



Informatica® PowerExchange
10.5.2

バルクデータ移動ガイド

本ソフトウェアおよびマニュアルは、使用および開示の制限を定めた個別の使用許諾契約のもとでのみ提供されています。本マニュアルのいかなる部分も、いかなる手段（電子的複製、写真複製、録音など）によっても、Informatica LLC の事前の承諾なしに複製または転載することは禁じられています。

米政府の権利プログラム、ソフトウェア、データベース、および関連文書や技術データは、米国政府の顧客に配信され、「商用コンピュータソフトウェア」または「商業技術データ」は、該当する連邦政府の取得規制と代理店固有の補足規定に基づきます。このように、使用、複製、開示、変更、および適応は、適用される政府の契約に規定されている制限およびライセンス条項に従うものとし、政府契約の条項によって適当な範囲において、FAR 52.227-19、商用コンピュータソフトウェアライセンスの追加権利を規定します。

Informatica、Informatica ロゴ、PowerCenter、および PowerExchange は、米国およびその他の国における Informatica LLC の商標または登録商標です。Informatica の商標の最新のリストは、次の Web サイトにあります <https://www.informatica.com/trademarks.html>。その他の企業名および製品名は、それぞれの企業の商標または登録商標です。

オプトアウトの権利の制限の下、本ソフトウェアによって、本ソフトウェアがデプロイされているコンピューティングおよびネットワーク環境に関する情報、デプロイメントのデータ使用状況およびシステム統計情報が米国の Informatica に自動的に送信されます。この送信は Informatica のプライバシーポリシーの下にサービスの一部と見なされ、Informatica は、<https://www.informatica.com/in/privacy-policy.html> のプライバシーポリシーに従って、この情報を使用または処理します。使用状況の収集は、Administrator ツールで無効にすることができます。

本ソフトウェアまたはドキュメンテーション（あるいはその両方）の一部は、第三者が保有する著作権の対象となります。必要な第三者の通知は、製品に含まれています。

特許については、<https://www.informatica.com/legal/patents.html> を参照してください。

本マニュアルの情報は、予告なしに変更されることがあります。このドキュメントで問題が見つかった場合は、infa_documentation@informatica.com までご報告ください。

Informatica 製品は、それらが提供される契約の条件に従って保証されます。Informatica は、商品性、特定目的への適合性、非侵害性の保証等を含めて、明示的または黙示的ないかなる種類の保証をせず、本マニュアルの情報を「現状のまま」提供するものとします。

発行日: 2022-07-07

目次

序文	9
Informatica のリソース.....	9
Informatica Network.....	9
Informatica ナレッジベース.....	9
Informatica マニュアル.....	10
Informatica 製品可用性マトリックス.....	10
Informatica Velocity.....	10
Informatica Marketplace.....	10
Informatica グローバルカスタマサポート.....	10
 第 1 章 : PowerExchange バルクデータ移動の概要	11
PowerExchange バルクデータ移動の概要.....	11
バルクデータ移動のコンポーネント.....	12
バルクデータ移動の構成例.....	13
バルクデータ移動のセキュリティ.....	14
バルクデータ移動のタスクフロー.....	14
 第 2 章 : PowerExchange Listener	16
PowerExchange Listener の概要.....	16
バルクデータ移動のための PowerExchange Listener 構成タスク.....	17
PowerExchange DBMOVER ファイルのカスタマイズ.....	17
APPBUFSIZE 文.....	18
APPBUFSIZEDYN 文.....	18
CMDNODE 文.....	19
CREDENTIALS_CASE 文.....	20
DATAMAP_SERVER 文.....	21
DM_SUBTASK 文.....	22
DMX_DIR 文.....	23
LISTENER 文.....	23
LOGPATH 文.....	26
MAXTASKS 文.....	26
NODE 文.....	27
SECURITY 文.....	29
SUBMITTIMEOUT 文.....	32
SVCNODE 文.....	33
TCPIPVER 文.....	34
TRACING 文.....	35
TCP/IP ポート番号の構成.....	41
複数のログファイルの構成.....	42
代替のコンフィギュレーションファイルまたはライセンスキーファイルの指定.....	42

z/OS での PowerExchange リスナ JCL の構成.	43
CA TCPAccess 用の PowerExchange TCP/IP ソケットの構成.	43
PowerExchange と PowerCenter との環境変数の非互換性.	44
PowerExchange Listener の起動.	45
PowerExchange リスナの起動 (i5/OS)	45
PowerExchange リスナの起動 (Linux および UNIX)	45
PowerExchange リスナの起動 (z/OS)	46
PowerExchange リスナの起動 (Windows)	46
PowerExchange Listener の管理.	47
PowerExchange Listener の停止.	47
リモートの PowerExchange Listener のテスト.	47
PowerExchange リスナの制御.	48
第 3 章 : Adabas バルクデータ移動.	50
Adabas バルクデータ移動の概要.	50
Adabas バルクデータ移動に関する考慮事項.	50
Adabas バルクデータ移動の構成.	51
リモートの Adabas ソースまたはターゲットへの接続の設定およびテスト.	51
DBMOVE 構成メンバへの Adabas 固有の文の追加.	52
Adabas LOAD ライブラリへのアクセスの構成 (z/OS)	58
ADARUN 制御文のデフォルト SVC のオーバーライド (z/OS)	58
暗号コードで暗号化された Adabas ソースの復号化.	59
Adabas バルクデータの移動.	60
第 4 章 : Datacom バルクデータ移動.	62
Datacom バルクデータ移動の概要.	62
Datacom バルクデータ移動の構成.	62
リモートの Datacom ソースへの接続の設定およびテスト.	62
Datacom バルクデータの移動.	63
第 5 章 : DB2 for i5/OS のバルクデータ移動.	65
DB2 for i5/OS バルクデータ移動の概要.	65
DB2 for i5/OS バルクデータ移動に関する考慮事項.	66
DB2 for i5/OS バルクデータ移動用にサポートされているデータタイプ.	67
DB2 for i5/OS バルクデータ移動の構成.	68
DB2 for i5/OS ソースまたはターゲットへの接続の設定およびテスト.	68
DBMOVE 構成メンバへの DB2 固有の文の追加.	68
DB2 for i5/OS のバルクデータ移動.	71
DB2 for i5/OS バルクデータの移動 - DB2 アクセス方式.	71
DB2 for i5/OS バルクデータの移動 - SEQ アクセス方式.	72
ソースまたはターゲットをトラブルシューティング用に再作成するための SQL 文の生成.	72
i5/OS のアップグレード後の PowerExchange 環境のリフレッシュ.	73

第 6 章 : DB2 for Linux、UNIX、および Windows バルクデータ移動.....	75
DB2 バルクデータ移動の概要.....	75
DB2 for Linux、UNIX、および Windows バルクデータ移動用にサポートされているデータタイプ..	76
DB2 バルクデータ移動の構成.....	77
リモート DB2 ソースまたはターゲットへの接続の設定およびテスト.....	77
DBMOVER コンフィギュレーションファイルの DB2 文の構成.....	77
DB2 バルクデータの移動.....	79
 第 7 章 : DB2 for z/OS のバルクデータ移動.....	 80
DB2 for z/OS バルクデータ移動の概要.....	80
DB2 for z/OS バルクデータ移動用にサポートされているデータタイプ.....	81
DB2 for z/OS の TIMESTAMP データ型.....	82
DB2 for z/OS LOB データ型.....	82
DB2 for z/OS バルクデータ移動の構成.....	83
リモートの DB2 for z/OS ソースまたはターゲットへの接続の設定およびテスト.....	83
DBMOVER 構成メンバへの DB2 固有の文の追加.....	84
DB2 リソースへのアクセスに必要な権限.....	93
DB2 リソースにアクセスするための権限の付与.....	94
DB2 複数行 FETCH 文および INSERT 文.....	94
リレーショナルソース定義またはターゲット定義を含む DB2 for z/OS バルクデータ移動.....	95
リレーショナルソース定義またはターゲット定義を含む DB2 for z/OS バルクデータの移動.....	96
イメージコピーソースを含む DB2 for z/OS バルクデータ移動.....	96
DB2 for z/OS イメージコピーの作成.....	97
データソースとしての圧縮イメージコピー.....	97
DB2 イメージコピーからのターゲットのマテリアライズ.....	98
非リレーショナルソース定義を含む DB2 for z/OS バルクデータ移動.....	99
バルクデータソースとしての DB2 アンロードファイル.....	99
非リレーショナルソース定義を含む DB2 for z/OS バルクデータの移動.....	100
DB2 LOAD ユーティリティを使用したバルクデータのロード.....	101
DB2 LOAD ユーティリティ用の PowerExchange テンプレート.....	101
構成手順の概要.....	102
DB2 LOAD ユーティリティのための DBMOVER 構成メンバのカスタマイズ.....	103
バルクデータロード用の PowerExchange テンプレートのカスタマイズ.....	103
PowerExchange リスナ JCL のカスタマイズ.....	104
PowerCenter における DB2 バルクデータロードセッションの構成.....	104
DB2 バルクロード操作のためのユーティリティ ID.....	109
ローダー JCL テンプレートの置換変数.....	109
DB2 LOAD ユーティリティオプション.....	110
単一のパーティションおよび複数のパーティションへのバルクロード.....	110
 第 8 章 : IDMS バルクデータ移動.....	 112
IDMS バルクデータ移動の概要.....	112

IDMS に関する PowerExchange の構成.	112
PowerExchange for IDMS Security.	113
リモートの IDMS ソースへの接続の設定およびテスト.	113
DBMOVER 構成メンバへの IDMS 固有の文の追加.	114
IDMS ロードライブラリのコピーの APF 許可.	116
APF 許可アクセスに対する z/OS プログラム呼び出しサービスルーチンの使用.	117
IDMS バルクデータ移動用の Netport ジョブの設定.	117
IDMS バルクデータの移動.	118
第 9 章 : IMS バルクデータ移動.	120
IMS バルクデータ移動の概要.	120
IMS データのアクセス方式.	121
PowerExchange におけるグループソース処理.	121
IMS カタログの使用.	121
IMS バルクデータ移動の構成.	122
構成に関する一般的な考慮事項.	122
IMS カタログを使用するための設定タスク.	122
DL/1 バッチデータマップを使用したバルクデータ移動用の PowerExchange の構成.	125
IMS ODBA データマップを使用したバルクデータ移動用の PowerExchange の構成.	125
IMS ソースまたはターゲットへの接続の設定とテスト.	126
IMS バルクデータ移動の実装.	126
前提情報.	126
実装に関する考慮事項.	127
IMS ソースへのアクセスに影響を与える設定.	128
ルックアップトランスフォーメーション.	129
IMS データベースからのターゲットのマテリアライズ.	129
バルクデータ移動と IMS アンロードファイル.	130
データソースとしての IMS アンロードデータセットの使用.	130
IMS アンロードデータセット用の DBMOVER 構成文.	132
IMS アンロードデータセットからのターゲットのマテリアライズ.	132
IMS アンロードデータセットへの複数レコードの書き込み.	133
第 10 章 : Microsoft SQL Server バルクデータ移動.	138
Microsoft SQL Server バルクデータ移動の概要.	138
SQL Server バルクデータ移動用にサポートされているデータタイプ.	139
Microsoft SQL Server バルクデータ移動の構成.	140
DBMOVER コンフィギュレーションファイルへの Microsoft SQL Server 文の追加.	141
SQL Service のソースまたはターゲットへの接続の設定およびテスト.	141
Microsoft SQL Server バルクデータの移動.	142
SQL Server バルクロードユーティリティを使用したバルクデータのロード.	143
構成手順の概要.	143
Microsoft SQL Server バルクロードの PWX MSSQLServer 接続属性.	143
ターゲットロードタイプセッションプロパティ.	144

SQL Server バルクロードセッションの PowerCenter セッションログ.....	144
バルクロードに関するチューニングの考慮事項.....	144
第 11 章 : Oracle バルクデータ移動.....	145
Oracle バルクデータ移動の概要.....	145
Oracle バルクデータ移動用にサポートされているデータタイプ.....	146
Oracle バルクデータ移動の構成.....	147
リモートの Oracle ソースまたはターゲットへの接続の設定およびテスト.....	147
DBMOVER コンフィギュレーションファイルへの Oracle 固有の文の追加.....	148
Oracle バルクデータの移動.....	149
第 12 章 : シーケンシャルファイルのバルクデータ移動.....	151
シーケンシャルファイルのバルクデータ移動の概要.....	151
リモートのソースまたはターゲットへの接続の設定およびテスト.....	152
シーケンシャルファイルのバルクデータの移動.....	152
シーケンシャルファイルのデータマップのプロパティ.....	153
シーケンシャルファイルの概念.....	153
i5/OS ファイル.....	155
Linux および UNIX ファイル.....	155
Windows ファイル.....	156
z/OS ファイル.....	156
Linux、UNIX、Windows での読み取り用の z/OS ファイルのダウンロード.....	158
ユーザーアクセス方法の例.....	159
複数レコードのシーケンシャルファイルのグループソース処理.....	160
シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込み.....	161
シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みのル ールとガイドライン.....	161
シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みの実行.....	163
第 13 章 : VSAM バルクデータ移動.....	166
VSAM バルクデータ移動の概要.....	166
VSAM バルクデータ移動の構成.....	167
リモートのソースまたはターゲットへの接続の設定およびテスト.....	167
DBMOVER 構成メンバへの VSAM 固有の文の追加.....	168
VSAM バルクデータの移動.....	169
複数レコードの VSAM データセットのグループソース処理.....	171
シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込み.....	171
VSAM ターゲットまたはシーケンシャルファイルターゲットへの複数レコード書き込みのル ールとガイドライン.....	172
シーケンシャルターゲットまたは VSAM ターゲットへの複数レコード書き込みの実行.....	173
第 14 章 : フォールトトレランスを持つデータ書き込み.....	177
PowerExchange ターゲットへのデータ書き込みのモード.....	177

フォールトトレランスの概要.	178
フォールトトレランスを持つエラー処理.	179
エラー処理の構成.	179
エラーしきい値の構成.	179
カスタマイズされたエラー処理用のエラーアクションファイルの作成.	180
フォールトトレランスを持つ拒否ファイル.	181
拒否ファイルの構造.	181
拒否ファイルの命名規則.	183
拒否ファイルの作成の無効化.	186
第 15 章 : 監視およびチューニングオプション.	187
バルクデータ移動セッションの監視.	187
PowerExchange におけるバルクデータ移動セッションの監視.	187
バルクデータ移動セッションのチューニング概要.	193
PowerExchange DBMOVE 文によるバルクデータ移動セッションのチューニング.	194
接続属性によるバルクデータ移動セッションのチューニング.	196
接続プール.	197
データマップのキャッシュ.	198
データマップのキャッシュの有効化.	198
単一ジョブモードまたは複数ジョブモードで実行するためのデータマップのキャッシュの設定.	199
データマップのキャッシュの設定 - 例.	200
バルクデータオフロードとマルチスレッド処理.	201
バルクデータオフロード処理に関するルールおよびガイドライン.	202
マルチスレッド処理に関するルールおよびガイドライン.	202
バルクデータ移動セッションのためのオフロード処理とマルチスレッド処理の有効化.	203
バルクデータ移動セッション内のパイプラインパーティション.	205
Reader パーティション化.	206
Writer パーティション化.	209
パーティション化されたバルクデータ移動セッションのパフォーマンスの評価.	215
パーティション化されたセッションのチューニング.	215
PowerExchange zIIP Exploitation.	216
PowerExchange zIIP Exploitation の DBMOVE 文.	217
PowerExchange zIIP Exploitation の z/OS システムログメッセージ.	217
作業を zIIP にオフロードするように PowerExchange を設定.	218
PowerExchange リスナの WLM サービスクラスへの割り当て.	219
PowerExchange リスナのサブタスク.	219
PowerExchange リスナのリソースの使用量.	219
索引.	220

序文

『Informatica® PowerExchange® バルクデータ移動ガイド』を使用して、データソースタイプごとに PowerExchange バルクデータ移動操作を設定する方法について説明します。PowerExchange はデータを特定のポイントインタイムで PowerExchange のリレーショナルソースと非リレーショナルソースから読み取り、そのデータを PowerCenter バッチセッションでターゲットに抽出し、ロードできるようにすることができます。

Informatica のリソース

Informatica は、Informatica Network やその他のオンラインポータルを通じてさまざまな製品リソースを提供しています。リソースを使用して Informatica 製品とソリューションを最大限に活用し、その他の Informatica ユーザーや各分野の専門家から知見を得ることができます。

Informatica Network

Informatica Network は、Informatica ナレッジベースや Informatica グローバルカスタマサポートなど、多くのリソースへの入口です。Informatica Network を利用するには、<https://network.informatica.com> にアクセスしてください。

Informatica Network メンバーは、次のオプションを利用できます。

- ナレッジベースで製品リソースを検索できます。
- 製品の提供情報を表示できます。
- サポートケースを作成して確認できます。
- 最寄りの Informatica ユーザーグループネットワークを検索して、他のユーザーと共同作業を行えます。

Informatica ナレッジベース

Informatica ナレッジベースを使用して、ハウツー記事、ベストプラクティス、よくある質問に対する回答など、製品リソースを見つけることができます。

ナレッジベースを検索するには、<https://search.informatica.com> にアクセスしてください。ナレッジベースに関する質問、コメント、ご意見の連絡先は、Informatica ナレッジベースチーム (KB_Feedback@informatica.com) です。

Informatica マニュアル

Informatica マニュアルポータルでは、最新および最近の製品リリースに関するドキュメントの膨大なライブラリを参照できます。マニュアルポータルを利用するには、<https://docs.informatica.com> にアクセスしてください。

製品マニュアルに関する質問、コメント、ご意見については、Informatica マニュアルチーム (infa_documentation@informatica.com) までご連絡ください。

Informatica 製品可用性マトリックス

製品可用性マトリックス (PAM) には、製品リリースでサポートされるオペレーティングシステム、データベースなどのデータソースおよびターゲットが示されています。Informatica PAM は、<https://network.informatica.com/community/informatica-network/product-availability-matrices> で参照できます。

Informatica Velocity

Informatica Velocity は、Informatica プロフェッショナルサービスが開発したヒントとベストプラクティスのコレクションで、多数のデータ管理プロジェクトから得た実体験に基づいています。Informatica Velocity には、世界中の組織と連携してデータ管理ソリューションを計画、開発、デプロイ、管理する Informatica コンサルタントによる集合知を表しています。

Informatica Velocity リソースには、<http://velocity.informatica.com> からアクセスしてください。Informatica Velocity についての質問、コメント、またはアイデアがある場合は、ips@informatica.com から Informatica プロフェッショナルサービスにお問い合わせください。

Informatica Marketplace

Informatica Marketplace は、お使いの Informatica 製品を拡張したり強化したりするソリューションを検索できるフォーラムです。Marketplace で、Informatica デベロッパーやパートナーからの多数のソリューションを活用すれば、生産性を向上したり、プロジェクトでの実装時間を短縮したりできます。Informatica Marketplace は、<https://marketplace.informatica.com> からアクセスしてください。

Informatica グローバルカスタマサポート

電話または Informatica Network を介してグローバルカスタマサポートに連絡できます。

各地域の Informatica グローバルカスタマサポートの電話番号は、Informatica Web サイト (<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>) を参照してください。

Informatica Network のオンラインサポートリソースを見つけるには、<https://network.informatica.com> にアクセスして eSupport オプションを選択します。

第 1 章

PowerExchange バルクデータ移動の概要

この章では、以下の項目について説明します。

- [PowerExchange バルクデータ移動の概要, 11 ページ](#)
- [バルクデータ移動のコンポーネント, 12 ページ](#)
- [バルクデータ移動の構成例, 13 ページ](#)
- [バルクデータ移動のセキュリティ, 14 ページ](#)
- [バルクデータ移動のタスクフロー, 14 ページ](#)

PowerExchange バルクデータ移動の概要

PowerExchange では、異なるオペレーティングシステム間で、大量のデータを 1 回の操作で効率的に移動できます。このバルクデータ移動機能を使用して、データターゲットの具体化または完全なリフレッシュ、データソースおよびデータターゲットの再同期、またはデータウェアハウス、データマート、運用データストアの作成を行うことができます。

ターゲットを具体化した後、PowerExchange Change Data Capture (CDC) を使用して（このオプションがサイトで使用可能な場合）、ターゲットを継続的に更新できます。変更データだけを移動することによって、ターゲット上のデータをより速くリフレッシュすることができます。

PowerExchange では、同一のオペレーティングシステムまたは異なるオペレーティングシステム上に存在する異なるタイプのソースとターゲット間で、バルクデータを移動することができます。PowerExchange は PowerCenter® と連結して、リレーショナルデータベース、非リレーショナルデータベース、フラットファイル、および順次データセットとの間でデータの抽出および書き込みを行います。バルクデータ移動に使用できるデータソースおよびデータターゲットのタイプは、PowerExchange ライセンスによって異なります。

Linux、UNIX、および Windows システムでは、PowerExchange で以下のタイプのデータソースおよびターゲットに対してバルクデータを移動できます。

- DB2 for Linux, UNIX, and Windows テーブル
- Oracle テーブル
- Microsoft SQL Server テーブル
- フラットファイル

i5/OS システムでは、PowerExchange で DB2 for i5/OS テーブルおよびフラットファイルに対してバルクデータを移動できます。

z/OS システムでは、特に指定のない限り、PowerExchange で以下のタイプのデータベースおよびファイルをバルクデータのソースまたはターゲットとして使用できます。

- Adabas ファイル
- CA Datacom データベース（データソース専用）
- DB2 for z/OS テーブル
- CA IDMS データベース（データソース専用）
- IMS データベース
- VSAM データセット
- シーケンシャルデータセット

DB2 for z/OS では、PowerExchange で DB2 LOAD ユーティリティを使用してデータを DB2 テーブルにロードできます。IMS では、PowerExchange で IMS アンロードデータセットをソースとして使用することができます。

ソースデータがテープ上に存在する場合、PowerExchange では、32,760 バイトを超えるブロックサイズを持つデータセットを含めて、テープデータセットからデータを読み取ることができます。

制限: 別の記載がないかぎり、PowerExchange がバルクデータを移動できるソースレコードの最大長は 144 KB です。

バルクデータ移動のコンポーネント

PowerExchange は、以下のコンポーネントの一部または全部を使用してバルクデータを移動します。

PowerExchange Client for PowerCenter (PWXPC)

PowerExchange で制御されるデータに PowerCenter からアクセスして各種ターゲットに書き込みできるようにするため、PowerCenter と PowerExchange を統合する PowerCenter コンポーネントです。また、PowerExchange ODBC ドライバを使用することもできます。ただし、Informatica では ODBC ドライバではなく PWXPC の使用をお勧めします。PWXPC では追加された機能性、信頼性、およびパフォーマンスを用意しています。

PWXPC および ODBC ドライバの詳細については、『*PowerCenter 用の PowerExchange インタフェース*』を参照してください。

PowerExchange Navigator

非リレーショナルのデータソースとデータターゲットおよびユーザー定義のフィールドと式を必要とする DB2 テーブルのデータマップを定義、管理するためのグラフィカルなユーザーインタフェースです。PowerExchange は、この定義を使用して処理するデータソースおよびデータターゲットを判断します。また、PowerExchange Navigator では、オプションのパーソナルメタデータプロファイルを作成して、データベース行テストを実行できます。データマップで行テストを実行すると、ソースデータを表示して PowerExchange からアクセス可能であることを確認できます。パーソナルメタデータプロファイルで行テストを実行すると、データソースのメタデータを表示できます。

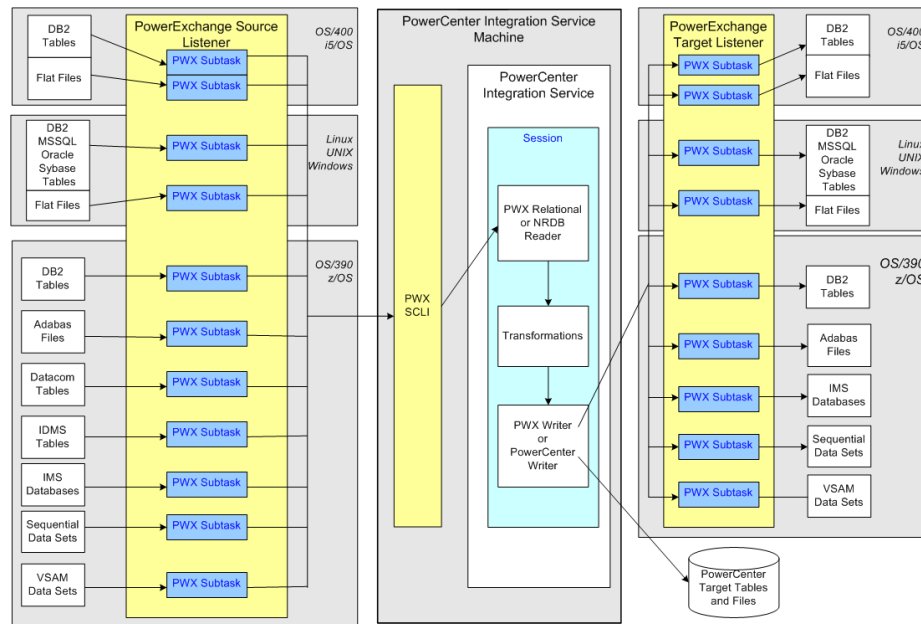
PowerExchange リスナ

非リレーショナルファイルおよび DB2 テーブルのデータマップを管理し、それを DATAMAPS ファイルに保持する PowerExchange コンポーネントです。また、PowerExchange リスナでは、PowerCenter からのバルクデータ抽出要求も処理します。

データソースまたはターゲットが使用中の PowerExchange からリモートのシステムに存在する場合は、そのリモートシステムで追加の PowerExchange リスナを実行して PowerExchange と通信する必要があります。

バルクデータ移動の構成例

以下の図に、PowerCenter を使用した PowerExchange のバルクデータ移動の構成例を示します。この例には、z/OS、i5/OS、および分散システム上に存在するソースおよびターゲットが含まれています。



この例では、PowerExchange で PWXPC を使用して PowerCenter と統合します。PowerExchange は、以下のタイプのデータベースおよびファイルとの間でバルクデータを抽出、ロードします。

- Linux、UNIX、および Windows 上のリレーショナルデータベースおよびフラットファイル
- i5/OS 上の DB2 テーブルおよびフラットファイル
- z/OS 上のリレーショナルデータベース、非リレーショナルデータベース、およびシーケンシャルデータセット

PowerExchange は、z/OS 上のリレーショナルデータベース、非リレーショナルデータベース、およびシーケンシャルデータセット、i5/OS 上の DB2 テーブルおよびフラットファイルとの間でバルクデータの読み込みおよび書き込みを行うために PowerExchange のアクセス方法を使用します。Linux、UNIX、および Windows システムでは、PowerCenter Reader および PowerCenter Writer が、リレーショナルデータソース、リレーショナルデータターゲット、およびフラットファイルのデータを抽出、ロードします。

注: また、PowerExchange は、Linux、UNIX、Windows 上にある DB2、Microsoft SQL Server、Oracle テーブルとデータの読み込みおよび書き込みを行うこともできます。

ターゲットシステムはソースシステムからリモートにあるため、PowerExchange リスナはこれらすべてのシステムに対して実行します。

バルクデータ移動のセキュリティ

PowerExchange ユーザーには、PowerExchange リスナに接続し、PowerExchange リソース、データソース、およびデータターゲットにアクセスできる適切なレベルの権限が必要です。

z/OS および i5/OS システムの場合、DBMOVER 構成メンバ内に SECURITY 文を含めて、PowerExchange リソース、データソース、およびデータターゲットにアクセスする際に、接続されたクライアントからユーザー ID とパスワードを要求することができます。

サポートされている Linux、UNIX、および Windows システムで、関連する文とともに DBMOVER 構成メンバに SECURITY 文を含めることによって、PowerExchange リスナまたは PowerExchange ロgger に接続しようとしているクライアントの資格情報を LDAP 認証で検証することができます。

z/OS では、データソースまたはターゲットのタイプに応じて、以下の追加レベルのセキュリティが使用可能です。

- DB2 for z/OS データの場合、DBMOVER 構成メンバ内に MVSDB2AF 文を含めて、呼び出し接続機能 (CAF) またはリカバリリソースサービス接続機能 (RRSAF) のいずれかを使用して DB2 に接続するかを制御することができます。DB2 認証方法を使用して DB2 テーブルへのユーザーアクセスを制御するには、RRSAF を選択します。PowerExchange リスナのユーザー ID を使用してユーザーアクセスを確認するには、CAF を選択します。
- Datacom データでは、RACF リソースプロファイルを定義して、Datacom テーブルへの不正な読み込みアクセスを防止できます。
- Adabas データでは、以下のどのセキュリティ方法でも使用できます。
 - ファイルレベルの Adabas パスワードセキュリティ
 - Adabas Security by Value 方法
 - Adabas データベースおよびファイルへの書き込みアクセスを制限する RACF リソースプロファイル

セキュリティの詳細については、『PowerExchange リファレンスマニュアル』を参照してください。

バルクデータ移動のタスクフロー

PowerExchange のインストール後、タスクを実行して、PowerExchange と PowerCenter を併用するバルクデータ移動を実装します。

次の表では、実行する上位レベルタスクを一覧表示し、タスクを実行するためのマニュアルへの参照を示します。

ステップ	タスク	参照
1	PowerExchange Listener を構成します。	第 2 章, 「PowerExchange Listener」 (ページ 16)
2	PowerExchange リスナを起動します。	第 2 章, 「PowerExchange Listener」 (ページ 16)
3	必要に応じてターゲットデータベース構造を作成します。	DBMS マニュアル

ステップ	タスク	参照
4	PowerExchange Navigator で、ユーザ定義のフィールドと式を必要とする非リレーショナルソースおよび DB2 ソースに対してデータマップを作成します。	<i>PowerExchange ナビゲータユーザガイド</i>
5	(データマップを使用している場合のオプション) PowerExchange Navigator で、データソースごとにデータベース行テストを実行して、PowerExchange からデータにアクセスし使用中のデータマップをテストできることを確認します。	<i>PowerExchange ナビゲータユーザガイド</i>
6	PowerCenter Designer で、マッピングを作成します。	<ul style="list-style-type: none"> - <i>PowerCenter 用の PowerExchange インタフェース</i> - <i>PowerCenter Designer ガイド</i>
7	PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、接続を構成した後、ワークフローを実行してバルクデータ移動を実行します。	<ul style="list-style-type: none"> - <i>PowerCenter 用の PowerExchange インタフェース</i> - <i>PowerCenter ワークフローベリックガイド</i>

注: DBMS タイプおよびバルクデータ移動方針によって、タスクフローが大幅に変わる可能性があります。たとえば、DB2 for z/OS の場合は、DB2 イメージコピーをデータソースとして使用したり、DB2 LOAD ユーティリティを使用してデータをターゲットにロードしたりできます。IMS および IDMS の場合は、Netport ジョブを使用できます。

第 2 章

PowerExchange Listener

この章では、以下の項目について説明します。

- [PowerExchange Listener の概要, 16 ページ](#)
- [バルクデータ移動のための PowerExchange Listener 構成タスク, 17 ページ](#)
- [PowerExchange DBMOVER ファイルのカスタマイズ, 17 ページ](#)
- [TCP/IP ポート番号の構成, 41 ページ](#)
- [複数のログファイルの構成, 42 ページ](#)
- [代替のコンフィギュレーションファイルまたはライセンスキーファイルの指定, 42 ページ](#)
- [z/OS での PowerExchange リスナ JCL の構成, 43 ページ](#)
- [CA TCPAccess 用の PowerExchange TCP/IP ソケットの構成, 43 ページ](#)
- [PowerExchange と PowerCenter との環境変数の非互換性, 44 ページ](#)
- [PowerExchange Listener の起動, 45 ページ](#)
- [PowerExchange Listener の管理, 47 ページ](#)

PowerExchange Listener の概要

PowerExchange Listener には、以下のサービスが備えられています。

- データソースのデータマップを格納および管理します。
- データを抽出およびロードするため PowerCenter でバルクデータを使用できるようにします。
- データベース行のテストを行うために PowerExchange Navigator でバルクデータを使用できるようにします。

PowerExchange Listener では、次の PowerExchange コンポーネントとのやり取りが行われます。

- PowerExchange Navigator
- 他の PowerExchange Listener

バルクデータ移動のための PowerExchange Listener 構成タスク

使用環境での必要性に応じて、以下の PowerExchange Listener 構成タスクの一部またはすべてを実行します。

- DBMOVER コンフィギュレーションファイル内に文を定義することによって、PowerExchange Listener を構成します。デフォルトの設定を受け入れることも、あるいは使用環境に合わせて構成文をカスタマイズすることもできます。
- デフォルトのポート番号である 2480 が使用不可能であるか、複数の PowerExchange Listener を使用し追加ポートを定義する必要がある場合は、TCP/IP ポート番号を構成します。
- メッセージおよびトレース情報用の代替ロギングを構成します。デフォルトでは、PowerExchange でログファイルが 1 つ提供されます。1 つのファイルにデータをロギングするのは効率的でなく、ログファイルが満杯になる可能性があるため、『*PowerExchange リファレンスマニュアル*』に記載されている PowerExchange の代替ロギング方法を使用します。代替ロギングを実装するには、DBMOVER コンフィギュレーションファイルで TRACING パラメータを指定する必要があります。

z/OS システムでは、PowerExchange と共に出荷されるデフォルトの DBMOVER コンフィギュレーションファイルに TRACING 文が含まれます。また、Netport ジョブなどの PowerExchange ジョブで提供される JCL にも、ログデータセットを割り当てるための以下の文が含まれます。

```
DTLLOG01 DD SYSOUT=*
```

通常、このデフォルトのロギング設定で十分です。ただし、他にもいくつかのロギング方法を使用できます。

- Linux、UNIX、または Windows では、dbmover.cfg および license.key ファイルの抽出コピーまたは変更済みコピーをデフォルト以外の場所に配置してください。このようにすると、新しいリリースにアップグレードした場合に、元のファイルが保持されます。代替ファイルを PowerExchange Listener に定義するには、PowerExchange Listener の起動時に、dtllst コマンドの CONFIG および LICENSE 文内に各ファイルのフルパスとファイル名を入力します。あるいは、PWX_CONFIG および PWX_LICENSE 環境変数で代替ファイルを指定することもできます。
- z/OS では、PowerExchange Listener JCL を構成します。
- z/OS で IBM z/OS TCP/IP ではなく CA TCPAccess を使用する場合は、TCPAccess を使用するように TCP/IP ソケットを設定します。

PowerExchange DBMOVER ファイルのカスタマイズ

PowerExchange の DBMOVER コンフィギュレーションファイルは、カスタマイズする必要がある可能性があります。コンフィギュレーションファイルには、PowerExchange リスナに影響を与える以下のパラメータが含まれます。

- APPBUFSIZE
- APPBUFSIZEDYN
- CMDNODE
- CREDENTIALS_CASE[CREDENTIALS_CASE]
- DATAMAP_SERVER
- DM_SUBTASK
- DMX_DIR

- LISTENER
- LOGPATH
- MAXTASKS
- NODE
- SECURITY
- SUBTIMEOUT
- SVCNODE
- TCPIVER
- TRACING

DBMOVER コンフィギュレーションファイルの編集後、変更を有効にするには PowerExchange リスナをリスタートします。

各 DBMOVER 文の説明については、『*PowerExchange リファレンスマニュアル*』を参照してください。

APPBUFSIZE 文

APPBUFSIZE 文は、データの読み取りまたは書き込みに使用する最大バッファサイズをバイト単位で指定します。

バッファサイズに達すると、PowerExchange はデータを必要とするシステムにバッファデータをネットワークを介して送信し、別のバッファの使用を開始します。

動的アプリケーションバッファのサイズ決定が有効な状態では、APPBUFSIZE 文で最大バッファサイズの初期値が指定されます。動的アプリケーションバッファのサイズ決定はデフォルトで有効になっています。この決定は、DBMOVER 構成ファイルの APPBUFSIZEDYN 文で「Y」を指定することによって明示的に有効にすることができます。

