session to report performance details, you can monitor the progress of the session in Workflow Monitor.

For more information about PowerCenter monitoring options, see the PowerCenter Performance Tuning Guide.
Session Log Messages

You can use messages that PWXPC and PowerCenter write to the session log to monitor the progress of CDC sessions.

When PWXPC flushes change data, PWXPC writes one of the following messages in the session log to indicate the reason for the flush:

- PWXPC_10081 [INFO] [CDCDispatcher] raising real-time flush with restart tokens [restart1], [restart2] because the UOW Count [count] is reached
- PWXPC_10082 [INFO] [CDCDispatcher] raising real-time flush with restart tokens [restart1], [restart2] because Real-time Flush Latency [latency] is reached
- PWXPC_12128 [INFO] [CDCDispatcher] raising real-time flush with restart tokens [restart1], [restart2] because the Maximum Rows Per commit [count] is reached

You can use the restart tokens in these PWXPC flush messages to monitor the processing of the change data.

For each PWXPC flush message, PowerCenter writes a WRT_8160 message after committing change data to the targets. This message displays the source-based commit statistics.

Performance Details in Workflow Monitor

In Workflow Monitor, you can view performance details in the run properties for a CDC session to assess the efficiency of the CDC session and extraction processing.

If session performance is degraded, you can use data in the Performance Counter column to determine the bottleneck.

PWXPC does not store performance details in the repository so you cannot view performance details for previous executions of a CDC session.

To enable the collection of performance details, select Collect performance data on the Properties tab of the CDC session properties.

When the CDC session runs, PWXPC refreshes performance statistics every 10 seconds.

If you enable a resume recovery strategy for the CDC session, PWXPC displays data for all Performance Counter fields.

The following table describes the Performance Counter fields:

<table>
<thead>
<tr>
<th>Performance Counter Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| 1 PowerExchange CDC Reader Status: | Current status of the PWXPC reader, as indicated by one of the following values:  
  - **No Data To Process.** In the last read, PowerExchange did not pass data to PWXPC.  
  - **Restart Advance.** PowerExchange passed restart tokens to PWXPC but did not pass change data.  
  - **Processing Data.** PowerExchange passed change data and restart tokens to PWXPC for processing. |
<p>| 1.1 Time Last Data Row Read | Time, in milliseconds, that PWXPC took to read the data last received from PowerExchange. |
| 1.2 Data Rows In Current Interval | Number of change records received from PowerExchange during the current statistics interval. |</p>
<table>
<thead>
<tr>
<th>Performance Counter Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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, based on the quantity of change data:</td>
</tr>
<tr>
<td></td>
<td>- If PowerExchange is reading large amounts of change data from the change stream, this value is usually large and reflects the maximum PowerExchange throughput.</td>
</tr>
<tr>
<td></td>
<td>- If PowerExchange is waiting for change data at the end of the change stream, this value is small.</td>
</tr>
<tr>
<td>The following factors can increase this value:</td>
<td>- Large network bandwidth</td>
</tr>
<tr>
<td></td>
<td>- CDC offload processing</td>
</tr>
<tr>
<td></td>
<td>- Multithreaded processing</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>2 PowerCenter Processing Status:</td>
<td>Overall status of the CDC session, as indicated by one of the following values:</td>
</tr>
<tr>
<td></td>
<td>- Idle. Waiting for change data.</td>
</tr>
<tr>
<td></td>
<td>- Processing Data. Data is being processed.</td>
</tr>
<tr>
<td></td>
<td>- Recovery Disabled. If a resume recovery strategy is not enabled, the PWXPC CDC reader cannot get PowerCenter status information.</td>
</tr>
<tr>
<td>2.1 Time Of Last Commit</td>
<td>Time stamp of the last commit to a target.</td>
</tr>
<tr>
<td>2.2 Rows Processed To Commit In Current Interval</td>
<td>Number of change records that the PWXPC reader flushed during the current statistics interval. This count includes the change records in all committed UOWs. Some of these UOWs might have started before the current statistics interval began.</td>
</tr>
<tr>
<td>2.3 Commit Rate In Current Interval (rows/sec)</td>
<td>Processing rate, in number of change records per second, for the change records for the UOW that was last committed during the current statistics interval. This processing includes reading the UOW from PowerExchange and committing the change data to the targets.</td>
</tr>
<tr>
<td></td>
<td>The following factors can affect this rate:</td>
</tr>
<tr>
<td></td>
<td>- Number of available DTM buffers</td>
</tr>
<tr>
<td></td>
<td>- Responsiveness of the target</td>
</tr>
<tr>
<td></td>
<td>- Number of transformations in the pipeline</td>
</tr>
<tr>
<td>2.4 Mean Commit Rate (rows/sec)</td>
<td>Mean number of change records per second for the rate displayed in 2.3 Commit Rate In The Current Interval.</td>
</tr>
<tr>
<td></td>
<td>This value differs from the 2.6 Mean Throughput Rate value in that it takes into account only the time when the session is actively processing data. This value does not reflect processing overlap in PowerCenter.</td>
</tr>
<tr>
<td>2.5 Max Commit Rate (rows/sec)</td>
<td>Maximum number of change records per second for the commit rate displayed in 2.3 Commit Rate In The Current Interval, from the start of the CDC session.</td>
</tr>
<tr>
<td>Performance Counter Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>2.6 Mean Throughput (rows/sec)</td>
<td>Mean rate of processing for the CDC session.</td>
</tr>
<tr>
<td>2.7 Max Throughput (rows/sec)</td>
<td>Maximum throughput for the CDC session.</td>
</tr>
<tr>
<td>2.8 Commits In Current Interval</td>
<td>Number of commits processed to completion by the target during the current statistics interval.</td>
</tr>
<tr>
<td>2.9 Commits Pending</td>
<td>Number of commits that the PWXPC reader issued but that have not yet reached the targets. A large value might indicate problems with target responsiveness.</td>
</tr>
<tr>
<td>3 Capture Timestamps</td>
<td>-</td>
</tr>
<tr>
<td>3.1 Timestamp On Last End Packet Read</td>
<td>The capture timestamp, DTL__CAPXTIMESTAMP, from the last UOW read for a source in the CDC session.</td>
</tr>
<tr>
<td>3.2 Timestamp On Last Target Commit</td>
<td>The capture timestamp, DTL__CAPXTIMESTAMP, from the last UOW committed to the target.</td>
</tr>
<tr>
<td>4 Totals</td>
<td>-</td>
</tr>
<tr>
<td>4.1 Elapsed Time</td>
<td>Total elapsed time for the CDC session.</td>
</tr>
<tr>
<td>4.2 Rows Read</td>
<td>Total number of change records read from PowerExchange.</td>
</tr>
<tr>
<td>4.3 End Packets Read</td>
<td>Total number of UOWs read.</td>
</tr>
<tr>
<td>4.4 Time in PowerExchange Processing</td>
<td>Total time of PowerExchange processing for the CDC session.</td>
</tr>
<tr>
<td>4.5 Rows Processed</td>
<td>Total number of change records processed through PowerCenter and committed to the targets.</td>
</tr>
<tr>
<td>4.6 Commits to Target</td>
<td>Total number of flushes that the PWXPC reader issued and that were committed to the targets.</td>
</tr>
<tr>
<td>4.7 TS on Last Commit minus TS at Commit (2.1-3.2)</td>
<td>Result from subtracting 3.2 Timestamp On Last Target Commit value from the 2.1 Time Of Last Commit value. If this result is negative, the value is enclosed in parentheses.</td>
</tr>
</tbody>
</table>

**Viewing Performance Details in Workflow Monitor**

In Workflow Monitor, view performance details for a CDC session to assess the efficiency of the CDC sessions.

1. In Workflow Monitor, right-click a session and select **Get Run Properties**.
2. In the **Properties** window, click the **Performance** area.
   
   The **Performance Counter** column displays a source qualifier from the CDC session. The **Counter Value** column displays the PowerCenter node name.
3. To view performance details, select the source qualifier.

**Note:** For PWXPC to display performance details for a CDC session that ended, you must select performance details while the session is still running.

PowerCenter displays data in the *Performance Counter* fields in the *Performance* area.
Tuning CDC Sessions

This chapter includes the following topics:

- **Tuning Overview, 371**
- **PowerExchange DBMOVER Statements for Tuning CDC Sessions, 372**
- **PowerCenter Connection Attributes and Session Properties, 375**
- **CDC Offload Processing, 378**
- **Multithreaded Processing, 380**

## Tuning Overview

PowerExchange and PowerCenter provide options that you can use to tune CDC sessions. These tuning options can help you increase throughput, reduce overhead on the source system, and improve CDC efficiency.

Use any of the following options to tune CDC sessions:

- PowerExchange DBMOVER statements. Customize certain statements in the DBMOVER configuration file to make tuning adjustments such as changing buffer sizes or disabling compression or traces.

- PowerCenter connection attributes. Customize PWX CDC application connection attributes to make tuning adjustments such as disabling encryption or compression, reducing commit processing, or enabling offload processing and multithreaded processing.

- Buffer memory. Set the PowerCenter **DTM Buffer Size** and **Default Buffer Block Size** session properties to generate a large number of small blocks. For CDC, this strategy improves session performance and prevents wasted buffer space.

- Offload processing. Use offload processing to transfer column-level extraction processing from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine. Also, if the data source type requires use of the UOW Cleanser (UOWC), offloading transfers UOWC processing to the PowerCenter Integration Service machine. Offloading helps increase throughput when resources available for the PowerExchange Listener are constrained on the source system.

- Multithreaded processing. Enable the use of multiple worker threads for resource-intensive, column-level extraction processing. You can use multithreading on the source system to process data from Linux, UNIX, or Windows data sources if the PWX connection for the CDC session has a location of local. You can also use multithreading for extracting change data from the systems other than the source system when offload processing is in effect. Enable multithreading only when extractions appear to be CPU bound.
**Note:** You can also log data to a PowerExchange Logger for Linux, UNIX, and Windows instance on a system that is remote from the source system. In certain situations, this configuration can reduce resource consumption on the source system, move column-level and UOW Cleanser processing to the remote system, and reduce the network overhead of data transfer. For more information, see Chapter 14, "Remote Logging of Data" on page 297.

**Related Topics:**

- "PowerCenter Connection Attributes for Tuning CDC Sessions" on page 375
- "PowerExchange DBMOVER Statements for Tuning CDC Sessions" on page 372
- "Tuning Commit Processing" on page 377

### PowerExchange DBMOVER Statements for Tuning CDC Sessions

You can customize certain statements in the dbmover.cfg configuration file to tune CDC sessions.

Customize any of the following parameters to try to increase throughput or reduce CPU use:

**APPBUFSIZE=**<br>

The maximum application data buffer size, in bytes, that PowerExchange uses to read or write data. This buffer type can exist on a source or target system.

If you use a remote target system, PowerExchange usually writes change data to its application data buffer on the source system until the buffer is full. PowerExchange then sends the change data to a sending TCP/IP buffer on the source system. TCP/IP transports the change data to a receiving TCP/IP buffer on the target system. PowerExchange on the target system reads the change data from the TCP/IP buffer into its application data buffer. PWXPC then reads the change data and passes it to PowerCenter. PowerCenter processes the data and applies it to the targets.

Enter an APPBUFSIZE value that is greater than the maximum size of any single data row to be sent.

Valid values are from 34816 through 8388608. Default is 256000.

If the target is remote, enter the same APPBUFSIZE value in the DBMOVER configuration files on the source and target systems.

When the APPBUFSIZE value is not optimal, PowerExchange writes message PWX-01295 in the PowerExchange message log file on the source system. This message recommends a minimum application buffer size.

If dynamic application buffer sizing is enabled, the APPBUFSIZE statement defines the initial size of the application data buffer for all connections made during a PowerExchange Listener run. PowerExchange resize the application data buffer dynamically for individual connections as needed. Dynamic application buffer sizing is enabled by default. You can explicitly enable it by specifying Y for the APPBUFSIZEDYN statement in the DBMOVER configuration file.

**APPBUFSIZEDYN=**<br>

Specifies whether to enable dynamic application buffer sizing.

The DBMOVER APPBUFSIZE statement defines the initial size of the application buffer for all connections made during a PowerExchange Listener run. If APPBUFSIZEDYN=Y, PowerExchange resize the application buffers for individual connection as needed.
The APPBUFSIZEDYN statement applies to PowerExchange connections to data sources with either fixed-length or variable-length records. A variable-length record is a record with at least one variable-length field. A variable-length field has a datatype of VARCHAR or VARBIN.

For each connection to a data source with variable-length records, PowerExchange resizes the application buffer when it encounters a record that is too large to fit into the buffer. PowerExchange increases the size of the application buffer to a value of ten times the size of the record that has overflowed, up to the maximum application buffer size of 8 MB. The new size remains in effect for the duration of the Listener run or until the application buffer is resized again. PowerExchange never decreases the application buffer size for a connection after the Listener run has started.

For each connection to a data source with fixed-length records, PowerExchange determines the record length when the connection is opened and resizes the application buffer once, up to the maximum application buffer size of 8 MB, as needed.

CAPI_CONNECTION=(..., (TYPE={UDB|UOWC}, MEMCACHE=cache_size, ...))

The maximum memory cache size, in kilobytes, that PowerExchange can allocate to reconstruct complete UOWs. This MEMCACHE parameter is specified only in the UDB or UOWC CAPI_CONNECTION statements.

Enter a number from 0 through 2147483647. Default is 1024. If you enter 0, the memory cache size is unlimited.

PowerExchange keeps all changes in each UOW in cache until processing the end-UOW record. PowerExchange incrementally allocates memory cache up to the limit that this parameter specifies. If the MEMCACHE value is too small to hold all of the changes in a UOW in cache, the changes spill to a disk file.

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

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

The default value of 1024 is appropriate if the change stream contains many small UOWs. If you have UOWs larger than 1024 KB, increase this value or enter 0. PowerExchange processes a UOW more efficiently if all of the changes are cached in memory. For most environments, a good starting value is 10240.

Attention: PowerExchange allocates memory cache for each connection for change data extraction processing. To prevent excessive memory use, use a MEMCACHE value that is reasonable for the extraction processing load and number of CDC sessions that run concurrently. If the value is too large and you run many concurrent sessions, memory constraints might occur.

CAPI_CONNECTION=(..., (TYPE={MSQL|UDB|UOWC}, RSTRADV=rstr_seconds, ...))

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

This RSTRADV parameter is specified only in CAPI_CONNECTION statements of the following types:

- MSQL
- UDB
- UOWC
Enter a number from 0 through 86400.

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

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

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

For sources with low change activity, you can use the RSTRADV parameter to periodically advance to the restart tokens for those sources. Advancing the restart tokens speeds up restart processing for CDC sessions by minimizing the amount of change data that must be reprocessed.

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

A low value can cause the UOW Count option on the PWX CDC connection to match more quickly than expected. When the UOW counter matches, PWXPC flushes the data buffer and commits restart tokens to the targets. Excessive flush activity can adversely affect performance on the PowerCenter Integration Service machine and on the target databases.

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

**LISTENER=(node_name,TCPIP,send_bufsize,receive_bufsize,send_size,receive_size, ...)**

A TCP/IP port on which a named PowerExchange Listener process listens for work requests.

The *send_bufsize* and *receive_bufsize* positional parameters define the data portion of the TCP/IP send and receive buffer sizes that PowerExchange uses. If you do not specify these values, PowerExchange uses the operating system defaults.

To increase throughput, try increasing the *send_bufsize* and *receive_bufsize* values in the LISTENER statement in the DBMOVER configuration file on the source system. For help in determining the best values to use, contact your network administrator.

**NODE=(node_name,TCPIP,host_name,port,send_bufsize,receive_bufsize,send_size,receive_size, ...)**

A TCP/IP host name and port that PowerExchange uses to contact a PowerExchange Listener process.

The *send_bufsize* and *receive_bufsize* positional parameters define the data portion of the send and receive buffer sizes that PowerExchange uses. If you do not specify these values, PowerExchange uses the operating system defaults.

To increase throughput, try increasing the *send_bufsize* and *receive_bufsize* values in the NODE statement in the DBMOVER configuration file on the target system. For help in determining the best values to use, contact your network administrator.

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

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

TRACE statements can severely impact PowerExchange performance. Use these statements only at the direction of Informatica Global Customer Support.
After Informatica Global Customer Support diagnoses the problem, remove or comment out all TRACE statements in the DBMOVER configuration files on all systems.

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

PowerCenter Connection Attributes and Session Properties

You can use certain PowerCenter connection attributes and session properties for tuning CDC sessions.

PowerCenter Connection Attributes for Tuning CDC Sessions

In PowerCenter, you can customize some attributes on the PWX CDC connections to tune CDC sessions.