オペレーティングシステム: すべて

関連する文: APPBUFSIZEDYN

必須: 不要

構文:

APPBUFSIZE={*buffer_size*|256000}

値: *buffer_size* 変数には、単一行の最大サイズよりも大きな値を入力します。有効な値は、34816～8388608 です。デフォルトは 256000 です。

使用上の注意:

- reader または writer パーティションを使用するバルクデータ移動セッションの場合は、APPBUFSIZE 値を増やしてセッションのパフォーマンスを向上することができます。

APPBUFSIZEDYN 文

APPBUFSIZEDYN 文は、動的アプリケーションバッファのサイズ決定を有効にするかどうかを指定します。

DBMOVER APPBUFSIZE 文は、PowerExchange リスナの実行中に行われるすべての接続の、アプリケーションバッファの初期サイズを定義します。APPBUFSIZEDYN=Y の場合、必要に応じて個々の接続のアプリケーションバッファのサイズが PowerExchange によって変更されます。

APPBUFSIZEDYN 文は、固定長または可変長のレコードを含むデータソースへの PowerExchange 接続に適用されます。可変長レコードは、1 つ以上の可変長フィールドが含まれるレコードです。可変長フィールドのデータ型は、VARCHAR または VARBIN です。

可変長レコードを含むデータソースへの接続ごとに、大きすぎてバッファに収まりきれないレコードが発生した場合は、PowerExchange によってアプリケーションバッファのサイズが変更されます。PowerExchange によって、アプリケーションバッファのサイズは、オーバーフローしたレコードのサイズの 10 倍の値に増やされます（最大 8MB まで）。新しいサイズは、リスナが実行されている期間またはアプリケーションバッファのサイズが再度変更されるまで有効のまま維持されます。リスナの実行が開始された後で、接続のためのアプリケーションバッファのサイズが PowerExchange によって減らされることはありません。

固定長レコードを含むデータソースへの接続ごとに、接続が開かれる時点で PowerExchange によってレコード長が確認されます。この際、必要に応じて、最大 8MB のバッファサイズとなるように、アプリケーションバッファのサイズが一度変更されます。

オペレーティングシステム: すべて

データソース: すべて

関連する文: APPBUFSIZE

必須: いいえ

構文:

```
APPBUFSIZEDYN={N|Y}
```

有効な値:

- **N.** PowerExchange は、動的アプリケーションバッファのサイズ決定を行いません。
- **Y.** PowerExchange が、動的アプリケーションバッファのサイズ決定を行います。

デフォルトは Y です。

CMDNODE 文

CMDNODE 文は、pwxcmd コマンドのターゲットである PowerExchange プロセスの接続情報を指定します。

pwxcmd コマンドの発行元の Linux、UNIX、または Windows システム上にある dbmover.cfg ファイルに、CMDNODE 文を含めます。

オペレーティングシステム: Linux、UNIX、および Windows

関連した文: SVCNODE

必須: いいえ

構文:

```
CMDNODE=(service_name
        ,{CONDENSE|ECCR|LISTENER}
        ,host_name
        ,connect_port
)
```

パラメータ:

service_name

必須。pwxcmd コマンドを送る送信先の PowerExchange プロセスのコマンド処理サービスのユーザー定義名。この PowerExchange プロセスに pwxcmd コマンドを発行するときに、このサービス名を使用します。z/OS 上で PowerExchange ログベースの ECCR または Datacom テーブルベースの ECCR に接続する場合、この値は ECCR 名です。この値が、対応する SVCNODE 文で指定された ECCR 名に一致していることを確認してください。

{CONDENSE|ECCR|LISTENER}

必須。PowerExchange サービスタイプ。

次のオプションがあります。

- **CONDENSE**。PowerExchange Condense または PowerExchange ロgger (Linux、UNIX、Windows 用)。
- **ECCR**。z/OS 上の PowerExchange Adabas、IDMS、または IMS ログベースの ECCR または Datacom テーブルベースの ECCR。
- **LISTENER**。PowerExchange リスナ。

デフォルト値は指定されていません。

host_name

必須。コマンドを送る送信先のターゲットシステムのホスト名または IP アドレス。

connect_port

必須。PowerExchange プロセスのコマンド処理サービスが pwxcmd コマンドをリスンするポート番号。このポート番号は、対応する SVCNODE 文のポート番号と一致する必要があります。

CREDENTIALS_CASE 文

CREDENTIALS_CASE 文は、オペレーティングシステムのユーザー ID およびパスワードに PowerExchange で使用される文字の大文字/小文字の区別を制御します。

オペレーティングシステム: すべて

必須: いいえ

構文:

CREDENTIALS_CASE={A|D|S}

有効な値:

- **A**。z/または IBM i/OS では、PowerExchange は、以下のようにユーザー ID およびパスワードを処理し、オペレーティングシステムに渡して認証を行います。
 1. PowerExchange はユーザー ID を大文字に変換します。
 2. PowerExchange は、大文字と小文字が混在したパスワードを処理するようにオペレーティングシステムが設定されているかどうかを確認します。
 - 大文字と小文字が混在するパスワードを処理するようにオペレーティングシステムが設定されている場合、PowerExchange はパスワードの大文字と小文字をそのまま保持します。
 - 大文字と小文字が混在するパスワードを処理するようにオペレーティングシステムが設定されていない場合、PowerExchange はパスワードを大文字に変換します。z/OS では、パスワードが 8 文字より長い場合（これはパスフレーズであることを示します）、PowerExchange はパスワードに入力された大文字と小文字を保持します。
 3. PowerExchange は、ユーザー ID とパスワードをオペレーティングシステムに渡して認証を行います。

Linux、UNIX、または Windows では、PowerExchange は、入力時の大文字/小文字のままでユーザー ID とパスワードをオペレーティングシステムに渡して認証を行います。
- **D**。IBM i または z/OS では、PowerExchange は、ユーザー ID とパスワードを大文字に変換してから、オペレーティングシステムに渡して認証を行います。

Linux、UNIX、または Windows では、PowerExchange は、入力時の大文字/小文字のままでユーザー ID とパスワードをオペレーティングシステムに渡して認証を行います。

- **S.** IBM i または z/OS では、PowerExchange は、ユーザー ID を大文字に変換し、パスワードは入力時の大文字/小文字のままにします。続いて、ユーザー ID とパスワードをオペレーティングシステムに渡して認証を行います。

Linux、UNIX、または Windows では、PowerExchange は、入力時の大文字/小文字のままでユーザー ID とパスワードをオペレーティングシステムに渡して認証を行います。

デフォルトは D です。

DATAMAP_SERVER 文

DATAMAP_SERVER 文は、データマップを格納するファイルへのアクセス要求に対してサーバーとしての役割を果たす PowerExchange Listener のノード名を指定します。

この文は、複数の PowerExchange Listeners が稼動しておりデータマップファイルでオープンおよびクローズ要求を頻繁に行う環境でオーバーヘッドを削減するために使用します。

オペレーティングシステム: z/OS

関連した文: DM_SUBTASK および NODE

必須: No

構文:

DATAMAP_SERVER=*node_name*

値: *node_name* 変数には、DATAMAPS データセットにアクセスする PowerExchange Listener を示す DBMOVER メンバの NODE 文からのノード名を入力します。

一般に、データマップサーバーとして 1 つの PowerExchange Listener を指定します。指定するには、データマップサーバーに接続する PowerExchange Listener の DBMOVER メンバ内で、DATAMAP_SERVER 文を定義します。

使用上の注意:

- 同じ DATAMAPS データセットを共有する 2 つ以上の PowerExchange Listener がある場合、最初に開始した PowerExchange Listener をデータマップサーバーとして使用するように PowerExchange を設定できます。同じ DATAMAPS データセットを共有する 3 つ以上の PowerExchange Listener がある場合は、そのいずれかをデータマップサーバーとして指定する必要があります。

PowerExchange Listener をデータマップサーバーとして使用するには、PowerExchange Listener ごとに DBMOVER メンバ内で以下の文を設定します。

- DATAMAP_SERVER 文で、他の PowerExchange Listener のノード名を指定します。
- DM_SUBTASK 文に Y と指定します。

最初に開始した PowerExchange Listener がデータマップサーバーになり、他の PowerExchange Listener はそこからデータマップにアクセスします。PowerExchange は、PowerExchange メッセージログファイルに以下のメッセージを書き込みます。

- PWX-02804 メッセージ。書き込み先は、データマップサーバーとしての役割を果たす PowerExchange Listener の PowerExchange メッセージログファイル。
- PWX-02800 および PWX-02805 メッセージ。書き込み先は、データマップサーバーとしての役割を果たしていない PowerExchange Listener の PowerExchange メッセージログファイル。PWX-02805 メッセージは、データマップサーバーである PowerExchange Listener のノード名、IP アドレス、およびポート番号を示します。

- DM_SUBTASK=Y も指定し、PowerExchange Listener が DATAMAPS データセットに排他的ロックを保持する場合、DATAMAP_SERVER=*node_name* と入力して、IMS 同期 ECCR や Netport ジョブなどの他のタスクが、データセットにアクセスできるようにします。この設定を行わないと、タスクは失敗します。

あるいは、データマップキャッシュを使用します。パフォーマンスを高め、PowerExchange Listener データマップサーバーが停止した場合に起こる可能性のあるアクセス問題を防止するために、このアプローチをお勧めします。データマップキャッシュを実装するには、DM_SUBTASK=R および DMXCACHE_MAX_MEMORY_MB=20 と指定します。このように設定すると、PowerExchange Listener は読み取り専用モードでデータセットを開き、他のタスクはデータセットにアクセスできます。

DM_SUBTASK 文

DM_SUBTASK 文は、PowerExchange Listener が、データマップを格納したファイルにアクセスする方法を制御します。

この文は、複数の PowerExchange Listeners が稼動しておりデータマップファイルでオープンおよびクローズ要求を頻繁に行う環境でオーバーヘッドを削減するために使用します。

オペレーティングシステム: z/OS

関連した文: DATAMAP_SERVER、DM_RESOURCE、DMXCACHE_DELETEECSA、DMXCACHE_MAX_MEMORY_MB、DMXCACHE_MULTIPLEJOBS、RACF_CLASS、および SECURITY

必須: No

構文:

DM_SUBTASK={N|R|Y}

有効な値:

- N.** PowerExchange は、必要に応じて、PowerExchange Listener サブタスクで DATAMAPS データセットを開いたり閉じます。
- R.** データマップの読み取りパフォーマンスを高めるために、PowerExchange Listener は、DATAMAPS データセットを読み取りモードで開くサブタスクを開始します。データマップの挿入および削除で処理またはパフォーマンスの変更は起こりません。

R オプションは、N オプションよりも高速ですが、Y オプションより低速です。

- Y.** 単一の PowerExchange サブタスクが、DATAMAPS データセットを所有し開きます。他のサブタスクはすべて、データマップサブタスクを使用して、データマップにアクセスします。DATAMAPS データセットは、PowerExchange Listener がシャットダウンするまで開いたままです。

リソースの使用率を減らし、PowerExchange Listener のパフォーマンスを高めるには、Y と入力します。PowerExchange は、要求ごとにデータセットを開けて閉じるのではなく、PowerExchange Listener アドレス空間が有効な間に一度 DATAMAPS データセットを開きます。

Y と入力し、DATAMAP_SERVER 文も定義した場合、PowerExchange は PowerExchange Listener をデータマップサーバーに指定します。

デフォルトは N です。

使用上の注意:

- DM_SUBTASK=Y と指定し、さらに SECURITY 文の最初のパラメータに 2 と指定した場合、PowerExchange は、このリソースプロファイルをチェックして DATAMAPS データセットへのアクセスを許可するかどうかを判断します。この場合、セキュリティシステムでリソースプロファイルをコード化して、z/OS 上のデータマップを読み書きするすべてのユーザーにアクセスを許可する必要があります。

デフォルトで、PowerExchange は、FACILITY クラスの DTL.DATAMAP.DATASET リソースプロファイルを確認します。DM_RESOURCE 文および RACF_CLASS 文で他の値を指定した場合、プロファイル名とクラスは異なることがあります。

- DM_SUBTASK=Y と指定し、PowerExchange Listener が DATAMAPS データセットに排他的ロックを保持する場合も、DATAMAP_SERVER=node_name と入力して、IMS 同期 ECCR や Netport ジョブなどの他のタスクがデータセットにアクセスできるようにします。この設定を行わないと、タスクは失敗します。

あるいは、データマップキャッシュを使用します。パフォーマンスを高め、PowerExchange Listener データマップサーバーが停止した場合に起こる可能性のあるアクセス問題を防止するために、このアプローチをお勧めします。データマップキャッシュを実装するには、DM_SUBTASK=R および DMXCACHE_MAX_MEMORY_MB=20 と指定します。このように設定すると、PowerExchange Listener は読み取り専用モードでデータセットを開き、他のタスクはデータセットにアクセスできます。

DMX_DIR 文

DMX_DIR 文は、PowerExchange がデータマップを格納する場所を指定します。

オペレーティングシステム: すべて

必須: いいえ

構文:

DMX_DIR=directory

値: directory 値については、使用しているオペレーティングシステムに応じて、次のいずれかのタイプの値を入力します。

- IBM i (i5/OS) システムの場合は、データマップのライブラリ名を入力します。最大長は 10 文字です。
デフォルト値は STDATAMAPS です。
- Linux、UNIX、または Windows システムの場合は、データマップが含まれるディレクトリのパスおよび名前を入力します。最大長は 512 文字です。
Linux および UNIX の場合のデフォルト値は、/datamaps です。Windows の場合のデフォルト値は、\datamaps です。
Windows ネットワークパスの場合は、PowerExchange がネットワークパスを正しく解析できるように、先頭に 3 つのバックスラッシュ (\\\) を使用します。
- z/OS システムの場合は、DATAMAPS データセットを指す JCL の DD 文の名前を入力します。最大長は 8 文字です。
デフォルト値は DATAMAP です。

LISTENER 文

LISTENER 文は、指定された PowerExchange リスナプロセスが作業要求をリスンする TCP/IP ポートを定義します。

DBMOVER コンフィギュレーションファイルには最大で 10 個の LISTENER 文を定義できます。

z/OS の Netport ジョブの場合、一意のポートで LISTENER 文を定義し、そのポートを参照する NETPORT を定義します。

オプションで、SSL 認証と、TCP/IP バッファサイズおよび待機時間を制御する追加パラメータとを指定できます。

オペレーティングシステム: すべて

関連した文: Netport ジョブの場合は NETPORT、SSL 認証の場合は SSL

必須: いいえ

構文:

```
LISTENER=( { listener_node|node1}
            ,TCPIP
            , {port|2480}
            [, {send_bufsize|65536}]
            [, {receive_bufsize|65536}]
            [, {send_size|4096}]
            [, {receive_size|4096}]
            [,receive_timeout]
            [,ip_address]
            [,SSL]
            [,message_queue]
          )
```

パラメータ:

{listener_node|node1}

必須。PowerExchange リスナプロセスがリスンする TCP/IP ポートのノード名。

このノード名を使用して、以下のように、PowerExchange リスナプロセスの開始時に LISTENER 文を選択します。

- Linux、UNIX、および Windows では、dtllst コマンドでリスナノード名を指定します。
- i5/OS では、DTLLST プログラムを実行する SBMJOB コマンドでリスナノード名を指定します。詳細については、『PowerExchange コマンドリファレンス』を参照してください。
- z/OS では、JCL の EXEC カードの PARM フィールドでリスナノード名を指定します。

デフォルトは node1 です。

TCPIP

必須。通信プロトコル。TCPIP が唯一の有効なオプションです。

{port|2480}

必須。作業要求をリスンするために使用される TCP/IP ポート。有効な値は 1~65535 です。デフォルトは 2480 です。

{send_bufsize|65536}

オプション。TCP/IP 送信バッファのデータ部分のサイズ（バイト単位）。有効な値は 1024~1048576 です。デフォルトは 65536 です。

{receive_bufsize|65536}

オプション。TCP/IP 受信バッファのデータ部分のサイズ（バイト単位）。有効な値は 1024~1048576 です。デフォルトは 65536 です。

{send_size|4096}

オプション。PowerExchange が TCP/IP に一度に送信するデータのブロックの最大サイズ（バイト単位）。データがこのサイズを超えた場合、PowerExchange は、すべてのデータが送信されるまで、複数のブロックにデータを分割します。有効な値は 512~1048576 です。デフォルトは 4096 です。

ヒント: TCP/IP 送信バッファサイズ以下である値を入力します。

{receive_size|4096}

オプション。PowerExchange が TCP/IP から 1 つの操作で処理するデータのブロックの最大サイズ（バイト単位）。データがこのサイズを超えた場合、PowerExchange は、すべてのデータが受信されるまで、複数のブロックにデータを分割します。有効な値は 512~1048576 です。デフォルトは 4096 です。

ヒント: TCP/IP 受信バッファサイズ以上である値を入力します。

receive_timeout

オプション。長時間の待機が必要なときに、PowerExchange が受信タイムアウト値として使用する秒数。PowerExchange は、この要求に対してのみこの値を使用します。有効な値は 1～14400 です。

ip_address

オプション。ソケットのバインド操作で PowerExchange が使用する IP アドレス。

IP アドレスを指定しない場合、PowerExchange がバインド操作で INADDR_ANY を使用します。これにより、TCP/IP は、ホスト上のすべてのネットワークインタフェースにバインドします。複数のネットワークインタフェースがあり、PowerExchange リスナが特定のインタフェースをリスンするポートを制限する場合に、このパラメータを使用します。

SSL

オプション。PowerExchange が SSL 認証を使用することを指定します。SSL 通信中に SSL サーバとして機能する Linux、UNIX、または Windows のリスナマシンでこのパラメータを指定します。

IBM i で TLS ネットワーキングを使用するには、SSL を 10 番目のコンマの位置で指定します。TLS ネットワークを使用する場合は、LISTENER 文の 6 番目と 7 番目のコンマ位置の送信サイズと受信サイズを 16384 を超えない値に設定する必要があります。

message_queue

オプション。IBM i の LISTENER 文にメッセージキューを追加します。IBM i は、PowerExchange Listener ごとに異なるメッセージキューを指定することにより、同じ PowerExchange 環境から複数のリスナを実行できます。以下に例を示します。

```
/* Listener using default message queue (LISTENER)
LISTENER=(node1,TCP,2480,262144,262144,262144,262144)
/* Listener using user defined message queue (LISTENER2)
LISTENER=(node2,TCP,12480,262144,262144,262144,262144,,,LISTENER2)
```

値が指定されていない場合、既存のデフォルトのメッセージキューである LISTENER を使用します。

メッセージキューの作成については、『PowerExchange インストールおよびアップグレードガイド』を参照してください。

使用上の注意:

- PowerExchange リスナサービスを作成する場合、次のように LISTENER 文で定義するノード名を使用します。
 - Informatica Administrator から Listener サービスを作成する場合、**開始パラメータ**プロパティで指定するノード名値は、LISTENER 文で定義するノード名に一致する必要があります。
 - infacmd pwx CreateListenerService コマンドから Listener サービスを作成する場合、コマンドの -StartParameters オプションに指定するノード名値は、LISTENER 文で定義するノード名に一致する必要があります。
- リスナサービスを作成するときに、サービスマネージャは、ノード上の PowerExchange リスナプロセスにそのサービスを関連付けます。リスナサービスの設定および作成の詳細については、『Informatica アプリケーションサービスガイド』を参照してください。
- pwxcmd コマンドまたは infacmd pwx コマンドを PowerExchange リスナプロセスに発行する場合、LISTENER 文で定義するノード名は、DBMOVER 構成ファイルの SVCNODE 文で定義したノード名と一致する必要があります。この場合、LISTENER 文の *listener_node* パラメータに最大 12 文字の文字列を入力します。これは、SVCNODE 文で指定できるサービス名の最大長です。

LOGPATH 文

LOGPATH 文は、PowerExchange メッセージログファイルを格納する Linux、UNIX、または Windows システム上のディレクトリへのパスを指定します。

オペレーティングシステム: Linux、UNIX、および Windows

必須: いいえ

構文:

LOGPATH=*path*

値: メッセージログファイルを格納するディレクトリへのフルパスを入力します。

Windows ネットワークパスを指定する場合、PowerExchange でネットワークパスが正しく解析されるように、先頭にバックスラッシュ (\\) を 3 つ追加してください。以下に例を示します。

LOGPATH=\\host\Shared Folders\C\CDC_SHARED\msglogs

デフォルトの場所は現在の作業ディレクトリです。

使用上の注意:

- メッセージログファイルの検索が簡単になるように、現在の作業ディレクトリとは別のディレクトリを指定することをお勧めします。
- また、DETAIL_LOGPATH 環境変数に値を指定する場合、この環境変数は LOGPATH 文を上書きします。

MAXTASKS 文

MAXTASKS 文は、PowerExchange Listener で同時に実行できるタスクの最大数を定義します。

オペレーティングシステム: すべて

必須: No

構文:

MAXTASKS={*maximum_tasks*|5}

値: *maximum_tasks* 変数に、1~255 の数値を入力します。デフォルトは 5 です。

使用上の注意:

- PowerExchange Listener のタスクによるリソースの使用が使用可能なリソースを超えると、Listener は異常終了します。この場合は、MAXTASKS 値を減らして、同時実行タスク数を少なくすることができます。各タスクのリソース要件は異なりますが、この文は、PowerExchange Listener タスクが使用するリソースの量を制限し、このような異常終了を避けるのに役立ちます。
- MAXTASKS パラメータは、オペレーティングシステムのパフォーマンスやチューニングのパラメータとなることを目的としていません。複数の同時実行タスクのワークロードのバランスをとるには、オペレーティングシステムのワークロード管理機能、例えば z/OS の Workload Management (WLM) を使用します。
- 複数のスレッドと PowerExchange Listener サブタスクを発生させる PowerExchange チューニング機能（パイプラインのパーティション化やマルチスレッド処理など）を使用するときに、PowerExchange の処理速度が低下するかハングする場合、MAXTASKS 値を増やす必要がある場合があります。また、PowerCenter ワークフローを追加する場合も、この値を増やす必要がある場合があります。同時実行タスク数が MAXTASKS の制限に達すると、PowerExchange Listener は追加タスクの要求を拒否し、メッセージ PWX-00609 が表示されます。

```
PWX-00650 10.3.0.111:3720 : Listener 10.33.40.42 -> 10.3.0.1 on port 1234 socket 51
PWX-00609 Listener has temporarily stopped accepting connections.
```

同時実行タスク数 MAXTASKS 制限を下回ると、PowerExchange Listener は追加タスクの要求を受け付け始めます。

- 接続プールを使用する場合、MAXTASKS 値が、接続プールのサイズに対応できるだけ十分に大きいことを確認します。
- PowerExchange Listener がサポートできる同時タスクの最大数は、z/OS での仮想ストレージ制限などオペレーティングシステムのリソース制約のために、実質的には 255 未満になる場合があります。

NODE 文

NODE 文は、PowerExchange リスナプロセスに接続するために PowerExchange が使用する TCP/IP ホスト名およびポートを定義します。

DBMOVER 構成ファイルに最大 128 個の NODE 文を指定できます。

オプションで、SSL 認証と、TCP/IP バッファサイズおよび待機時間を制御する追加パラメータを指定します。さらに、必要に応じて、PowerExchange リスナサービスを特定する *service_name* パラメータを指定します。

オペレーティングシステム: すべて

関連した文: Netport ジョブの場合は NETPORT、SSL 認証の場合は SSL

必須: いいえ

構文:

```
NODE=({node_name|node1}
      ,TCPIP
      ,host_name
      ,port|2480}
      [, {send_bufsize|65536}]
      [, {receive_bufsize|65536}]
      [, {send_size|4096}]
      [, {receive_size|4096}]
      [, receive_timeout]
      [, {SSL|ZOSSL}]
      [, service_name]
      )
```

パラメータ:

node_name|node1

必須。この NODE 文に対するユーザー定義の一意の名前。名前は、PowerExchange リスナプロセスの名前に一致する必要はありません。文が指し示す PowerExchange リスナプロセスに接続するには、PowerExchange リスナの場所を指定するように求めるユーザーインターフェース（以下のインターフェースなど）にこの名前を入力します。

- Informatica Developer または PowerCenter Workflow Manager における接続定義内の **【場所】** 属性
- PowerCenter Designer におけるソース定義またはターゲット定義内の **【場所】** 属性
- [PowerExchange Navigator] ダイアログボックスの **【場所】** フィールド

デフォルトは node1 です。

TCPIP

必須。通信プロトコル。TCPIP が唯一の有効なオプションです。

host_name

必須。*port* パラメータで指定されたポートでリスンする PowerExchange リスナプロセスの TCP/IP ホスト名または IP アドレス。*service_name* パラメータを指定すると、*host_name* は無視されます。

`{port|2480}`

必須。*host_name* で指定されたシステム上で稼働している PowerExchange リスナがリスンする TCP/IP ポート。有効な値は 1～65535 です。デフォルトは 2480 です。

`{send_bufsize|65536}`

オプション。TCP/IP 送信バッファのデータ部分のサイズ（バイト単位）。有効な値は 1024～1048576 です。デフォルトは 65536 です。

`{receive_bufsize|65536}`

オプション。TCP/IP 受信バッファのデータ部分のサイズ（バイト単位）。有効な値は 1024～1048576 です。デフォルトは 65536 です。

`{send_size|4096}`

オプション。PowerExchange が TCP/IP に一度に送信するデータのブロックの最大サイズ（バイト単位）。データがこのサイズを超えた場合、PowerExchange は、すべてのデータを送信するまで、複数のブロックにデータを分割します。有効な値は 512～1048576 です。デフォルトは 4096 です。

ヒント: TCP/IP 送信バッファサイズ以下である値を入力します。

`{receive_size|4096}`

オプション。PowerExchange が TCP/IP から 1 つの操作で処理するデータのブロックの最大サイズ（バイト単位）。データがこのサイズを超えた場合、PowerExchange は、すべてのデータを受信するまで、複数のブロックにデータを分割します。有効な値は 512～1048576 です。デフォルトは 4096 です。

ヒント: TCP/IP 受信バッファサイズ以上である値を入力します。

receive_timeout

オプション。長時間の待機が必要なときに、PowerExchange が受信タイムアウト値として使用する秒数。PowerExchange は、この要求に対してのみこの値を使用します。有効な値は 1～14400 です。

`{SSL|ZOSSL}`

オプション。PowerExchange が SSL 認証を使用するように指定します。

z/OS システムと通信している Linux、UNIX、または Windows の SSL クライアントのみでの ZOSSL オプションを指定します。それ以外の場合は、Linux、UNIX、または Windows の SSL クライアントでの SSL オプションを指定します。

service_name

オプション。Informatica ドメイン内の PowerExchange リスナサービスの場所を見つけるように Informatica クライアントツールまたは Informatica 統合サービスを構成するには、*service_name* パラメータにそのリスナサービスの名前を指定します。

クライアントツールは、Developer ツールまたは PowerCenter Client です。統合サービスは、PowerCenter 統合サービスまたはデータ統合サービスです。

このパラメータを含めると、Informatica クライアントツールまたは統合サービスは、NODE 文の *host_name* パラメータを無視し、*service_name* パラメータと *port* パラメータを使用して Informatica ドメイン内のリスナサービスを見つけます。

PowerExchange リスナサービスの詳細については、『*Informatica アプリケーションサービスガイド*』を参照してください。

SECURITY 文

SECURITY 文は、PowerExchange ユーザーの認証およびリソースやコマンドへのアクセスを制御します。

DBMOVER コンフィギュレーションファイルで SECURITY 文を使用して、以下のタイプのセキュリティを設定します。

- PowerExchange にアクセスするためのユーザー認証
- z/OS および i5/OS 上での、PowerExchange ジョブおよびタスクによる、ファイルやデータセットへのアクセス
- Informatica ドメイン内の PowerExchange アプリケーションサービスに infacmd pwx コマンドを発行するためのユーザー認証
- PowerExchange プロセスに pwxcmd コマンドを発行するためのユーザー認証
- PowerExchange ナビゲータから PowerExchange リスナ LISTTASK および STOPTASK コマンドを発行するためのユーザー認証

オペレーティングシステム: すべて

関連する文: DM_RESOURCE、MVSD2AF、RACF_CLASS

必須: いいえ

構文:

```
SECURITY=({0|1|2}  
           ,{N|Y}  
           [,LDAP]  
           [,ORACLE_LDAP|OPEN_LDAP])
```

パラメータ: 最初の位置パラメータには、以下の有効な値があります。

{0|1|2}

PowerExchange がオペレーティングシステムの有効なユーザー ID とパスワードまたはパスフレーズの入力をユーザーに要求するかどうかを制御します。また、PowerExchange がユーザー入力資格情報をチェックして、ファイルおよびデータベースリソースへのアクセスおよび特定の PowerExchange コマンドの発行を制御するかどうかを制御します。

次のいずれかのオプションを入力します。

- **0.**PowerExchange は、オペレーティングシステムの有効なユーザー ID とパスワードを指定することをユーザーに要求せず、ユーザーが入力した資格情報を無視します。

z/OS および i5/OS では、PowerExchange は PowerExchange リスナまたは PowerExchange Condense タスクを実行するユーザー ID を使用してファイルリソースへのアクセスを制御します。PowerExchange はこのユーザー ID をデータベースシステムに渡します。

Linux、UNIX、および Windows では、PowerExchange は PowerExchange リスナタスクを実行するユーザー ID を使用してファイルリソースへのアクセスを制御します。RDBMS セキュリティはデータベースリソースへの PowerExchange アクセスを、ユーザーが PWX 接続時に、または PowerExchange ロgger CAPTURE_NODE_UID パラメータで指定したユーザー ID に基づいて制御します。

すべてのオペレーティングシステムで、PowerExchange はコマンドを発行するユーザー認証をチェックしません。すべてのユーザーがコマンドを発行できます。

- **1.**z/OS および i5/OS では、PowerExchange がユーザーにオペレーティングシステムの有効なユーザー ID とパスワードまたは PowerExchange パスフレーズを指定することを要求します。PowerExchange は、タスクの開始時にこれらの資格情報をチェックします。その後、PowerExchange はファイルリソースへのアクセスを、オプション 0 と同じ方法で制御します。ファ

イルアクセスについては、PowerExchange は PowerExchange リスナまたは PowerExchange Condense タスクを実行するユーザー ID を使用し、このユーザー ID をデータベースシステムに渡します。

Linux、UNIX、および Windows では、サポートされているシステムの SECURITY 文の 3 番目のパラメータに LDAP を指定しなければ、PowerExchange は、有効なオペレーティングシステムのユーザー ID とパスワードを指定してファイルまたはデータベースのリソースにアクセスするようにユーザーに要求しません。また、これらの資格情報があるかどうかをチェックしません。オプション 0 では、PowerExchange は PowerExchange リスナタスクを実行するユーザー ID を使用してファイルリソースへのアクセスを制御します。RDBMS セキュリティはデータベースリソースへの PowerExchange アクセスを、ユーザーが PWX 接続時に、または PowerExchange ロgger CAPTURE_NODE_UID パラメータで指定したユーザー ID に基づいて制御します。

すべてのオペレーティングシステムで、PowerExchange はコマンドを発行するユーザー認証をチェックしません。すべてのユーザーがコマンドを発行できます。

- **2.最も厳密なレベルのセキュリティを提供します。**

- z/OS では、オプション 2 を使用することをお勧めします。PowerExchange は、1) MVS ユーザー ID/パスワードまたは PowerExchange パスフレーズおよび 2) RACF や ACF2 などの z/OS セキュリティ製品のアクセス制御機能に基づいてアクセスを制御します。

変更ストリームから変更データを読み取るには、ECCR が z/OS の有効なユーザー ID とパスワードまたはパスフレーズを使用する必要があります。PowerExchange リスナは、ECCR タスクまたはジョブの開始時にこれらの資格情報をチェックします。データベースにアクセスしてデータを読み取るために、PowerExchange は、データベース固有のセキュリティチェックのために z/OS のユーザー ID とパスワードまたはパスフレーズをデータベースシステムに渡します。z/OS のセキュリティ製品と MVS システム許可機能 (SAF) を組み合わせて、PowerExchange は CAPX.REG.*リソースプロファイルと照合して z/OS のユーザー ID とパスワードまたはパスフレーズをチェックし、キャプチャ登録へのアクセスを制御します。

変更データを抽出するには、z/OS の有効なユーザー ID とパスワードまたはパスフレーズを指定する PWXPC 接続で PowerCenter CDC セッションを実行します。抽出マップにアクセスするセッションでは、これらのユーザークレデンシャルは PowerExchange リスナ JCL の DTLCAMAP DD 文で定義された PowerExchange データセットへの READ アクセスを持っている必要があります。

注: 呼び出し接続機能 (CAF) を使った DB2 for z/OS への接続は、セキュリティの設定に関係なく、PowerExchange リスナのユーザー ID を使って実行されます。DB2 は、接続タイプが Recoverable Resource Manager Service 接続機能 (RRSAF) である場合、またはオフロード処理が有効な場合にのみ、接続時に指定されたユーザー ID を使用します。

また、PowerExchange はリソースプロファイルを使用して、以下のタイプのコマンドを実行できるユーザーを制御します。

- Linux、UNIX、Windows システムから発行される PowerExchange リスナまたは PowerExchange Condense プロセス用の pwxcmd コマンド

- PowerExchange ナビゲータまたは DTLUTSK ユーティリティから発行される PowerExchange リスナ LISTTASK および STOPTASK コマンド

- i5/OS と z/OS の場合、PowerExchange はユーザーに対して、有効なオペレーティングシステムユーザー ID とパスワードまたはパスフレーズを指定するように要求します。PowerExchange は、タスクの開始時にこれらの資格情報をチェックします。PowerExchange リスナサブタスクプロセスは、指定されたユーザー ID とパスワードまたはパスフレーズで実行されます。PowerExchange はこのユーザー ID とパスワードまたはパスフレーズを使用して、PowerExchange ファイルへのアクセスを制御します。また、PowerExchange はこのユーザー ID とパスワードまたはパスフレーズをデータアクセスのためにデータベースシステムに渡します。

PowerExchange はセキュリティオブジェクトを使用して、以下のタイプのコマンドを実行できるユーザーを制御します。

- Linux、UNIX、Windows システムから発行される PowerExchange リスナまたは PowerExchange Condense プロセス用の pwxcmd コマンド
- SNDLSTCMD インタフェース、PowerExchange ナビゲータ、または DTLUTSK ユーティリティから発行される PowerExchange リスナ LISTTASK および STOPTASK コマンド
- Linux、UNIX、および Windows では、サポートされているシステムの SECURITY 文の 3 番目のパラメータに LDAP を指定しなければ、PowerExchange は、オペレーティングシステムのユーザー ID とパスワードを指定して PowerExchange のファイルまたはデータベースにアクセスするようにユーザーに要求しません。PowerExchange は、PowerExchange リスナを実行するユーザー ID およびパスワード、または PowerExchange ロgger（Linux、UNIX、Windows 用）が使用するユーザー ID およびパスワードを使用して、PowerExchange ファイルへのアクセスを制御します。RDBMS セキュリティは、データベースへのアクセスを制御します。

ただし、以下のタイプのコマンドを実行するには、オペレーティングシステムの有効なユーザー ID とパスワードを指定する必要があります。

- Informatica ドメイン内の PowerExchange アプリケーションサービスへの infacmd pwx コマンド
- PowerExchange プロセスへの pwxcmd コマンド

PowerExchange はサインオンファイルの USER および AUTHGROUP COMMANDS 文と照合してこれらのユーザークレデンシャルをチェックし、ユーザーが infacmd pwx または pwxcmd コマンドを発行する権限を付与されているかどうかを判断します。この場合、SECURITY 文の 2 番目の位置パラメータは無視されます。

デフォルトは 0 です。

2 番目の位置パラメータには、以下の有効な値があります。

{N|Y}

PowerExchange リスナへの接続をユーザーに許可するための、PowerExchange の選択的サインオンファイルの使用を制御します。

次のいずれかのオプションを入力します。

- **N**。PowerExchange は選択的サインオンファイルを使用しません。
- **Y**。PowerExchange は選択的サインオンファイルで ALLOW および IP サブパラメータを指定して USER 文を使用し、PowerExchange リスナに接続できるユーザーを制限します。

注: Y を指定して、SECURITY 文の最初のパラメータを 1 に設定した場合、PowerExchange はサインオンファイルの USER 文で TASKCNTRL パラメータを使用して、PowerExchange ナビゲータから発行された PowerExchange リスナ LISTTASK および STOPTASK コマンドへのアクセスを制御します。

デフォルトは N です。

オプションの 3 番目の位置パラメータには、以下の有効値があります。

LDAP

3 番目の位置パラメータに LDAP を指定し、最初の位置パラメータとして 1 または 2 を指定すると、PowerExchange はサポートされている Linux、UNIX、および Windows システムで LDAP 認証を使用します。

3 番目のパラメータを含めない場合、PowerExchange は LDAP 認証を使用しません。

4 番目の位置パラメータには、以下の有効な値があります。

{ORACLE_LDAP|OPEN_LDAP}

3 番目の位置パラメータに LDAP を指定する場合は、ロードする LDAP クライアントライブラリのセットを指定してください。

次のいずれかのオプションを入力します。

- **ORACLE_LDAP**。PowerExchange は Oracle LDAP クライアントライブラリをロードします。
このオプションは、Oracle LDAP がインストール済みの場合のみ選択してください。PowerExchange は、Oracle LDAP クライアントライブラリを提供していません。
- **OPEN_LDAP**。PowerExchange は OpenLDAP クライアントライブラリをロードします。

デフォルトは ORACLE_LDAP です。

使用上の注意:

- z/OS Installation Assistant の **【汎用パラメータ】** ページで **【詳細パラメータ】** をクリックすると、SECURITY_LEVEL および SECURITY_PWX パラメータを定義できます。SECURITY_LEVEL パラメータは、SECURITY 文の最初のパラメータに対応します。SECURITY_PWX パラメータは、SECURITY 文の 2 番目のパラメータに対応します。
- z/OS で、SECURITY 文の先頭のパラメータを 1 または 2 に設定した場合は、PowerExchange リスナおよび Netport ジョブについて、STEPLIB を APF 許可する必要があります。そうしないと、PowerExchange がユーザー認証またはリソースアクセスの制御を実行できず、このパラメータが 0 に設定されたかのように動作します。
- z/OS データソースのカラムレベルの処理を PowerCenter 統合サービスが実行されている Linux、UNIX、Windows システムにオフロードする場合、PowerCenter CDC セッションは接続時に指定した **【マップの場所のユーザー】** および **【マップの場所のパスワード】** の値を使用して、すべてのリソースへのアクセスを制御します。接続はオフロード処理が有効になっている PWX NRDB CDC アプリケーション接続または PWX DB2zOS CDC アプリケーション接続であることが必要です。
- z/OS データソースから取得したデータをリモートの PowerExchange ロgger (Linux、UNIX、Windows 用) ログファイルに記録する場合、z/OS 上の DBMOVER コンフィギュレーションメンバで SECURITY オプションを 2 に設定します。PowerExchange ロgger (Linux、UNIX、Windows 用) のコンフィギュレーションファイル (PWXCCL) のユーザー ID とパスワードが、z/OS セキュリティチェックに合格できる z/OS の有効なユーザー ID とパスワードであることを確認します。PowerExchange ロgger (z/OS 用) ログファイルにキャプチャされたデータを読み取るには、上記のユーザー資格情報に、FACILITY クラスの CAPX.REG.*リソースプロファイルに対する READ アクセス権が割り当てられている必要があります。これは、z/OS セキュリティ製品によって管理されます。また、CDC セッションでログファイルからデータを抽出するには、PWXP 接続時に **【マップの場所のユーザー】** および **【マップの場所のパスワード】** 接続属性で z/OS のユーザー ID とパスワードを指定する必要があります。これらのユーザークレデンシャルは、CAPX.CND.*リソースプロファイルへの READ アクセスを持っている必要があります。

SUBMITTIMEOUT 文

SUBMITTIMEOUT 文は、PowerExchange Listener が生成したバッチジョブから開始を知らせる通知を受信するまで待機する時間 (秒数) を指定します。

オペレーティングシステム: z/OS

関連する文: LOADJOBFILE および NETPORT

必須: 不要

構文:

SUBMITTIMEOUT={*timeout_seconds*|60}

値: *timeout_seconds* 変数には、1 から 86400 までの数値を入力します。デフォルトは 60 です。

使用上の注意:

- デフォルトでは、PowerExchange Listener は生成したジョブが開始するまで 60 秒間待ちます。この時間を超過すると、そのジョブはタイムアウトとなり、PowerExchange Listener でのタスクは停止され、PWX-00426 メッセージが PowerExchange メッセージログに書き込まれます。
- この文は、PowerExchange Listener が生成したすべてのバッチジョブに適用されます。それには以下のジョブも含まれます。
 - Netport ジョブ
 - DB2 LOAD ユーティリティジョブ
 - CA IDMS/DB メタデータジョブ
 - DTLREXE ユーティリティの PROG=SUBMIT オプションを使ってサブミットされたジョブ

SVCNODE 文

SVCNODE 文は、PowerExchange プロセスがコマンドでリスニングする TCP/IP ポートを指定します。プロセスタイプには、PowerExchange リスナ、PowerExchange Condense、PowerExchange ログベースまたはテーブルベースの ECCR、PowerExchange ロガー（Linux、UNIX、Windows 用）が含まれます。

以下のプログラムを使用して、PowerExchange プロセスに対してコマンドを発行できます。

- infacmd pwx プログラムを使用し、Linux、UNIX、または Windows 上のプロセスに対してコマンドを発行する。
- pwxcmd プログラムを使用し、i5/OS、Linux、UNIX、Windows、または z/OS 上のプロセスに対してコマンドを発行する。

DBMOVER 構成ファイルには最大で 30 個の SVCNODE 文を含めることができます。

オペレーティングシステム: すべて

関連する文: CMDNODE、LISTENER、NODE

必須: いいえ

構文:

```
SVCNODE=(service_name
          ,listen_port
)
```

パラメータ:

service_name

必須。PowerExchange のサービス名は最大 12 文字です。この名前は、以下のいずれかの値と一致しなければならない。

- PowerExchange Condense プロセスの場合、CAPTPARM コンフィギュレーションメンバまたはファイルの CONDENSENAME 文に指定したサービス名を使用します。
- z/OS 上の PowerExchange Adabas、IDMS、IMS ログベースの ECCR プロセスまたは Datacom テーブルベースの ECCR プロセスでは、DBMOVER 構成ファイルの CMDNODE 文で指定された ECCR 名を使用します。
- PowerExchange ロガー（Linux、UNIX、および Windows プロセス用）の場合、pwxccl.cfg ファイルの CONDENSENAME 文に指定したサービス名を使用します。
- PowerExchange リスナプロセスの場合、DBMOVER 構成ファイルの LISTENER 文にある *listener_node* パラメータに指定した名前を使用します。

リスナアプリケーションサービスを使用してリスナに接続するために infacmd pwx コマンドを発行するためには、この名前が次に示す値のいずれかに一致する必要があります。

- Informatica Administrator を使用してアプリケーションサービスを作成した場合は、**開始パラメータ**プロパティに指定したサービス名を使用します。
- infacmd pwx CreateListenerService コマンドを使用してアプリケーションサービスを作成した場合は、コマンドの -StartParameters オプションにある *service_name* パラメータに指定した名前を使用します。

ロガーアプリケーションサービスを使用して PowerExchange ロガー（Linux、UNIX、Windows 用）に接続するために infacmd pwx コマンドを発行する場合は、ロガーサービスが使用する pwxcl.cfg ファイルに正しい CONDENSENAME 値が含まれていることを確認してください。ロガーサービスの作成時には、必要に応じて次に示す方法でインストールディレクトリにあるデフォルトの pwxcl.cfg ファイルとは異なる構成ファイルを指定できます。

- Informatica Administrator を使用してアプリケーションサービスを作成する場合は、**開始パラメータ**プロパティに *cs* 値を指定します。
- infacmd pwx CreateLoggerService コマンドを使用してアプリケーションサービスを作成した場合は、コマンドの -StartParameters オプションに *cs* 値を指定します。

listen_port

必須。PowerExchange リスナ、PowerExchange ログベースまたはテーブルベースの ECCR、PowerExchange ロガー（Linux、UNIX、Windows 用）、または PowerExchange Condense がコマンドをリスンする一意のポート番号。

リスナアプリケーションサービスまたはロガーアプリケーションサービスを使用してリスナまたはロガーに接続するために infacmd pwx コマンドを発行するためには、SVCNODE 文に指定するポート番号が次に示す値のいずれかに一致する必要があります。

- Informatica Administrator を通じてアプリケーションサービスを作成した場合は、**SVCNODE ポート番号**プロパティで指定された値。
- infacmd pwx CreateListenerService コマンドまたは CreateLoggerService コマンドを使ってアプリケーションサービスを作成した場合は、そのコマンドの -SvcPort オプションで指定した値。

TCPIPVER 文

TCPIPVER 文は、PowerExchange が標準の IBM TCP/IP UNIX ソケットコードの代わりに使用する代替 TCP/IP ソケットコードを指定します。

デフォルトでは、PowerExchange は IBM TCP/IP UNIX ソケットを使用します。この文は、CA TCPAccess を使用する場合に定義します。

オペレーティングシステム: z/OS

必須: 不要

構文:

TCPIPVER={2|3}

有効な値:

- **2.** PowerExchange は、z/OS 上で IBM TCP/IP UNIX ソケットコードの代わりに CA TCPAccess ソケットコードを使用します。
- **3.** PowerExchange はネイティブ MVS ソケットコードを使用します。この値は、Informatica グローバルカスタマサポートからの指示に従って指定します。

TRACING 文

TRACING 文は、PowerExchange の代替ログを有効にし、代替ログファイルの属性を指定します。PowerExchange では、メッセージを格納するために、デフォルトの PowerExchange メッセージログファイルではなく、代替ログファイルが使用されます。

オペレーティングシステム: すべて

関連する文: LOGPATH

必須: いいえ

構文:

```
TRACING=(PFX=prefix
[ ,APPEND={N|Y}]
[ ,BUFFERS={number_of_buffers|100}]
[ ,FILENUM={number_of_files|5}]
[ ,FLUSH={flush_interval|99}]
[ ,RECLN={record_length|80}]
[ ,SIZE={log_size|100}]
[ ,VIEW={N|Y}]
)
```

パラメータ:

PFX=*prefix*

必須。代替ログファイル名のプレフィックスを指定します。

PowerExchange は、以下のシステム別のルールに従って代替ログファイル名を作成します。

i5/OS

PowerExchange は、PFX 値を使用して、PowerExchange データライブラリ内のログファイルのメンバ名を作成します。PowerExchange リスナ、PowerExchange Condense、または他の PowerExchange ジョブのうちどれがログファイルを作成するかによって、生成されるログファイル名は異なります。

- PowerExchange リスナは、次のファイル命名規則を使用します。
datalib/Plistener_port(prefixnnn)
- PowerExchange Condense とその他の PowerExchange ジョブは、次のファイル命名規則を使用します。
datalib/JOBJjob_number(prefixnnn)

これらの命名規則には、以下の変数が含まれます。

- *datalib* は、PowerExchange のインストール時に指定された PowerExchange データライブラリ名です。
- *listener_port* は、PowerExchange リスナのポート番号です。
- *job_number* は、PowerExchange Condense または他の PowerExchange ジョブの下で実行されるトレースサブタスク DTLTRTSK の i5/OS ジョブ番号です。
- *prefixnnn* は、PFX パラメータ値に 001~999 の連番を付加したものです。

例えば、リスナポート番号が 2480、PFX の値が PWXLOG、FILENUM の値が 3 の PowerExchange リスナは、以下のログファイルを作成します。

```
datalib/P02480(PWXLOG001)
datalib/P02480(PWXLOG002)
datalib/P02480(PWXLOG003)
```

PFX の値の最大長は 7 文字です。

Linux、UNIX、および Windows

PowerExchange は、PFX の値を、ログファイルを収めるサブディレクトリの名前として使用します。PowerExchange は、dbmover.cfg ファイル内の LOGPATH 文を使用して、このログサブディレクトリを収めるディレクトリを決定します。

PowerExchange リスナ、PowerExchange Condense、または他の PowerExchange タスクのうちどれがログファイルを作成するかによって、生成されるログファイル名は異なります。

- PowerExchange リスナは、以下のファイル命名規則を使用します。

Linux および UNIX:

logpath/prefix/DTLLST1.plistener_port.nnnn.log

Windows Listener サービス:

logpath\prefix\DTLLSTNT.plistener_port.nnnn.log

Windows Listener:

logpath\prefix\DTLLST1.plistener_port.nnnn.log

- PowerExchange ロgger（Linux、UNIX、Windows 用）は、以下のファイル命名規則を使用します。

Linux および UNIX:

logpath/prefix/PWXCL.tyyyyymmddhhmmss.ppid.nnnn.log

Windows:

logpath\prefix\PWXCL.tyyyyymmddhhmmss.ppid.nnnn.log

- その他のタスクの場合、PowerExchange は次のファイル命名規則を使用します。

Linux および UNIX:

logpath/prefix/module.tyyyyymmddhhmmss.ppid.nnnn.log

Windows:

logpath\prefix\module.tyyyyymmddhhmmss.ppid.nnnn.log

変数は、それぞれ以下の値を表します。

- *logpath* は、dbmover 構成ファイル内の LOGPATH 文の値です。
- *prefix* は、PFX パラメータの値です。
- *module* は、実行中の PowerExchange モジュールの名前です。例えば DTLURDMO ユーティリティの場合は DTLURDMO、PowerCenter 操作の場合は DTLODBCDRV です。
- *listener_port* は、PowerExchange リスナのポート番号です。
- *yyyyymmddhhmmss* は、ファイルが作成された時点のタイムスタンプです。
- *pid* は、PowerExchange タスクのプロセス ID です。
- *nnn* は、001 から 999 までの通し番号です。

例えば、ポート番号が 2480、PFX の値が PWXLOG、FILENUM の値が 3 の、UNIX 上で動作している PowerExchange リスナは、以下のログファイルを作成します。

logpath/PWXLOG/DTLLST1.p02480.n001.log
logpath/PWXLOG/DTLLST1.p02480.n002.log
logpath/PWXLOG/DTLLST1.p02480.n003.log

PFX の値の最大長は 210 文字です。

z/OS

PowerExchange は、PFX の値を、動的に割り当てられる代替ログデータセットの上位修飾子 (HLQ) として使用します。これらのデータセットはシーケンシャルデータセットです。別の方法として、PowerExchange タスクの JCL に DTLLOGnn DD 文を指定して代替ログデータセットを割り当てることもできます。デフォルトでは、PowerExchange は動的に割り当てられた代替ログデータセットを使用します。

拡張シーケンシャルデータセットを割り当てる DTLLOGnn DD 文を使用すると、PowerExchange は各追跡でメッセージを 1 つだけ書き込みます。DD 文が通常のシーケンシャルデータセットを割り当てる場合、PowerExchange は各データブロックに 1 つのメッセージを書き込みます。

注: 代替ログデータセットに対して DFSMS 圧縮を使用しないでください。

動的に割り当てられるログデータセットの場合、PowerExchange リスナバッチジョブ、他の PowerExchange バッチジョブ、開始済みタスクのうちどれがファイルを作成するかによって、生成されるデータセット名が異なります。

- PowerExchange リスナは、次のファイル命名規則を使用します。

prefix.sysid.listener_port.nnnn

- それ以外の PowerExchange バッチジョブおよび開始済みタスクはすべて、次のファイル命名規則を使用します。

prefix.job_name.job_number.sysid.nnnn

変数は、それぞれ以下の値を表します。

- *prefix* は、PFX パラメータに指定する上位修飾子です。プレフィックス全体の最大長は 16 文字です。
- *sysid* は、バッチジョブまたは開始済みタスクが実行されている z/OS システムのシステム ID です。
- *listener_port* は、PowerExchange リスナのポート番号です。
- *job_name* は、バッチジョブまたは開始済みタスクのジョブ名です。
- *job_number* は JES ジョブ番号です。バッチジョブの場合は JOB で始まり、開始済みタスクの場合は STC で始まります。
- *nnn* は、001 から 999 までの通し番号で生成されます。

例えば、ポート番号が 2480、PFX の値が PWXLOG、FILENUM の値が 3 の、MVS1 システム上で動作している PowerExchange リスナは、以下のログファイルを作成します。

```
PWXLOG.MVS1.P02480.N001
PWXLOG.MVS1.P02480.N002
PWXLOG.MVS1.P02480.N003
```

APPEND={N|Y}

オプション。メッセージを発行する PowerExchange コンポーネントの起動時に PowerExchange がメッセージログファイルをどのように使用するかを制御します。

次のオプションがあります。

- **N**。PowerExchange は、新規のログファイルを開くか、最も古いログファイルをログファイルとして上書きします。

例えば、動的に割り当てられた3つのログファイルを使用するために FILENUM=3 と設定している場合、PowerExchange リスナは起動時に、まずログファイル1を開こうと試み、次にログファイル2、3の順に試します。その後、PowerExchange は以下のアクションのどちらかを実行します。

- 動的に割り当てられる1つ以上のログファイルが存在しない場合、PowerExchange は存在しない最初のログファイルを初期ログファイルとして使用します。例えば、ログファイル1と2が存在していてログファイル3が存在しない場合、PowerExchange はログファイル3を初期ログファイルとして使用します。ログファイルが1つもない場合、PowerExchange はログファイル1を初期ログファイルとして使用します。
- 3つのログファイルがすべて存在する場合、PowerExchange は最も古いログファイルを初期ログファイルとして使用し、そのファイルを完全に上書きします。

注: z/OS 上で代替ログ用に GDG を使用する場合は、PowerExchange リスナが起動したときに新しい世代が作成されます。

- **Y.** ログファイルが存在している場合、PowerExchange は最新のログファイルを開き、その末尾にログメッセージを追加していきます。ログファイルが存在しない場合は、新規のログファイルが PowerExchange によって開かれます。

例えば、3つのログファイルを使用するために FILENUM=3 と設定している場合、PowerExchange リスナは起動時に、まずログファイル1を開こうと試み、次にログファイル2、3の順に試します。その後、PowerExchange は以下のアクションのどちらかを実行します。

- ログファイルが1つ以上存在する場合、PowerExchange は最新のログファイルを開き、その末尾にログメッセージを追加していきます。

z/OS 上で代替ログ用に GDG を使用していて、PowerExchange リスナ JCL の DTLLOGnn DD 文に GDG(0)を指定した場合は、現行世代の末尾にメッセージが追加されていきます。GDG(0)を使用しない場合、PowerExchange はこのパラメータを無視します。

- ログファイルが1つも存在しない場合、PowerExchange は新規のログファイル1を開き、そのファイルをログファイルとして使用します。

デフォルトはYです。

`BUFFERS={number_of_buffers}100`

オプション。PowerExchange サブタスクからのメッセージとトレース情報を受信するために PowerExchange が割り当てるバッファの数を指定します。バッファスペースがいっぱいの場合、メッセージとトレース情報を生成した PowerExchange サブタスクはバッファスペースが使用可能になるまで待機します。PowerExchange プログラムは、このバッファスペースを内部的に使用します。

有効な値は5~9999です。デフォルトは100です。

このパラメータは、Informatica グローバルカスタマサポートの指示の下でのみ指定します。

`FILENUM={number_of_files}5`

オプション。ログファイルが動的に割り当てられている場合に、PowerExchange が作成して使用する代替ログファイルの数を指定します。1つのログファイルがいっぱいになると、PowerExchange は最も古い代替ログファイルに切り替えて、そのファイルを上書きします。

有効な値は1~99です。デフォルトは5です。

注: z/OS 上では、代替ロギングに GDG を使用する場合、または DTLLOG メッセージを発行する PowerExchange コンポーネントに JCL で DTLLOGnn DD 文を指定する場合、FILENUM パラメータは無視されます。GDG については、JCL の DTLLOGnn DD 文で GDG(0)または GDG(+1)のどちらを指定したかに関係なく、パラメータは無視されます。

FLUSH={flush_interval}99}

オプション。PowerExchange がログレコードをいくつ収集したらディスク上のログファイルにフラッシュするかを指定します。PowerExchange は、空き領域不足から回復できるようにするために、ログレコードを定期的にフラッシュする必要があります。フラッシュの値を小さく設定すると、ログファイルに対する I/O アクティビティが増加します。

有効な値は 1～99 です。デフォルトは 99 です。

RECLEN={record_length}80}

オプション。PowerExchange がログレコードをログファイルに書き込むために使用するレコード長を指定します。メッセージの長さがこのレコード長を超えた場合、ログレコードは PowerExchange によって複数行にわたって書き込まれます。

有効な値は 80～255 です。デフォルトは 80 です。

注: RECLEN パラメータを指定せず、LOG_LINE_LIMIT 文に 80 を超える値を入力すると、PowerExchange はその LOG_LINE_LIMIT 値を RECLEN 値として使用します。

SIZE={log_size}100}

オプション。代替ログファイルに PowerExchange が書き込むログデータのおおよその量を MB 単位で指定します。書き込んだログデータの量がこの値に達すると、PowerExchange は現在のログファイルを閉じて次のログファイルを開き、ログレコードの書き込みを続行します。

有効な値は 1～2048 です。デフォルトは 100 です。

注: z/OS では、手動で割り当てたデータセットが SIZE 値よりも大きい場合は、PowerExchange により、データセットに書き込むログデータ量を SIZE 値に制限されます。データセットが SIZE 値より小さい場合は、そのデータセットサイズが、PowerExchange が書き込むことができるログデータ量の上限になります。空き領域不足が発生する場合、PowerExchange は手動で割り当てられている次のメッセージデータセットに切り替えます。

VIEW={N|Y}

オプション。FLUSH 間隔が終了すると、PowerExchange が現在のログファイルを定期的に閉じて再度開くかどうかを制御します。このパラメータはすべてのオペレーティングシステムで指定できますが、最も役に立つのは z/OS の場合です。z/OS では、ログデータセットが閉じられるまで、代替メッセージログレコードを表示できません。z/OS 以外のオペレーティングシステムでは、PowerExchange が FLUSH 間隔に基づいてログレコードをディスクにフラッシュした後に、ログレコードを表示できます。現在のログファイルは、ログレコードを表示するために閉じる必要はありません。

次のオプションがあります。

- **N。**PowerExchange が現在のログファイルを定期的に閉じてまた開くことはありません。
- **Y。**PowerExchange は現在のログファイルを定期的に閉じてまた開きます。

ヒント: z/OS では、VIEW=Y に指定して、代替ログデータセットを定期的に閉じて再度開くようにして、ログレコードを表示できるようにすることをお勧めします。

z/OS で VIEW=Y を指定した場合、次の考慮事項が適用されます。

- z/OS で代替ログ用に GDG を使用する場合は、PowerExchange リスナ JCL の単一の DTLLOGnnDD 文で GDG(0)と DISP=SHR を指定します。また、PowerExchange リスナを起動する前に、GDG データセットの少なくとも 1 つの世代を作成して割り当てる必要があります。
- データセットの割り当てを操作するサードパーティの手順を使用する場合は、それらの製品が VIEW=Y 処理を妨げる可能性があります。例えば、その製品が SYSDSN ENQ を EXCLUSIVE モードに変更することにより、データセットの表示が妨げられる場合があります。

- データセットを開く操作と閉じる操作が頻繁に要求されることにより、データを代替ログデータセットに書き込む PowerExchange ジョブのパフォーマンスが低下する可能性があります。パフォーマンスの低下を最小限に抑えるには、FLUSH パラメータでデフォルト値の 99 を使用します。

デフォルトは N です。

使用上の注意:

- 例えば継続モードで実行される PowerExchange ロgger (Linux、UNIX、Windows 用) プロセスのように長期間にわたって実行されるジョブについてログに記録されるデータの量をカスタマイズし、ロギングのパフォーマンスを向上させるために、代替ログを使用します。
- 動的代替ログが有効になっている場合、PowerExchange プロセスごとに別々のディレクトリの中に代替ログファイルのセットが作成されます。

場所、ログファイルの数、およびログファイルのサイズ (MB 単位) を指定できます。使用中のログファイルが指定されたサイズに達すると、PowerExchange は次のログファイルに切り替えて、そのファイルの中のデータを上書きし始めます。

- TRACING 文を定義した場合は、LOGPATH 文も定義して、Linux、UNIX、または Windows システム上で代替ログファイルが保存されるディレクトリを指定します。
- PowerExchange ジョブまたは開始済みタスクに JCL で DTLLOGnn DD 文を定義した場合を除き、PowerExchange は代替ログデータセットを動的に割り当てます。
- z/OS では、メッセージを発行するすべての PowerExchange ジョブまたは開始済みタスクに対して、JCL で使用する DTLLOG01 DD 文で SYSOUT=* を指定することをお勧めします。以下に例を示します。

```
//DTLLOG01 DD SYSOUT=*
```

このストラテジでは、単一の SYSOUT オプションが指定された 1 つの DTLLOG01 DD 文のみを定義するため、設定が簡単になります。また、このストラテジにより、特定のジョブ実行またはタスク実行のメッセージ出力の検索が容易になります。PowerExchange がすべてのメッセージ出力を、他のジョブ出力で利用できる単一の SYSOUT データセットに書き込むためです。

- z/OS で代替ログ用に GDG を使用する場合は、PowerExchange リスナ JCL の DTLLOGnn DD 文で GDG(0) を指定します。以下に例を示します。

```
DTLLOG01 DD DSN=USER1.V901.TRCGDG(0),DISP=SHR
```

GDG(0)を使用することにより、APPEND=Y を使用して現行世代へのメッセージの書き込みを再開することができます。また、VIEW=Y を使用して、PowerExchange リスナタスクがアクティブな間に GDG 内のログレコードを表示することもできます。APPEND=N を指定すると、PowerExchange リスナが起動したときに新しい世代が作成されるようになります。

代わりに GDG(+1)を使用する場合は、TRACING 文の APPEND、および VIEW パラメータが無視され、PowerExchange リスナが起動するたびに新しい世代が作成されるようになります。

また、GDG を使用している場合、PowerExchange リスナを起動する前に、GDG の少なくとも 1 つの世代を作成して割り当てる必要があります。

- z/OS では、代替ログ用に GDG を使用する場合、または PowerExchange ジョブまたは開始済みタスクに対して JCL で DTLLOG01 DD 文を指定する場合、FILENUM パラメータは無視されます。

GDG の例:

z/OS で GDG の現行世代にメッセージを追加し (GDG(0))、定期的にメッセージを表示できるようにするには、以下の手順を実行します。

1. 以下のような JCL 文を含むバッチジョブを実行して、少なくとも 1 つの世代別データセットを GDG 内に割り当てて作成します。

```
//DJEGDG@ JOB (ACCOUNT),'GDG',NOTIFY=&SYSUID
//JSTEP01 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
```

```

DEFINE GDG-
(NAME(USER1.V901.TRCGDG)-
  LIMIT(4)-
  NOEMPTY-
  SCRATCH)
//STEP2 EXEC PGM=IEFBR14
//DLLOG01 DD DSN=USER1.V901.TRCGDG(+1),DISP=(NEW,CATLG),
           DCB=(BLKSIZE=32718,LRECL=132,RECFM=VB),
           SPACE=(CYL,(1,1))

```

2. DBMOVER コンフィギュレーションメンバで、VIEW=Y および APPEND=Y を含む TRACING 文を定義します。以下に例を示します。

```
TRACING=(PFX=PWX,APPEND=Y,VIEW=Y)
```

この文は、GDG の現行世代の末尾にメッセージを追加し、メッセージを表示できるようにするために定期的に世代別データセットを閉じてまた開くように、PowerExchange を設定します。

3. GDG の現行の世代別データセットを参照するように、PowerExchange リスナ JCL 内の DTLLOG nn DD 文を設定します。以下に例を示します。

```
DTLLOG01 DD DSN=USER1.V901.TRCGDG(0),DISP=SHR
```

この文の中で DISP=SHR を使用します。

4. PowerExchange リスナを起動します。

PowerExchange が、現行の世代別データセットの末尾にメッセージを追加し始めます。

現行世代が存在しない場合は JCL エラーが発生し、PowerExchange リスナは起動しません。

TCP/IP ポート番号の構成

PowerExchange のインストールの際、PowerExchange Listener のデフォルトの TCP/IP ポート番号は、DBMOVER コンフィギュレーションファイルで 2480 に設定されます。

```
LISTENER=(node_name,TCPIP,2480)
```

TCP/IP ポート 2480 は、PowerExchange 専用に使われる Internet Assigned Numbers Authority (IANA) に登録されます。このポートがサイトで PowerExchange 用に使用可能かどうかを確認するようネットワーク管理者に依頼してください。

別の製品または PowerExchange の以前のインストールでポート 2480 を使用している場合は、別の有効な TCP/IP ポート番号をネットワーク管理者から取得する必要があります。そのポート番号を DBMOVER コンフィギュレーションファイルの LISTENER 文で入力します。

複数の PowerExchange Listener に複数のポートを必要とする場合は、PowerExchange で使用する他の有効なポート番号を取得するようネットワーク管理者に依頼してください。また、ポートが他の製品で使用するよう登録されているかどうかについても、IANA Web サイトで確認します。

ヒント: IANA 情報は一般ガイドとして使用してください。IANA に登録されていないポートが他の製品で使われている可能性があります。

ポート番号を識別したら、DBMOVER コンフィギュレーションファイルでポート番号ごとに LISTENER または NODE 文を追加します。NODE 文は、PowerExchange がデータを読み取る元となるリモートプラットフォーム上の PowerExchange Listener に対してのみ適用されます。以下の例に示すように、それぞれの文に、ソースプラットフォームの IP アドレスまたはホスト名を含めます。

```
NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)
```

HOSTENT ユーティリティを使用して、ソースの IP アドレスまたはホスト名を判断できます。また、このユーティリティを使用して、IP アドレスが PowerExchange ライセンスキーファイル内のものと一致していることを確認できます。

複数のログファイルの構成

デフォルトでは、PowerExchange は、ログメッセージおよびトレース情報を 1 つのログファイルに書き込みます。1 つのファイルからはエラーメッセージを簡単に検索できますが、ログファイルが満杯になる可能性があります。PowerExchange Listener などの長時間実行するジョブが存在する場合は、その可能性が特に高くなります。空き領域不足を解決するには、通常、PowerExchange を停止する必要があります。

ログファイルの空き領域不足から生じる障害を回避し、ロギングをより効率的にするため、PowerExchange の代替ロギングを使用できます。代替ロギングを実装するには、DBMOVER コンフィギュレーションファイルに TRACING 文を含める必要があります。この文を使用すると、生成された数値サフィックスを持つ再利用可能な一連のログファイルに、PowerExchange がメッセージおよびトレース情報を書き込むことができますようになります。再利用可能なログファイルが満杯になると、PowerExchange は、数値サフィックスに基づいたシーケンスで次のログファイルに情報の書き込みを開始します。

z/OS システムでは、デフォルトの DBMOVER 構成メンバに PowerExchange の TRACING 文が含まれます。また、PowerExchange ジョブのデフォルト JCL にも、ログデータセットを割り当てるための以下の文が含まれます。

```
DTLLOG01 DD SYSOUT=*
```

このデフォルトのロギング設定が十分ではない場合は、z/OS で使用できる複数のロギング方法のいずれかを使用できます。

複数のログファイルを構成する手順

- ▶ DBMOVER コンフィギュレーションファイルに、TRACING 文を入力します。以下の構文を使用します。
TRACING=(PFX=*prefix*, [APPEND=Y|N,] [BUFFERS=*nnn*,] [FILENUM=*nnn*,] [FLUSH=*nn*,] [RECLN=*nnn*,] [SIZE=*nnnn*,] [VIEW=Y|N])
この構文には、使用可能なすべてのパラメータが含まれます。使用するパラメータは、ロギングとトレースの要件およびシステムによって異なります。

代替のコンフィギュレーションファイルまたはライセンスキーファイルの指定

i5/OS、Linux、UNIX、または Windows システムで PowerExchange Listener を起動するとき、CONFIG および LICENSE 文を起動コマンドに含めることができます。これらの文は、デフォルトの DBMOVER コンフィギュレーションファイルと license.key ファイルではなく、PowerExchange Listener の実行で使用する代替のコンフィギュレーションファイルとライセンスキーファイルを指定します。通常、代替ファイルを作成するには、デフォルトのファイルを別の場所または異なるファイル名でコピーしてから、そのコピーをカスタマイズします。このようにすると、PowerExchange をアップグレードまたは再インストールしても、カスタマイズしたコンフィギュレーションファイルおよびライセンスキーファイルが上書きされることはありません。

Linux、UNIX、または Windows システムでは、代替のコンフィギュレーションファイルおよびライセンスキーファイルを PWX_CONFIG および PWX_LICENSE 環境変数で指定することもできます。この指定を行うと、現行の PowerExchange Listener の実行については、CONFIG および LICENSE 文で指定される代替ファイルによって、環境変数で指定されている代替ファイルがオーバーライドされます。

z/OS での PowerExchange リスナ JCL の構成

PowerExchange には、RUNLIB ライブラリの以下のメンバ内に PowerExchange リスナ JCL が用意されています。

- PWXLSTNR は、PowerExchange を開始されたタスクとして実行します。
- STARTLST は、PowerExchange をバッチジョブとして実行します。

注: インストール中に XIZZZ998 ジョブを実行した場合、PWXLSTNR メンバが PROCLIB ライブラリにコピーされます。

PowerExchange リスナを初めて起動する場合は、起動前に PowerExchange リスナ JCL を確認します。

z/OS で PowerExchange リスナ JCL を構成する手順

1. PowerExchange リスナ JCL の STEPLIB DD 連結にソース DBMS のロードライブラリが含まれていることを確認します。
STARTLST メンバの STEPLIB 連結には、すべてのデータソースのロードライブラリが適切な順序で含まれます。
2. DBMOVER 構成ファイルの LISTENER 文で定義されたノード名が PARM で指定されていることを確認します。
3. EXEC 文の REGION パラメータに、領域サイズを入力します。領域サイズは、PowerExchange リスナを実行するために割り当てられるストレージ量を示します。
サンプル JCL では、EXEC 文で 384 MB の領域サイズを指定しています。
0M の領域サイズをお勧めします。これは、z/OS により 16 MB より上および下の使用可能なすべてのストレージにジョブが提供されることを示します。0M の領域サイズの指定に関する重要な考慮事項については、『z/OS MVS JCL Reference』を参照してください。
REGION パラメータを省略すると、システムデフォルトが適用されます。
4. EXEC 文の TIME パラメータが NOLIMIT に設定されていることを確認します。NOLIMIT に設定すると、PowerExchange リスナジョブが時間制限なしで実行できるため、異常終了コード S322 で発生する可能性のある時間関連の異常終了を防ぐことができます。
5. DBMOVER コンフィギュレーションファイル内の SECURITY 文が 1 または 2 に設定されている場合は、PowerExchange の LOADLIB ライブラリおよび PowerExchange リスナ JCL の STEPLIB DD 連結内の他のロードライブラリを APF 許可します。