The following table describes the connection attributes that you can optionally use for tuning:

<table>
<thead>
<tr>
<th>Connection Option</th>
<th>Description</th>
<th>Tuning Suggestion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compression</td>
<td>Controls whether to compress source data during the PowerCenter session. Default disables compression.</td>
<td>Do not use compression.</td>
</tr>
<tr>
<td>Encryption Type</td>
<td>Type of data encryption that PowerExchange uses. Default is None for no encryption.</td>
<td>Do not use encryption.</td>
</tr>
<tr>
<td>Connection Option</td>
<td>Description</td>
<td>Tuning Suggestion</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>------------------</td>
</tr>
</tbody>
</table>
| **Image Type**    | Indicates how PWXPC passes captured Updates to CDC sessions that extract and apply the updates to the target. Options are:  
- **AI**. Process Updates as Update operations. PWXPC passes each Update as a single Update record. An Update record includes after images of the data only, unless you add before image (BI) and change indicator (CI) fields to the extraction map that you import for the source definition for the CDC session.  
- **BA**. Process Updates as Deletes followed by Inserts. PWXPC passes each Update as a Delete record followed by an Insert record. The Delete record contains the before image of the data, and the Insert record contains the after image.  
Default is BA.  
If you specify AI, you can still use before images of the data, if available, in extraction processing. PWXPC can embed before-image data and after-image data in the same Update row. To embed before-image data, you must complete the following configuration tasks:  
- In the PowerExchange Navigator, add BI and CI fields to the extraction map that you plan to import for the source definition in PowerCenter.  
- If you use batch or continuous extraction mode, enter BA for the CAPT_IMAGE parameter in the PowerExchange Condense or PowerExchange Logger for Linux, UNIX, and Windows configuration file. This setting stores both before and after images in the PowerExchange Logger log files or PowerExchange Condense condense files. When CDC sessions run, they extract data from these files. | Set to AI. |
| **UOW Count**     | 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. | To improve efficiency on the PowerCenter Integration Service machine and the target databases, increase this value to reduce commit processing. |
| **Real-time Flush Latency in milliseconds** | 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 2 seconds. | To improve efficiency on the PowerCenter Integration Service machine and the target databases, increase this value to reduce commit processing. |
| **PWX Latency in seconds** | Maximum time, in seconds, that the PowerExchange instance on the source waits for more change data before flushing data to PWXPC on the PowerCenter Integration Service machine. Default is 2. | Use the default value. |
### Connection Options

<table>
<thead>
<tr>
<th>Connection Option</th>
<th>Description</th>
<th>Tuning Suggestion</th>
</tr>
</thead>
<tbody>
<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 ignores this option.</td>
<td>If UOWs usually contain few changes, increase this value to increase the size of the UOWs. This practice can improve efficiency on the PowerCenter Integration Service machine and on the target databases by reducing commit processing.</td>
</tr>
<tr>
<td>Offload Processing</td>
<td>Controls whether PowerExchange uses CDC offload processing. Offload processing transfers resource-intensive column-level and UOW Cleanser processing from the source system to another system. Default is No.</td>
<td>If resource constraints exist on the source system and you need to increase CDC throughput, consider enabling offload processing.</td>
</tr>
<tr>
<td>Worker Threads</td>
<td>Controls whether PowerExchange uses multiple threads for resource-intensive, column-level extraction processing. You can use multithreading on the source system to process data from Linux, UNIX, or Windows data sources, or on another system for extraction processing when offload processing is in effect. Enable multithreading only when extractions appear to be CPU bound. Enter the number of threads that you want PowerExchange to use. Valid values are 1 through 64. Default is 0, which causes PowerExchange to not use multithreaded processing.</td>
<td>Enter a number greater than 0.</td>
</tr>
<tr>
<td>Array Size</td>
<td>If the Worker Threads value is greater than zero, indicates the size of the storage array, in number of records, for the threads. Valid values are from 25 through 100000. Default is 25.</td>
<td>Use 25. <strong>Attention:</strong> If you enter a large array size value, have large records, or run many sessions that use multithreading concurrently, memory shortages might occur on the PowerCenter Integration Service machine.</td>
</tr>
</tbody>
</table>

For more information about PWX CDC connection attributes, see *PowerExchange Interfaces for PowerCenter*.

### Tuning Commit Processing

To tune commit processing and performance of CDC sessions, you can adjust commitment control attributes on the PWX CDC application connection.

If the session log for a CDC session contains PWXPC flush messages followed by PowerCenter source-based commit messages, the session might be reading change data faster than the data is applied to the targets. To try to resolve this issue, adjust the following commitment control attributes on the PWX CDC connection, based on the most prevalent type of flush message in the session log:

- If PWXPC_10081 flush messages are the most prevalent messages, try increasing the **UOW Count**.
- If PWXPC_10082 flush messages are the most prevalent messages, try increasing the **Real-time Flush Latency in milli-seconds**.
If PWXPC flushes change data too frequently, too many commitment control attributes might be specified on the PWX CDC connection. In this case, specify a single commitment control attribute and disable the other ones.

If the change stream contains many small UOWs, you can use the Minimum Rows Per commit option to create larger UOWs of more uniform size. PowerExchange and PWXPC can process a few large UOWs more efficiently than many small UOWs. By using the Minimum Rows Per commit option to increase the size of UOWs, you can improve CDC processing efficiency.

Also, performance of the target database can impact the performance of the CDC session. Contact your database administrator to verify that database access is optimal.

PowerCenter Session Properties for Tuning Buffer Memory

When you run a CDC session, the PowerCenter Data Transformation Manager (DTM) allocates buffer memory to the session based on the DTM Buffer Size value on the Properties tab of the session properties. The DTM divides the memory into buffer blocks based on the Default Buffer Block Size setting on the Config Object tab of the session properties.

If you suspect that buffer memory is insufficient, enable the collection of performance details in the CDC session. Then review the difference between the performance counters 4.1 Time in PowerExchange Processing and 4.4 Elapsed Time. If the elapsed time is much larger that the PowerExchange processing time, buffer memory constraints might exist. To improve performance of the CDC session, try adjusting the DTM Buffer Size and Default Buffer Block Size properties.

For optimal CDC performance, set these session properties to create a large number of small blocks. Informatica recommends the following settings:

- For the DTM Buffer Size, specify 128 MB, 256 MB, 512 MB, 1 GB, or 2 GB.
- For the Default Buffer Block Size, specify 32 KB.

Do not set these session properties to auto. The auto option creates a small number of large blocks, which can degrade CDC session performance. The auto option is intended for bulk data load processing.

CDC Offload Processing

CDC offload processing transfers column-level processing of change data from the PowerExchange Listener on the source system to the PowerExchange client on the PowerCenter Integration Service machine.

For data sources for which PowerExchange uses the UOW Cleanser (UOWC), offload processing also transfers UOWC processing to the PowerCenter Integration Service machine. These data sources include z/OS data sources, DB2 for i5/OS, and Oracle CDC with LogMiner.

Use offload processing when resources on the source system are constrained. In this situation, offload processing can help increase throughput for CDC sessions.

Related Topics:

- "Rules and Guidelines for CDC Offload Processing" on page 379
- "Enabling Offload Processing for CDC Sessions" on page 379
- "Example of CDC Offload Processing with a z/OS Source" on page 380
Rules and Guidelines for CDC Offload Processing

Before implementing CDC offload processing, review the following rules and guidelines:

- You must copy the appropriate source-specific CAPI_CONNECTION statements from the DBMOVER configuration file on the source system to the PowerCenter Integration Service machine.
- PowerExchange does not support CDC offload processing for capture registrations that you create from data maps that use any of the following options:
  - User access methods
  - User-defined fields that invoke programs by using the CALLPROG function
  - Record-level exits

Enabling Offload Processing for CDC Sessions

To use CDC offload processing, you must configure some PWX CDC connection attributes. Also, you must add the source-specific CAPI_CONNECTION statements to the DBMOVER configuration file on the PowerCenter Integration Service machine.

1. Configure the attributes for offload processing on the PWX CDC Real Time application connection for the CDC session.
   
   The following table describes these attributes:

<table>
<thead>
<tr>
<th>Connection Attribute</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 match the name of a NODE statement in the dbmover.cfg configuration file on the PowerCenter Integration Service machine.</td>
</tr>
</tbody>
</table>
   | Offload Processing    | Controls whether PowerExchange uses CDC offload processing. When offload processing is enabled, PowerExchange transfers column-level processing of the change data and any UOW Cleanser (UOWC) processing from the source system to the PowerCenter Integration Service machine. Options are:
   |                      | - No. Disables offload processing. |
   |                      | - Yes. Enables offload processing. |
   |                      | - Auto. PowerExchange determines whether to enable or disable offload processing. Default is No. |
   | CAPI Connection Name | Specifies the name of the source CAPI_CONNECTION statement in the dbmover.cfg on the PowerCenter Integration Service machine. |

2. Copy the source-specific CAPI_CONNECTION statements from the DBMOVER configuration file on the source system to the dbmover.cfg configuration file on the PowerCenter Integration Service machine.
   For z/OS data sources, copy CAPI_CONNECTION statements of the types LRAP and UOWC.

3. Remove all z/OS-specific parameters from the UOWC CAPI_CONNECTION statement in the dbmover.cfg file on the PowerCenter Integration Service machine.

Related Topics:

- "Example of CDC Offload Processing with a z/OS Source " on page 380
- "CDC Offload Processing" on page 378
Example of CDC Offload Processing with a z/OS Source

In this example, you enable a CDC session with a PWX CDC Real Time connection to use offload processing for change data extraction from a data source on a z/OS system.

The source data remains on z/OS but all column-level and UOW Cleanser (UOWC) processing is offloaded to the PowerCenter Integration Service machine.