CA TCPAccess 用の PowerExchange TCP/IP ソケットの構成

デフォルトでは、PowerExchange によって IBM TCP/IP ソフトウェアが使用されます。ただし、z/OS システムでは、必要に応じて、CA TCPAccess ソフトウェアを使用するように PowerExchange を構成できます。PowerExchange は、バージョン 5.3 以降の TCPAccess をサポートしています。

注: TCPAccess をバージョン 5.3 から 6.0 にアップグレードする場合も、PowerExchange 構成を変更する必要はありません。

CA TCPAccess 用の PowerExchange TCP/IP ソケットを構成する手順

- PowerExchange Listener JCL を次のように編集します。
 - TCPAccess の LOAD および LINK ライブラリを、STEPLIB DD 連結に追加します。
 - 以下の DD 文を追加します。

```
//SYSTCPD DD DISP=SHR,hlq.TCPIP.DATA
```
- TCPAccess の LOAD および LINK ライブラリを、PowerExchange RUNLIB ライブラリの次のいずれかのメンバに追加します。
 - PowerExchange Listener をバッチジョブとして実行する場合は STARTLST
 - PowerExchange Listener を開始されたタスクとして実行する場合は PWXLSTNR
- TCPAccess の LOAD および LINK ライブラリを、Netport ジョブと通信する任意の PowerExchange ジョブの STEPLIB DD 連結に追加します。
- 以下の文を、RUNLIB ライブラリ内の DBMOVER 構成メンバに追加します。
 TCP/IPVER=2

PowerExchange と PowerCenter との環境変数の非互換性

PowerCenter[®]と PowerExchange が同じ Linux、UNIX、または Windows マシンにインストールされている場合、場合によっては、PATH や LD_LIBRARY_PATH 環境変数の要件が競合することがあります。これらの場合に正しく実行するには、PowerExchange と PowerCenter は別々の環境で実行する必要があります。

この要件は、PowerCenter 統合サービスまたは PowerCenter リポジトリサービスが、以下の PowerExchange コンポーネントのいずれかと同じマシンで実行されている場合に適用されます。

- PowerExchange リスナ
- PowerExchange ロgger (Linux、UNIX、Windows 用)
- PowerExchange Navigator
- createdatamaps ユーティリティ以外の任意の PowerExchange ユーティリティ

以下の表に、PowerExchange 環境および PowerCenter 環境において PATH 変数および LD_LIBRARY_PATH 変数に適用される制限を示します。

環境	PATH	LD_LIBRARY_PATH
PowerExchange	\$INFA_HOME が \$PWX_HOME より先行してはいけません。それ以外の場合は、PowerExchange リスナまたはロggerをコマンドラインから開始することはできません。	LD_LIBRARY_PATH に PowerCenter のエントリが含まれてはいけません。この要件は、PowerExchange ユーティリティはライブラリを必ず \$PWX_HOME からのみピックアップすることを保証します。
PowerCenter	\$PWX_HOME エントリが \$INFA_HOME エントリより先行してはいけません。	\$LD_LIBRARY_PATH 変数定義は必ず \$INFA_HOME と \$PWX_HOME の両方を含み、\$INFA_HOME が \$PWX_HOME より先行である必要があります。以下に例を示します。 \$INFA_HOME/server/bin:\$PWX_HOME:\$LD_LIBRARY_PATH

PowerExchange インスタンスまたは PowerCenter インスタンスの正しい環境を同じマシンに設定するには、以下のいずれかのストラテジを使用します。

- PowerExchange と PowerCenter は常に別のアカウントで起動し、各アカウントに適切な環境変数を設定します。
- PowerExchange コンポーネントを開始するたびに、`pwxssettask.sh` または `pwxssettask.bat` スクリプトを実行します。

PowerExchange Listener の起動

PowerExchange Listener が動作するプラットフォームに有効な起動コマンドを使用します。

PowerExchange リスナの起動 (i5/OS)

i5/OS では、SBMJOB コマンドを使用して、PowerExchange リスナの DTLLST プログラムを呼び出します。

PowerExchange リスナを開始する前に、次の前提条件が満たされていることを確認します。

- QMLTTHDACN システム値が 1 または 2 に設定されている。QMLTTHDACN システム値の詳細については、i5/OS に関する IBM インフォメーションセンターを参照してください。
- JOBD の記述に複数スレッドを許可する `ALWMLTTHD(*YES)` パラメータが含まれている。
- システムジョブテーブルが最大サイズを超えないようにするために、JOBD の記述に `SPLFACN(*DETACH)` パラメータが含まれている。

以下のコマンド構文を使用します。

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

この構文には、以下の変数が含まれます。

- *dtllib* は、インストール時に入力された PowerExchange ソフトウェアライブラリの名前。
- *node_name* は、*datalib*/CFG(DBMOVER) コンフィギュレーションメンバの LISTENER 文で指定される PowerExchange リスナのノード名。
- *job_name* は、PowerExchange リスナのジョブまたは開始タスクの名前です。
- *datalib* は、インストール時に入力された PowerExchange データライブラリのユーザー指定の名前。

SBMJOB コマンドは、コマンドラインから入力できます。

あるいは、自動化されたスケジューラ、CL プログラム、または REXX プロシージャを使用して SBJOB コマンドを実行することもできます。例えば、SBMJOB コマンドを STARTLST という名前の REXX メンバに含めてから、次の文を使用して PowerExchange リスナを開始します。

```
STRREXPRC SRCMBR(STARTLST) SRCFILE(datalib/REXX)
```

注: PowerExchange リスナを開始するために `pwxcmd` プログラムを使用することはできません。

PowerExchange リスナの起動 (Linux および UNIX)

PowerExchange リスナの dtllst プログラムをコマンドプロンプトから実行します。

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

元の dbmover.cfg ファイルと license.key ファイルをオーバーライドするコンフィギュレーションファイルとライセンスキーファイルを指定する場合は、オプションの config パラメータと license パラメータを含めません。

最後にアンパサンド (&) を付けることにより、dtllst をバックグラウンドモードで実行できます。また、プレフィックス nohup を追加し、永続的に dtllst を実行することも可能です。

または、PowerExchange と共に出荷された startlst スクリプトを使用します。startlst スクリプトは、detail.log ファイルを削除して PowerExchange リスナを起動します。

注意: 同じユーザーアカウントを使用して PowerExchange と PowerCenter を同じマシンで実行する場合は、PowerExchange と PowerCenter に個別の環境を作成する必要があります。適切な PowerExchange 環境を作成して PowerExchange リスナを起動するには、pwxsettask.sh スクリプトを実行します。

以下の構文を使用します。

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

引用符はオプションです。

詳細については、[「PowerExchange と PowerCenter との環境変数の非互換性」 \(ページ 44\)](#)を参照してください。

PowerExchange リスナの起動 (z/OS)

PowerExchange リスナを開始タスクとして実行している場合は、標準の MVS START (S) コマンドを発行します。

```
S task_name
```

PowerExchange リスナをバッチジョブとして実行している場合は、RUNLIB ライブラリの STARTLST メンバで JCL をサブミットします。

PowerExchange リスナの起動 (Windows)

Windows システムで PowerExchange リスナを起動するには、以下に示す方法のどれかを使用します。

- 次のいずれかのアクションを実行することで、PowerExchange リスナを Windows サービスとして実行します。

- Windows の [スタート] メニューで、[スタート] > [プログラム] > [Informatica PowerExchange] > [PowerExchange リスナの起動] の順にクリックします。

- dtllstsi プログラムを使用して、Windows のコマンドプロンプトで起動コマンドを入力します。

```
dtllstsi start "service_name"
```

- dtllst を入力します。

構文は、&オペランドと nohup オペランドがサポートされていない以外は、Linux と UNIX の構文と同じです。dtllst を手動で実行するには、製品ライセンスが必要です。

Informatica ドメインで PowerExchange リスナをアプリケーションサービスとして実行する場合、Informatica Administrator ツールから PowerExchange リスナサービスを有効にして起動します。詳細については、『Informatica アプリケーションサービスガイド』を参照してください。

注意: 同じユーザーアカウントを使用して PowerExchange と PowerCenter を同じマシンで実行する場合は、PowerExchange と PowerCenter に個別の環境を作成する必要があります。適切な PowerExchange 環境を作成して PowerExchange リスナを起動するには、pwxsettask.bat スクリプトを実行します。

以下の構文を使用します。

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

引用符は必須です。

詳細については、[「PowerExchange と PowerCenter との環境変数の非互換性」 \(ページ 44\)](#)を参照してください。

PowerExchange Listener の管理

PowerExchange Listener コマンドを使用して、PowerExchange Listener の処理を制御できます。ここでは、必要となる可能性の高いコマンドを示します。

PowerExchange Listener の停止

PowerExchange Listener を停止するには、PowerExchange Listener を実行しているシステム上で CLOSE または CLOSE FORCE コマンドを実行します。

CLOSE および CLOSE FORCE コマンドは、コマンドラインから発行できます。あるいは、Linux、UNIX、または Windows システムから、システムで実行中の PowerExchange Listener まで pwxcmd close または closeforce コマンドを発行することもできます。

CLOSE または pwxcmd close コマンドは、以下のサブタスクが完了した後、PowerExchange Listener を停止します。

- バルクデータ移動サブタスク
- CDC サブタスク (Unit of Work (UOW) の次のコミットで停止)
- PowerExchange Listener のサブタスク

CLOSE FORCE または pwxcmd closeforce コマンドは、すべてのユーザーサブタスクを強制的にキャンセルした後、PowerExchange Listener を停止します。このオプションは、PowerExchange Listener 上で長時間実行するジョブが存在する場合に便利です。FORCE パラメータを指定すると、PowerExchange は以下の処理を実行します。

1. PowerExchange Listener のアクティブなサブタスクの有無をチェックします。
2. アクティブなサブタスクが存在する場合は、アクティブなサブタスクの数を毎秒ポーリングします。
3. この待機中に、TCP/IP ネットワーク入力を待機しているサブタスクを終了します。
4. 30 秒が経過した後で、残っているアクティブなサブタスクをキャンセルします。
5. PowerExchange Listener を停止します。

注: MVS の CANCEL コマンドは使用しないでください。PowerExchange Listener をキャンセルすると、TCP/IP でクリーンアップが完了するまでそのポートを使用できなくなります。TCP/IP がこのクリーンアップを完了する前に別の PowerExchange Listener ジョブをサブミットすると、PowerExchange によってエラーメッセージが発行されます。アクティブな PowerExchange Navigator セッションが PowerExchange Listener を使用しているときに z/OS で PowerExchange Listener を終了すると、MVS CANCEL コマンドの場合と同じエラーメッセージが PowerExchange から発行されます。この場合は、PowerExchange Navigator を終了し、z/OS で PowerExchange Listener をリスタートします。

コマンドの構文は、オペレーティングシステムと、それらの発行方法によって異なります。

リモートの PowerExchange Listener のテスト

リモート PowerExchange Listener の接続および状態をテストするには、**ping** プログラムで DTLREXE ユーティリティを実行します。リモート PowerExchange Listener の NODE 文を、DBMOVER コンフィギュレーションファイルで構成します。

ユーティリティを実行するための構文は、オペレーティングシステムによって異なります。

Linux、UNIX、および z/OS では、以下の一般的な構文を使用します。

```
dtlrexec loc=node_name prog=ping [uid=user_id] [pwd=password]
```

i5/OS では、以下の構文を使用します。

```
CALL PGM(DTLREXE) PARM('prog=ping loc=node_name [uid=user_id] [pwd=password]')
```

z/OS または i5/OS で SECURITY 文の最初のパラメータを 1 または 2 に設定して PowerExchange Listener を実行している場合は、ユーザー ID とパスワードを含める必要があります。

UNIX の場合、パスワードにドル記号 (\$) が含まれていると、DTLREXE ではドル記号 (\$) までのパスワード部分しか取得されません。この問題を解決するには、エスケープ文字としての役割を果たす一重引用符でパスワードを囲みます。

```
'pwd=yourpwd'
```

DTLREXE から PowerExchange Listener に正常に接続された場合、PowerExchange から以下のメッセージが発行されます。

```
PWX-00750 DTLREXE Input LOC=node_name, PROG=PING, PARMS=none, UID=user_id.  
PWX-00650 Listener 127.127.127 -> 127.127.127 on port 1234 socket 4  
PWX-00591 Tasks now active = 1.  
PWX-00753 DTLREXEL Input received LOC=node_name, PROG=PING, PARMS=none.  
PWX-00754 DTLREXEL User Return Codes 1=0, 2=0.  
PWX-00755 DTLREXE Command OK!
```

PowerExchange リスナの制御

PowerExchange リスナのコマンドを使用し、アクティブなタスクをすべて表示したり、特定のタスクを停止したりできます。

アクティブな PowerExchange リスナタスクのリストには、次の情報が含まれます。

- TCP/IP アドレス
- ポート番号
- アプリケーション名
- アクセスタイプ
- ステータス

次の表に、すべてのアクティブ PowerExchange リスナタスクをリスト表示するコマンドと、それらのコマンドを実行できるシステムの種類を示します。

コマンド	システム
DISPLAY ACTIVE	- i5/OS - z/OS - Linux、UNIX、および Windows
LISTTASK	- z/OS - Windows 上の PowerExchange Navigator から、任意のシステム上の PowerExchange リスナまで
pwxcmd listtask {-service -sv} service	Linux、UNIX、または Windows システムから、任意のシステムの PowerExchange リスナまで

次の表に、PowerExchange リスナタスクを停止するコマンドと、それらのコマンドを実行できるシステムの種類を示します。

コマンド	システム
STOPTASK	<ul style="list-style-type: none">- すべて- Windows 上の PowerExchange Navigator から、任意のシステム上の PowerExchange リスナまで
<code>pwxcmd stoptask {-service -sv} service</code>	Linux、UNIX、または Window システムから、任意のシステムの PowerExchange リスナまで

詳細については、『*PowerExchange コマンドリファレンス*』を参照してください。

第 3 章

Adabas バルクデータ移動

この章では、以下の項目について説明します。

- [Adabas バルクデータ移動の概要, 50 ページ](#)
- [Adabas バルクデータ移動に関する考慮事項, 50 ページ](#)
- [Adabas バルクデータ移動の構成, 51 ページ](#)
- [Adabas バルクデータの移動, 60 ページ](#)

Adabas バルクデータ移動の概要

PowerExchange では、PowerCenter と連携して、Adabas データベースとの間でバルクデータを移動できます。

Adabas はリレーショナルデータベースではないため、データマップを作成する必要があります。PowerExchange は、そのデータマップを基に Adabas データおよびメタデータにアクセスして、リレーショナル行タイプのレコードビューを作成します。PowerExchange では、SQL 型の文を使用してバルクデータを読み書きするのにリレーショナルビューが必要になります。

Adabas ソースデータベースを PowerExchange Navigator、PowerCenter Client、および PowerCenter Integration Service システムからリモートにあるシステムで実行する場合は、別の PowerExchange Listener をリモートシステム上で実行する必要があります。例えば、Adabas を z/OS で実行する場合は、z/OS 上に PowerExchange Listener をインストールおよび設定する必要があります。また、PowerExchange と PowerCenter がリモートの PowerExchange Listener と通信できることも検証してください。

Adabas バルクデータ移動に関する考慮事項

Adabas ソースまたはターゲットにバルクデータ移動を実装する前に、以下の考慮事項を確認します。

- PowerExchange では、デフォルトの長さが 1,024 バイトである Long Alpha (LA) フィールドをインポートします。データマップを編集することによって、PowerExchange Navigator からこのデフォルトの長さをオーバーライドすることができます。Adabas ファイルの **【レコード】** ウィンドウを開いて、LA フィールドの **【フィールドプロパティ】** ダイアログボックスを開きます。**【長さ】** フィールドに、最大 16,381 のオーバーライド値を入力することができます。
- Adabas ソースまたはターゲットの Adabas フィールド定義テーブル (FDT) がパスワードで保護されている場合、PowerCenter のバッチセッション中に Adabas ソースまたはターゲットに接続するには、Adabas FDT パスワードが必要です。Adabas FDT パスワードを入力するには、Task Developer でセッションを編

集します。【ソース】または【ターゲット】の【タスクの編集】ダイアログボックスの【マッピング】タブで、Adabas ソースまたはターゲット定義をクリックします。【プロパティ】の右ペインで、【Adabas パスワード】属性に FDT パスワードを入力します。

- z/OS で Adabas 8.2.2 以降を使用する場合、PowerExchange は、スパンレコードを含む Adabas ファイルをデータソースとしてサポートしています。スパンレコードは、単一の物理プライマリレコードと、最大 4 個のセカンドセカンダリレコードで構成される論理レコードです。各レコードは、個別のデータストレージブロックに保存されます。
- Adabas 記述子フィールドに NULL の抑制 (NU)属性が設定されている場合、フィールドにアクセスする PowerCenter ワークフローが予期しない結果を返すことがあります。NU 属性を持つ記述子フィールドはインデックスから除外されます。ワークフローがキーフィールド値に基づいて書き込みを実行しようとする、SQLCODE=256 のエラーコードまたは他のエラー状態が返される場合があります。

Adabas バルクデータ移動の構成

バルクデータ移動操作用に PowerExchange を構成するには、PowerExchange 管理者が以下のタスクを実行します。

- Adabas z/OS システム上の PowerExchange リスナへの接続を設定し、テストします。
- DBMOVER 構成メンバに Adabas 固有の文を定義することにより、Adabas バルクデータ移動を構成し、Adabas の読み取り処理を最適化します。
- Adabas LOAD ライブラリへアクセスする PowerExchange リスナを設定します。
- z/OS システム上の PowerExchange リスナで、次の Adabas ロードモジュールの再利用率可能なバージョンが使用されていることを確認します。ADARUN、ADAIOR、ADAIOS、ADAMLF、ADAPRF、および ADALNK。
- 必要に応じて、ADARUN 制御文のデフォルトの SVC パラメータ値をオーバーライドします。
- 必要に応じて、暗号コードで暗号化された Adabas データベースを読み込むように PowerExchange を設定します。

リモートの Adabas ソースまたはターゲットへの接続の設定およびテスト

リモート z/OS システム上の Adabas データソースまたはターゲットにアクセスするには、PowerExchange でリモートシステム上の PowerExchange Listener と通信する必要があります。

リモートの Adabas ソースまたはターゲットへの接続を設定およびテストする手順

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用して Adabas ソースまたはターゲットがあるリモートシステムへのネットワークアクセスをテストします。
2. 各 PowerExchange および PowerCenter システム上の dbmover.cfg ファイルに以下の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを定義します。
`NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)`
ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。
3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER 構成メンバへの Adabas 固有の文の追加

UNIX、Windows、または z/OS システム上にある DBMOVER コンフィギュレーションファイルに、以下の Adabas 固有の文を含めることにより、Adabas バルクデータ移動を設定し、Adabas の読み取り処理を最適化します。

- ADA_L3_ALLOW
- ADABAS_DEFAULT_DBID
- ADABAS_MU_SEARCH
- ADABAS_PREFIX
- ADABASCODEPAGE
- ADAOPT
- ADAPREFETCH
- ADASTATS
- START_UP_USER_EXIT

DBMOVER コンフィギュレーションファイルの編集後、PowerExchange リスナを再起動して変更内容を適用する必要があります。

各 DBMOVER 文の説明については、『*PowerExchange リファレンスマニュアル*』を参照してください。

ADA_L3_ALLOW 文

ADA_L3_ALLOW 文は、PowerExchange が Adabas L3 コマンドを使用して、記述子の値による論理的シーケンスでレコードをファイルから読み取るかどうかを制御する Adabas 最適化文です。

オペレーティングシステム: z/OS

データソース: Adabas

関連した文: ADAOPT および ADASTATS

必須: いいえ

構文:

ADA_L3_ALLOW={N|Y}

有効な値:

- **N**。PowerExchange は、稼働している Adabas バージョンが開始および終了範囲をサポートしているかどうかを判断しようと試みます。この Adabas バージョンが範囲をサポートしていると判断すると、PowerExchange は L3 コマンドを使用します。それ以外の場合は、L2 コマンドを使用します。
- **Y**。SQL SELECT 文の WHERE 句で Adabas 記述子を指定すると、PowerExchange は L3 コマンドを使用して、Adabas ファイルから論理的順序でレコードを読み取ります。SQL 文に WHERE 句が含まれていない場合は、PowerExchange は L2 コマンドを使用して、Adabas が保存した物理的な順序でファイルからレコードを読み取ります。

使用中の Adabas バージョンが L3 コマンドでの開始および終了記述子キー範囲をサポートしているかどうかは確認しません。

PowerExchange が稼働中の Adabas バージョンを正しく判別できない場合は、Y と指定します。

デフォルトは N です。

使用上の注意:

- L3 コマンドを使用するには、ADAOPT 文に Y と指定する必要があります。
- ADAOPT 文とは異なり、ADA_L3_ALLOW 文では、PowerExchange は Adabas バージョン 7 以降が稼働しているかどうか確認しません。ADA_L3_ALLOW 文は、ユーザーが Adabas に変更を加えた結果、インストールされている Adabas バージョンを PowerExchange が確認できない場合に使用します。
- PowerExchange ではデフォルトで、データマップの【最適化レベル】リストで【範囲のみ】が選択されています。【最適化レベル】リストで【OFF】を選択した場合、PowerExchange は、ADA_L3_ALLOW 文を無視し、そのデータマップのデータを処理するときに L3 コマンドを使用しません。
- ADASTATS 文で Y と指定すると、PowerExchange は PowerExchange メッセージログファイルにメッセージ PWX-02196 を書き込みます。このメッセージは、L3 コマンドと最適化のステータスを使用できるかどうかを示します。

ADABAS_DEFAULT_DBID 文

ADABAS_DEFAULT_DBID 文は、データマップで【データベース ID】プロパティが 0 に指定され、バルクデータ移動セッションでオーバーライド値が指定されていない場合に、PowerExchange で使用される DBID 値を指定します。

オペレーティングシステム: z/OS

データソース: Adabas

必須: いいえ

構文:

ADABAS_DEFAULT_DBID={*dbid*|0}

値: *dbid* 変数には、0～65535 の数値を入力します。デフォルトは 0 です。

使用上の注意:

- DBID 値には先頭のゼロを含められます。例えば、デフォルトの DBID として 100 を指定する場合、以下のどの文でも定義できます。
 - ADABAS_DEFAULT_DBID=100
 - ADABAS_DEFAULT_DBID=0100
 - ADABAS_DEFAULT_DBID=00100
- ある PowerExchange 環境から別の環境に簡単にデータマップを移行するには、データマップの【データベース ID】プロパティに 0 を入力します。続いて、各環境の ADABAS_DEFAULT_DBID 文を定義して DBID 値を指定します。あるいは、データマップで指定された DBID 値を上書きするには、PowerCenter のバルクデータ移動セッションの【データベース ID の上書き】属性に値を設定するか、ODBC を使用している場合は ODBC パラメータに値を設定します。

ADABAS_MU_SEARCH 文

ADABAS_MU_SEARCH 文は、Adabas 複数値 (MU) フィールドが検索関数に含まれているどうかを指定します。MU フィールドは、レコード内の出現回数変動する 1 つのフィールドです。MU フィールド名は、先頭に \$ 記号が付き、検索で許可されています。MU フィールドは、Adabas 記述子フィールドにする必要があります。これは、Adabas レコードの記述でタイプ MU として表示されます。

オペレーティングシステム: z/OS

データソース: Adabas

必須: いいえ

構文:

ADABAS_MU_SEARCH={Y|N}

有効な値:

- **Y**。PowerExchange は検索で Adabas 記述子フィールドを使用します。SQL SELECT 文の WHERE 句で、複数値フィールドを指定すると、PowerExchange はこのフィールドをスキャンし、指定された値を検索します。
- **N**。PowerExchange は、Adabas 複数値フィールドを検索に使用しません。

デフォルトは N です。

使用上の注意:

- 既存のデータマップ定義で、\$接頭辞を複数値フィールドに追加して、添字インデックスを削除すれば、複数値記述子 (DE) フィールドのフィールド名を使用できます。
- 複数値フィールドを含める検索条件を、EQUAL コンパレータまたは BETWEEN 句に制限します。OR コンパレータや非記述子検索値を使用する複雑な検索はサポートされていません。次の SELECT 文は、複数値フィールドの簡単な検索条件を示しています。

```
SELECT * from table_name where $MU_field = 'JONES'
SELECT * from table_name where $MU_field >= 10 and $MU_field <= 20
SELECT * from table_name where $MU_field is between 10 and 20
SELECT AA_field, AB_field, $MU_field where $MU_field = 10
```
- 複数値フィールドが検索条件で使用されている場合は、SQL SELECT 文でもこのフィールドを使用できます。SQL SELECT 文は、検索条件を満たす複数値配列データの内容を表示します。
- ADABAS_MU_SEARCH 文を Y に設定する場合は、必ず ADAOPT 文も Y に設定し、L3 コマンドを使用して、論理シーケンスで記述子値に基づいてレコードをファイルから読み取れるようにしてください。検索を最適化できない場合、また SQL で OR 条件が存在する場合、検索は失敗します。

ADABAS_PREFIX 文

ADABAS_PREFIX 文は、Adabas ファイルにアクセスするユーザー ID の作成に PowerExchange で使用されるプレフィックスを指定します。

オペレーティングシステム: z/OS

データソース: Adabas

関連した文: ADAUSER (Netport ジョブ用)

必須: いいえ

構文:

ADABAS_PREFIX={*prefix*|DTL0}

値: *prefix* 変数には、1~4 文字のアルファベットのプレフィックスを入力します。ユーザー ID を作成するため、PowerExchange によってさらに 4 文字が加えられ、PowerExchange リスナが開く Adabas ファイルごとに一意の値が生成されます。デフォルト値は DTL0 です。

使用上の注意:

- 別々のユーザー ID で同時に同じ Adabas ファイルにアクセスするには、一意のユーザー ID を指定します。Adabas ファイルにアクセスするために PowerExchange が作成する各ユーザー ID が一意でない場合、1 つ以上の PowerExchange リスナが、Adabas ファイルにアクセスできなくなる可能性があります。この場合、読み取り操作が失敗し、Adabas 応答コード 48 サブコード 8 と、PowerExchange メッセージ PWX-00416 が返されます。

- 複数の PowerExchange リスナが同じ Adabas ファイルにアクセスするときに、一意の Adabas ユーザー ID が使用されるようにするには、ADABAS_PREFIX 文を使用して、PowerExchange リスナごとに異なるプレフィックスを指定します。ユーザー ID が一意でないと、Adabas ファイルへのアクセスを試みる最初の PowerExchange リスナは成功しますが、2 番目の PowerExchange リスナは失敗します。また、PowerCenter セッションも失敗します。
- Netport ジョブを使用して Adabas ファイルにアクセスする場合、Netport ジョブごとに一意のユーザー ID を確保するように、ADAUSER 文を定義します。

ADABASCODEPAGE 文

ADABASCODEPAGE 文は、Adabas データベースに使用するシングルバイトおよび複数バイトのコードページを指定します。

DBMOVER コンフィギュレーションファイルに、最大 20 個の ADABASCODEPAGE 文を入力します。

オペレーティングシステム: z/OS

データソース: Adabas

関連した文: CODEPAGE

必須: いいえ

構文:

```
ADABASCODEPAGE=( dbid
                  [,single_cp]
                  [,multi_cp])
```

パラメータ:

dbid

必須。1 つまたは複数のコードページが適用される Adabas データベース識別子 (DBID)。

Adabas のデフォルトデータベースを示す 0 を指定します。

single_cp

オプション。シングルバイトコードページの名前。

multi_cp

オプション。マルチバイトコードページの名前。データベースに WIDECHAR フィールドが含まれる場合、マルチバイトコードページを入力します。

使用上の注意:

- 通常、デフォルトのコードページや CODEPAGE 文で設定されたコードページ以外のコードページを必要とする WIDECHAR フィールドを持つデータベースにこの文を指定します。

データマップを定義するときに、この文で指定するコードページを上書きできます。データマップで、特定のソースファイルのコードページ、すべての WIDECHAR フィールドのワイド文字コードページ、または各フィールドの特定のコードページを指定します。フィールドレベルのコードページは、データマップコードページを上書きし、データマップコードページは、この文または CODEPAGE 文で指定するすべてのコードページを上書きします。

ADAOPT 文

ADAOPT 文は、PowerExchange が Adabas L3 コマンドを使用して、記述子の値による論理的シーケンスでレコードをファイルから読み取るかどうかを制御する Adabas 最適化文です。

オペレーティングシステム: z/OS

データソース: Adabas

関連した文: ADA_L3_ALLOW、および ADASTATS

必須: いいえ

構文:

ADAOPT={N|Y}

有効な値:

- **N**。PowerExchange は、最適化を無効にし、L2 コマンドを使用してファイルからレコードを読み取ります。
- **Y**。SQL SELECT 文の WHERE 句で Adabas 記述子を指定すると、PowerExchange は L3 コマンドを使用して、Adabas ファイルから論理的順序でレコードを読み取ります。SQL 文に WHERE 句が含まれていない場合は、PowerExchange は L2 コマンドを使用して、Adabas によって保存された物理的な順序でファイルからレコードを読み取ります。

PowerExchange は、使用中の Adabas バージョンが L3 コマンドでの開始および終了記述子キー範囲をサポートしているかどうかを確認します。

Adabas にユーザー変更をインストールしている場合、インストールされているバージョンの Adabas を PowerExchange で正しく特定できない場合があります。この場合、L3 コマンドを指定するために、ADA_L3_ALLOW 文に Y と指定する必要もあります。

デフォルトは Y です。

使用上の注意:

- PowerExchange ではデフォルトで、データマップの **【最適化レベル】** リストで **【範囲のみ】** が選択されています。**【最適化レベル】** リストで **【OFF】** を選択した場合、PowerExchange はこの文を無視し、そのデータマップのデータを処理するときに L3 コマンドを使用しません。
- ADASTATS 文で Y と指定すると、PowerExchange は PowerExchange メッセージログファイルにメッセージ PWX-02196 を書き込みます。このメッセージは、L3 コマンドと最適化のステータスを入力できるかどうかを示します。

ADAPREFETCH 文

ADAPREFETCH 文は、レコードの読み取り時にパフォーマンスを高めるため、PowerExchange が Adabas プレフェッチ機能を使用するかどうかを制御します。

オペレーティングシステム: z/OS

データソース: Adabas

必須: いいえ

構文:

ADAPREFETCH={N|Y}

有効な値:

- **N**。PowerExchange は、レコードの読み取り時に Adabas プレフェッチ機能を使用しません。
- **Y**。PowerExchange は、レコードの読み取り時に Adabas プレフェッチ機能を使用して、読み取りパフォーマンスを高めます。

デフォルトは N です。

注意: UNIX および Windows の場合、Adabas は ACBX インタフェースを使用したプリフェッチ処理を許可しません。ACBX および ADAPREFETCH=Y をこれらのシステムで使用している場合、PowerExchange は、32

KB より大きいレコードを読み取ることができない ACB 呼び出しに復帰します。ACBX を UNIX または Windows で使用する場合は、ADAPREFETCH=Y 文を指定しないでください。

ADASTATS 文

ADASTATS 文は、PowerExchange が、Adabas 操作に関する統計情報を PowerExchange メッセージログファイルに書き込むかどうかを制御します。

オペレーティングシステム: z/OS

データソース: Adabas

関連した文: ADA_L3_ALLOW、および ADAOPT

必須: いいえ

構文:

ADASTATS={N|Y}

有効な値:

- **N**。PowerExchange は、Adabas 統計メッセージを PowerExchange メッセージログファイルに書き込みません。
- **Y**。PowerExchange は、Adabas 統計メッセージを PowerExchange メッセージログファイルに書き込みます。

レコードの読み取り前に、PowerExchange は、以下の Adabas 情報を含むメッセージを書き込みます。

- PowerExchange がプレフェッチを使用するかどうか

- PowerExchange がデータの読み取りに L2 コマンドを使用するか、L3 コマンドを使用するか

PowerExchange が L3 コマンドを使用する場合、PowerExchange は、L3 コマンドで使用されるキーおよびキー値を含んだ追加メッセージを書き込みます。

- PowerExchange が使用するフォーマットバッファ

レコードの読み取り後、PowerExchange は、データの読み取りに使用した L2 および L3 コマンド数を表示したメッセージを発行します。

デフォルトは N です。

START_UP_USER_EXIT 文

START_UP_USER_EXIT 文は、PowerExchange リスナが起動または停止するたびに PowerExchange が呼び出すユーザー定義イグジットプログラムの名前とプログラミング言語を指定します。

この文を使用して、PowerExchange リスナが Adabas 暗号コードで暗号化された Adabas データベースを復号化できるようにします。指定するユーザーイグジットプログラムは、暗号コードといくつかの追加情報を含む結果セットを提供する必要があります。

オペレーティングシステム: z/OS

必須: いいえ

構文:

START_UP_USER_EXIT=(PROGRAM_NAME=*program_name*, LANGUAGE=*language*)

パラメータ:

PROGRAM_NAME=*program_name*

必須。ユーザーイグジットプログラムの名前。

LANGUAGE=*language*

必須。ユーザーイグジットプログラムの記述で使用するプログラミング言語。次のオプションがあります。

- **A。** アセンブリ言語。
- **C。** C 言語。

使用上の注意:

- それぞれ別個のイグジットプログラムを指す、最大 10 個の文を指定できます。PowerExchange リスナが起動すると、DBMOVER 構成ファイルに関連する START_UP_USER_EXIT 文が発生する順序でユーザーイグジットプログラムが実行されます。リスナがシャットダウンすると、逆の順序でユーザーイグジットプログラムが実行されます。

関連項目：

- [「暗号コードで暗号化された Adabas ソースの復号化」 \(ページ 59\)](#)

Adabas LOAD ライブラリへのアクセスの構成 (z/OS)

MVS システムでは、PowerExchange リスナで Adabas LOAD ライブラリにアクセスしてバルクデータを移動することが必要です。

z/OS Installation Assistant を実行した場合、Adabas LOAD ライブラリを入力しています。z/OS Installation Assistant では、入力した Adabas LOAD ライブラリで、RUNLIB ライブラリの PowerExchange リスナの STARTLST メンバおよび PWXLSTNR メンバをカスタマイズしています。

注: PowerExchange リスナでは、ADARUN、ADAIOR、ADAIOS、ADAMLF、ADAPRF、ADALNK の各 Adabas ロードモジュールの再利用不可能なバージョンが必要です。

1. PowerExchange リスナ JCL の STEPLIB DD 連結に Adabas LOAD ライブラリが含まれていることを確認します。
この JCL は、RUNLIB ライブラリの STARTLST メンバまたは PWXLSTNR メンバにあります。
2. PowerExchange リスナの LOADLIB ライブラリを APF 許可した場合は、Adabas LOAD ライブラリも APF 許可します。

ADARUN 制御文のデフォルト SVC のオーバーライド (z/OS)

z/OS では、PowerExchange Listener は Adabas デフォルト SVC 番号の 249 をデフォルトで使用します。PowerExchange Listener で別の SVC 番号を使用する場合は、別々のメンバで ADARUN 制御文のオーバーライド SVC パラメータを定義する必要があります。