On the z/OS source system, the DBMOVER member in the RUNLIB library includes the following CAPI_CONNECTION statements:

```plaintext
CAPI_CONNECTION=(NAME=MV2UOWC,TYPE=(UOWC,CAPINAME=M2_LRAP,RSTRADV=600,MEMCACHE=20480,DATACLAS=UOWC))
CAPI_CONNECTION=(NAME=MV2_LRAP,TYPE=(LRAP,LOG=MV2L,AGENT=MV2A))
```

1. Copy the UOWC and LRAP CAPI_CONNECTION statements from the DBMOVER member on z/OS to the dbmover.cfg configuration file on the PowerCenter Integration Service machine.
   Remove z/OS-specific parameters, such as DATACLAS, from the UOWC CAPI_CONNECTION statement.
   This example uses the following CAPI_CONNECTION statements in the dbmover.cfg file on the PowerCenter Integration Service machine:
   ```plaintext
   CAPI_CONNECTION=(NAME=MV2UOWC,TYPE=(UOWC,CAPINAME=M2_LRAP,RSTRADV=600,MEMCACHE=20480))
   CAPI_CONNECTION=(NAME=MV2_LRAP,TYPE=(LRAP,LOG=MV2L,AGENT=MV2A))
   ```

2. Stop the CDC session.

3. Update the following attributes on the PWX CDC Real Time application connection for the CDC session:
   - For the **Offload Processing** option, select **Yes**.
   - For the **CAPI Connection Name** attribute, enter the name of the UOWC CAPI_CONNECTION statement. In this example, the name is MV2UOWC.

4. Restart the CDC session.

**RELATED TOPICS:**
- "Enabling Offload Processing for CDC Sessions" on page 379
- "CDC Offload Processing" on page 378
- "Rules and Guidelines for CDC Offload Processing" on page 379

Multithreaded Processing

Multithreaded processing uses multiple worker threads to distribute resource-intensive, column-level processing across multiple CPUs. Use multithreading if a single CPU cannot optimally handle extraction processing.

By default, PWXPC uses a single thread to process change data on the PowerCenter Integration Service machine. When you enable multithreading, PWXPC uses multiple threads to process change records.
Rules and Guidelines for Multithreaded Processing

Multithreaded processing can help improve performance for CDC sessions in specific situations. Use the following rule and guidelines to determine when multithreaded processing is useful and how to set the **Worker Threads** attribute:

- Use multithreaded processing when the PWX reader thread of a CDC session uses 100% of a single CPU on a multiple-CPU server on the PowerCenter Integration Service machine. In this situation, multithreading improves throughput by spreading PowerExchange processing across multiple threads. Otherwise, multithreading does not improve throughput.

- For optimal performance, verify that the value of the **Worker Threads** attribute does not exceed the number of installed or available processors on the PowerCenter Integration Service machine.

- When defining the PWX CDC application connection, you must either set the **Location** attribute to "local" to enable the extraction to access the source locally, or set the **Offload Processing** attribute to **Yes** to offload extraction processing.

- If processing slows or hangs for CDC sessions that use multiple worker threads, increase the MAXTASKS value in the DBMOVER configuration file to help improve performance.

Enabling Multithreaded Processing for CDC Sessions

To use multithreaded processing, you must configure some PWX CDC connection attributes.

The following table describes the PWX CDC Real Time application connection attributes that are required to enable multithreaded processing for a CDC session:

<table>
<thead>
<tr>
<th>Connection Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Worker Threads</td>
<td>Specifies the number of threads that PowerExchange uses on the PowerCenter Integration Service machine to process change data. Default is 0.</td>
</tr>
<tr>
<td>Array Size</td>
<td>If the <strong>Worker Threads</strong> value is greater than zero, specifies the size of the storage array, in number of records, for each thread. Default is 25.</td>
</tr>
</tbody>
</table>
Chapter 20

zIIP Exploitation

This chapter includes the following topic:

- PowerExchange zIIP Exploitation, 382

PowerExchange zIIP Exploitation

The IBM System z Integrated Information Processor (zIIP) is designed to help free up general computing capacity and lower overall total cost of computing for certain data and transaction processing workloads on z/OS. The zIIP can execute any suitable workload provided that software is designed to run the workload on the zIIP.

If you have one or more zIIPs installed, you can configure the PowerExchange Listener on z/OS so that some of its work is offloaded to a zIIP. If multiple PowerExchange Listeners are running, you can configure each of them to offload work to a zIIP.

To be eligible to run on a zIIP, work must meet the following requirements:

- Run in a WorkLoad Manager enclave which has been classified as being able to offload to a zIIP, also called a System Utility Processor (SUP)
- Run in an enclave System Request Block (SRB)

Programs that run in an SRB must meet the following requirements:

- Run in Supervisor state, key zero
- Do not issue any SVCs, except for SVC 13 (ABEND)
- For subroutines, do not call another subroutine

DBMOVER Statements for PowerExchange zIIP Exploitation

The following DBMOVER statements control zIIP configuration:

**SUP_SSNAME=subsystem_name**

Defines the subsystem name that identifies the PowerExchange Listener started task to the IBM Workload Manager for offloading work to a zIIP. If your system includes multiple Listeners, you can define a different name for each Listener. Enter a name of up to eight characters.

Default is PWXLSTNR.
**SUP_SSTYPE**=subsystem_type

Defines the name that the IBM Workload Manager uses as the subsystem type for the enclave SRB under which work is dispatched on the zIIP. Enter a name of up to four characters.

Default is PWX.

**USESUP**(Y/N)

Controls whether PowerExchange offloads zIIP-enabled PowerExchange Listener functions to a zIIP. Specify USESUP=Y to enable offloading to a zIIP.

**WORKCLASS**

Defines the transaction name for Workload Manager classification. Enter a name of up to eight characters.

Default is PWXWORK.

### z/OS System Log Messages for PowerExchange zIIP Exploitation

PowerExchange issues messages to the z/OS system log to report the status of PowerExchange zIIP operations.

The message IDs have the following form:

PWXmmm34xxs

The string mmm represents the calling routine and might be helpful to Informatica Global Customer support if you receive an error message.

xx are the last two digits of the message number.

The code s is either I for an informational message or E for an error message.

Use these messages to determine whether zIIP configuration was successful, as follows:

- **Informational messages** indicate successful configuration. The absence of these messages might indicate that prerequisites for zIIP offloading were not satisfied. For more information, see "Configuring PowerExchange to Offload Work to a zIIP" on page 383.
- **Error messages** indicate an error condition that, in most cases, requires you to call Informatica Global Customer Support.
- Messages PWXmmm3412E and PWXmmm3414E indicate possible error conditions but might not require you to contact Informatica Global Customer Support if rc = 4.

For more information, See PowerExchange Message Reference Volume 1.

### Configuring PowerExchange to Offload Work to a zIIP

Before you configure PowerExchange to offload work to a zIIP, verify that the following prerequisites are satisfied:

- The system callable services library SYS1.CSSLIB is available through the LNKLST concatenation or the LPALIB data set.
- The projected usage function (PROJECTCPU) in the IEAOPTxx member in the system PARMLIB is enabled.
  If you enable zIIP usage on a system without a zIIP, and PROJECTCPU is set to FALSE, the system does not project CPU usage as if a zIIP were present, and PowerExchange reports RC = 4 from IWM4EOCT. PowerExchange continues to run zIIP-enabled functions in SRB mode.
- All the libraries in the PowerExchange Listener STEPLIB concatenation are APF-authorised.
The DBMOVER configuration member does not include any TRACE statements.

1. Include the USESUP=Y statement in the DBMOVER configuration file on z/OS, and optionally include the following statements:
   - SUP_SSNAME
   - SUP_SSTYPE
   - WORKCLASS

2. Add PWX to the IBM Workload Manager for z/OS (WLM):
   a. From the main menu of the WLM ISPF application, add PWX as a subsystem type, or specify whatever value you specified for the SUB_SSTYPE statement in the DBMOVER configuration member.
   b. For each PowerExchange Listener, add a work qualifier with a type of SI (system instance) to the list. The name must match the value in the DBMOVER SUP_SSNAME statement (default PWXLSTNR).
   c. Optionally, change the default transaction name by using the TN qualifier type. This value must match the value in the DBMOVER WORKCLASS statement (default PWXWORK).
   d. Check the job log to verify that zIIP enablement is successful.

   If zIIP enablement is successful, the z/OS system log includes informational messages such as the following ones:

   PMXDSF3400I Checking processors...
   PMXDSF3401I Cpu 00 Serial FF04EEC52098 Type CP Rel. Speed 1
   PMXDSF3401I Cpu 01 Serial FF04EEC52098 Type CP Rel. Speed 1
   PMXDSF3401I Cpu 06 Serial FF04EEC52098 Type zIIP Rel. Speed 1
   PMXDSF3403I 1 Processor available for zIIP offload
   PMXNC03405I Connect to WLM Sub = PWX Subn = GRAPWX token = 140C211B
   PMXNC03409I Classify work to WLM Service class = 00038000
   PMXCE3411I WLM Create Enclave function = PMXFUNC enclave token = 0000000300000033
   PMXN8E3415I WLM Set Rules tok = PMXR id = IMMOCY ver = 00 cnt = 01 Dur = 100000 Pct = 100
   DTL-06603 Listener NODE1 VERM 9.5.0 Build DEV_BUILD started.

   If the job log does not include messages that indicate that zIIP was successfully enabled, verify that the prerequisites for zIIP enablement are satisfied. If not all libraries in the PowerExchange Listener STEPLIB concatenation are APF-authorized, or if the DBMOVER configuration member includes a TRACE statement, zIIP exploitation is disabled.
APPENDIX A

CDC for z/OS Troubleshooting

This appendix includes the following topics:

- CDC for z/OS Troubleshooting Overview, 385
- Problems Related to Propagating Data, 385

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

Before you contact Informatica Global Customer Support, gather information about the problem and your CDC environment for diagnostic use.

The following table identifies the information that you should gather, depending on your system characteristics:

<table>
<thead>
<tr>
<th>System Characteristic</th>
<th>Required Information</th>
</tr>
</thead>
</table>
| Problem description and related output | Description of the problem.  
Message output.  
Description of your troubleshooting procedure. |
| Processor                   | CPU type.                                                                             |
| z/OS operating system       | z/OS operating system version, release, and maintenance level including APARs.         |
| SMS use                     | Whether SMS is being used.                                                            |
| System security             | Security product.                                                                     |
|                             | Security package version and release.                                                 |
| PowerExchange version       | PowerExchange product version, release, and any installed hotfix or EBF.              |
| PowerExchange CDC source    | Source database type, version and release, and any maintenance that is applied.       |
| CDC target                  | Target operating system type, version and release, and any maintenance that is applied.  
The target operating system can be a Linux, UNIX, or Windows system or another z/OS system.  
Target database type, version and release, and any maintenance that is applied.  
The target can be a PowerCenter target. |
| PowerExchange Agent         | A copy of all PowerExchange Agent output.                                             |
| PowerExchange Logger for MVS| A copy of all PowerExchange Logger output.                                             |
| PowerCenter version         | PowerCenter version, release, and any maintenance.                                    |
This appendix includes the following topic:

- **Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source, 387**

### Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source

The time stamp that PowerExchange reports in the generated DTL__CAPXTIMESTAMP field in change records depends on the data source type and on certain parameter settings.

For PowerExchange data sources on z/OS and for PowerExchange Oracle CDC with LogMiner sources, the TIMESTAMP parameter in the UOWC_CAPI_CONNECTION controls the type of time stamp that PowerExchange reports in the DTL__CAPXTIMESTAMP field. If you set the TIMESTAMP parameter to COMMIT, PowerExchange reports the time stamp of the transaction commit on the source for all changes in the transaction. If you use the default parameter value of LOG, PowerExchange retrieves the time stamp from the source database logs. In this case, the time stamp type depends on the source type.

The following table describes the time stamps that PowerExchange reports when you use the default value of LOG for the TIMESTAMP parameter:

<table>
<thead>
<tr>
<th>Data Source Type</th>
<th>Time Stamp Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adabas</td>
<td>The HDDATE time stamp from the PLOG block header, which indicates when the block was written. <strong>Note:</strong> In Adabas environments with a low level of update activity, the same time stamp might be reported for multiple updates that occurred at different times.</td>
</tr>
<tr>
<td>Datacom table-based CDC</td>
<td>The Coordinated Universal Time (UTC) time or local time when the change record was written to the Datacom LXX log. The LOCAL_TIME parameter in the ECCR configuration member, ECCRDCMP, controls whether the UTC or local time is used.</td>
</tr>
<tr>
<td>DB2 for i5/OS</td>
<td>An i5/OS journal time stamp that reflects when the change was recorded in the journal.</td>
</tr>
<tr>
<td>DB2 for z/OS</td>
<td>The time at which the DB2 ECCR captured the change data record. Each record in a UOW has a different time stamp. Usually, this time stamp is a UTC value that reflects the time zone of the DB2 for z/OS system.</td>
</tr>
<tr>
<td>Data Source Type</td>
<td>Time Stamp Type</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IDMS</td>
<td>The time at which the change data record was written to the IDMS log file. This time stamp is equivalent to the storeclock (STCK) time stamp. It does not reflect the local time zone.</td>
</tr>
<tr>
<td>IMS log-based CDC</td>
<td>The time at which the change was recorded in the IMS logs.</td>
</tr>
<tr>
<td>IMS synchronous CDC</td>
<td>The time at which the change occurred.</td>
</tr>
<tr>
<td>Oracle CDC with LogMiner</td>
<td>The time stamp of the change on the source database, as recorded in the redo logs. This time reflects the local time zone.</td>
</tr>
<tr>
<td>Batch VSAM and CICS/VSAM</td>
<td>The time at which the change record was captured. Each record in a UOW has a different time stamp. Usually, this time stamp is a UTC value.</td>
</tr>
</tbody>
</table>

For other data sources that do not use the UOWC CAPI_CONNECTION statement, PowerExchange determines the appropriate time stamp to report in the DTL__CAPXTIMESTAMP field. For PowerExchange Express CDC for Oracle sources, the TIME_STAMP_MODE parameter in the OPTIONS statement of the Express CDC configuration file controls the time stamp type.

The following table describes the time stamp types that PowerExchange reports for these data sources:

<table>
<thead>
<tr>
<th>Data Source Type</th>
<th>Time Stamp Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2 for Linux, UNIX, and Windows</td>
<td>The time stamp of the transaction commit. This time stamp is an ascending virtual time stamp (VTS) of the DB2 system, which usually corresponds to the UTC value.</td>
</tr>
<tr>
<td>Microsoft SQL Server</td>
<td>The time at which the change was written to the distribution database.</td>
</tr>
<tr>
<td>PowerExchange Express CDC for Oracle</td>
<td>The time stamp type is controlled by the TIME_STAMP_MODE parameter setting in the OPTIONS statement of the Express CDC configuration file.</td>
</tr>
<tr>
<td></td>
<td>- If you use the default value of LOGTIME, PowerExchange reports the time stamp of the change on source database, as recorded in the redo logs. This time stamp reflects the local time zone.</td>
</tr>
<tr>
<td></td>
<td>- If you specify COMMITTIME, PowerExchange reports the time stamp of the transaction commit on the source database.</td>
</tr>
<tr>
<td></td>
<td>- If you specify BEGINTIME, PowerExchange reports the time stamp of the begin UOW log record.</td>
</tr>
</tbody>
</table>
INDEX

A
active batch ECCR 159
Adabas CDC
capturing changes from Adabas spanned records 140
components and data flow 137
configuring Adabas PLOG archiving JCL 140
ECRR requirements for accessing multiple Adabas databases 139
operational considerations 139
overview 137
PCAT Utility (DTLCCADW) 154
SAMPUEX2 member 141
testing installation and configuration 150
Adabas ECCR
ADAECP1 parameters 142
ADASEL_DSN parameter 144
CAPT_STATS_INTVL parameter 145
CAPT_STATS_TERSE parameter 145
COLDSTART parameter 146
COLL_END_LOG parameter 146
customizing JCL 149
DB_TYPE parameter 147
DBID parameter 147
ECRRNAME parameter 147
IGNORENOCHANGEUPDATES parameter 147
NO_DATA_WAIT parameter 148
NO_DATA_WAIT2 parameter 148
ON_SUSPENSION_ERROR_CONTINUE parameter 149
requirements for accessing multiple Adabas databases 139
Adabas log-based ECCR
adding a capture registration) 152
CAPT_STATS parameter 144
deleting a capture registration) 152
REFRESH_ALLOWED parameter 149
ADAECP1 member
Adabas ECCR parameters 142
ADASEL_DSN parameter
Adabas ECCR 144
adding active log data set definitions 78
Agent Security 54
AGENTGEN 45
AgentID 45
AGENTREP data set
BackToBackDelay parameter 47
Cache1 parameter 47
Cache2 parameter 47
Location parameter 47
RestartInterval parameter 47
UpdateInterval parameter 47
allocating restart data sets 77
Application Name attribute 330
application names 340
application recovery
batch VSAM considerations 162
archive log rules and guidelines 79
backing up PowerExchange Condense files 134
batch extraction mode
use for terminating CDC sessions 352
batch job requirements 158
batch mode 101
batch VSAM ECCR
interactions with other PowerExchange components 155
before indicator (BI) fields
use cases 316
BYPASS_VERSION_CHECKING parameter
IMS log-based ECCR 265