注: ADARUN 制御文は、Adabas 操作環境を定義して開始し、Adabas ユーティリティを開始し、データベース I/O をすべて実行する ADAIOR モジュールをロードします。

1. ADARUN 制御文のオーバーライドパラメータ値を保持するメンバを作成します。
2. 新しいメンバでは、PowerExchange Listener が使用するオーバーライド SVC パラメータを定義します。
以下の構文を使用します。

ADARUN SVC=*nnn*

nnn 変数は、オーバーライド SVC 番号です。有効な値は、200~255 です。Adabas デフォルトは 249 です。

3. 作成したオーバーライドメンバをポイントする PowerExchange Listener JCL に DDCARD DD 文を追加します。以下に例を示します。

//DDCARD DD DSN=*library.file(member)*,DISP=SHR

ADARUN 制御文およびそのパラメータの詳細については、Adabas のマニュアルを参照してください。

暗号コードで暗号化された Adabas ソースの復号化

PowerExchange は、暗号コードで暗号化された Adabas データベースにバルクデータ移動サポートを提供します。

Adabas 暗号コードのサポートを有効にするには、次のアクションを実行します。

- PowerExchange が復号化の実行で必要とする情報とともに結果セットを返すユーザーイグジットプログラムをアセンブリ言語または C で記述します。
- PowerExchange リスナマシン上の DBMOVER 構成ファイルに START_UP_USER_EXIT 文を含めます。この文は、ユーザーイグジットプログラムとプログラミング言語を識別します。

PowerExchange は、PowerExchange リスナの起動またはシャットダウンのたびにユーザーイグジットプログラムを呼び出します。リスナの起動時、Adabas 暗号コードで保護された 1 つ以上の Adabas データベースにアクセスするための情報をイグジットプログラムが提供します。リスナのシャットダウン時、イグジットプログラムはイグジットプログラムが割り当てて使用したリソースをクリーンアップします。

暗号コードがメモリダンプに表示されないように、PowerExchange リスナが実行している間、PowerExchange は暗号コードを暗号化形式でメモリに格納します。

ユーザーイグジットプロセスは、次のいずれかの戻りコードを返します。

- 0=プログラムは正常に完了しました。
- 4=エラーが発生しましたが、PowerExchange リスナは引き続き実行します。リスナは結果セットを無視します。
- 他の値=エラーが発生し、リスナのタスクは終了します。

DBMOVER 構成ファイルで SECURITY 文の最初のパラメータを 2 に設定することをお勧めします。この設定により、PowerExchange リスナに割り当てられたユーザー ID を、個々のデータアクセスタスクとは異なる RACF 承認および認証で実行できます。この方法でセキュリティ設定を定義することで、要求が PowerExchange Navigator、PowerCenter ワークフロー、Informatica クライアントツールのいずれから行われたかに関係なく、PowerExchange はリスナのサブタスクを PowerExchange リスナ要求で提供される RACF ユーザー ID に切り替えます。PowerExchange に転送されるすべてのデータアクセス要求は、要求を出したユーザーアカウントの RACF 承認を使用して実行されます。

z/OS の SRCLIB ライブラリに、LSUUXADC などのサンプルユーザーイグジットプログラムが用意されています。

注意事項:

- Netport ジョブに暗号コードで暗号化された Adabas ソースがある場合、Netport ジョブの実行に使用するユーザー ID は、DBMOVER 構成ファイルの RACF_CLASS 文で指定されたクラスのリソース DTL.LISTENER.AMVALUES に対する READ 権限を持っている必要があります。Netport ジョブは AMTSK (listamvalues) を介して PowerExchange リスナと通信して、Adabas 暗号を取得します。
- Netport ジョブに暗号コードで暗号化された Adabas ソースがある場合、DBMOVER 構成ファイルで OUSP を Y に設定します。
- 暗号コードで暗号化された Adabas ソースにスパンレコードが含まれる場合、SAG 修正プログラム AN826117 および AU826076 を適用する必要があります。そうしないと、スパンレコードを復号化する際に問題が発生する可能性があります。

関連項目：

- [「START_UP_USER_EXIT 文」 \(ページ 57\)](#)

ユーザーイグジットプログラムの結果セット

ユーザーイグジットプログラムは、セミコロンで区切られた文で構成される結果セットを返します。

例えば、SRCLIB ライブラリで提供される C 言語でのサンプルユーザーイグジットプログラムは、次の結果セットを返します。

```
char results_set_without_terminators[] =
{
  "ADA,DBID=8242,FILENUM=104,ActionFlag=1,ActionValue=12345678;"
  "ADA,DBID=8242,FILENUM=105,ActionFlag=1,ActionValue=12345678;"
  "ADA,DBID=8262,FILENUM=105,ActionFlag=1,ActionValue=12345678;"
  "ADA,DBID=0,FILENUM=0,ActionFlag=1,ActionValue=87654321;"
};
```

結果セットの各文には、次のカンマ区切りフィールドが含まれます。

- ADA。必須。ユーザーイグジットプログラムによって適用されるソースタイプを識別します。ADA はサポートされる唯一の値です。
- DBID=*nnn*。オプション。Adabas 暗号コードで保護されたデータベースのデータベース ID。DBID が指定されていないか 0 である場合、暗号コードはすべての DBID に適用されます。
- FILENUM=*nnn*。オプション。Adabas 暗号コードで保護された Adabas ファイルのファイル番号。FILENUM が指定されていないか 0 である場合、暗号コードはすべての FILENUM 番号に適用されます。
- ActionFlag=*n*。必須。ユーザーイグジットが実行するアクションのタイプ。値 1 は、Adabas 暗号コードを使用したデータ復号化を示します。
- ActionValue=*cipher_code*。必須。最大 8 桁の数値の Adabas 暗号コード。

例えば、次の文は、暗号コードをデータベース ID 83 の Adabas データベースにのみ使用することを示します。

```
ADA,DBID=83,ActionFlag=1,ActionValue=12345678
```

次の文は、DBID 値を省略して、暗号コードを z/OS システム上のすべての Adabas データベースで使用することを示します。

```
ADA,ActionFlag=1,ActionValue=12345678
```

注: 両方の例で FILENUM 値が指定されていないため、暗号コードはデータベース内の暗号化されたすべての Adabas ファイルに適用されます。

Adabas バルクデータの移動

Adabas バルクデータを移動するには、以下の手順を使用します。

始める前に、以下の情報を収集します。

- Adabas データベースを搭載しているホストの TCP/IP アドレスまたはホスト名
- ADABAS データベース ID
- アクセスする各ファイルの Adabas ファイル番号
- z/OS のユーザー ID およびパスワード (DBMOVER コンフィギュレーションファイル内で SECURITY 文のセキュリティ設定に必要な場合)

Adabas バルクデータを移動する手順

1. PowerExchange Navigator でデータマップを作成します。

【アクセス方式】 リストで **【ADABAS】** をクリックします。データソースレコードおよびフィールドに関するメタデータをインポートするには、**【レコード定義のインポート】** オプションと **【キーフィールド/FDT のインポート】** オプションのどちらか一方または両方を選択します。

ヒント: **【レコード定義のインポート】** と **【キーフィールド/FDT のインポート】** のどちらも選択しない場合は、後でメタデータをインポートできます。リソースエクスプローラでデータマップを開き、**【ファイル】** - **【コピーブックのインポート】** をクリックします。

2. リソースエクスプローラでデータマップを開き、リレーショナルでない「レコード」またはリレーショナルな「テーブル」のビューをクリックして、メタデータが正常に取得されたことを確認します。
3. データマップがデータソース用のものである場合は、リモートの Adabas システム上の PowerExchange Listener にそのデータマップを送信します。リソースエクスプローラでデータマップを選択し、**【ファイル】** - **【リモートノードへの送信】** をクリックします。この手順により、PowerExchange 抽出ルーチンでランタイム中にデータマップにアクセスすることができます。
4. **【場所】** フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定された PowerExchange Listener ノード名を入力します。必要に応じて、他のフィールドも実行します。
注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。
5. メタデータのテーブルビューに対してデータベース行のテストを実行して、Adabas ソースデータベースから実際のデータを返すことができることをテストします。
6. PowerCenter Designer で、PowerExchange データマップをインポートし、マッピングを作成します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアプリケーション接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

第 4 章

Datacom バルクデータ移動

この章では、以下の項目について説明します。

- [Datacom バルクデータ移動の概要, 62 ページ](#)
- [Datacom バルクデータ移動の構成, 62 ページ](#)
- [Datacom バルクデータの移動, 63 ページ](#)

Datacom バルクデータ移動の概要

PowerExchange では、MVS システム上の CA Datacom ソースからバルクデータを読み込むことができます。一方、Datacom ターゲットにバルクデータを書き込むことはできません。

PowerExchange は、Datacom をリレーショナルでない DBMS として扱います。このため、PowerExchange Navigator から Datacom データソースのデータマップを作成する必要があります。PowerExchange は、そのデータマップを基に、Datacom ソースデータおよびメタデータにアクセスして、処理対象のソースレコードのリレーショナルビューを作成します。

Datacom データベースは MVS システムにあるため、PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作する Windows システムからはリモートとなります。このため、そのリモート MVS システムで追加の PowerExchange Listener を実行し、PowerExchange および PowerCenter がリモート MVS システムと通信できるようにする必要があります。

Datacom バルクデータ移動の構成

バルクデータ移動操作用に PowerExchange を構成するには、PowerExchange 管理者がリモートの MVS システム上の PowerExchange Listener への接続を設定し、テストする必要があります。

リモートの Datacom ソースへの接続の設定およびテスト

リモート z/OS システム上の Datacom データソースにアクセスするには、PowerExchange および PowerCenter でリモートシステム上の PowerExchange Listener と通信する必要があります。

リモートの Datacom ソースへの接続を設定およびテストする手順

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用して Datacom データベースがあるリモートシステムへのネットワークアクセスをテストします。

2. 各 PowerExchange または PowerCenter システム上の dbmover.cfg ファイルに以下の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを特定します。

NODE=(node_name,TCP/IP, ipaddress_or_hostname,port_number)

ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。

3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

Datacom バルクデータの移動

Datacom バルクデータ移動を実行するには、以下の手順を実行します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を収集します。

- Datacom ソースデータベースが搭載されているホストの TCP/IP アドレス
- Datacom データベースの名前および ID
- MVS ユーザー ID とパスワード（PowerExchange の SECURITY 文の設定により必要な場合）

Datacom バルクデータを移動する手順

1. PowerExchange Navigator でデータマップを作成します。

【アクセス方式】 リストで **【Datacom】** をクリックします。レコードおよびフィールドに関するメタデータをインポートするには、**【レコード定義のインポート】** オプションも選択します。

ヒント: **【レコード定義のインポート】** オプションを選択しない場合は、後でメタデータをインポートできます。リソースエクスプローラでデータマップを開き、**【ファイル】** - **【コピーブックのインポート】** をクリックします。

2. リソースエクスプローラでデータマップを開き、リレーショナルでない「レコード」またはリレーショナルな「テーブル」のビューをクリックして、メタデータが正常に取得されたことを確認します。
3. リモートの Datacom システム上の PowerExchange Listener にデータマップを送信します。リソースエクスプローラでデータマップを選択し、**【ファイル】** - **【リモートノードへの送信】** をクリックします。この手順により、PowerExchange 抽出ルーチンでランタイム中にデータマップにアクセスすることができます。

【場所】 フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定された PowerExchange Listener ノード名を入力します。必要に応じて、他のフィールドも実行します。

注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。

4. メタデータのテーブルビューに対してデータベース行のテストを実行して、Datacom ソースデータベースからデータを返すことができることをテストします。

5. PowerCenter Designer で、PowerExchange データマップをインポートします。**【ソース】 - 【PowerExchange からインポート】** をクリックします。次に、以下の必須の属性を入力します。
 - **【場所】** フィールドに、dbmover.cfg ファイルの NODE 文で指定した PowerExchange Listener ノード名を入力します。
 - SECURITY 文の設定に応じて、**【ユーザー名】** フィールドおよび **【パスワード】** フィールドに、z/OS ユーザー ID およびパスワードを入力します。
 - **【ソースタイプ】** リストで、**【DATACOM】** を選択します。
必要に応じてオプションの属性に値を設定します。次に、PowerExchange に接続し、データマップをインポートします。
6. PowerCenter Designer でマッピングを作成します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアップリケーション接続を設定します。次に、バルクデータ移動処理を実行するワークフローを開始します。

第 5 章

DB2 for i5/OS のバルクデータ移動

この章では、以下の項目について説明します。

- [DB2 for i5/OS バルクデータ移動の概要, 65 ページ](#)
- [DB2 for i5/OS バルクデータ移動に関する考慮事項, 66 ページ](#)
- [DB2 for i5/OS バルクデータ移動用にサポートされているデータタイプ, 67 ページ](#)
- [DB2 for i5/OS バルクデータ移動の構成, 68 ページ](#)
- [DB2 for i5/OS のバルクデータ移動, 71 ページ](#)
- [ソースまたはターゲットをトラブルシューティング用に再作成するための SQL 文の生成, 72 ページ](#)
- [i5/OS のアップグレード後の PowerExchange 環境のリフレッシュ, 73 ページ](#)

DB2 for i5/OS バルクデータ移動の概要

PowerExchange では、PowerCenter と連携して、DB2 for i5/OS ソースデータベースまたはターゲットデータベースとの間でバルクデータを移動できます。

PowerExchange DB2 または SEQ のどちらかのデータベースアクセス方式を使用できます。どちらのアクセス方式も一般に、データベーステーブルおよびデータを処理するための DB2 for i5/OS SQL またはシステムファイルアクセス方式に対応しています。DB2 アクセス方式はソースまたはターゲットをリレーショナル DB2 オブジェクトとして処理するのに対して、SEQ アクセス方式はソースまたはターゲットをライブラリ、ファイル、およびメンバとして処理します。SEQ 方式では、バルクデータ移動処理の対象としてメンバの全部または一部を指定できます。

選択するアクセス方式によっては、データマップを作成する必要がありません。バルクデータ移動を定義するには、以下の方法を使用します。

- DB2 アクセス方式を使用する場合は、PowerCenter のみでバルクデータ移動を完全に定義できます。データマップは必要ありません。PowerCenter Designer で、PowerExchange が「DB2」ソースタイプとして DB2 カタログから読み込むメタデータをインポートします。次に、マッピングを作成します。PowerCenter Workflow Manager で、バルクデータ移動を実行するワークフローを作成し、実行します。
また、PowerExchange Navigator では、「DB2」アクセス方式を指定するデータマップを作成することもできます。次に、そのデータマップを非リレーショナルソース定義またはターゲット定義として PowerCenter Designer にインポートします。データマップでは、ユーザー定義フィールドを定義し、そのフィールドに値を代入する式を作成できます。
- SEQ アクセス方式を使用するには、PowerExchange Navigator から、「SEQ」アクセス方式を指定するデータマップを作成する必要があります。PowerCenter Designer で、データマップを非リレーショナルソースまたはターゲットとしてインポートし、マッピングを作成します。PowerCenter Workflow Manager で、バルクデータ移動を実行するワークフローを作成し、実行します。SEQ アクセス方式では、データベースファイルのメンバの全部または一部を処理したり、ファイルリスト処理を使用したりできます。

DB2 データベースは i5/OS システム上で動作するため、PowerExchange Navigator および PowerCenter Integration Service が動作するシステムからはリモートとなります。このため、そのリモート i5/OS システムで追加の PowerExchange Listener を実行し、PowerExchange がリモート i5/OS システムと通信できるようにする必要があります。

DB2 for i5/OS バルクデータ移動に関する考慮事項

DB2 for i5/OS バルクデータ移動操作には、次の考慮事項が適用されます。

- PowerExchange のバルクデータ移動操作に含めることができる LOB カラムを除いた DB2 for i5/OS レコードの最大長は、32,766 バイトです。レコードに変長のカラムや Null 可能なカラムが含まれている場合、DB2 ではそのレコードに追加バイトを加えます。これらの追加バイトと共に、すべてのカラムの合計の長さが 32,766 バイトを超えないことが必要です。詳細については、DB2 for i5/OS のマニュアルを参照してください。

- PowerExchange のバルクデータ移動操作に含めることができる LOB カラムを含めた DB2 for i5/OS レコードの最大長は、8MB です。LOB カラムを含むレコードには、以下の制限が適用されます。

- NRDB SEQ データマップまたは DB2 データマップを使用してデータにアクセスできません。

- LOB カラムが含まれているターゲットテーブルに挿入を実行することはできません。

ヒント: SUBSTRING コマンドを使用して、PowerExchange が LOB カラムから読み取るデータ量を減らすことができます。次のような文を使用します。

```
select SUBSTRING(LOBVALUE,1,900) from myschema.mytable
```

- DB2 アクセス方式を使用する DB2 for i5/OS バルクデータ移動操作の場合、PowerExchange では DB2 の複数行の FETCH 文を使用して、ソーステーブルから一度に複数行のデータを取得します。この機能は、ソースデータの読み取りのためにデータベースにアクセスする回数を減らすことで、パフォーマンスを向上させます。デフォルトでは 25 行が取得されます。PowerCenter で、PWXPC が使用する PWX DB2i5OS リレーショナル接続の**配列サイズ**属性を設定することにより、取得する行数を構成できます。

次の条件がすべて当てはまるとき、PowerExchange は配列サイズを動的に削減します。

- データベースタイプが DB2。

- テーブルに LOB カラムが含まれている。

- **【配列サイズ】** の値が 1 よりも大きい。

- 行サイズ × **【配列サイズ】** の値が 16000000 バイトよりも大きい。

これらの条件を満たす場合、PowerExchange は配列サイズを減らして、クライアントおよび PowerExchange リスナマシンの両方でメッセージ PWX-00186 をログ記録します。

- DB2 for i5/OS ソースファイル内にあるレコードの相対レコード番号 (RRN) の値をプロパゲートするには、SEQ アクセス方式を持つデータマップを作成し、PowerExchange GetDatabaseKey() 式関数によって入力されるユーザー定義フィールドを追加します。この関数により、ソースレコードの RRN でユーザー定義フィールドが入力されます。
- バルクデータ移動は、データベース管理者が機密の DB2 データの可視性を制御するために作成できる Row and Column Access Control (RCAC) ルールをサポートします。これらのルールは、DB2 for i5/OS 7.2 で導入されました。

DB2 for i5/OS バルクデータ移動用にサポートされているデータタイプ

PowerExchange ではバルクデータ移動用のほとんどの DB2 for i5/OS データタイプをサポートしています。

以下の表に、PowerExchange がバルクデータ移動でサポートする DB2 for i5/OS のデータタイプとサポートされないデータタイプを示します。

DB2 データ型	バルクデータ移動のサポートの可否
BIGINT	はい
バイナリ	はい
BLOB	はい（ソースのみ）
CHAR	はい
CLOB	はい（ソースのみ）
DATALINK	いいえ
DATE	はい
DBCLOB	はい（ソースのみ）
DECFLOAT	いいえ
DECIMAL	はい
DISTINCT（ユーザー定義）	いいえ
DOUBLE	はい
FLOAT	はい
GRAPHIC	はい
INTEGER	はい
LONG VARCHAR	はい
LONG VARGHAPHIC	はい
NUMERIC	はい
REAL	はい
ROWID	いいえ
SMALLINT	はい
TIME	はい
TIMESTAMP	はい

DB2 データ型	バルクデータ移動のサポートの可否
VARBINARY	はい
VARCHAR	はい
VARGRAPHIC	はい

DB2 for i5/OS バルクデータ移動の構成

バルクデータ移動操作用に PowerExchange を構成するには、PowerExchange 管理者が以下のタスクを実行します。

- リモートの DB2 for i5/OS システム上の PowerExchange Listener への接続を設定し、テストします。
- DBMOVER 構成メンバに DB2 固有の文を定義することにより、DB2 for i5/OS バルクデータ移動を構成します。

DB2 for i5/OS ソースまたはターゲットへの接続の設定およびテスト

リモートの i5/OS システム上の DB2 データにアクセスするには、PowerExchange がそのリモートシステム上の PowerExchange Listener と通信する必要があります。リモートの i5/OS システムへの接続を設定およびテストするには、以下の手順を実行します。

DB2 for i5/OS ソースまたはターゲットへの接続を設定およびテストするには：

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用して DB2 ソースまたはターゲットがあるリモートシステムへのネットワークアクセスをテストします。
2. 各 PowerExchange または PowerCenter システム上の dbmover.cfg ファイルに以下の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを特定します。

```
NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)
```

ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。
3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER 構成メンバへの DB2 固有の文の追加

DB2 for i5/OS バルクデータ移動を構成するには、i5/OS ソースまたはターゲットシステムの DBMOVER 構成メンバに、以下の DB2 固有の文を含めることができます。

- DB2_BIN_AS_CHAR
- DB2_BIN_CODEPAGE

- DB2_ERROR_FILE
- RDBMSINSRTDFLT

DBMOVER 構成メンバを編集した後、変更を有効にするには PowerExchange Listener をリスタートする必要があります。

各 DBMOVER 文の説明については、『*PowerExchange リファレンスマニュアル*』を参照してください。

DB2_BIN_AS_CHAR 文

DB2_BIN_AS_CHAR 文は、FOR BIT DATA 句で定義された CHAR カラムおよび VARCHAR カラム内のデータを、PowerExchange が文字データと見なすかどうかを制御します。

間違って CCSID 65535 が割り当てられた文字データを上書きするには、DB2_BIN_CODEPAGE 文と一緒にこの文を使用します。

オペレーティングシステム: i5/OS

データソース: DB2 for i5/OS

関連した文: DB2_BIN_CODEPAGE

必須: No

構文:

DB2_BIN_AS_CHAR={N|Y}

有効な値:

- **N**。FOR BIT DATA 句で定義された CHAR カラムおよび VARCHAR カラム内のデータはバイナリデータです。

PowerExchange は、バイナリデータのコードページ変換を実行しません。

- **Y**。FOR BIT DATA 句で定義された CHAR カラムおよび VARCHAR カラム内のデータは文字データです。

デフォルトは N です。

DB2_BIN_CODEPAGE 文

DB2_BIN_CODEPAGE 文は、DB2_BIN_AS_CHAR 文に Y と指定した場合に、FOR BIT DATA 句で定義されたカラム内の文字データを処理するために PowerExchange で使用されるシングルバイトおよびマルチバイト CCSID 値を定義します。

間違って CCSID 65535 が割り当てられた文字データを上書きするには、DB2_BIN_AS_CHAR 文と一緒にこの文を使用します。

オペレーティングシステム: i5/OS

データソース: DB2 for i5/OS

関連した文: DB2_BIN_AS_CHAR

必須: No

構文:

```
DB2_BIN_CODEPAGE=(sbcscsccsid
                   ,dbcsccsid
)
```

パラメータ:

sbcscs_ccsid

必須。シングルバイトデータの CCSI 値。有効な値は 1～65534 です。

dbcs_ccsid

必須。マルチバイトデータの CCSI 値。有効な値は 1～65534 です。

例: 以下の文は、日本語データの CCSID 値を定義します。

DB2_BIN_CODEPAGE=(8482,1390)

DB2_ERRORFILE 文

DB2_ERRORFILE 文は、PowerExchange が DB2 バルクデータ移動操作に使用する、ユーザーがカスタマイズした SQL エラーファイルの名前を指定します。

オペレーティングシステム: すべて

データソース: DB2

必須: いいえ

構文:

DB2_ERRORFILE=*file*

値: *file* 値には、PowerExchange でリカバリ可能または致命的エラーとして処理する SQL エラーコードを含むファイルまたはメンバの名前を入力します。

値のタイプは、Db2 が実行されているオペレーティングシステムによって変わります。

- IBM i (i5/OS) システムの場合は、ライブラリとファイル名を指定し、オプションでメンバ名を指定します。ファイル名とメンバ名は二重引用符 (") で囲みます。
- z/OS システムの場合は、完全修飾シーケンシャルファイル名または PDS 名、およびメンバ名を指定します。ファイル名またはメンバ名は二重引用符 (") で囲みます。
- Linux、UNIX、および Windows システムの場合は、パスとファイル名を指定します。Windows ネットワークパスを指定する場合、PowerExchange でネットワークパスが正しく解析されるように、先頭にバックスラッシュ (\\) を 3 つ追加してください。

使用上の注意: PowerExchange では、次の表に示すように、Db2 データソースの各タイプに対して、サンプルのエラーファイルまたはメンバが用意されています。

データソース	エラーファイルの例
Db2 for z/OS	RUNLIB での DB2ERR メンバ
Db2 for i	<i>datalib</i> /CFG での DB2ERR メンバ
Db2 for Linux, UNIX, and Windows	PowerExchange インストールディレクトリ内の db2err.act

RDBMSINSRTDFLT 文

RDBMSINSRTDFLT 文は、PowerExchange が RDBMS の WITH DEFAULT クラスで定義するカラムにデフォルト値を使用するかどうかを制御します。

オペレーティングシステム: すべて

データソース: DB2、Microsoft SQL Server、および Oracle のターゲット

必須: いいえ

構文:

RDBMSINSRTDFLT={N|Y}

有効な値:

- **N**。PowerExchange は、WITH DEFAULT クラスで定義するカラムにデータを書き込むときに、PowerExchange デフォルトを使用します。
- **Y**。PowerExchange は、WITH DEFAULT クラスで定義するカラムにデータを書き込むときに、RDBMS デフォルトを使用します。

RDBMS がデフォルトを与えられるようにする句を使用してカラムを定義する必要があります。このようにしないと、SQL エラーが発生します。

デフォルトは N です。

DB2 for i5/OS のバルクデータ移動

DB2 for i5/OS バルクデータを移動するには、次のいずれかのアクセス方式を使用します。

- DB2 アクセス方式
- SEQ アクセス方式

DB2 for i5/OS バルクデータの移動 - DB2 アクセス方式

DB2 for i5/OS バルクデータを移動するには、以下の手順を実行します。この手順では、DB2 アクセス方式と、データマップのない PWXPC を使用するものとします。

始める前に、以下の情報を収集します。

- DB2 データベース名およびスキーマ名
- 次の形式のうち 1 つの DB2 テーブル名: *schema.table* (SQL テーブル形式を使用している場合) または *library.file* (シーケンシャルファイル形式を使用している場合)
- i5/OS のユーザー ID およびパスワード (DBMOVER コンフィギュレーションファイル内で SECURITY 文のセキュリティ設定に必要な場合)

また、まず PowerExchange Navigator から DB2 データをプレビューする場合は、パーソナルメタデータプロファイルを作成し、そのプロファイルに対してデータベース行テストを実行できます。PowerExchange Navigator には、DB2 カラムごとにメタデータが表示されます。メタデータには、データ型、日付形式、CCSID などのカラム属性が表示されます。DB2 for i5/OS では、複数の日付形式に日付およびタイムスタンプを格納できます。

DB2 アクセス方式を使用して、DB2 for i5/OS バルクデータを移動する手順

1. PowerCenter Designer で、DB2 データソースに **[ソース] - [PowerExchange からインポート]**、または、DB2 データターゲットに **[ターゲット] - [PowerExchange からインポート]** をクリックします。
2. [PowerExchange からインポート] ダイアログボックスで、ソースタイプとして **[DB2i5OS]** を選択します。また、データベースの場所と DB2 データベース名の他、データベースへのアクセスに必要な場合にはユーザー ID とパスワードも入力します。必要に応じて、他のフィールドも実行します。

注: [場所] フィールドに、PowerExchange システムの DBMOVER コンフィギュレーションファイルの NODE 文で指定した PowerExchange Listener ノード名を入力します。

3. PowerCenter Designer で、DB2 for i5/OS ソースまたはターゲットが含まれているマッピングを作成します。

4. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX DB2i5OS バッチリレーショナルデータベース接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

DB2 for i5/OS バルクデータの移動 - SEQ アクセス方式

DB2 for i5/OS バルクデータを移動するには、以下の手順を実行します。この手順では、SEQ アクセス方式および PWXPC を使用するものとします。

SEQ アクセス方式を使用して、DB2 for i5/OS バルクデータを移動する手順

1. PowerExchange Navigator でデータマップを作成します。
アクセス方式として **[SEQ]** を選択します。レコードおよびフィールドに関するメタデータをインポートするには、**[レコード定義のインポート]** オプションも選択します。
ヒント: **[レコード定義のインポート]** オプションを選択しない場合は、後でメタデータをインポートできます。そのためには、リソースエクスプローラでデータマップを選択し、**[ファイル] - [コピーブックのインポート]** をクリックします。
2. **[ファイル名]** フィールドに、以下の形式のうち 1 つを使用して i5/OS ファイルを指定します。
 - ライブラリ/ファイル
 - ライブラリ/ファイル (メンバ)
3. リソースエクスプローラでデータマップを開き、**レコード**または**テーブル**のビューをクリックして、メタデータが正常に取得されたことを確認します。
4. リモートの i5/OS システム上の PowerExchange Listener にデータマップを送信するには、リソースエクスプローラでデータマップを選択し、**[ファイル] - [リモートノードへの送信]** をクリックします。
[場所] フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定された値を入力します。必要に応じて、他のフィールドも実行します。
注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。
5. メタデータのテーブルビューに対してデータベース行のテストを実行して、Datacom ソースデータベースから実際のデータを返すことができることをテストします。
6. PowerCenter Designer で、PowerExchange データマップをインポートし、マッピングを作成します。
ヒント: データマップをインポートする場合、**[場所]** フィールドに、ローカルシステムにある dbmover.cfg の NODE 文で指定された PowerExchange Listener ノード名を入力します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアップリケーション接続を設定します。次に、バルクデータ移動処理を実行するワークフローを開始します。

ソースまたはターゲットをトラブルシューティング用に再作成するための SQL 文の生成

PowerExchange には、PowerExchange 環境で使用される SQL ソースまたはターゲットオブジェクトを再作成する SQL 文の生成コマンドが用意されています。このコマンドは、Informatica グローバルカスタマサポートがバルクデータ移動または CDC の問題のトラブルシューティングに使用できる SQL 文を生成します。

i5/OS SQL オブジェクトの SQL 文を生成するには、PowerExchange がインストールされている i5/OS システムから RTVSQLSTMT コマンドを入力します。PowerExchange *dtllib* ライブラリが i5/OS ジョブの現在のライブラリリストに含まれている必要があります。RTVSQLSTMT コマンドでは、生成される SQL 文を制御する一

連のパラメータを入力するように求められます。エラーのリスクを軽減するため、RTVSQLSTMT コマンドにより入力が検証されます。

重要: いつでも i5/OS コンソールでヘルプ情報を表示し、フィールドの説明、コマンドの例、コマンドの実行に必要な権限を確認できます。

このコマンドは、エイリアス、制約、インデックス、プロシージャ、スキーマ（ライブラリまたはコレクション）、テーブル、トリガ、ビューを含む、多くのタイプのオブジェクトを再作成する DDL 文を生成できます。また、オブジェクトの使用に必要な権限を付与する GRANT 文も生成できます。生成する SQL 文のタイプを制御するオプションを使用できます。例えば、選択したオブジェクトタイプに対して CREATE 文に先行する DROP 文を生成するかどうかを制御できます。また、SQL で作成されていないオブジェクトに対しても SQL 文を生成できます。例えば、物理および論理データベースファイルの定義にデータ記述仕様（DDS）を使用した場合、このコマンドでそれらのファイルを再作成する同等の SQL CREATE 文を生成できます。

次のサンプルコマンドには、すべての必須パラメータとオプションパラメータが示されています。

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

注: パラメータ名は太字で強調表示されています。これらのパラメータとそのすべてのオプションの説明は、ヘルプを参照してください。

このコマンドを実行すると、指定したソースファイルメンバに SQL 文が書き込まれます。ソースメンバ名と同じ名前を持つメンバがすでに存在する場合、置き換えるかどうかを示す RPLSRCMBR パラメータを設定できます。

SQL 文を含むソースファイルメンバを Informatica グローバルカスタマサポートに送信します。これによりサポート担当者が別のマシンでその SQL 文を実行し、エラーが発生した環境を再作成できます。

インタフェースを使用しているときに PowerExchange が生成する検証メッセージの詳細については、『PowerExchange メッセージリファレンスボリューム 3』のメッセージ DTL5001~DTL5008 を参照してください。

i5/OS のアップグレード後の PowerExchange 環境のリフレッシュ

PowerExchange がインストールされている DB2 システム、または PowerExchange がデータにアクセスするリモート DB2 システムで i5/OS オペレーティングシステムをアップグレードする場合、システムメタデータへの変更内容を有効にするには、いくつかのコマンドを実行して PowerExchange 環境をリフレッシュする必要があります。i5/OS のアップグレードを完了後、PowerExchange リスナジョブまたは PowerExchange Condense ジョブを再実行してバルクデータ移動または CDC 処理を再開する前に、コマンドを実行します。

コマンドは、次のいずれかのタイプのユーザープロファイルで実行する必要があります。

- IBM が提供する QSECOFR ユーザープロファイル
- *SECOFR の USRCLS および *USRCLS の SPCAUT が定義されたユーザープロファイル
- 少なくとも、*SECADM、*ALLOBJ、および *JOBCTL の SPCAUT 値が定義されたユーザープロファイル

1. 次の ADDLIBLE コマンドを発行します。

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

dtllib 変数は、PowerExchange ソフトウェアライブラリです。

2. PowerExchange 環境をリフレッシュするには、最後の PowerExchange フルインストール時に CRTPWXENV コマンドに使用したパラメータと同じパラメータで、次のいずれかの CRTDTLENVF コマンドを発行します。

- DB2 for i5/OS ソースデータベースが PowerExchange インストールと同じ i5/OS サーバー上にある場合は、以下を使用します。

```
CRTDTLENVF DTLLIB(dtllib) DATALIB(datalib) RMTRDBDIRE(*LOCAL) OSLEVEL(*LOCAL)
```

- DB2 for i5/OS ソースデータベースが PowerExchange をインストールしたサーバーとは別のサーバー上にある場合は、以下を使用します。

```
CRTDTLENVF DTLLIB(dtllib) DATALIB(datalib) RMTRDBDIRE(database_name) RMTSYSNAME(host_name)  
RMTOSLEVEL(os_level)
```

パラメータの説明については、『PowerExchange インストール&アップグレードガイド』の第 4 章を参照してください。

3. 最後の PowerExchange フルインストール時に *dtllib*、*datalib*、*condlib*、および *cpplib* ライブラリなどのオブジェクトの所有権を変更した場合は、コマンドを再度実行してオブジェクトの所有権を変更する必要があります。

詳細については、『PowerExchange インストール&アップグレードガイド』の第 4 章を参照してください。

第 6 章

DB2 for Linux、UNIX、および Windows バルクデータ移動

この章では、以下の項目について説明します。

- [DB2 バルクデータ移動の概要, 75 ページ](#)
- [DB2 for Linux、UNIX、および Windows バルクデータ移動用にサポートされているデータタイプ, 76 ページ](#)
- [DB2 バルクデータ移動の構成, 77 ページ](#)
- [DB2 バルクデータの移動, 79 ページ](#)

DB2 バルクデータ移動の概要

PowerExchange では、PowerCenter と連携して、DB2 for Linux、UNIX、および Windows データベースとの間でバルクデータを移動できます。

DB2 はリレーショナルデータベースであるため、PowerExchange Navigator からデータマップを作成する必要があります。PowerCenter Designer および Workflow Manager のみでバルクデータ移動を完全に定義できます。PowerCenter Designer では、PowerExchange がソース定義またはターゲット定義を作成するために DB2 カタログから読み込む DB2 ソースまたはターゲットのメタデータをインポートできます。