C
calculating the data set size 75
canceling the condense job 130
CAPI Connection Name Override attribute 328
CAPI connection statements
LRAP parameters 30
MEMCACHE parameter 372
RSTRADV parameter 372
UOWC parameters 32
CAPI_CONNECTION - LRAP statement
DBMOVER configuration file 30
CAPI_CONNECTION - UOWC statement
DBMOVER configuration file 32
CAPT_IMAGE parameter
PowerExchange Condense 109
CAPT_STATS parameter
Adabas log-based ECCR 144
Datacom table-based ECCR 184
IDMS log-based ECCR 243
IMS log-based ECCR 265
CAPT_STATS_INTVL parameter
Adabas ECCR 145
Datacom table-based ECCR 184
IDMS log-based ECCR 244
IMS log-based ECCR 266
CAPT_STATS_TERSE parameter
Adabas ECCR 145
Datacom table-based ECCR 185
IDMS log-based ECCR 244
IMS log-based ECCR 267
CAPTIMS member
IMS log-based ECCR parameters 263
CAPTPARM 105, 127
CAPTPARM member
PowerExchange Condense parameters 107
RESTART_TOKEN parameter 127
SEQUENCE_TOKEN 127
CAPTPARM parameters
PowerExchange Condense 107
capture registrations
  suspending and reactivating Adabas registrations 153
  suspending and reactivating Datacom registrations 194
  suspending and reactivating IDMS registrations 252
  suspending and reactivating IMS registrations 277
CCT
  DTLAMCPR 103
  CCVACTIVE 45
CDC sessions
  adding sources with DTLUAPPL-generated CURRENT_RESTART tokens 354
  adding sources with special override CURRENT_RESTART tokens 354
  changing and restarting 353
  cold starting 350
  commit processing 320
  default restart points 341
  defining terminating conditions 352
  methods of starting 340
  monitoring in PowerCenter 366
  monitoring in PowerExchange 359
  multithreaded processing 380
  offload processing 378
  performance details in Workflow Monitor 367
  PowerExchange Logger for LUW logging of data from remote source 297
  processing of multiple source definitions 318
  recovering 355
  recovery example 356
  recovery start 351
  restart points for warm starts 342
  restart token file 345
  session and connection attributes for CDC 326
  start methods 349
  stop command processing 352
  stopping 351
  tuning 371
  tuning buffer memory 377
  tuning overview 321
  warm starting 350
CDC tables
  Datacom 179
  CDC_BASE parameter
    Datacom table-based ECCR 185
  CDC_ID parameter
    Datacom table-based ECCR 185
  CDCL program 180
  CDCM program 180
  CDCT file
    tracking information for condense files 104
  CDCU program 180
change data extraction
  connection attributes for Logger for LUW log files remote from the source 305
  creating restart tokens 343
  extraction modes 311
  monitoring in PowerCenter 366
  monitoring in PowerExchange 359
  multithreaded processing 380
  offload processing 378
  overview 310
  overview of extracting change data 322
  task flow 323
  testing extraction maps 324
  tuning CDC sessions 371
  change indicator (CI) fields 316
  changing the size of existing active log data sets 79
  changing VSAM structures 162
checkpoint files 105
checkpoint record type 105
CHKPT_BASENAME 127
CHKPT_BASENAME parameter
  PowerExchange Condense 109
CHKPT_FILE_CTL parameter
  PowerExchange Condense 110
CHKPT_NUM parameter
  PowerExchange Condense 110
CHKPT_PRIM_ALLOC parameter
  PowerExchange Condense 111
CHKPT_SCND_ALLOC parameter
  PowerExchange Condense 111
CHKPT_VOLSERS parameter
  PowerExchange Condense 111
CICS regions
  configuring for CDC 169
CICS/VSAM CDC
  changing the structure of a VSAM source 177
  overview 164
  requirements and restrictions 165
  use of CICS global and task-related exit points 165
CICS/VSAM data sets
  stopping change capture for a data set 177
CICS/VSAM ECCR
  activating 172
  controlling with EDMC keywords 173
  defining to CICS 169
  example LOG START report 172
  stopping 176
CLEANUP parameter
  Datacom table-based ECCR 186
CLEANUP_INTERVAL parameter
  Datacom table-based ECCR 186
CLEANUP_STATISTICS parameter
  Datacom table-based ECCR 186
  close (pwxcmd) 37
CLOSE command
  PowerExchange Listener 37
CLOSE FORCE command
  PowerExchange Listener 37
  closeforce (pwxcmd) 37
  closing a VSAM dataset 161
  CmdAuthCheck 45
  CmdPrefix 45
  cold start 127
  cold starts
  CDC sessions 350
  determining restart tokens 341
COLDSTART parameter
  Adabas ECCR 146
  Datacom table-based ECCR 187
  IDMS log-based ECCR 245
  IMS log-based ECCR 267
  COLL_END_LOG parameter
  Adabas ECCR 146
  PowerExchange Condense 111
commands
  DISPLAY SUBSYS 293
  IMS 293
  SSR xEDP-ABORT 292
  SSR xEDP-CONTINUE 292
  SSR xEDP-STAT 292
  SSR xEDP-STATWTO 292
commit processing
  commitment control attributes 333
  examples 335
  in CDC sessions 320
commit processing (continued)
  target latency 332
  tuning 377
COND_CDCT_RET_P parameter
  PowerExchange Condense 111
condense (pxwcmd) 134
CONDENSE command
  PowerExchange Condense 101
condense files 104
CONDENSE_SHUTDOWN_TIMEOUT parameter
  PowerExchange Condense 112
CONDENSENAME parameter
  PowerExchange Condense 112
CONDF_FULL_FILE_CTL parameter
  PowerExchange Condense 112
CONDF_PART_BLKSZ parameter
  PowerExchange Condense 113
CONDF_PART_DATACLAS parameter
  PowerExchange Condense 113
CONDF_PART_LRECL parameter
  PowerExchange Condense 113
CONDF_PART_MGMTCLAS parameter
  PowerExchange Condense 114
CONDF_PART_STORCLAS parameter
  PowerExchange Condense 114
CONDF_PRIM_ALLOC parameter
  PowerExchange Condense 114
CONDF_SCND_ALLOC parameter
  PowerExchange Condense 114
CONDF_TYPE parameter
  PowerExchange Condense 115
CONDF_UNIT parameter
  PowerExchange Condense 115
CONDF_VOL parameter
  PowerExchange Condense 115
Configuration Parameters
  40
configuring
  post-log merge 92
  the post-log merge job 93
CONN_OVR parameter
  PowerExchange Condense 116
connection attributes
  Application Name 330
  attributes to set for CDC 326
  CAPI Connection Name Override 328
  commitment control attributes 333
  Event Table 331
  Idle Time 328
  Image Type attribute 327
  PWX Latency in seconds 331
  restart control attributes 330
  RestartToken File Folder 330
  RestartToken File Name 330
  continuous extraction mode 311
  Customer Support
    information Support requires for diagnosing problems 386
D
data set size determination 75
data spaces
  SCOPE=COMMON 40
data sharing environment 207
database row tests
  testing data access with an extraction map 324
Datacom table-based CDC
  ECCR functions 180
Datacom table-based CDC (continued)
  managing table definition changes 195
  overview 178
Datacom table-based ECCR
  adding a capture registration 193
  CAPT_STATS parameter 184
  CAPT_STATS_INTVL parameter 184
  CAPT_STATS_TERSE parameter 185
  CDC_BASE parameter 185
  CDC_ID parameter 185
  CLEANUP parameter 186
  CLEANUP_INTERVAL parameter 186
  CLEANUP_STATISTICS parameter 186
  COLDSTART parameter 187
  DB_TYPE parameter 187
  deleting a capture registration) 194
  ECCRDMP parameters 181
  ECCRNAME parameter 187
  LOCAL_TIME parameter 188
  MONITOR parameter 188
  MONITOR_INTERVAL parameter 188
  MUF parameter 189
  NO_DATA_WAIT parameter 189
  NO_DATA_WAIT2 parameter 189
  ON_SUSPENSION_ERROR_CONTINUE parameter 190
  REFRESH_ALLOWED parameter 191
  REG_MUF parameter 190
  RESTART_ADVANCE_ACTIVE parameter 191
  starting 193
  stopping 193
DATAMAP DD statement 273
DB_TYPE parameter
  Adabas ECCR 147
  Datacom table-based ECCR 187
  IDMS log-based ECCR 245
  IMS log-based ECCR 267
  PowerExchange Condense 116
DB2 catalog tables
  DATA CAPTURE CHANGES requirement 207
DB2 data sharing environments
  ECCR configuration considerations 204
DB2 ECCR
  capture directory tables 201
  configuration statements in REPL2OPT DD data set 210
  controlling data amount 227
  required access to DB2 catalog tables 209
DB2 for z/OS
  schema verification 231
  target tables materialized from image copies 218
DB2 for z/OS CDC
  altering columns in source tables 233
  changing the qualifier for table spaces with source tables 234
  changing the schema of registered source tables 232
  datatypes supported for CDC 197
  FIELDPROC and EDITPROC exit routines 200
  manually generating an event marker for the QUIESCE utility 228
  migrating a DB2 environment from data sharing to non-data-sharing 229
  migrating a DB2 subsystem to DB2 10 new-function mode 204
  migrating a DB2 subsystem to DB2 11 205
  migrating to a DB2 data sharing environment 229
  operational considerations 198
  overview 196
  recovering from unplanned schema changes to source tables 232
  replacing a table with another that has the same name 228
  stopping change capture 230
DB2 for z/OS ECCR
  altering DB2 system tables to set DATA CAPTURE CHANGES 224