DB2 データベースが、PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムからリモートとなるシステムにある場合、そのリモートシステムで追加の PowerExchange Listener を実行し、PowerExchange がそのリモートシステムと通信できることを確認する必要があります。

また、以下のバルクデータ移動方法を使用することもできます。

- PowerCenter のみでバルクデータ移動を完全に実行します。
- PowerExchange Navigator で DB2 データマップを作成してから、リレーショナルでないデータマップと同じように PowerCenter にインポートします。ユーザー定義フィールドを作成し、そのフィールドに値を代入する式を作成する場合には、この方法を使用することをお勧めします。

DB2 for Linux、UNIX、および Windows バルクデータ移動用にサポートされているデータタイプ

PowerExchange ではバルクデータ移動用のほとんどの DB2 for Linux, UNIX and Windows のデータタイプをサポートしています。例外は、LOB、DECFLOAT、または XML の各データタイプのカラムです。

次の表には、DB2 カタログで使用される可能性がある DB2 データタイプの一覧と、それらのデータタイプが PowerExchange でバルクデータ移動用にサポートされているかどうかが表示されています。

DB2 データ型	バルクデータ移動のサポートの可否
BIGINT	○
BLOB	いいえ
CHAR	○
CLOB	いいえ
DATE	○
DBCLOB	いいえ
DECFLOAT	×
DECIMAL	○
DOUBLE	○
GRAPHIC	いいえ
INTEGER	○
LONG VARCHAR	はい
LONG VARGRAPHIC	いいえ
REAL	○
REF	いいえ
SMALLINT	○
TIME	○
TIMESTAMP	○
UDT ¹	いいえ
VARCHAR	○
VARGRAPHIC	いいえ

DB2 データ型	バルクデータ移動のサポートの可否
XML	×
1. DISTINCT や STRUCT などのユーザー定義のデータタイプ。	

DB2 バルクデータ移動の構成

バルクデータ移動操作に PowerExchange を構成するには、PowerExchange 管理者が以下のタスクを実行します。

- DB2 データソースまたはターゲットがリモートシステムに存在する場合は、リモートシステム上の PowerExchange Listener に対して接続を設定し、テストします。
- PowerExchange フォールトトレランス動作をカスタマイズするためのエラーアクションファイルを作成した場合は、dbmover.cfg ファイルに DB2_ERRORFILE 文を追加します。
- コードページを指定するには、次の環境変数 DBCODEPAGE=1208 を定義する必要があります。

リモート DB2 ソースまたはターゲットへの接続の設定およびテスト

DB2 データソースまたはターゲットがリモートシステムにある場合は、PowerExchange および PowerCenter がそのリモートシステム上の PowerExchange Listener と通信できる必要があります。

リモート DB2 ソースまたはターゲットへの接続を設定およびテストするには：

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用して DB2 ソースまたはターゲットがあるリモートシステムへのネットワークアクセスをテストします。
2. 各 PowerExchange および PowerCenter システム上の dbmover.cfg ファイルに次の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを特定します。
`NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)`
ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。
3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER コンフィギュレーションファイルの DB2 文の構成

DB2 for Linux、UNIX、および Windows バルクデータ移動を構成するには、DB2 for Linux、UNIX、および Windows ソースまたはターゲットシステムの dbmover.cfg ファイルに、以下の DB2 固有の文を含めることができます。

- DB2_ERRORFILE
- RDBMSINSRTDFLT

DBMOVER コンフィギュレーションファイルの編集後、変更を有効にするには PowerExchange Listener をリスタートします。

各 DBMOVER 文の説明については、『PowerExchange リファレンスマニュアル』を参照してください。

DB2_ERRORFILE 文

DB2_ERRORFILE 文は、PowerExchange が DB2 バルクデータ移動操作に使用する、ユーザーがカスタマイズした SQL エラーファイルの名前を指定します。

オペレーティングシステム: すべて

データソース: DB2

必須: いいえ

構文:

DB2_ERRORFILE=*file*

値: *file* 値には、PowerExchange でリカバリ可能または致命的エラーとして処理する SQL エラーコードを含むファイルまたはメンバの名前を入力します。

値のタイプは、Db2 が実行されているオペレーティングシステムによって変わります。

- IBM i (i5/OS) システムの場合は、ライブラリとファイル名を指定し、オプションでメンバ名を指定します。ファイル名とメンバ名は二重引用符 (“) で囲みます。
- z/OS システムの場合は、完全修飾シーケンシャルファイル名または PDS 名、およびメンバ名を指定します。ファイル名またはメンバ名は二重引用符 (“) で囲みます。
- Linux、UNIX、および Windows システムの場合は、パスとファイル名を指定します。Windows ネットワークパスを指定する場合、PowerExchange でネットワークパスが正しく解析されるように、先頭にバックスラッシュ (\\) を 3 つ追加してください。

使用上の注意: PowerExchange では、次の表に示すように、Db2 データソースの各タイプに対して、サンプルのエラーファイルまたはメンバが用意されています。

データソース	エラーファイルの例
Db2 for z/OS	RUNLIB での DB2ERR メンバ
Db2 for i	<i>datalib</i> /CFG での DB2ERR メンバ
Db2 for Linux, UNIX, and Windows	PowerExchange インストールディレクトリ内の db2err.act

RDBMSINSRTDFLT 文

RDBMSINSRTDFLT 文は、PowerExchange が RDBMS の WITH DEFAULT クラスで定義するカラムにデフォルト値を使用するかどうかを制御します。

オペレーティングシステム: すべて

データソース: DB2、Microsoft SQL Server、および Oracle のターゲット

必須: いいえ

構文:

RDBMSINSRTDFLT={N|Y}

有効な値:

- **N**。PowerExchange は、WITH DEFAULT クラスで定義するカラムにデータを書き込むときに、PowerExchange デフォルトを使用します。
- **Y**。PowerExchange は、WITH DEFAULT クラスで定義するカラムにデータを書き込むときに、RDBMS デフォルトを使用します。

RDBMS がデフォルトを与えられるようにする句を使用してカラムを定義する必要があります。このようにしないと、SQL エラーが発生します。

デフォルトは N です。

DB2 バルクデータの移動

DB2 for Linux、UNIX、および Windows データベースとの間でバルクデータを移動するには、以下の手順を実行します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を取得します。

- DB2 データベース名
- DB2 ソーステーブルまたはターゲットテーブルのテーブル名
- データベースのユーザー ID とパスワード

ヒント: PowerExchange Navigator では、DB2 ソースデータを移動する前にその内容をプレビューできます。パーソナルメタデータプロファイルを作成し、それに対してデータベース行テストを実行します。PowerExchange Navigator には、DB2 カラムごとに、データ型、日付形式、CCSID などのメタデータが表示されます。

DB2 バルクデータを移動する手順

1. PowerCenter Designer で、DB2 データソースに **[ソース] - [PowerExchange からインポート]**、または、DB2 データターゲットに **[ターゲット] - [PowerExchange からインポート]** をクリックします。
2. **[PowerExchange からインポート]** ダイアログボックスで、以下の必須の属性を入力します。
 - PowerCenter が動作するローカルシステムに DB2 データベースが存在する場合は、**[場所]** フィールドに、ローカルの dbmover.cfg ファイルの LISTENER 文で指定した PowerExchange Listener ノード名を入力します。DB2 データベースが別のシステムにある場合は、ローカルの dbmover.cfg の NODE 文に指定されているノード名を入力します。
 - **[ユーザー名]** フィールドおよび **[パスワード]** フィールドに、DB2 ソーステーブルまたはターゲットテーブルへのアクセスに必要なユーザー ID およびパスワードを入力します。
 - **[ソースタイプ]** リストから、**[DB2LUW]** を選択します。
 - **[データベース名]** フィールドに、DB2 データベース名を入力します。
必要に応じてオプションの属性に値を設定します。
3. PowerCenter Designer で、DB2 ソースまたはターゲットが含まれているマッピングを作成します。
4. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX DB2LUW リレーションデータベース接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

第 7 章

DB2 for z/OS のバルクデータ移動

この章では、以下の項目について説明します。

- [DB2 for z/OS バルクデータ移動の概要, 80 ページ](#)
- [DB2 for z/OS バルクデータ移動用にサポートされているデータタイプ, 81 ページ](#)
- [DB2 for z/OS バルクデータ移動の構成, 83 ページ](#)
- [リレーショナルソース定義またはターゲット定義を含む DB2 for z/OS バルクデータ移動, 95 ページ](#)
- [イメージコピーソースを含む DB2 for z/OS バルクデータ移動, 96 ページ](#)
- [非リレーショナルソース定義を含む DB2 for z/OS バルクデータ移動, 99 ページ](#)
- [DB2 LOAD ユーティリティを使用したバルクデータのロード, 101 ページ](#)

DB2 for z/OS バルクデータ移動の概要

PowerExchange では、PowerCenter と連携して、DB2 for z/OS テーブルとの間でバルクデータを移動できます。

必要に応じて、データソースの DB2 イメージコピーまたはアンロードファイルを作成してから、バルクデータ移動操作にそのイメージコピーまたはアンロードファイルを使用できます。PowerCenter で使用できるイメージコピーを作成するには、DB2 オンラインユーティリティ COPY を使用するか、または COPY 互換のイメージコピーを生成するユーティリティを使用する必要があります。DB2 ソーステーブルスペースまたはテーブルスペースパーティションのフルイメージコピーを作成します。

また、DB2 LOAD ユーティリティを使用して、DB2 for z/OS ターゲットにバルクデータをロードできます。PowerExchange には、DB2 LOAD ユーティリティを実行するための JCL テンプレートおよび制御カードテンプレートが用意されています。また、PowerExchange は LOAD ユーティリティが DB2 ターゲットにロードするデータを含むデータファイルを生成します。必要に応じて、バルクデータ LOAD 操作を Netport ジョブのタスクとして実行できます。

このようなユーティリティと PowerExchange を使用する場合は、バルクデータ移動用の設定作業を別途実行する必要があります。

DB2 for z/OS テーブルなどのリレーショナルソースおよびターゲットの場合、データマップを作成する必要はありません。PowerCenter では、PowerExchange がソース定義またはターゲット定義を作成するために DB2 カタログから読み込むメタデータをインポートできます。PowerCenter でバルクデータ移動を実行するとき、PowerExchange ではこれらの定義を使用してソーステーブルまたはターゲットテーブルにアクセスします。

ただし、PowerCenter では DB2 アンロードファイルなどの DB2 ソースをリレーショナルソースとして扱うことができません。PowerCenter で DB2 データソースをリレーショナルでないソースとして扱うには、PowerExchange Navigator で DB2 アクセス方式または DB2UNLD アクセス方式を備えたデータマップを作成する必要があります。次に、そのデータマップを PowerCenter にインポートして、リレーショナルでないタイプのソース定義を作成します。

DB2 for z/OS バルクデータ移動用にサポートされているデータタイプ

PowerExchange ではバルクデータ移動用のほとんどの DB2 for z/OS データタイプをサポートしています。

以下の表に、PowerExchange がバルクデータ移動でサポートする DB2 for z/OS のデータタイプとサポートされない DB2 for z/OS のデータタイプを示します。

DB2 データ型	バルクデータ移動のサポートの可否
BIGINT	○
BINARY	○
BLOB	○（ソースのみ）
CHAR	○
CLOB	○（ソースのみ）
DATE	○
DBCLOB	○（ソースのみ）
DECFLOAT	×
DECIMAL	○
DISTINCT（ユーザー定義）	×
DOUBLE	○
FLOAT	○
GRAPHIC	○
LONG VARCHAR	○
LONG VARGHAPHIC	○
INTEGER	○
REAL	○
ROWID	×
SMALLINT	○
TIME	○
TIMESTAMP	○（最大 12 桁の秒の小数部分をサポートする拡張精度 TIMESTAMP カラムを含む）
TIMESTAMP WITH TIME ZONE	×

DB2 データ型	バルクデータ移動のサポートの可否
VARBINARY	○
VARCHAR	○
VARGRAPHIC	○
XML	×

DB2 for z/OS の TIMESTAMP データ型

DB2 10 for z/OS では、最大 12 桁のサブ秒値を使用できる拡張精度 TIMESTAMP カラムがサポートされるようになりました。PowerCenter が拡張精度 TIMESTAMP カラムをどのようにトランスフォーメーションデータ型にマップするかは、スケールによって異なります。

以下の表に、拡張精度 TIMESTAMP データ型がトランスフォーメーションデータ型にどのようにマップするかを示します。

スケール	精度	トランスフォーメーションデータ型
6	26	date/time
0	19	string
1～5、または 7～12	20+scale	string

DB2 の TIMESTAMP データを DB2 ターゲットに書き込む際、ソースとターゲットの位取りが同じである必要があります。そうしなければ、予期しない結果が発生することがあります。

DB2 ソースの TIMESTAMP フィールドの位取りが 9 より大きく、対応する DB2 ターゲットの TIMESTAMP フィールドの位取りが 6 の場合、値はターゲットに書き込まれる前に切り詰められます。

拡張精度 TIMESTAMP データを非リレーショナルターゲットに書き込む場合は、**【タスクの編集】** ダイアログボックスで以下のセッション属性を定義して、すべての TIMESTAMP フィールドが同じ形式になるようにします。

- **【日付/時刻フォーマット文字列】** を **YYYY-MM-DD HH24:MI:SS** に設定します。
- **【85 以前のタイムスタンプの互換性】** の選択を取り消します。

DB2 for z/OS LOB データ型

LOB データを持つカラムを含んだ DB2 テーブルをバルクデータ移動セッションにソースとして含めることができます。LOB データ型には、BLOB、CLOB および DBCLOB があります。

データマップで定義できる、または PowerExchange のバルクデータ移動操作に含めることができる LOB カラムを持つ DB2 for z/OS 行の最大長は 8MB です。

LOB カラムを含むテーブルには、以下の制限が適用されます。

- アンロードファイルはデータソースとして使用できません。
- LOB カラムがインラインの場合にのみ、イメージコピーをデータソースとして使用できます。
- LOB カラムが含まれているターゲットテーブルに挿入を実行することはできません。

ヒント: DB2 からデータを読み取るときは、SUBSTRING コマンドを使用して、PowerExchange が LOB カラムから読み取るデータの量を減らすことができます。

BLOB または CLOB カラムのデータを選択するには、次のような文を使用します。

```
select SUBSTRING(LOBVALUE,1,900,OCTETS) from myschema.mytable
```

DBCLOB カラムのデータを選択するには、次のような文を使用します。

```
select SUBSTRING(DBCLOBVALUE,1,900,CODEUNITS16) from myschema.mytable
```

DB2 イメージコピーファイルまたは CDC データを処理するときに使用される PowerExchange 非リレーショナル SQL は、SUBSTRING()関数をサポートしていません。

DB2 for z/OS バルクデータ移動の構成

PowerExchange のバルクデータ移動操作を構成するには、PowerExchange 管理者が以下の構成タスクを行います。

1. リモートの DB2 for z/OS システム上の PowerExchange Listener への接続を設定し、テストする。
2. z/OS ソースまたはターゲットシステム上の DBMOVER 構成メンバに DB2 固有の文を定義することにより、DB2 for z/OS バルクデータ移動を構成する。
3. PowerExchange ユーザーアプリケーションで、必要とする DB2 データにアクセスするために必要な権限をユーザーに付与する。
4. オプション。PowerExchange で DB2 複数行フェッチ挿入機能をサポートする場合、この機能の要件が満たされていることを確認する。
5. オプション。DB2 LOAD ユーティリティを使用して DB2 ターゲットにバルクデータをロードする場合、LOAD ユーティリティを使用するように PowerExchange を構成する。

関連項目：

- [「DBMOVER 構成メンバへの DB2 固有の文の追加」 \(ページ 84\)](#)
- [「DB2 リソースへのアクセスに必要な権限」 \(ページ 93\)](#)
- [「DB2 複数行 FETCH 文および INSERT 文」 \(ページ 94\)](#)
- [「DB2 LOAD ユーティリティ用の PowerExchange テンプレート」 \(ページ 101\)](#)
- [「リモートの DB2 for z/OS ソースまたはターゲットへの接続の設定およびテスト」 \(ページ 83\)](#)

リモートの DB2 for z/OS ソースまたはターゲットへの接続の設定およびテスト

リモートの z/OS システム上の DB2 バルクデータにアクセスするには、PowerExchange がそのリモートシステム上の PowerExchange Listener と通信する必要があります。

リモートの DB2 for z/OS ソースまたはターゲットへの接続を設定およびテストするには：

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで **ping** コマンドを使用して DB2 ソースまたはターゲットがあるリモートの MVS システムへのネットワークアクセスをテストします。

2. 各 PowerExchange または PowerCenter システム上の dbmover.cfg ファイルに以下の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを特定します。

NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)

ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。

3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER 構成メンバへの DB2 固有の文の追加

DB2 for z/OS バルクデータ移動を構成するには、z/OS ソースまたはターゲットシステムの DBMOVER 構成メンバに、以下の DB2 固有の文を含めることができます。

- DB2_ERRORFILE
- DB2CODEPAGE
- DB2DEF_ENCODING
- DB2ID
- DB2PLAN
- MVSDDB2AF
- RDBMSINSRTDFLT
- SESSID

DBMOVER 構成メンバを編集した後、変更を有効にするには PowerExchange Listener をリスタートする必要があります。

各 DBMOVER 文の説明については、『PowerExchange リファレンスマニュアル』を参照してください。

DB2_ERRORFILE 文

DB2_ERRORFILE 文は、PowerExchange が DB2 バルクデータ移動操作に使用する、ユーザーがカスタマイズした SQL エラーファイルの名前を指定します。

オペレーティングシステム: すべて

データソース: DB2

必須: いいえ

構文:

DB2_ERRORFILE=*file*

値: *file* 値には、PowerExchange でリカバリ可能または致命的エラーとして処理する SQL エラーコードを含むファイルまたはメンバの名前を入力します。

値のタイプは、Db2 が実行されているオペレーティングシステムによって変わります。

- IBM i (i5/OS) システムの場合は、ライブラリとファイル名を指定し、オプションでメンバ名を指定します。ファイル名とメンバ名は二重引用符 (") で囲みます。
- z/OS システムの場合は、完全修飾シーケンシャルファイル名または PDS 名、およびメンバ名を指定します。ファイル名またはメンバ名は二重引用符 (") で囲みます。

- Linux、UNIX、および Windows システムの場合は、パスとファイル名を指定します。Windows ネットワークパスを指定する場合、PowerExchange でネットワークパスが正しく解析されるように、先頭にバックスラッシュ (\\\) を 3 つ追加してください。

使用上の注意: PowerExchange では、次の表に示すように、Db2 データソースの各タイプに対して、サンプルのエラーファイルまたはメンバが用意されています。

データソース	エラーファイルの例
Db2 for z/OS	RUNLIB での DB2ERR メンバ
Db2 for i	<i>datalib</i> /CFG での DB2ERR メンバ
Db2 for Linux, UNIX, and Windows	PowerExchange インストールディレクトリ内の db2err.act

DB2CODEPAGE 文

DB2CODEPAGE 文は、バルクデータ移動で呼び出される DB2 for z/OS サブシステム内の CHAR および VARCHAR カラムデータの処理に PowerExchange が使用するシングルのバイト、グラフィック、およびミックスの CCSID を定義します。

オペレーティングシステム: z/OS

データソース: DB2 for z/OS

必須: いいえ

構文:

```
DB2CODEPAGE=(db2_subsystem
  [,DB2TRANS={P|N|R}]
  [,MIXED={N|Y}]
  [,EBCDIC_CCSID=({sbcscsccsid|037}
    ,{graphic_ccsid|037}
    ,{mixed_ccsid|037})]
  [,ASCII_CCSID=({sbcscsccsid|850}
    ,{graphic_ccsid|65534}
    ,{mixed_ccsid|65534})]
  [,UNICODE_CCSID=({sbcscsccsid|367}
    ,{graphic_ccsid|1200}
    ,{mixed_ccsid|1208})]
  [,PLAN_CCSID=({sbcscsccsid|037}
    ,{graphic_ccsid|037}
    ,{mixed_ccsid|037})]
  [,REMAP=({current_data_ccsid},{remapped_data_ccsid})
)
```

パラメータ:

db2_subsystem

必須。ソースまたはターゲットサブシステムの DB2 サブシステム識別子 (SSID)。

DB2TRANS=[P|N|R]

DB2 が PowerExchange とやり取りするデータのエンコーディングを変換するかどうかを示します。次のオプションがあります。

- **P**。DB2 は、カラムデータが保存されているコードページを PowerExchange 用にバインドされた DB2 プランで定義されているコードページに変換します。EBCDIC_CCSID パラメータおよび PLAN_CCSID パラメータ（オプション）も指定する必要があります。両方指定した場合は、PLAN_CCSID パラメータが優先されます。ASCII と Unicode のデータがある場合、EBCDIC コードページにマップするための ASCII_CCSID パラメータと UNICODE_CCSID パラメータを指定できます。
注: いずれかの*_CCSID パラメータを使用するには、DB2TRANS を P に設定する必要があります。
- **N**。DB2 は、カラムデータのコードページを対応する EBCDIC コードページに変換しません。PowerExchange は、データが保存されているネイティブコードページを使用します。EBCDIC_CCSID、ASCII_CCSID、UNICODE_CCSID、または PLAN_CCSID の各パラメータを定義する必要はありません。
- **R**。DB2 は、ユーザーが指定した特定のデータコードページを 1 つ以上の REMAP*n* パラメータで定義される他のコードページに変換します。REMAP*n* の各パラメータでは、最初の位置パラメータが再マップのためのデータコードページを識別し、2 番目の位置パラメータが使用するコードページを識別します。PowerExchange DB2 プランがバインドされているコードページ以外のコードページを使用します。

デフォルトは P です。

MIXED=[N|Y]

ASCII と EBCDIC 文字列を格納する DB2 カラムにミックスおよびグラフィックデータが含まれているかどうかを示しています。ミックスデータはシングルバイトおよびダブルバイト両方の文字で構成され、グラフィックデータはダブルバイト文字で構成されています。

次のオプションがあります。

- **N**。ASCII および EBCDIC 文字列があるカラムには、シングルバイト（SBCS）データのみが含まれます。ミックスおよびグラフィックデータは発生しません。
- **Y**。ASCII および EBCDIC 文字列があるカラムには、ミックスデータとグラフィックデータが含まれます。

デフォルトは N です。

EBCDIC_CCSID={sbcsc_ccsid[037],{graphic_ccsid[037],{mixed_ccsid[037]}}

DB2TRANS=P（デフォルト）の場合は必須です。EBCDIC シングルバイト、グラフィックダブルバイト、およびミックス（シングルおよびダブルバイト）データに使用する CCSID。

各位置サブパラメータの有効な値は 1~65534 です。値 65534 は、CCSID がいないことを示しています。

デフォルト値の 037,037,037 は、DB2CODEPAGE 文を指定していない場合、または z/OS Installation Assistant からの値を変更せずに受け入れた場合に使用されます。インストール中にこれらの値を変更した場合は、Installation Assistant により DBMOVER ファイル内の DB2CODEPAGE 文が、入力した値にカスタマイズされます。

EBCDIC コードページに ICU 変換テーブルがない場合、または EBCDIC コードページが PowerExchange の DB2 プランがバインドされているものとは異なる場合は、PLAN_CCSID パラメータを使用して、ICU 変換テーブルを持つ、または DB2 プランを反映した別の EBCDIC コードページにマップすることができます。

注: データが物理的に ASCII または Unicode コードページで格納されていても、DB2 は EBCDIC コードページで PowerExchange にデータを配布します。DB2 カタログテーブルは、CCSID 1208 に対応する

UTF-8 で格納されます。この CCSID は UNICODE_CCSID *mixed_ccsid* コードページ値から EBCDIC_CCSID *mixed_ccsid* コードページ値に再マップされます。

シングルバイトのシステムの場合は、SBCS コードページを 3 回繰り返すか、または 65534 と MIXED=N パラメータを使用します。以下に例を示します。

```
EBCDIC_CCSID=(1047,1047,1047)
```

または

```
DB2CODEPAGE=(D91G,EBCDIC_CCSID=(1047,65534,65534),MIXED=N)
```

次に、PowerExchange はシングルバイト EBCDIC コードページを使用します。

```
ASCII_CCSID=({sbcsc_ccsid|850},{graphic_ccsid|65534},{mixed_ccsid|65534})
```

オプション。ASCII シングルバイト、グラフィック、およびミックスデータ統合サービスに使用する CCSID。このパラメータは、バルクデータ移動で呼び出される ASCII でエンコードされたデータがサブシステムに含まれる場合にのみ指定します。PowerExchange は、これらの ASCII コードページを、EBCDIC_CCSID パラメータで定義されている同等の EBCDIC コードページにマップします。また、EBCDIC_CCSID パラメータを含みます。

各サブパラメータの有効な値は 1~65534 です。値 65534 は、CCSID がいないことを示しています。

インストール中に変更しない限り、z/OS Installation Assistant では、値 850、65534、65534 が挿入されます。

```
UNICODE_CCSID=({sbcsc_ccsid|367},{graphic_ccsid|1200},{mixed_ccsid|1208})
```

オプション。Unicode シングルバイト、グラフィック、およびミックスデータの CCSID。PowerExchange は、これらの Unicode コードページを、EBCDIC_CCSID で定義されている同等の EBCDIC コードページにマップします。また、EBCDIC_CCSID パラメータを含みます。

各サブパラメータの有効な値は 1~65534 です。値 65534 は、CCSID がいないことを示しています。

デフォルト値の 367,1208,1200 は、DB2CODEPAGE 文を指定しない場合、または z/OS Installation Assistant の値を変更せずに受け入れた場合に使用されます。通常は、これらのデフォルト値が適しています。

注: DB2 バージョン 8.1 以降の場合、DB2 カタログテーブルは、CCSID 1208 に対応する UTF-8 で格納されます。

```
PLAN_CCSID=({sbcsc_ccsid|037},{graphic_ccsid|037},{mixed_ccsid|037})
```

オプション。EBCDIC のシングルバイト、グラフィック、およびミックスのデータのために、EBCDIC_CCSID パラメータの CCSID の代わりに使用する CCSID。EBCDIC コードページを他の EBCDIC コードページにリダイレクトする必要がある場合は、このパラメータを使用します。

例えば、このパラメータは以下のような状況で実行できます。

- EBCDIC_CCSID コードページには、PowerExchange が ICU ベースのコードページ変換に使用できる ICU 変換テーブルはありません。
- EBCDIC_CCSID コードページは、DB2 サブシステム用に定義されたデフォルトのコードページと一致しますが、特定のソーステーブルまたはターゲットテーブルの EBCDIC コードページとは異なります。

デフォルトは 037,037,037 です。

```
REMAPR=(current_data_ccsid,remapped_data_ccsid)
```

オプション。DB2TRANS=R を指定すると、DB2 でこのパラメータを使用して、データが保存されているコードページを指定した別のコードページに再マップできます。例えば、DB2 プランがバインドされているコードページにマップされない ASCII データがあり、ICU コンバータがない場合は、このパラメータを使用して、ASCII コードページをサポートされている EBCDIC コードページに再マップします。

代わりに、DB2TRANS=N を指定すると、DB2 はデータを変換または再マップしません。ただし、PowerExchange は、REMAP n 文を使用して、不正なコードページを正しいコードページに置き換えることができます。例えば、DB2 カタログで定義されているコードページと一致しないデータコードページを DB2 がレポートすることがあります。これは、多くの場合、データが正しくロードされなかったことが原因です。この場合は、REMAP n パラメータに正しいコードページを指定できます。

DB2CODEPAGE 文では、それぞれ異なる DB2 テーブルに対して最大 6 つの REMAP n パラメータを指定できます。それぞれのパラメータ名が一意になるように、名前の最後の n 値をインクリメントします。

使用上の注意:

- DB2CODEPAGE 文が存在しない DB2 サブシステムに PowerExchange リスナが接続されている場合、PowerExchange は PowerExchange リスナのコードページを使用します。
- PowerExchange のインストール中に、バインドジョブは、PowerExchange の DB2 プランを EBCDIC として、CCSID を指定することなくバインドします。その結果、PowerExchange は、作成時に DB2 サブシステム用に定義されたデフォルトのアプリケーションプログラミング CCSID を使用します。PowerExchange は、これらの CCSID を DB2 カタログテーブルから取得し、DB2CODEPAGE パラメータとともに使用して使用するコードページを決定します。
- DB2CODEPAGE 文に定義した値は、アプリケーションプログラミングデフォルトパネルの DSNTIPF で DB2 サブシステム用に指定した値と一致する必要があります。DB2 でグラフィックおよびミックス CCSID を指定しなかった場合は、DB2CODEPAGE 文の *graphic_ccsid* および *mixed_ccsid* パラメータに 65534 を指定します。

DB2 サブシステム用に指定した CCSID が DB2CODEPAGE のデフォルト CCSID とは異なる場合は、DB2CODEPAGE 文を編集して DB2 CCSID と一致させる必要があります。

以下の表に、DB2CODEPAGE パラメータに対応する DSNTIPF パネルおよび DSNHDECP ロードモジュールの DB2 設定オプションを示します。

DB2CODEPAGE パラメータ	DSNTIPF フィールド	DSNHDECP パラメータ
EBCDIC_CCSID	EBCDIC CCSID	SCCSID (シングルバイト)、MCCSID (ミックス)、GCCSID (グラフィック)
ASCII_CCSID	ASCII CCSID	ASCCSID (シングルバイト)、AMCCSID (ミックス)、AGCCSID (グラフィック)
UNICODE_CCSID	UNICODE CCSID	USCCSID (シングルバイト)、UMCCSID (ミックス)、UGCCSID (グラフィック)
MIXED	MIXED DATA	MIXED

CCSID 値とその意味に関する詳細については、IBM の『*DB2 for z/OS Installation Guide*』(DB2 バージョン用)を参照します。

- z/OS Installation Assistant の[DB2 パラメータ]ページで[詳細パラメータ]をクリックすると、以下のパラメータの CCSID を入力することができます。
 - DB2CODEPAGE_ASCII_DBCS_CCSID
 - DB2CODEPAGE_ASCII_MIXED_CCSID
 - DB2CODEPAGE_ASCII_SBCS_CCSID
 - DB2CODEPAGE_EBCDIC_DBCS_CCSID

- DB2CODEPAGE_EBCDIC_MIXED_CCSID
- DB2CODEPAGE_EBCDIC_SBCS_CCSID
- DB2CODEPAGE_MIXED
- DB2CODEPAGE_UNICODE_DBCS_CCSID
- DB2CODEPAGE_UNICODE_MIXED_CCSID
- DB2CODEPAGE_UNICODE_SBCS_CCSID

これらのインストールパラメータは、DBMOVER ファイルの DB2CODEPAGE 文内に、EBCDIC_CCSID、ASCII_CCSID、および UNICODE_CCSID パラメータを追加します。これらは必要に応じて、DBMOVER ファイル内で編集することができます。

- DB2CODEPAGE パラメータが定義されていない場合、PowerExchange は、UNICODE_CCSID および EBCDIC_CCSID パラメータを含む DB2CODEPAGE の最小仕様を自動生成します。DB2 カタログテーブルは Unicode のエンコードを使用するため、UNICODE_CCSID が含まれます。
- DB2CODEPAGE 文は、バルクデータ移動操作に対してのみ適用されます。DB2 CDC の場合、PowerExchange は常に DB2 データのネイティブコードページを使用します。
- DB2CODEPAGE 文は CODEPAGE 文に影響しません。DB2DEF_ENCODING 文を使用する場合、状況によっては、DB2CODEPAGE 文を編集して、DB2DEF_ENCODING オプションに対応する ASCII_CCSID、EBCDIC_CCSID、または UNICODE_CCSID パラメータで *mixed_ccsid* 値を 037 に設定する必要があります。
- PowerExchange が、シングルバイトサブシステムにあるテーブルの VARGRAPHIC データまたは LOB データを含むカラムを読み取ろうとすると、DB2CODEPAGE のデフォルト設定の DB2TRANS=P が有効である場合は、次の SQL エラーが発生することがあります。

-332 CHARACTER CONVERSION BETWEEN CCSID *from_ccsid* TO *to_ccsid* REQUESTED BY *reason_code* IS NOT SUPPORTED

このような場合は、DB2CODEPAGE 文で DB2TRANS=N を指定してください。

DB2DEF_ENCODING 文

DB2DEF_ENCODING 文は、DB2 データマップを作成するときに PowerExchange がエンコードのスキームを持たない DB2 カラムに割り当てるデフォルトのエンコードのスキームを定義します。

オペレーティングシステム: z/OS

データソース: DB2 for z/OS

関連する文: DB2CODEPAGE

必須: No

構文:

DB2DEF_ENCODING={A|E|U}

有効な値:

- **A**。ASCII エンコード。
- **E**。EBCDIC エンコード。
- **U**。Unicode エンコード。

デフォルトは E です。

使用上の注意:

- 複数の DB2 以降の結果として、エンコードのスキーム SYSIBM.SYSDATABASE テーブルで指定されていない場合があります。

- 一般的に、DB2 に定義するエンコードは、アプリケーションプログラミングデフォルトパネル [DSNTIPF] の [DEF ENCODING SCHEME] フィールドで定義します。ただし、混在するコードページに対して DSNTIPF で 65534 が指定されている場合は、DB2CODEPAGE 文を編集して同等の ASCII、EBCDIC、または Unicode パラメータの *mixed_ccsid* 値を 037 に設定します。037 の値は、PowerExchange が DB2 カタログテーブルを読み取るために必要です。

例えば、DB2DEF_ENCODING 文にデフォルト値の E を使用し、[DSNTIPF] パネルで EBCDIC 混在コードページに 65534 が指定されている場合は、DB2CODEPAGE 文で EBCDIC_CCsid パラメータの最後に位置するサブパラメータに 037 を指定します。

EBCDIC_CCsid=(*sbcsc_ccsid,graphic_ccsid,037*)

DB2ID 文

DB2ID 文は、PowerExchange がバルクデータ移動用に DB2 for z/OS ソースからのデータの処理に使用する、DB2 サブシステム、プラン、およびアクセス方式ロードモジュールを定義します。

オペレーティングシステム: z/OS

データソース: DB2 for z/OS

関連した文: DB2PLAN

必須: いいえ

構文:

```
DB2ID=(db2_subsystem
      [,plan]
      [,{module_name|DTLAMV8F}])
```

パラメータ:

db2_subsystem

DB2 サブシステム識別子。値はインストール時に入力されます。別の DB2 サブシステム用に DB2ID 文を作成している場合は、この値を入力する必要があります。

プラン

PowerExchange バルクデータ移動操作の DB2 プラン名です。デフォルトは、DB2PLAN 文からの値です。

{module_name|DTLAMV8F}

PowerExchange DB2 アクセス方式ロードモジュールです。次のオプションがあります。

- DEFAULT**。デフォルトのアクセス方式ロードモジュール、DTLAMV8F です。
- DTLAMV8F**。PowerExchange が DB2 の複数行 FETCH 文と INSERT SQL 文を使用して、一度に複数行のデータを処理するモジュール。

デフォルトは DTLAMV8F です。

使用上の注意:

- インストール時に指定されたサブシステム ID を含む、この文のすべてのパラメータにデフォルトを使用できる場合、DB2ID 文を指定する必要はありません。そうでない場合は、DB2ID 文を定義する必要があります。
- DBMOVER メンバごとに DB2ID を最大で 25 個指定することができます。

DB2PLAN 文

DB2PLAN 文は、PowerExchange がバルクデータ移動処理に使用する DB2 プランを定義します。

オペレーティングシステム: z/OS

データソース: DB2 for z/OS

関連した文: DB2ID

必須: いいえ

構文:

DB2PLAN={*plan*|DTLPL*vrn*}

値: *plan* 変数には、PowerExchange バルクデータ移動操作の DB2 プラン名を入力します。デフォルトは DTLPL*vrn* です。ここで、*vrn* は、PowerExchange のバージョン、リリース、および変更レベルを表す数値です。例えば PowerExchange 8.6.1 の場合、*vrn* 値は 861 です。PowerExchange 9.5.0 の場合は、*vrn* は 950 です。

使用上の注意:

- PowerExchange は、DB2PLAN 文からのプラン名を使用して、DB2ID 文で定義されていない DB2 サブシステム、または DB2ID 文でプラン名が指定されていない DB2 サブシステムにアクセスします。異なる DB2 サブシステムで PowerExchange の一意のプラン名を使用する場合、DB2ID 文でサブシステムとその一意のプラン名を定義します。
- インストールプロセス中、PowerExchange は、z/OS Installation Assistant で指定したプラン名を使用して、DBMOVER メンバと XIDDB210 バインドジョブをカスタマイズします。

MVSDDB2AF 文

MVSDDB2AF 文は、PowerExchange で DB2 バルクデータ移動操作に使用される DB2 for z/OS 接続機能を指定します。

オペレーティングシステム: z/OS

データソース: DB2 for z/OS

必須: No

構文:

MVSDDB2AF={CAF|RRSAF}

有効な値:

- CAF。** PowerExchange は、DB2 呼び出し接続機能（CAF）を使用して DB2 に接続します。
CAF を使用する場合、PowerExchange と DB2 は、DB2 リソースの認証に、PowerExchange Listener または Netport ジョブのユーザー ID を使用します。
- RRSAF。** PowerExchange は、DB2 Resource Recovery Service 接続機能（RRSAF）を使用して、DB2 に接続します。DB2 テーブルへのユーザーアクセスを制御するには、RRSAF を指定します。
RRSAF を使用すれば、PowerExchange は DB2 にサインオンでき、バルクデータ移動操作を要求したユーザーのユーザー ID を DB2 リソースの認証に使用できます。
RRSAF を使用するには、z/OS システムで RRS を設定し実行する必要があります。

デフォルトは CAF です。

使用上の注意: RRSAF を指定した場合、PowerExchange は RRSAF にサインオンするとき correlation-id および accounting-token に次の値を使用します。

correlation-id

PowerExchange は次のいずれかの値を使用します。

- 指定した場合、PowerExchange は PowerCenter Workflow Manager または Informatica Developer ツールの相関 ID 接続プロパティに指定された値を使用します。
- 相関 ID 接続プロパティのクライアントに値を指定しない場合、PowerExchange は PowerExchange Listener マシン上にある DBMOVER コンフィギュレーションファイルの SESSID 文で指定された値を使用します。
- クライアントの相関 ID 接続プロパティまたは PowerExchange Listener マシン上の DBMOVER コンフィギュレーションファイルに値を指定しない場合、PowerExchange は DETAIL のデフォルト値を使用します。

accounting-token

PowerExchange は、次のように 8 バイトのジョブ名、8 バイトのユーザー ID、および 6 バイトのタスク ID を連結します。

jobname||*user_id*||*task_id*

RDBMSINSRTDFLT 文

RDBMSINSRTDFLT 文は、PowerExchange が RDBMS の WITH DEFAULT クラスで定義するカラムにデフォルト値を使用するかどうかを制御します。

オペレーティングシステム: すべて

データソース: DB2、Microsoft SQL Server、および Oracle のターゲット

必須: いいえ

構文:

RDBMSINSRTDFLT={*N*|*Y*}

有効な値:

- **N**。PowerExchange は、WITH DEFAULT クラスで定義するカラムにデータを書き込むときに、PowerExchange デフォルトを使用します。
- **Y**。PowerExchange は、WITH DEFAULT クラスで定義するカラムにデータを書き込むときに、RDBMS デフォルトを使用します。

RDBMS がデフォルトを与えられるようにする句を使用してカラムを定義する必要があります。このようにしないと、SQL エラーが発生します。

デフォルトは N です。

SESSID 文

SESSID 文は、DB2 要求の DB2 相関 ID として使用されるデフォルト値を指定します。

オペレーティングシステム: z/OS

必須: 不要

構文:

SESSID=*correlation_id*

値: *correlation_id* に、最大 8 文字の文字列を入力します。

使用上の注意:

- DB2 要求の DB2 相関 ID のデフォルト値を指定するには、PowerExchange Listener マシン上で DBMOVER コンフィギュレーションファイルに SESSID 文を含めます。特定の DB2 for z/OS 接続に対してこの値をオーバーライドするには、PowerCenter Workflow Manager または Informatica Developer ツールで相関 ID 接続プロパティを指定します。
- クライアントの相関 ID 接続プロパティまたは PowerExchange Listener マシンの DBMOVER コンフィギュレーションファイルの SESSID 文にも値が指定されていない場合、PowerExchange は DETAIL のデフォルト値を使用します。
- SESSID 文は、PowerExchange が RRSF を使用して DB2 に接続する場合にのみ適用されます。RRSF 接続を使用する場合は、MVSDB2AF 文を DBMOVER コンフィギュレーションファイルに含めます。

DB2 リソースへのアクセスに必要な権限

PowerExchange では、DB2 for z/OS リソースにアクセスするには次の権限レベルが必要です。

- DB2 プランに対する EXECUTE 権限
- DB2 ソーステーブルに対する SELECT 権限
- DB2 ターゲットテーブルに対する INSERT 権限
- DB2 LOAD ユーティリティによってロードされる DB2 ターゲットテーブルに対する LOAD 権限
- 次の DB2 カタログテーブルに対する SELECT 権限
 - SYSIBM.SYSCHECKS
 - SYSIBM.SYSCHECKDEP
 - SYSIBM.SYSCOLUMNS
 - SYSIBM.SYSDATABASE
 - SYSIBM.SYSDATATYPES
 - SYSIBM.SYSDUMMY1
 - SYSIBM.SYSFOREIGNKEYS
 - SYSIBM.SYSINDEXES
 - SYSIBM.SYSKEYS
 - SYSIBM.SYSPACKAGE
 - SYSIBM.SYSPACKSTMT
 - SYSIBM.SYSPARMS
 - SYSIBM.SYSRELS
 - SYSIBM.SYSROUTINES
 - SYSIBM.SYSSEQUENCES
 - SYSIBM.SYSSEQUENCESDEP
 - SYSIBM.SYSSYNONYMS
 - SYSIBM.SYSTABLES
 - SYSIBM.SYSTABLESPACE
 - SYSIBM.SYSTRIGGERS

注: PowerExchange では、バルクデータ移動セッションのマッピングを作成する際に Designer またはデベロッパツールにメタデータをインポートするには、これらのテーブルに対する SELECT 権限が必要です。

また、PowerExchange ユーザーから特定のテーブルの DB2 データへのアクセスを制御するために DB2 が使用するユーザー ID は、RUNLIB ライブラリの DBMOVER 構成メンバの SECURITY 文および MVSD2AF 文の設定によって異なります。

以下の表に、代替の文の設定で使用するユーザー ID を示します。

MVSD2AF 設定	SECURITY 設定	DB2 へのアクセスに使用されるユーザー ID
CAF	0、1、または 2。	PowerExchange Listener を実行しているユーザー ID
RRSAF	0 または 1	PowerExchange Listener を実行しているユーザー ID
RRSAF	2	ユーザーアプリケーションを実行している MVS ユーザー ID

DB2 リソースにアクセスするための権限の付与

適切な権限を PowerExchange に付与して、PowerExchange DB2 プランを実行し、DB2 カタログ、DB2 ソーステーブル、および DB2 ターゲットテーブルにアクセスします。

DB2 リソースにアクセスするための権限を付与するには：

- ▶ 以下の SQL GRANT 文を発行します。

```
GRANT EXECUTE on PLAN db2_plan_name to user_id [for the DB2 plan]
```

```
GRANT SELECT on SYSIBM.SYSTABLES to user_id [for the DB2 catalog]
```

```
GRANT SELECT on SYSIBM.SYSCOLUMNS to user_id
```

```
GRANT SELECT on creator.table to user_id [for each DB2 source table]
```

```
GRANT INSERT on creator.table to user_id [for each DB2 target table]
```

説明:

- *db2_plan_name* は、DBMOVER 構成メンバの DB2PLAN 文で指定されている DB2 プラン名です。
- *user_id* は、PowerExchange Listener による DB2 ソースデータまたはターゲットデータへのアクセスに対するユーザー要求に関連付けられている、z/OS ユーザー ID または PowerExchange Listener ユーザー ID です。
- *creator.table_name* には、バルクデータ移動に関係している特定の DB2 ソーステーブルまたはターゲットテーブルを指定します。

関連項目：

- [「DB2 リソースへのアクセスに必要な権限」 \(ページ 93\)](#)

DB2 複数行 FETCH 文および INSERT 文

複数行の FETCH 文および INSERT 文を使用する場合は、DB2 により複数行のデータが一度に取り出されるか、または挿入されます。その結果、PowerExchange がデータベースにアクセスする回数が減り、バルクデータ移動のパフォーマンスが向上します。

PowerExchange がサポートする DB2 のすべてのバージョンで、PowerExchange は複数行の FETCH 文および INSERT 文を使用できます。

PowerExchange のデフォルトのアクセス方式モジュールである DTLAMV8F は、複数行の FETCH 文および INSERT 文を使用します。

バルクデータ移動操作で複数行の文を使用するには、以下の設定要件が満たされていることを確認してください。

- DBMOVER 構成メンバでは、DB2ID 文の 3 番目の位置のパラメータを DTLAMV8F または DEFAULT に設定し、複数行の FETCH 文および INSERT 文を使用します。DB2ID 文の詳細については、[「DB2ID 文」 \(ページ 90\)](#)を参照してください。
- PWXLSTNR メンバまたは STARTLST メンバで、PowerExchange リスナの開始済みのタスクまたはジョブの領域サイズを、複数行 FETCH および INSERT 処理のストレージ要件が増えた場合でも十分に対応できるサイズに設定します。
- **【配列サイズ】** 接続属性を一度に取得または挿入する行数に設定します。デフォルトは 25 です。

PowerExchange は、DB2ID 文の DB2 サブシステムのバージョンが複数行 SQL 文をサポートしていることを確認します。

次の条件がすべて当てはまるとき、PowerExchange は配列サイズを動的に削減します。

- データベースタイプが DB2。
- テーブルに LOB カラムが含まれている。
- **【配列サイズ】** の値が 1 よりも大きい。
- 行サイズ × **【配列サイズ】** の値が 16000000 バイトよりも大きい。

これらの条件を満たす場合、PowerExchange は配列サイズを減らして、クライアントおよび PowerExchange リスナマシンの両方でメッセージ PWX-00186 をログ記録します。

DB2 複数行の FETCH 文および INSERT 文の詳細については、IBM DB2 for z/OS のマニュアルを参照してください。

リレーショナルソース定義またはターゲット定義を含む DB2 for z/OS バルクデータ移動

DB2 for z/OS ソースデータでバルクデータ移動を実行するには、リレーショナルソースデータベース接続を使用します ([「非リレーショナルソース定義を含む DB2 for z/OS バルクデータ移動」 \(ページ 99\)](#)に記載された条件を除く)。

DB2 イメージコピーをデータソースとして使用するときは、十分な考慮を払う必要があり、特殊な手順が適用されます。詳細については、[「イメージコピーソースを含む DB2 for z/OS バルクデータ移動」 \(ページ 96\)](#)を参照してください。

DB2 for z/OS ターゲットでバルクデータ移動を実行するには、リレーショナルターゲットデータベース接続を使用します。

バルクデータ移動を定義する前に、以下の情報を収集します。

- DB2 サブシステム ID
- *creatortable_name* 形式の DB2 テーブル名。
- z/OS のユーザー ID およびパスワード (DBMOVER 構成メンバ内で SECURITY 文のセキュリティ設定に必要な場合)

最初に PowerExchange Navigator から DB2 データをプレビューする場合は、パーソナルメタデータプロファイルを作成し、そのプロファイルに対してデータベース行のテストを実行することができます。

PowerExchange Navigator には、テーブルの DB2 カラムごとにメタデータが表示されます。メタデータには、データ型、日付形式、CCSID などのカラム属性が表示されます。

リレーショナルソース定義またはターゲット定義を含む DB2 for z/OS バルクデータの移動

以下の手順を使用して、PowerCenter で DB2 for z/OS ソース定義またはターゲット定義をリレーショナルデータとしてインポートし、バルクデータ移動を実行します。

1. PowerCenter Designer で、DB2 データソースに **[ソース] - [PowerExchange からインポート]**、または、DB2 データターゲットに **[ターゲット] - [PowerExchange からインポート]** をクリックします。
2. [PowerExchange からインポート] ダイアログボックスで、**[ソースタイプ]** フィールドとして **[DB2zOS]** を選択します（ターゲットにも表示されます）。また、場所、DB2 サブシステム ID、およびデータベースアクセスに必要なユーザー ID やパスワードを入力します。必要に応じて、他のフィールドも実行します。
注: **[場所]** フィールドに、ローカルの PowerCenter システムにある dbmover.cfg ファイルの NODE 文で指定した PowerExchange Listener ノード名を入力します。
3. PowerCenter Designer で、DB2 for z/OS ソースまたはターゲットが含まれているマッピングを作成します。
4. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX DB2zOS リレーショナルデータベース接続を設定します。次に、バルクデータを移動するワークフローを開始します。

イメージコピーソースを含む DB2 for z/OS バルクデータ移動

DB2 テーブルスペースまたはテーブルスペースパーティションのフルイメージコピーを PowerExchange バルクデータ移動セッションのデータソースとして使用できます。フルイメージコピーを作成するには、DB2 COPY ユーティリティを使用します。

たとえば、既存の DB2 ソースに基づいてターゲットをマテリアライズするものとします。DB2 ソースのフルイメージコピーを取得できます。次に、ターゲットをマテリアライズする PowerCenter ワークフローの入力としてそのイメージコピーを使用します。ターゲットの実体化が完了した後、PowerExchange 変更データキャプチャ（CDC）を使用してソースとターゲットの同期を確保できます。

フルイメージコピーをデータソースとして使用するときは、データマップを作成する必要はありません。代わりに、DB2 テーブル用にインポートするリレーショナルソース定義を使用できます。

データソースとして FlashCopy イメージコピーまたはシーケンシャルイメージコピーを使用することもできます。FlashCopy イメージコピーを使用する場合、それが 1 つのテーブルが含まれており、廃止されたタイプのテーブルスペースではないテーブルスペース用であることを確認します。

PowerCenter セッションのプロパティを定義するときは、読み取るイメージコピーを制御するプロパティを設定します。以下のいずれかのイメージコピーを読み取るように DB2 イメージコピー用の PowerExchange Reader を設定できます。

- 最新の SHRLEVEL REFERENCE のフルイメージコピー
- 指定したイメージコピー
- 異なる DB2 サブシステムで作成されたイメージコピー

注意事項:

- 圧縮されたテーブルスペースのイメージコピーを作成する場合、解凍のディクショナリが取得され、正しい順序で構築されている場合にのみ、圧縮された行をイメージコピーデータセットからアンロードできます。

- PowerExchange でイメージコピーデータセットを正常に処理するためには、イメージコピーが作成された時点でのオブジェクトの状態が、テーブルとテーブルが含まれるテーブルスペースについて説明した DB2 カタログの情報として正確に反映されていなければなりません。DB2 では、ALTER 文をいくつか発行し、後で REORG ユーティリティを実行して実装（マテリアライズ）できます。このため、DB2 カタログ内の情報と、テーブルスペースの最新のイメージコピーのコンテンツの間に不一致が生じることがあります。イメージコピーをバルクデータ移動のソースとして使用する前に、そのイメージコピーの作成後、テーブルまたはテーブルスペースに対して ALTER 文が発行されていないことを確認する必要があります。

DB2 for z/OS イメージコピーの作成

バルクデータ移動セッションの入力としてフルイメージコピーを作成するには、JCL および制御カードを定義します。次に、DB2 COPY ユーティリティのジョブを手動でサブミットするか、またはジョブスケジューラを使用します。

フルイメージコピーを作成する JCL は DB2 DSNUTILB プログラムを実行します。DB2 DSNUTILB プログラムは制御カードを読み込むと、DB2 COPY ユーティリティを呼び出してフルイメージコピーを作成します。

各 COPY TABLESPACE 文で以下のパラメータを含めます。

- FULL YES
- SHRLEVEL REFERENCE

PowerExchange CDC を使用して、実体化後のターゲットに変更を書き込むようにする場合は、COPY TABLESPACE 文よりも前に QUIESCE TABLESPACE 文を含めます。QUIESCE TABLESPACE 文を含めると、PowerExchange は PowerExchange ロgger ログファイルにイベントマーカを設定します。CDC のリスタートトークンを作成するには、イベントマーカが必要です。リスタートトークンによって、PowerExchange 抽出に対する変更ストリームの開始ポイントが決まります。

リスタートトークン作成の詳細については、『PowerCenter 用の PowerExchange インタフェース』を参照してください。イメージコピー作成の詳細については、適切な IBM のマニュアルを参照してください。

データソースとしての圧縮イメージコピー

圧縮されたイメージコピーを PowerExchange バルクデータ移動のデータソースとして使用する場合、以下の要件および考慮事項が適用されます。

- イメージコピー圧縮解除では、ESA/390 ハードウェア命令 CMPSC のハードウェアサポートが必要です。お使いのハードウェアがこの手順をサポートしていることを確認します。PowerExchange zIIP Exploitation が有効になっている場合、圧縮解除で zIIP 処理が使用されます（使用できる場合）。

zIIP 処理を使用する場合、PowerExchange は圧縮された行の配列を zIIP にディスパッチして展開します。

【配列サイズ】 接続属性はディスパッチされる行数を制御します。

より大きな配列サイズに割り当てられた追加のメモリが有益で、サーバーのパフォーマンスが低下していないかどうかをテストおよび判断できる場合を除き、デフォルトの配列サイズ（25）を使用することをお勧めします。これらの判断ができ、かつ PowerExchange バルクデータ移動のデータソースとして圧縮イメージコピーを使用し、z/OS で zIIP 処理を有効にしている場合は、配列サイズを 500～1000 にすることをお勧めします。

- バルクデータ移動セッションで圧縮イメージコピーを使用すると、メモリの使用量が増加します。JCL の DTLST の手順で指定した REGION のサイズが PowerExchange リスナジョブを実行できる十分な大きさであることを確認します。
- PowerExchange は、COPY TABLESPACE 文で SYSTEMLPAGES NO オプションを指定して作成された圧縮イメージコピーをサポートしません。この種のイメージコピーをデータソースとして使用すると、PowerExchange は以下のメッセージを発行します。

PWX-09278 Compressed record encountered before dictionary complete.

場合によっては、SOC7 異常終了が発生します。

- PowerExchange のバルクデータ移動では、Db2 ハフマン圧縮によるイメージコピーソースの圧縮がサポートされます。ハフマン圧縮を使用することでディスク容量と I/O コストを削減できます。IBM は、Db2 バージョン 12 の機能レベル 504 から、テーブルスペースの圧縮に対して、z14 メインフレームでハフマン圧縮のハードウェア サポートを開始しました。PowerExchange では、z14 以前のシステムのハードウェアおよびソフトウェアを使用してハフマンで圧縮されたイメージコピーもサポートされます。

DB2 イメージコピーからのターゲットのマテリアライズ

DB2 ソーステーブルスペースのフルイメージコピーを作成し、それを使用してターゲットをマテリアライズするには、以下の手順を使用します。

1. フルイメージコピーを作成するための JCL および制御カードを作成します。
2. DB2 COPY ユーティリティのジョブをサブミットするには、手動で行うか、またはジョブスケジューラやバッチジョブを使用します。DB2 DSNUTILB プログラムは制御カードを読み込むと、DB2 COPY ユーティリティを呼び出してフルイメージコピーを作成します。
3. テーブルなどターゲットデータベース構造を作成します。
4. PowerCenter Designer でマッピングを作成します。イメージコピーを作成した DB2 テーブルをソースとして含め、手順 3 で作成したデータベースオブジェクトをターゲットとして含めます。
5. PowerCenter Workflow Manager で、接続、ワークフロー、およびセッションを作成します。

新しいアプリケーション接続を作成し、接続タイプとして **[PWX NRDB Batch]** を選択します。オフロード処理を有効にする必要がある場合は、**[オフロード処理]** 属性に **[事後フィルタ]** を選択します。

セッションを構成する場合は、Reader として **[DB2 イメージコピー用の PowerExchange Reader]** を選択し、接続の一覧から作成したばかりのアプリケーション接続を選択します。

イメージコピーを指定するには、次のいずれかのアクションを実行します。

- 最新のイメージコピーを読み込むには、Reader の **イメージコピーデータセット** 属性にデータセット名を入力しません。
- 最新以外のフルイメージコピーを読み込むには、Reader の **イメージコピーデータセット** 属性にそのイメージコピーのデータセット名を入力します。
- 異なる DB2 サブシステムで作成されたフルイメージコピーファイルを読み込むには、Reader の **イメージコピーデータセット** 属性にそのイメージコピーのデータセット名を入力し、**[整合性確認の無効化]** を選択します。

警告: [一貫性チェックを無効にする] プロパティを選択し、異なる DB2 サブシステム上で作成されたイメージコピーや以前の DB2 バージョンで作成されたイメージコピーを使用すると、次のような追加の制限が適用されます。

- イメージコピーの読み取りに使用されている DB2 サブシステムのカタログ内のテーブル定義は、イメージコピーが作成された際のテーブル定義を正しく反映している必要があります。
 - イメージコピーには、1 つのテーブルのみのデータが含まれている必要があります。
 - イメージコピーは、イメージコピーの読み取りに使用されたサブシステムと同じ DB2 バージョンのサブシステムで作成する必要があります。
6. イメージコピーデータをターゲットテーブルに移動するワークフローを開始します。

PowerExchange CDC を使用してターゲットを更新する場合は、QUIESCE TABLESPACE 手順からのイベントマーカーに基づいてリスタートトークンを設定する必要があります。リスタートトークンの設定の詳細については、『PowerCenter 用の PowerExchange インタフェース』を参照してください。

非リレーショナルソース定義を含む DB2 for z/OS バルクデータ移動

DB2 for z/OS データソースを非リレーショナルソースとして定義できます。次の DB2 for z/OS データソースでこの方法を使用します。

- DB2 アンロードファイル
- 複数のフィールドに分割するカラムを含む DB2 テーブル
- ソースデータを処理する式で 1 つ以上の PowerExchange 関数を使用するフィールドを定義する DB2 テーブル
- データタイプを変更するカラムを含む DB2 テーブル（CHARACTER として定義されたがバイナリデータを含むカラムなど）

PowerCenter で DB2 for z/OS データソースを非リレーショナルソースとして扱う場合は、PowerExchange Navigator でデータマップを作成して、PowerCenter にそのデータマップをインポートします。

バルクデータソースとしての DB2 アンロードファイル

PowerExchange バルクデータ移動のために、DB2 アンロードファイルをデータソースとして使用できます。まずアップロードファイルを作成してから、それを使用して PowerCenter にインポートできるデータマップを追加します。データマップは DB2UNLD アクセス方式を使用します。

DB2 アンロードファイルを作成するには、以下のいずれかの IBM または BMC ソフトウェアユーティリティを使用します。

- DB2 for z/OS オンライン REORG TABLESPACE ユーティリティ（UNLOAD EXTERNAL）
- DB2 for z/OS オンラインユーティリティ UNLOAD
- DB2 for z/OS のサンプルの DSNTIAUL アンロードプログラム
- BMC UNLOAD PLUS for DB2

BMC UNLOAD PLUS for DB2 ユーティリティを使用する場合、以下のユーティリティオプションがエラーとなる可能性があります。

- AUTOTAG YES オプションを使用すると、ユーティリティは各 SELECT 文の各出力レコードの先頭に 4 バイトのフィールドを追加します。アンロードファイルに基づいて作成する DB2UNLD データマップにはこの 4 バイトのフィールドが含まれないため、後続のフィールドが正しくマッピングされません。次に PWX バッチセッション用に PowerCenter にデータマップをインポートしてセッションを実行しようすると、セッションが失敗します。この問題を回避するには、AUTOTAG YES オプションを使用しないようにするか、または各データマップの先頭に 4 バイトのフィールドを手動で追加します。
- FIXEDVARCHAR NO オプションを使用する場合、PowerExchange Navigator がアンロードファイルにアクセスしてデータマップを生成しようとしたときに、マッピングエラーが発行されます。この問題を回避するには、FIXEDVARCHAR YES オプションを使用してアンロードファイルを生成します。

DB2 バージョン 11 REORG TABLESPACE ユーティリティと UNLOAD EXTERNAL を使用する場合、デフォルトでは NOPAD オプションが使用されます。このオプションにより、アンロードレコードの可変長カラムが、パディングなしの実際のデータの長さを持つことになります。PowerExchange Navigator がアンロードファイルにアクセスしてデータマップを生成しようとした場合、レコードマッピングエラーが発生します。この問題を回避するには、NOPAD NO オプションを使用してアンロードファイルを生成します。

アンロードファイルに基づくデータマップの追加の詳細については、『PowerExchange Navigator ユーザーガイド』を参照してください。.

非リレーショナルソース定義を含む DB2 for z/OS バルクデータの移動

非リレーショナルソース定義と共に DB2 for z/OS バルクデータを移動するには、以下の手順を実行します。手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を収集します。

- DB2 サブシステム ID
- *creatortable_name* 形式の DB2 テーブル名。
- データソースとして使用しているアンロードファイル名（該当する場合）。
- MVS のユーザー ID およびパスワード（DBMOVER 構成メンバ内で SECURITY 文のセキュリティ設定に必要な場合）

1. PowerExchange Navigator から、データマップを作成します。**[アクセス方式]** リストで、データソースがデータマップを必要とする DB2 テーブルの場合に、データソースが DB2 アンロードファイルまたは **[DB2]** のときは、**[DB2UNLD]** を選択します。データソースレコードおよびフィールドに関するメタデータをインポートするには、**[レコード定義のインポート]** オプションも選択します。

ヒント: **[レコード定義のインポート]** を選択しない場合は、後でメタデータをインポートできます。リソースエクスプローラでデータマップを開き、**[ファイル]** - **[コピーブックのインポート]** をクリックします。

データマップ作成の詳細については、『*PowerExchange Navigator ユーザーガイド*』を参照してください。

2. リソースエクスプローラでデータマップを開き、「レコード」または「テーブル」のビューをクリックして、メタデータが正常に取得されたことを確認します。
3. リモートの DB2 for z/OS システム上の PowerExchange Listener にデータマップを送信します。リソースエクスプローラでデータマップを選択し、**[ファイル]** - **[リモートノードへの送信]** をクリックします。この手順により、PowerExchange 抽出ルーチンでランタイム中にデータマップにアクセスすることができます。
注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。
4. メタデータのテーブルビューに対してデータベース行のテストを実行して、DB2 ソースデータベースからデータを返すことができることを確認します。
5. PowerCenter Designer で、PowerExchange データマップをインポートします。ソースタイプの場合、アンロードデータマップに **[DB2UNLD]**、または DB2 テーブルに **[DB2MAP]** を選択します。**[場所]** フィールドに、ローカル dbmover.cfg ファイルの NODE 文で指定した PowerExchange Listener ノード名を入力します。必要に応じて、他のフィールドも実行します。
6. マッピングを作成します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアプリケーション接続を設定します。次のいずれかの Reader がデータソースに応じて自動的に選択されます。

- PowerExchange Reader for DB2 Datamaps
- PowerExchange Reader for DB2 Unload Datasets

次に、バルクデータ移動を実行するワークフローを開始します。

DB2 LOAD ユーティリティを使用したバルクデータのロード

DB2 LOAD ユーティリティを使用すると、PowerExchange または PowerCenter がデータソースから読み込んだ DB2 テーブルに大量のデータを効率よくロードできます。PowerExchange には、DB2 LOAD ユーティリティを実行するための JCL テンプレートおよび制御カードテンプレートが用意されています。PowerCenter で、PowerExchange Client for PowerCenter (PWXPC) 接続を使用して、バルクデータ移動を実装する PowerExchange と通信します。

PWXPC 接続属性の構成方法に応じて、PowerExchange では DB2 LOAD ユーティリティの以下の機能を実行できます。

- DB2 LOAD ユーティリティの入力として使用するデータファイルのみを作成してください。この場合、PowerExchange の外部で DB2 LOAD ユーティリティを実行する必要があります。手動でジョブをサブミットできます。または、ジョブスケジューラあるいは PowerExchange DTLREXE ユーティリティを使用できます。
- 以下の項目を作成します。
 - LOAD ユーティリティの入力として使用するデータファイル
 - LOAD ユーティリティを実行するロードカード
 - LOAD ユーティリティを別のジョブとしてサブミットする JCL

次に、DB2 LOAD ユーティリティジョブをサブミットし、ジョブが完了するのを待ちます。ユーティリティを別のジョブとして実行するには、PWXPC の **【モードタイプ】** 接続属性を JOB に設定します。

注: Informatica では、DB2 LOAD ユーティリティを、**【モードタイプ】** に JOB を指定する接続属性を持つ別のジョブで実行することをお勧めします。

- データファイルとロードカードを作成します。次に、PowerExchange リスナのアドレス空間で DB2 LOAD ユーティリティを実行します。ユーティリティを PowerExchange リスナのサブタスクとして実行するには、**【モードタイプ】** 接続属性を TASK に設定します。

【モードタイプ】 に TASK を指定した接続を設定することにより、PowerExchange リスナアドレス空間で DB2 LOAD ユーティリティの単一インスタンスを実行することは可能ですが、JCL は、ジョブで DD 名を複数回使用することを禁止しています。PowerExchange リスナアドレス空間で DB2 LOAD ユーティリティの 2 番目のインスタンスを実行しようとする、DD 名が SYSDISC、SYSERR、SYSIN、SYSMAP、SYSPRINT、SORTOUT、SYSUT1 であるデータセットの動的割り当ては、それらの名前がすでに他の DB2 LOAD インスタンスで使用されているため、失敗します。

DB2 LOAD ユーティリティ用の PowerExchange テンプレート

PowerExchange には、DB2 LOAD ユーティリティを使用するバルクデータロードジョブ用に、JCL テンプレートおよび制御カードテンプレートが用意されています。

以下の表に、その各テンプレートを示します。

RUNLIB メンバ	説明
DB2LDCTL	パーティション化されていないテーブル用の制御カードテンプレート。
DB2LDCTP	パーティション化されているテーブル用の制御カードテンプレート。

RUNLIB メンバ	説明
DB2LDJCL	パーティション化されていないテーブル用の JCL テンプレート。
DB2LDJCP	パーティション化されているテーブル用の JCL テンプレート。

パーティション化されているテーブルスペースの DB2 テーブルにバルクデータをロードするには、DB2LDCTP メンバおよび DB2LDJCP メンバを使用します。パーティション化されていないテーブルスペースの DB2 テーブルにバルクデータをロードするには、DB2LDCTL および DB2LDJCL を使用します。

これらのテンプレートを編集して他のロードユーティリティを使用したり、新たに別のテンプレートを作成したりできます。

注: PowerExchange でこれらのテンプレートを使用してバッチバルクロードジョブを生成するには、そのテンプレートが含まれているメンバの名前を PWX DB2zOS リレーショナルデータベース接続の PWXPC 接続属性に指定する必要があります。

関連項目：

- [「DB2 LOAD ジョブ用の PWX DB2zOS 接続属性」 \(ページ 105\)](#)

構成手順の概要

DB2 バルクロード操作を実行するように PowerExchange を構成するには、以下の手順を実行します。

1. DBMOVER 構成メンバをカスタマイズする。この手順は必須。
2. DB2 LOAD ユーティリティ用に PowerExchange JCL テンプレートおよび制御カードテンプレートをカスタマイズする。この手順は必須。
3. PowerExchange Listener JCL をカスタマイズする。この手順は必須。
4. 必要の場合は、DB2 LOAD ユーティリティ用に Netport JCL をカスタマイズする。
5. DB2 LOAD ジョブ用の PWXPC 接続属性または ODBC パラメータを設定します。この手順は必須。

関連項目：

- [「DB2 LOAD ユーティリティのための DBMOVER 構成メンバのカスタマイズ」 \(ページ 103\)](#)
- [「バルクデータロード用の PowerExchange テンプレートのカスタマイズ」 \(ページ 103\)](#)
- [「PowerExchange リスナ JCL のカスタマイズ」 \(ページ 104\)](#)

DB2 LOAD ユーティリティのための DBMOVER 構成メンバのカスタマイズ

DBMOVER 構成メンバで、DB2 LOAD ユーティリティ用の PowerExchange JCL テンプレートおよび制御カードテンプレートが含まれているライブラリを特定するため、以下の文を設定します。

以下の表に、それらの文を示します。

文	説明	デフォルト
LOADJOBFILE	DB2 LOAD ユーティリティを使用するバルクデータ移動ジョブ用の JCL テンプレートが含まれている PDS ライブラリの名前。	デフォルトは PowerExchange RUNLIB ライブラリです。 注: デフォルトのテンプレートメンバ名は DB2LDJCL です。
LOADCTLFILE	DB2 LOAD ユーティリティを使用するバルクデータ移動ジョブ用の制御カードテンプレートが含まれている PDS ライブラリの名前。	デフォルトは PowerExchange RUNLIB ライブラリです。 注: デフォルトのテンプレートメンバ名は DB2LDCTL です。

DBMOVER 構成メンバの他の文については、以下の考慮事項を確認してください。

- **PC_AUTH=Y**。PowerExchange Listener の DBMOVER メンバに PC_AUTH=Y を指定した場合は、PowerExchange Listener の制御下でバルクロード操作を実行することはできません。代わりに、この文を含まない DBMOVER メンバを指定する Netport ジョブを使用して、バルクロード操作を実行します。
- **MVSDDB2AF=RRSAF**。MVSDDB2AF=RRSAF を指定した場合は、PWX DB2zOS リレーショナル接続パラメータの**モードの種類**に **JOB** を指定して、独立したバッチジョブとしてバルクロード操作を実行します。バルクロード操作を PowerExchange Listener のアドレス空間のタスクとして実行するには、MVSDDB2AF=CAF を指定する必要があります。
- **SUBMITTIMEOUT**。PowerExchange Listener でバルクロード操作にバッチジョブをサブミットする場合は、SUBMITTIMEOUT 文の値により、バッチジョブの実行が開始されるために十分な時間が提供されていることを確認します。
- **MAXTASKS**。バルクロード操作に Netport ジョブを使用した場合は、MAXTASKS 文の値が十分であることを確認します。Netport ジョブを使用してバルクロード操作を実行するたびに、MAXTASKS 値から 3 つのタスクを使用します。

バルクデータロード用の PowerExchange テンプレートのカスタマイズ

DB2 LOAD ユーティリティのデフォルトの JCL 文および制御カードを受け入れた場合、JCL テンプレートおよび制御カードテンプレートを修正なしで使用できます。制御カードを変更したり、他のロードユーティリティを使用したりする場合には、これらのテンプレートを編集できます。

RUNLIB データセットには、デフォルトの DB2LDJCL または DB2LDJCP テンプレートメンバが含まれています。**[JCL テンプレート]** 属性が、接続内のこのメンバを参照していることを確認してください。