Index 391
EDMCMUOW
restart processing 159
EDMKOPER module 169
EDMLRPRM parameters 70
EDMPARMS 103
EDMSDIR module
configuring options 45
EDMSDIR options module
AGENTID option 41
CCERR option 41
CENTURY option 41
DATE option 41
ESLLIB option 41
LGWAITTO option 41
LOGGER option 41
LOGRGRP option 41
SYSOUT option 41
TIME option 41
EDMSLOG 51
EDMUPARM options module
configuration introduction 60
DEFINE statement 60
END statement 66
event table processing
guidelines for using 330
implementing 331
use for terminating CDC sessions 352
EDMUPARM options module 66
enqueues considerations 40
ERROR_LOG parameter
IMS log-based ECCR 268
ERT records 105
Event Table attribute 331
extending
fileswitch (pwxcdm) 105, 134
FILESWITCH command
PowerExchange Condense 101
flush latency 331–333
formatting log data sets 83
F
FILE_SWITCH_CRIT parameter
PowerExchange Condense 101, 117
FILE_SWITCH_VAL parameter
PowerExchange Condense 101, 117
FILESWITCH 105
FILESWITCH command
PowerExchange Condense 101
Flush latency 331–333
formatting log data sets 83
G
GetIMSRBAByLevel function
IMS synchronous ECCR 282
GROUPDEFS parameter
PowerExchange Condense 118
HELP keyword, EDMC 173
H
Idle Time attribute
use for terminating CDC sessions 352
IDMS log-based CDC
Log Catalog 238
managing schema changes 253
overview 235
IDMS log-based ECCR
adding a capture registration 251
CAPT_STATS parameter 243
CAPT_STATS_INVL parameter 244
CAPT_STATS_TERSE parameter 244
COLDSTART parameter 245
configuring ECCR JCL 248
DB_TYPE parameter 245
deleting a capture registration 252
ECCRIDLP parameters 241
ECCRNAME parameter 245
failure 257
LOGSID parameter 246
NO_DATA_WAIT parameter 246
NO_DATA_WAIT2 parameter 246
ON_SUSPENSION_ERROR_CONTINUE parameter 247
operational considerations 237
REFRESH_ALLOWED parameter 247
RESTART_ADVANCE_ACTIVE parameter 247
SR2 and SR3 records 250
starting 251
IFI306 statement
controlling amount of data sent to DB2 ECCR 227
implementing IFI306 for the DB2 ECCR 227
manually creating an event marker for the QUIESCE utility 228
IGNORECHANGEUPDATES parameter
Adabas ECCR 147
Image Type attribute 327
Index 393
IMS database types
not supported by IMS synchronous ECCR 281
IMS log-based CDC
APF-authorization of STEPLIB libraries 272
changing source schema 278
comparison with IMS synchronous CDC 258
IMS log-based change capture
stopping change capture 275
IMS log-based ECCR
adding a capture registration 276
BYPASS_VERSION_CHECKING parameter 265
CAPT_STATS parameter 266
CAPT_STATS_INTVL parameter 267
CAPTIMS parameters 263
capture overview 259
COLDSTART parameter 272
DB_TYPE parameter 267
DBID parameter 268
DD statements in JCL 273
deleting a capture registration 276
ECCR programs
activating the IMS ECCR 290
adding the CRG.LOAD library to the DBRC JCL 286
comparison with IMS synchronous CDC 279
configuring the IMS region JCL 285
IMS synchronous change capture
stopping change capture 294
IMS synchronous ECCR
accessing IMS external subsystem modules 289
BMC Software products with required components 283
considerations 282
non-keyed segments 282
restrictions 281
IMSID parameter
IMS log-based ECCR 269
INIT keyword, EDMC 173
InitAuthCheck 45
integration with PowerCenter 25

K
KEY_CHANGE_ALW parameter
PowerExchange Condense 118

L
linkage index 40
listtask (pwxcmd) 37, 362
LISTTASK command 37, 362
Local Mode
adding log restrictions 238
LOCAL_TIME parameter
Datacom table-based ECCR 188
log catalog
adding Logs in Order 238
Log Catalog (LOGSCAT)
IDMS log-based CDC 238
LogBuffLimit 45
LogClass 45
LOGCLOSE 52
LogHold 45
LogLimit 45
LOGOPEN 52
LOGSID parameter
IDMS log-based ECCR 246
LOGSPIN 52
LRAP CAPI_CONNECTION parameters
parameters and syntax 30

M
managing
log and restart data sets 73
Maximum Rows Per commit attribute 333
message log 51
message log data sets
described 24
Minimum Row Per commit attribute 333
MNT table 179
MONITOR parameter
Datacom table-based ECCR 188
MONITOR_INTERVAL parameter
Datacom table-based ECCR 188
monitoring CDC sessions
methods 359
performance details in Workflow Monitor 367
PowerCenter 366
PowerCenter session log messages 367
PowerExchange extraction statistics messages 360
PowerExchange multithreaded processing statistics 361
PowerExchange read progress messages 360
viewing performance details in PowerCenter 369
moving
log data sets to other devices 89
MSGLVL parameter
IMS log-based ECCR 270
MUF parameter
IMS log-based ECCR 270
MVS/DFP checkpoint/restart utility
IMS log-based ECCR 270
multiple instances of the PowerExchange Logger 58
multiple schemas
restrictions 236
multiple-source processing
in CDC sessions 318
multithreaded processing
guidelines for usage 381
overview 321, 380
statistics messages 361
MVS/DFP checkpoint/restart utility 162
NO_DATA_WAIT parameter (continued)
IMS log-based ECCR 270
PowerExchange Condense 101, 119
NO_DATA_WAITZ parameter
Adabas ECCR 148
Datacom table-based ECCR 189
IDMS log-based ECCR 246
IMS log-based ECCR 270
PowerExchange Condense 119
number of data sets 76

offload processing
enabling for CDC sessions 379
example of offload processing 380
overview 321, 378
rules and guidelines 379
ON_SUSPENSION_ERROR_CONTINUE parameter
Adabas ECCR 149
Datacom table-based ECCR 190
IDMS log-based ECCR 247
IMS log-based ECCR 270
PowerExchange Condense 119

PCAT file
populating for Adabas CDC 140
PCAT Utility 154
performance
CDC session performance details 369
multithreaded processing 380
PLT initialization list 160
point-in-time recovery
batch VSAM change data 162
Post-Log Merge
restrictions 91
PowerCenter Client for PowerCenter (PWXPC) 25
PowerCenter integration with PowerExchange 25
PowerExchange Agent
AGENTCTL parameter descriptions 45
cache data sets 54
customizing JCL statements and parameters 48
EDMPARMS DD statement in JCL 48
EDMSCTL DD statement in JCL 48
EDMSDIR options module 41
overview 38
sample JCL 49
STARTUP parameter in JCL 48
PowerExchange Agent commands 160
PowerExchange Condense
NO_DATA_WAIT parameter 119
backing up output files 134
CAPT_IMAGE parameter 109
CAPTPARM parameters 107
CHKPT_BASENAME parameter 109
CHKPT_FILE_CTL parameter 110
CHKPT_NUM parameter 110
CHKPT_PRIM_ALLOC parameter 110
CHKPT_SCND_ALLOC parameter 111

PowerExchange Condense (continued)
CHKPT_VOLSERS parameter 111
cold start processing 127
COND_CDC_RET_P parameter 111
CONDENSE_SHUTDOWN_TIMEOUT parameter 112
CONDENSENAME parameter 112
COND_BK_FILE_CTL parameter 112
COND_PART_BKJSZ parameter 113
COND_PART_BKACLAS parameter 113
COND_PART_BKRECL parameter 113
COND_PART_BKMGMTCLAS parameter 114
COND_PART_BKSTORCLAS parameter 114
COND_PART_BKPRIMALLOC parameter 114
COND_PART_BKSCNDALLOC parameter 114
COND_PART_BKUNIT parameter 115
COND_PART_BKVOL parameter 115
CONN_OVR parameter 116
continuous mode 101
DB_TYPE parameter 116
DBID parameter 116
DTLCA output 104
enabling registrations for condense processing 100
EXT_CAPT_MASK parameter 116
FILE_SWITCH_CRIT parameter 117
FILE_SWITCH_VAL parameter 117
GROUPDEF parameter 118
KEY_CHANGE_ALW parameter 118
messages from condense job 130
NO_DATA_WAIT parameter 119
OPER_WTO parameter 120
overview 99
RESTART_TOKEN parameter 120
SEQUENCE_TOKEN parameter 120
SIGNALLING parameter 121
VERBOSE parameter 121

PowerExchange Condense parameters
CAPTPARM member 107
PowerExchange configuration file
DTLFCFG 103
PowerExchange Listener
DD statements in Listener JCL 30
DISPLAY ACTIVE command 362
LISTTASK command 37, 362
starting 37
stopping 37
STOPTASK command 37
PowerExchange Logger
commands for controlling the Logger 69
PowerExchange Logger for Linux, UNIX, and Windows
capture registrations for logging data from remote sources 305
configuration parameters for logging z/OS source data to a remote
Logger 301
configuration tasks for remote logging 301
connection attributes for log files remote from the source 305
DBMOVES for logging data from remote sources 305
example of logging data from a remote source 306
logging data from remote sources 297
monitoring statistics 364
rules and guidelines for logging data from a remote source 300
PowerExchange Logger for MVS
active log and emergency restart data sets 66
archive options 63
configuration considerations 59
customizing Logger JCL 67

defining log data sets to ERDS 84
PowerExchange Logger for MVS (continued)
deleting log data sets from ERDS 67
log entries in the ERDS 67
logging options 65
planning considerations 59
Post-Log Merge 91
sample proc 68
SYSTEM_OPTIONS parameters 61
PowerExchange Logger overview 57
PowerExchange message data sets 106
PowerExchange-generated extraction map columns
  DTL__BIColumnname 311
  DTL__CAPXACTION 311
  DTL__CAPXASDELIND 311
  DTL__CAPXRESTART1 311, 344
  DTL__CAPXRESTART2 311, 344
  DTL__CAPXRROWID 311
  DTL__CAPXRRN 311
  DTL__CAPXTIMESTAMP 311
  DTL__CAPXUOW 311
  DTL__CAPXUSER 311
  DTL__CIColumnname 311
  DTL__CAPXROWID 311
  DTL__CAPXRRN 311
  DTL__CAPXTIMESTAMP 311
  DTL__CAPXUOW 311
  DTL__CAPXUSER 311
  DTL__CIColumnname 311
Post-Log Merge 91
sample proc 68
SYSTEM_OPTIONS parameters 61
PowerExchange-generated extraction map columns
  DTL__BIColumnname 311
  DTL__CAPXACTION 311
  DTL__CAPXASDELIND 311
  DTL__CAPXRESTART1 311, 344
  DTL__CAPXRESTART2 311, 344
  DTL__CAPXRROWID 311
  DTL__CAPXRRN 311
  DTL__CAPXTIMESTAMP 311
  DTL__CAPXUOW 311
  DTL__CAPXUSER 311
  DTL__CIColumnname 311
PWX Latency in seconds attribute 331
PWXcmmd
  close 37
  close command 37
  closeforce 37
  closeforce command 37
  condense command 134
  displaystatus 134
  filesswitch 105, 134
  listtask 362
  listtask command 362
  shutcond 134
  shutcond command 130
  shutdown 134
  shutdown command 105, 130
  stoptask 37
PWXPC 25
PWXCREG utility
  suspending and reactivating Adabas registrations 153
  suspending and reactivating Datacom registrations 194
  suspending and reactivating IDMS registrations 252
  suspending and reactivating IMS registrations 277
PWXUCREG utility
  suspending and reactivating Adabas registrations 153
  suspending and reactivating Datacom registrations 194
  suspending and reactivating IDMS registrations 252
  suspending and reactivating IMS registrations 277

R
real-time extraction mode 311
Real-Time Flush Latency in milli-seconds attribute 333
RECID parameter
  IMS log-based ECCR 271
recover, point-in-time
  batch VSAM change data 162
recovery
  IDMS log-based ECCR 257
  recovery
    CDC sessions 355
    example of session recovery processing 356
    PM_REC_STATE table 338, 339
    PM_RECOVERY table 338
    PM_TGT_RUN_ID table 338
    recovery information for nonrelational targets 339
    recovery state file for nonrelational targets 339
    recovery tables for relational targets 338
    recovery and restart processing 337
    recovery scenarios 95
    recovery start
      CDC sessions 351
    REFRESH_ALLOWED parameter
      Adabas log-based ECCR 149
      Datacom table-based ECCR 191
      IDMS log-based ECCR 247
      IMS log-based ECCR 271
    Refreshsscvt 45
    REG_MUF parameter
      Datacom table-based ECCR 190
      remote logging of z/OS data using the PowerExchange Logger for Linux, UNIX, and Windows 301
    REPCLOSE 52
    REPDB2CT member
      CA NAME statement 209
      STOPAFT statement 209
    UOWPREFIX statement 209
    REPDBZOP member
      CHKSCHM statement 210
      COMMITINT statement 210
      EC PERMIL statement 210
      IFI306 statement 210
      MODE statement 210
      ROWNOTDECOMPRESSED statement 210
      START keyword 210
      STAT LEV statement 210
      TRACE statement 210
    REPL2CTL DD data set
      CA NAME statement 209
      STOPAFT statement 209
    UOWPREFIX statement 209
    REPL2OPT DD data set
      CA NAME statement 209
      STOPAFT statement 209
    UOWPREFIX statement 209
    REPOPEN 52
    RepositoryDSN 45
    REPOSITORYDSN 52
    RepositoryMode 45
    REPSTATUS 52
    resolving in-doubt units of work 71
restart
  default restart points 341
  earliest restart points 341
  methods of starting CDC sessions 349
  restart points with null restart tokens 341
  warm starting CDC sessions 350
  warm starts
    CDC sessions 350
restart control options
  Application Name connection attribute 330
  RestartToken File Folder attribute 330
  RestartToken File Name attribute 330
  restart of CDC sessions
    processing by start type 340
  restart token file
    $PMRootDir/Restart 344
    comment statements 347
    example file 347
  explicit override statements 346
  overview 317
  special override statement 346

396 Index
restart token file (continued) statement types 345
syntax 345
restart tokens
creating for DB2 target tables materialized from image copies 218
creating for extraction sessions 343
determining for cold starts 341
displaying in DTL__CAPXRESTART2 column 344
overview 317
recovery state file 339
recovery state table 339
RESTART_ADVANCE_ACTIVE parameter
Datacom table-based ECCR 191
IDMS log-based ECCR 247
RESTART_TOKEN parameter
PowerExchange Condense 120
restarting processing
EDMCUOW DD statement 159
RestartToken File Folder attribute 330
RestartToken File Name attribute 330
RESUME 52
row tests
testing data access with an extraction map 324

S
SAMPEXTU member
JCL for populating the Adabas CDC PCAT file 140
Sample Messages 50
SAMPUEX2 member
customizing for Adabas CDC 141
JCL for populating the Adabas CDC PCAT file 140
security
extracting z/OS data from remote PowerExchange Logger for LUW log files 300
sequence tokens
displaying in DTL__CAPXRESTART1 column 344
SEQUENCE_TOKEN parameter
PowerExchange Condense 120
session attributes
attributes to set for CDC 326
SHOW_THREAD_PERF parameter 361
shutcmd (pwxcmd) 134
SHUTCOND command
PowerExchange Condense 101
SHUTDOWN 52, 105
shutdown (pwxcmd) 105, 134
SHUTDOWN command
PowerExchange Condense 101
shutting down
STOP command 130
shutting down Condense 130
SIGNALING parameter
PowerExchange Condense 121
size and number of active log data sets 74
SR2OUT
DTLUCSR2 DD Card 250
SR2TOTAL
DTLUCSR2 DD Card 250
SRT record 105
START
PowerExchange Agent command 160
starting the PowerExchange Logger 69
STARTTIME parameter
IMS log-based ECCR 272
Startup 45
STOP
PowerExchange Agent command 160
STOP command
shutting down 130
stopping the ECCR 161
stopping the PowerExchange Logger 69
stoptask (pwxcmd) 37
STOPTASK command
stopping CDC sessions 351
synchronous ECCR
IMS Batch Backout utility 295
MVS checkpoint and restart 295
overview 280
recovery considerations 295
system requirements 91

T
target latency 332
task flow
extracting change data 323
TaskLimit 45
terminating conditions
Idle Time attribute for CDC sessions 328
testing Adabas CDC configuration 150
timed checkpoint considerations for dormant member Loggers 94
tracks per cylinder and bytes per track 76
TSN table 179
tuning
overview of tuning options 321
tuning CDC sessions
APPBFSIZE statement 372
buffer memory 377
CAPI_CONNECTION MEMCACHE parameter 372
CAPI_CONNECTION RSTRADV parameter 372
commit processing attributes 377
DBMOVER tuning parameters 372
methods 371
NODE and LISTENER buffer size parameters 372
PowerCenter CDC connection attributes 375
TRACE statement 372

U
UOW Count attribute 333
UOWC CAPI_CONNECTION parameters
parameters and syntax 32
use cases 316
using post-log merge 90
utilities
DTLCUIML 275

V
VERBOSE parameter
PowerExchange Condense 121
VSAM batch change capture
overview 155
VSAM batch ECCR
batch VSAM ECCR restrictions 156
output 160
restrictions 156
stopping 161
W
warm starts
  restart points used 342
WRITE_RESTART_SECS parameter
  IMS log-based ECCR 272

X
XCF groups 59

Z
zIIP exploitation
  DBMOVER statements 382
  zIIP, offloaidng work to 382