JCL では、置換変数はパーセント記号 (%) で始まります。PowerExchange は DB2 LOAD ジョブを z/OS へサブミットする前に、定義された置換変数を適切な値を使って解決します。詳細については、「[ローダー JCL テンプレートの置換変数](#)」(ページ 109)を参照してください。

DBMOVER 構成メンバで SECURITY 文が 2 に設定されている場合は、JCL に USER=%USER および PASSWORD=%PWD が含まれていることを確認してください。

JCL の SYSDISC DD を次の変更元から次の変更先に変更することにより、破棄された更新レコードを短時間保持できます。変更元:

```
//SYSDISC DD UNIT=SYSDA,SPACE=(CYL,(100,100),RLSE)
```

変更先:

```
//SYSDISC DD DISP=SHR,DSN=%SYSDISC
```

この編集されたステートメントは、HLQ.basename.SYSDISC という形式の名前で破棄ファイルを作成します。

注: ベース名はタスク ID に由来します。同じタスクスロットで実行される次の DB2 LOAD ユーティリティジョブは、破棄ファイルを上書きします。

Informatica では、SYSPRINT ファイルからの DB2 LOAD ユーティリティ制御レポートを呼び出し元のセッションログに書き込むことをお勧めします。SYSPRINT ファイルをセッションログに書き込むには、**[PWX 上書き]** 接続属性で RETLOGINFOMSG=Y を設定します。

DB2 LOAD ユーティリティのための DD 文の要件の詳細については、IBM DB2 のマニュアルを参照してください。

PowerExchange リスナ JCL のカスタマイズ

次の情報を使用して、PowerExchange リスナ JCL をカスタマイズします。

DB2 LOAD ユーティリティが使用する DD 名が、リスナ JCL に表示されてはいけません。モードタイプが JOB である DB2 LOAD ユーティリティを実行する場合、必要な DD 名は別のジョブでサブミットされた JCL で定義され、リスナ JCL の DB2 LOAD ユーティリティの DD 名は使用されません。モードタイプが TASK（非推奨）である DB2 LOAD ユーティリティを実行する場合、必要な DD 名を持つデータセットは SVC99 を使用して動的に割り当てられます。DD 名がリスナ JCL に表示されている場合、SVC99 サービスは失敗します。

DB2 LOAD ユーティリティは、以下の DD 名を持つデータセットを使用します。

- **SYSDISC**。キーの重複や参照整合性エラーなど何らかの理由でロードされなかった、破棄された更新レコードのデータセット。
- **SYSERR**。エラー処理用の作業データセット。
- **SYSIN**。ロードする入力データレコード。
- **SYSMAP**。参照整合性処理用の作業データセット。
- **SYSPRINT**。DB2 LOAD ユーティリティからの制御レポート。これには、制御カウントと破棄された更新の警告が含まれます。**[PWX 上書き]** 接続属性で RETLOGINFOMSG=Y オプションを指定すると、レポートをセッションログに書き込むことができます。
- **SORTOUT**。キー用の作業データセット。
- **SYSUT1**。キー用の作業データセット。

PowerCenter における DB2 バルクデータロードセッションの構成

DB2 LOAD ユーティリティを使用するバルクデータロードを制御するには、PowerCenter で、関連する PWX DB2zOS リレーショナルデータベースの接続属性を、PowerCenter のバルクデータ移動セッション用に設定する必要があります。

PowerExchange ODBC ドライバを使用する場合は、odbc.ini ファイル内の関連する ODBC パラメータを設定します。

DB2 LOAD ジョブ用の PWX DB2zOS 接続属性

以下の表に、DB2 LOAD ユーティリティを使用するバルクデータロード操作を制御するために設定できる、各接続属性を示します。

PWXPC 接続属性	説明	有効な値
バルクロード	選択すると、PowerExchange で DB2 LOAD ユーティリティを使用して DB2 for z/OS ターゲットにデータをロードします。	デフォルトは [いい] です。
CTL テンプレート	PowerExchange ターゲットシステムの RUNLIB ライブラリにある、DB2 LOAD ユーティリティ用の制御ファイルテンプレートのメンバ名です。	デフォルトは DB2LDCTL です。
JCL テンプレート	PowerExchange ターゲットシステムの RUNLIB ライブラリにある、DB2 LOAD ユーティリティ用の JCL テンプレートのメンバ名です。	デフォルトは DB2LDJCL です。
ロードオプション	DB2 LOAD ユーティリティバルクデータロード操作に LOAD REPLACE 制御カードを使用するかどうかを示すオプション。	<p>次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> - INSERT。現在テーブルスペースまたはテーブルスペースパーティションにあるデータに、新たにロードするデータを追加するには、このオプションを選択します。LOAD RESUME YES 制御カードに対応しています。 - REPLACE。現在テーブルスペースまたはテーブルスペースパーティションにあるデータを、新たにロードするデータで置き換えるには、このオプションを選択します。LOAD REPLACE 制御カードに対応しています。 <p>デフォルトは INSERT です。</p>

PWXPC 接続属性	説明	有効な値
モードの時間	DB2 LOAD ユーティリティジョブの実行に関する待機期間。	<p>次のいずれかの値を入力します。</p> <ul style="list-style-type: none"> - WAIT。ジョブが終了するのを待ってから、PowerCenter に制御を返します。[モードの種類] が JOB または TASK の場合にのみ、このオプションを指定します。 - NO WAIT。ジョブの終了を待たずに PowerCenter に制御を返します。[モードの種類] が JOB または NOSUBMIT の場合にのみ、このオプションを指定します。 - TIMED。TIME 接続属性に指定されている秒数だけ待ってから、PowerCenter に制御を返します。[モードの種類] を JOB に指定した場合にのみ、このオプションを指定します。 - DATAONLY。データファイルのみを作成します。DB2 LOAD ユーティリティを実行するファイルや JCL は作成しません。このオプションは通常、[モードの種類] が [NOSUBMIT] の場合に使用します。 <p>デフォルトは WAIT です。</p> <p>注意事項:</p> <ul style="list-style-type: none"> - WAIT を入力した場合、PWX オーバーライド接続属性の TCPIP_OP_TIMEOUT パラメータで指定される値とは関係なく、PowerExchange により操作に対するネットワーク操作のタイムアウトが無効になります。TIMED を入力した場合、PowerExchange により TIME 接続属性の値に 300 秒が追加されます。 - WAIT を入力し、ワークフローの実行時にサブミットされたジョブが失敗した場合、z/OS システム上の PowerExchange リスナは実行し続けます。リスナを停止するには、次のいずれかの操作を実行します。 <ul style="list-style-type: none"> - z/OS オペレータコンソールで次のコマンドを入力します。 <pre>F task_name,STOPTASK TASKID=task_id</pre> <p>詳細については、『PowerExchange コマンドリファレンス』の「STOPTASK コマンド」を参照してください。</p> - PowerExchange Navigator の [データベース行のテスト] ダイアログボックスに STOPTASK コマンドを入力します。[DB の種類] リストから [TASK_CNTL] を選択し、[Fetch] ボックスで [タスクの停止] を選択します。 [SQL Statement] ボックスには stoptask taskid=が表示されます。タスク ID を入力します。 <p>詳細については、『PowerExchange Navigator ユーザーガイド』の「データベース行のテスト内の PowerExchange リスナコマンドの発行」を参照してください。</p>

PWXPC 接続属性	説明	有効な値
モードの種類	PowerExchange が DB2 テーブルにデータをロードするために DB2 LOAD ユーティリティを実行するときのモード。	次のいずれかの値を入力します。 <ul style="list-style-type: none"> - タスク。LOAD ユーティリティを PowerExchange リスナのサブタスクとして実行します。 - JOB。DB2 LOAD ユーティリティを実行する独立したジョブをサブミットします。 - NOSUBMIT。[モードの時間] が [DATAONLY] の場合を除き、DB2 LOAD ユーティリティを実行するファイルや JCL を作成します。ロードジョブはサブミットしません。ジョブを手動でサブミットする必要があります。 デフォルトは TASK です。
時間	モードの種類に JOB、モードの時間に TIMED を選択したときの待機時間（秒単位）です。	有効な値は 1～99998 です。 デフォルトは 0 です。
ファイル名	PowerExchange で DB2 LOAD ユーティリティのバルクデータロード操作の一時ファイルを作成するために使用する、データセット名または上位修飾子です。 詳細については、「 DB2 バルクロード操作のためのデータセット名 」（ページ 108）を参照してください。	PowerExchange でパーティション化されているテーブルの 1 つのパーティション用にデータファイルを作成する必要がある場合は、以下の構文を使用します。 <i>filename/partxxxx</i> <i>xxxx</i> 変数はパーティション番号です。
一時ファイルの削除	DB2 LOAD ユーティリティで DB2 テーブルにデータをロードする際に作成される一時ファイルを PowerExchange で処理する方法です。	次のいずれかの値を選択します。 <ul style="list-style-type: none"> - NO では、一時ファイルは削除されません。 - BEFORE では、ユーティリティの実行前に一時ファイルが削除されます。 - AFTER SUCCESS ONLY では、ユーティリティが戻りコード 0 で終了すると、ユーティリティの実行後に一時ファイルが削除されます。 - AFTER では、ユーティリティの実行後に一時ファイルが削除されます。 デフォルトは [いいえ] です。
PWX オーバーライド	セミコロンで区切られた、PowerExchange 接続オーバーライド。	失敗した LOAD ジョブからの出力や DB2 LOAD 拒否データセットの名前などの情報メッセージを PowerCenter セッションログに書き込むには、以下の上書きを指定します。 RETLOGINFOMSG=Y 拒否ファイルの名前は、PWX-00409 メッセージに表示されます。 デフォルトでは、PWXPC はセッションログに PowerExchange のエラーメッセージと警告メッセージを書き込みますが、情報メッセージは書き込みません。

注: DB2 バルクデータロード操作の場合、書き込みモード接続属性に書き込み確認オンまたは書き込み確認オフを指定します。

PowerCenter における PWX DB2zOS リレーショナル接続の詳細については、『PowerCenter 用の PowerExchange インタフェース』を参照してください。

DB2 バルクロード操作のためのデータセット名

PowerExchange により、DB2 LOAD ユーティリティのバルクデータロード操作の一時ファイルが作成されます。生成されるデータセットとその名前は、選択する【モードの時間】および【モードの種類】接続属性によって異なります。

後に続くデータセット名の説明には、次の変数が含まれます。

- *filename*: 【ファイル名】 接続属性に指定する上位修飾子またはデータセット名。
- *nnnn*: タスクの PID。
- *xxxx*: 次の構文を使用して 【ファイル名】 接続属性に指定するパーティション番号。 *filename/partxxxx*。

次の表では【モードの種類】 接続属性で【JOB】を選択した場合に PowerExchange が作成するファイルについて説明します。

ファイル名	説明
<i>filename.DTLnnnnnn</i> または <i>filename.PRNxxxx</i>	パーティション化されていないテーブルにロードする DB2 LOAD ユーティリティのデータを含むファイル または パーティション化されたテーブルにロードする DB2 LOAD ユーティリティのデータを含むファイル
<i>filename.DTLnnnnnn.CTL</i>	LOAD コマンドを含むファイル
<i>filename.DTLnnnnnn.SQL</i>	CREATE TABLE 文の SQL テンプレートを含むファイル
<i>filename.DTLnnnnnn.SYSPRINT</i>	バルクロードジョブの出力を含むファイル

次の表では【モードの種類】 接続属性で【NOSUBMIT】を選択した場合に PowerExchange が作成するファイルについて説明します。

ファイル名	説明
<i>filename</i> または <i>filename.PRNxxxx</i>	パーティション化されていないテーブルにロードする DB2 LOAD ユーティリティのデータを含むファイル または パーティション化されたテーブルにロードする DB2 LOAD ユーティリティのデータを含むファイル
<i>filename.CTL</i>	LOAD コマンドを含むファイル
<i>filename.SQL</i>	CREATE TABLE 文の SQL テンプレートを含むファイル
<i>filename.SYSPRINT</i>	バルクロードジョブの出力を含むファイル

次の表では【モードの時間】 接続属性で【DATAONLY】を選択した場合に PowerExchange が作成するファイルについて説明します。

ファイル名	説明
<i>filename</i> または <i>filename.PRNXxxx</i>	パーティション化されていないテーブルにロードする DB2 LOAD ユーティリティのデータを含むファイル または パーティション化されたテーブルにロードする DB2 LOAD ユーティリティのデータを含むファイル

DB2 バルクロード操作のためのユーティリティ ID

バルクロード操作のユーティリティ ID を生成するために、JCL テンプレートには以下のパラメータが含まれています。

PWX%N5

PowerExchange は、ジョブを z/OS ヘサブミットする前に、%N5 を増分する 5 桁の数値に置き換えます。この方法より、ユーティリティ ID でジョブを一意に識別できます。

ローダー JCL テンプレートの置換変数

JCL テンプレート DB2LDJCL および DB2LDJCP には、置換変数が含まれています。PowerExchange は、ジョブを z/OS ヘサブミットする前に、定義された置換変数を適切な値を使って解決します。

以下の表で、各変数について説明します。

変数	説明
%ENDJOB	DTLNLS の 16 文字の入力文字列。PowerExchange リスナは、ジョブが終了したかどうかを判断できます。
%JOBCLASS	ローダーパラメータで指定されたジョブクラスに置き換えられます。
%JOBFILE	ジョブの作成に使用される JCL スケルトンを含むファイルの名前に置き換えられます。
%N <i>n</i>	増分する 1 から 7 の有効桁数の数値に置き換わります。
%PARTDDN	パーティションの入力データの DDNAME を生成します。PAR <i>nnnn</i> という形式となり、 <i>nnnn</i> はパーティション番号を表します。スケルトン JCL には、この変数の複数のインスタンスを含めることができます。未使用のインスタンスは PRT <i>nnnn</i> に設定されます。
%PN <i>n</i>	ロードがパーティション化されたテーブル用の場合のみ、1 から 4 の有効桁数のパーティション番号に置き換わります。パーティション化されていないテーブルのローダー JCL スケルトンでは、使用しないでください。
%PWD	JOB 文でパスワードまたはパスフレーズを指定します。PowerExchange リスナがセキュリティレベル 2 で実行され、Db2 LOAD ジョブが PowerExchange リスナジョブとは異なる資格情報で実行される場合、PowerExchange はこのパスワードまたはパスフレーズを使用します。z/OS は、パスワードまたはパスフレーズを表示しません。 テンプレートは、%PWD を JOB 文の最後の部分として定義する必要があり、次の仕様を含む別の行にある必要があります。 // PASSWORD=%PWD 詳細については、『PowerExchange リファレンスマニュアル』を参照してください。

変数	説明
%SSID	ロードターゲットの DB2 サブシステム ID。
%STRTJOB	DTLNLS の 16 文字の入力文字列。PowerExchange リスナは、ジョブが開始されたかどうかを判断できます。
%SYSDISC	破棄されたレコードが書き込まれるファイルの名前を置換します。
%SYSIN	ローダー制御文を含むファイルの名前に置き換わります。
%SYSPRINT	ローダーメッセージが書き込まれるファイルの名前に置き換わります。
%SYSREC	入力データの DSNAME に置き換わります。パーティション化されたロードの未使用のインスタンスの場合、この値は NULLFILE に設定されます。
%USER	ユーザー ID。
%UID	JOB 文でユーザー ID を指定します。PowerExchange リスナがセキュリティレベル 2 で実行され、Db2 LOAD ジョブが PowerExchange リスナジョブとは異なる資格情報で実行される場合、PowerExchange はこのユーザー ID を使用します。

DB2 LOAD ユーティリティオプション

【モードの時間】 接続属性に **[DATAONLY]** を選択した場合を除き、PowerExchange は DB2LDCTL または DB2LDCTP 制御カードテンプレートを使用して DB2 LOAD ユーティリティのロード制御ファイルを生成します。**【モードの時間】** 接続属性に **[DATAONLY]** を選択する場合、ロード制御ファイルを指定する必要があります。

PowerExchange は、パーティション化されていないテーブル用に次のロードオプションを生成します。

```
LOAD DATA RESUME NO REPLACE INTO TABLE table_name
```

または

```
LOAD DATA RESUME YES INTO TABLE table_name
```

PowerExchange は、パーティション化されたテーブル用に次のロードオプションを生成します。

```
LOAD DATA INTO TABLE table_name PART n INDDN PAR0000n REPLACE
```

または

```
LOAD DATA INTO TABLE table_name PART n INDDN PAR0000n RESUME YES
```

単一のパーティションおよび複数のパーティションへのバルクロード

PowerExchange で DB2 LOAD ユーティリティを使用して、データをテーブルパーティションにロードできます。パーティション化されていないテーブルと同様に、PowerExchange では LOAD ユーティリティを実行するように JCL およびロードカードをカスタマイズし、ロード対象のデータを含むデータファイルを生成し、ターゲットにデータをロードするバルクロードジョブを自動的にサブミットできます。

ただし、PowerExchange では、複数のテーブルパーティションにデータをロードするための単一のバルクロードジョブを生成してサブミットすることはできません。単一のロードジョブで複数のパーティションに書き込むために、パーティションごとに PowerCenter で個別ターゲットを定義し、接続属性を定義してターゲットごとにロードする DB2 LOAD ユーティリティ用のデータファイルのみを作成できます。ローダー制御ファイルおよび各パーティションにデータをロードする単一の DB2 LOAD ジョブ用の JCL を提供できます。PowerCenter ワークフローでは、データがパーティションごとのデータファイルに書き込まれた後、

DTLREXE を使用して DB2 LOAD ジョブをサブミットする個別の PowerCenter コマンドを含めることができます。または、ジョブスケジューラなどの機能を使用して PowerCenter 外のジョブをサブミットすることもできます。

第 8 章

IDMS バルクデータ移動

この章では、以下の項目について説明します。

- [IDMS バルクデータ移動の概要, 112 ページ](#)
- [IDMS に関する PowerExchange の構成, 112 ページ](#)
- [IDMS バルクデータの移動, 118 ページ](#)

IDMS バルクデータ移動の概要

PowerExchange では、z/OS システム上の CA IDMS ソースからバルクデータを読み取ることができます。一方、IDMS ターゲットにバルクデータを書き込むことはできません。

PowerExchange は、IDMS をリレーショナルでない DBMS として扱います。このため、PowerExchange Navigator から IDMS データソースのデータマップを作成する必要があります。PowerExchange は、そのデータマップを基に、IDMS ソースデータおよびメタデータにアクセスして、処理対象のソースレコードのリレーショナルビューを作成します。

IDMS データベースは z/OS システムで動作するため、PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムからリモートとなります。したがって、そのリモート z/OS システムで追加の PowerExchange Listener を実行し、PowerExchange および PowerCenter でリモート z/OS システムと通信できることを確認する必要があります。

また、DBMOVER コンフィギュレーション文の設定方法に応じて、PowerExchange では、IDMS ロードライブラリまたはそのロードライブラリの PowerExchange コピーを APF 許可する必要があります。

IDMS に関する PowerExchange の構成

バルクデータ移動操作用に PowerExchange を構成するには、PowerExchange 管理者が以下のタスクを実行します。

- リモートシステム上の PowerExchange Listener への接続を設定し、テストします。
- IDMS ソースまたは IDMS ターゲットシステム上の DBMOVER 構成メンバに IDMS 固有の文を定義することにより、IDMS バルクデータ移動を構成します。
- IDMS データにアクセスするためのセキュリティ要件を評価し、それに応じて PowerExchange を構成します。
- DBMOVER 構成メンバで SECURITY 文の最初のパラメータを 2 に設定している場合は、IDMS バルクデータ移動用の Netport ジョブを設定します。

PowerExchange for IDMS Security

セキュリティ要件は、主に z/OS システム上の DBMOVER コンフィギュレーションメンバでの SECURITY 文の最初のパラメータの設定方法により異なります。

SECURITY 文の最初のパラメータを 1 または 2 に設定した場合、PowerExchange Listener を APF 許可して実行する必要があります。しかし、IDMS ロードライブラリは通常、APF 許可されていません。この状況に対処するには、以下の方法のいずれかを使用します。

- PowerExchange インストール中に XIDIDM10 ジョブを実行し、XIDIDM10 ジョブが作成した IDMS ロードライブラリのコピーを APF 許可します。次に、これらのライブラリが PowerExchange リスナの STEPLIB DD 文に指定されていることを確認します。
- z/OS システム上の DBMOVER 構成メンバで PC_AUTH 文を Y に設定します。このように設定すると、PowerExchange で、z/OS プログラム呼び出し (PC) サービスルーチンを使用して PowerExchange リスナで必要とされる APF 許可を取得ができます。

注: DBMS ロードライブラリを STEPLIB DD 文ではなく DTLLOAD DD 文に定義する必要があります。
Netport ジョブを使用してデータにアクセスする場合は、ロードライブラリを Netport JCL の DTLLOAD DD 文に定義します。

IDMS ロードライブラリのコピーの保持および APF 許可を必要としない場合には、この方法を使用します。

SECURITY 文の最初のパラメータを 2 に設定した場合は、セキュリティチェックを正しく実行できるように IDMS バルクデータ要求ごとに Netport ジョブを設定する必要があります。

SECURITY 文の最初のパラメータを 0 に設定した場合は、PowerExchange Listener を APF 許可して実行する必要はありません。PowerExchange Listener JCL の STEPLIB DD 文に、IDMS ロードライブラリの PowerExchange コピーではなく、IDMS ロードライブラリを指定できます。

注: SECURITY 以外の DBMOVER 構成文の中には、たとえば STATS のように、一部の機能を APF 許可する必要があるものもあります。

関連項目：

- [「IDMS ロードライブラリのコピーの APF 許可」 \(ページ 116\)](#)
- [「APF 許可アクセスに対する z/OS プログラム呼び出しサービスルーチンの使用」 \(ページ 117\)](#)
- [「IDMS バルクデータ移動用の Netport ジョブの設定」 \(ページ 117\)](#)

リモートの IDMS ソースへの接続の設定およびテスト

リモート z/OS システム上の IDMS データソースにアクセスするには、PowerExchange および PowerCenter でリモートシステム上の PowerExchange Listener と通信する必要があります。

リモートの IDMS ソースへの接続を設定およびテストする手順

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用してリモートシステムへのネットワークアクセスをテストします。
2. 各 PowerExchange または PowerCenter システム上の dbmover.cfg ファイルに以下の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを特定します。
`NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)`
ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。
3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER 構成メンバへの IDMS 固有の文の追加

IDMS バルクデータ移動を構成するには、IDMS ソースまたはターゲットシステム上の DBMOVER 構成メンバに、以下の IDMS 固有の文を含めることができます。

- LOADJOBFILE
- PC_AUTH
- TEMPHLQ

DBMOVER 構成メンバを編集した後、変更を有効にするには PowerExchange Listener をリスタートする必要があります。

注: UNIT および VOLSER 文を設定する場合、IDMS メタデータ用の一時データセットを IDMSMJCL ジョブで割り当てることができるようにする値を入力します。

各 DBMOVER 文の説明については、『*PowerExchange リファレンスマニュアル*』を参照してください。

LOADJOBFILE 文

LOADJOBFILE 文は、DB2 for z/OS LOAD ユーティリティおよび CA IDMS/DB メタデータ取得バッチジョブの JCL テンプレートメンバを含んだ PDS データセットを指定します。

オペレーティングシステム: z/OS

データソース: CA IDMS/DB および DB2 for z/OS

関連した文: SUBMITTIMEOUT

必須: No

構文:

LOADJOBFILE={*pds_name*|A}

値: *pds_name* 変数には、DB2 for z/OS LOAD ユーティリティおよび CA IDMS/DB メタデータ取得バッチジョブの JCL テンプレートメンバを含んだ PDS データセットを入力します。DB2 LOAD ユーティリティ操作の場合、PowerExchange は、バルクロードを実行するシステムでこのデータセットを読み取ります。デフォルトは A です。

使用上の注意:

- PowerExchange をインストールすると、z/OS Installation Assistant により、DBMOVER メンバの LOADCTLFILE 文に LOADJOBFILE データセット名が含まれます。
- RUNLIB には、次の JCL テンプレートメンバが用意されています。
 - **DB2LDJCL**。パーティション化されていないテーブル用の DB2 LOAD ユーティリティジョブの JCL のサンプル。
 - **DB2LDJCP**。パーティション化されているテーブル用の DB2 LOAD ユーティリティジョブの JCL のサンプル。
 - **IDMSMJCL**。CA IDMS/DB メタデータ取得の JCL のサンプル。
 - **IDMSMJCX**。サブスキーマロードモジュール用の一時ロードライブラリを作成する、CA IDMS/DB メタデータ取得の JCL のサンプル。

- デフォルトでは、PowerExchange Listener は、生成されたジョブが開始するまで 60 秒間待機します。SUBMITTIMEOUT 文を定義することによって、このタイムアウト期間を増やすことができます。バッチジョブがタイムアウト期間内に開始しない場合、PowerExchange はジョブをタイムアウトし、PowerExchange Listener のタスクを停止し、PWX-00426 メッセージを PowerExchange メッセージログに書き込みます。

PC_AUTH 文

PC_AUTH 文は、PowerExchange リスナがその MVS プログラム呼び出し（PC）サービスルーチンを使用して、DBMS ロードライブラリにアクセスする許可を得るかどうかを制御します。

オペレーティングシステム: z/OS

データソース: すべて

必須: いいえ

構文:

PC_AUTH={N|Y}

有効な値:

- **N**。PowerExchange リスナは、APF 許可された状態で実行し、DBMS ロードライブラリの APF 許可済みコピーを PowerExchange リスナの STEPLIB DD 文に含める必要があります。
- **Y**。PowerExchange リスナは APF 許可された状態で実行し、PowerExchange リスナは PowerExchange に用意されたプログラム呼び出し（PC）サービスルーチンを使用して、DBMS ロードライブラリへのアクセスに必要な許可を得ます。

注: DBMS ロードライブラリを STEPLIB DD 文ではなく DTLLOAD DD 文に定義する必要があります。Netport ジョブを使用してデータにアクセスする場合は、ロードライブラリを Netport JCL の DTLLOAD DD 文に定義します。

DBMS ロードライブラリの APF 許可済み PowerExchange コピーをどれも保持しない場合は、Y と指定します。

デフォルトは N です。

使用上の注意:

- PC_AUTH DBMOVER 文と DTLLOAD DD 文を使用して、APF 許可が必要ないロードライブラリにアクセスできます。例えば、これらの文を使用して CA IDMS/DB ロードライブラリにアクセスしたり、データマップの CallProg 関数によって起動されるユーザー定義プログラムにアクセスして、レコード内のソースデータを処理したりできます。

TEMPHLQ 文

TEMPHLQ 文は、PowerExchange が CA IDMS/DB メタデータ用の一時ファイルを作成するときにデフォルトで使用する高レベル修飾子をオーバーライドします。

PowerExchange Listener のユーザー ID を使用してデータセットが作成されないようにする場合は、この文を定義します。PowerExchange は、SECURITY 文の先頭のパラメータに 2 が指定された場合には TEMPHLQ 文を無視します。

オペレーティングシステム: z/OS

データソース: CA IDMS/DB

関連する文: SECURITY

必須: 不要

構文:

TEMPHLQ=*hlq*

値: *hlq* 変数には、PowerExchange が CA IDMS/DB メタデータ用に作成する一時データセット用の高レベル修飾子 (HLQ) を入力します。有効な値は、1~17 文字の文字列です。デフォルトでは、PowerExchange Listener のユーザー ID が、メタデータ一時データセット用の HLQ として使用されます。

TEMPHLQ 文を使用するには、SECURITY 文の先頭のパラメータに 0 または 1 を指定する必要があります。

例: 以下の SECURITY 文と TEMPHLQ 文を定義した場合:

```
SECURITY=(0,N)
TEMPHLQ=B.C
```

IDMS コピーブックインポートプロセスの最中に CA IDMS/DB メタデータを保持するために、次のデータセットが作成されます。

B.C.METADATA.DTL000001

IDMS ロードライブラリのコピーの APF 許可

PowerExchange APF 許可要件を満たすには、PowerExchange のインストール中にオプションで作成されたシステム IDMS ロードライブラリのコピーを APF 許可します。

警告: システム IDMS ロードライブラリにメンテナンスやスキーマの変更を施した場合は、そのロードライブラリのコピーに同じ変更を施すか、XIDIDM10 ジョブを再実行してコピーを再作成する必要があります。そうしないと、バルクデータ移動によってデータの整合性が失われることがあります。

IDMS ロードライブラリのコピーの保持および APF 許可を必要としない場合は、PowerExchange に用意されている MVS プログラム呼び出しサービスルーチンを使用して、APF 許可要件を満たすことができます。

IDMS ロードライブラリのコピーを APF 許可する手順

1. PowerExchange のインストール中に、ジョブを実行する前に以下の文を SETUPIDM JCL に追加します。

```
//STEP1 EXEC SUBMJOB, MEMBER=XIDIDM10
```

XIDIDM10 ジョブは *hlq*.IDMS.LOADLIB および *hlq*.IDMS.DBA.LOADLIB というデフォルト名で IDMS ロードライブラリのコピーを作成しています。
2. PowerExchange インストール時に XIDIDM10 ジョブによって作成された IDMS ロードライブラリのコピーを APF 許可します。
3. PowerExchange Listener JCL 内の STEPLIB DD に *hlq*.IDMS.LOADLIB ライブラリと *hlq*.IDMS.DBA.LOADLIB ライブラリが含まれていることを確認します。

PowerExchange Listener をバッチジョブとして実行した場合、JCL は RUNLIB ライブラリの STARTLST メンバ内にあります。PowerExchange Listener を開始されたタスクとして実行した場合、JCL は PROCLIB ライブラリの PWXLSTNR メンバにあります。
4. PowerExchange Listener JCL の STEPLIB DD 文にある他のライブラリがすべて APF 許可されていることを確認します。これらのライブラリに APF 許可されていないものがあると、PowerExchange Listener を介して IDMS データにアクセスする PowerExchange 許可が無効になり、PowerExchange Listener が開始されません。

関連項目：

- [「APF 許可アクセスに対する z/OS プログラム呼び出しサービスルーチンの使用」 \(ページ 117\)](#)

APF 許可アクセスに対する z/OS プログラム呼び出しサービスルーチンの使用

PowerExchange Listener を実行するための PowerExchange 許可要件を満たすには、PowerExchange に用意されている z/OS プログラム呼び出し (PC) サービスルーチンを使用できます。z/OS PC サービスルーチンは、システム IDMS ロードライブラリの PowerExchange コピーではなく、システム IDMS ロードライブラリで動作します。z/OS PC サービスルーチンは、IDMS ロードライブラリの PowerExchange コピーの保持および APF 許可を必要としない場合、z/OS PC サービスルーチンを使用します。PowerExchange Listener JCL の STEPLIB DD 文内のライブラリはすべて APF 許可する必要があります。また、PowerExchange Listener JCL の DTLLOAD DD 文にシステム IDMS ロードライブラリを指定する必要があります。

APF 許可アクセスに z/OS PC サービスルーチンを使用する手順

1. PowerExchange Listener JCL に DTLLOAD DD 文を追加し、この DD にシステム IDMS ロードライブラリを指定します。

PowerExchange Listener をバッチジョブとして実行した場合、JCL は RUNLIB ライブラリの STARTLST メンバ内にあります。PowerExchange Listener を開始されたタスクとして実行した場合、JCL は PROCLIB ライブラリの PWXLSTNR メンバにあります。
2. RUNLIB ライブラリの DBMOVER 構成メンバに、PC_AUTH=Y という文を追加します。この文は、PowerExchange に z/OS PC サービスルーチンを使用するように指示するものです。
3. 該当する場合は、PowerExchange インストール時にオプションで作成された IDMS ロードライブラリのコピー、*hlq.IDMS.LOADLIB* および *hlq.IDMS.DBA.LOADLIB* を、PowerExchange Listener STEPLIB DD 文から削除します。z/OS PC サービスルーチンでは、代わりにシステム IDMS ロードライブラリを使用します。
4. PowerExchange Listener JCL の STEPLIB DD 文にあるライブラリがすべて APF 許可されていることを確認します。

関連項目：

- [「IDMS ロードライブラリのコピーの APF 許可」 \(ページ 116\)](#)

IDMS バルクデータ移動用の Netport ジョブの設定

DBMOVER 構成メンバ内の SECURITY 文を 2 に設定した場合は、Netport ジョブを使用して IDMS バルクデータ移動ジョブを実行する必要があります。SECURITY 文を 2 に設定するには、RACF などの z/OS セキュリティ管理システムで、IDMS データへのアクセスを試みる各ユーザーアプリケーションのログオンユーザー ID を確認する必要があります。

IDMS データソースにアクセスするための Netport ジョブを設定するには、RUNLIB ライブラリの TAPEJCL メンバ内にあるサンプル Netport ジョブをテンプレートとして使用します。

IDMS バルクデータ移動用の Netport ジョブを設定する手順

1. TAPEJCL メンバをコピーして、IDMS Netport ジョブの基礎となるものを作成します。
2. z/OS PC サービスルーチンを使用していない場合は、IDMS ロードライブラリ *hlq.IDMS.LOADLIB* および *hlq.IDMS.DBA.LOADLIB* の PowerExchange コピーが Netport JCL の STEPLIB DD 文に指定されていることを確認します。
3. IDMS Netport ジョブに以下の DD 文を追加します。

```
//SYSIDMS DD DSN=&HLQ..RUNLIB(DTLDMCL),  
// DISP=(SHR)
```

4. IDMS Central Version を実行している場合は関連する SYSCTL 文が Netport JCL に含まれていることを確認し、ローカルモードで IDMS を実行している場合は以下の DD 文が Netport ジョブに含まれていることを確認します。

```
//IDMSDCT    INCLUDE MEMBER=IDMSDICT  
//IDMSFIL    INCLUDE MEMBER=IDMSFILE
```

注: IDMSDICT メンバおよび IDMSFILE メンバには、関連するディクショナリ定義およびデータベースファイル定義が必要です。

5. DBMOVER 構成メンバ内で、TAPEJCL 用に提供される LISTENER 文と NETPORT 文を編集します。各文に、Netport ジョブの有効なポート番号を入力します。NETPORT 文では、指定したポートに Netport ジョブメンバを割り当てます。

PowerExchange Navigator からデータベース行のテストを実行して IDMS データを読み取る場合は、以下のように、PowerExchange Navigator が動作するシステム上の dbmover.cfg に NODE 文を追加します。

```
NODE=(idmsnet,TCPIP,ipaddress_or_hostname,port_number)
```

必要に応じて、他の文を編集します。

6. z/OS PC サービスルーチンを使用する場合は、Netport ジョブで使用する DBMOVER 構成メンバ内に PC_AUTH=Y を指定します。また、システム IDMS ロードライブラリの PowerExchange コピーではなく、システム IDMS ロードライブラリ自体を指す DTLLOAD DD 文を Netport ジョブに追加します。
7. PowerExchange で新しい Netport ジョブを検出するように、PowerExchange Listener をリスタートします。

IDMS バルクデータの移動

PowerExchange Listener または Netport ジョブを使用して IDMS バルクデータを移動するには、以下の手順を実行します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を収集します。

- IDMS ソースデータベースが含まれているホストの TCP/IP アドレス
- PowerExchange Listener のポート番号および Netport ジョブのポート番号
- サブスキーマ名やデータベース名などの IDMS メタデータ
- z/OS ユーザー ID とパスワード（PowerExchange の SECURITY 文の設定により必要な場合）

IDMS バルクデータを移動するには：

1. PowerExchange Navigator から、データマップを作成します。アクセス方式として **[IDMS]** を選択します。レコードおよびフィールドに関するメタデータをインポートするには、**[レコード定義のインポート]** オプションも選択します。

ヒント: **[レコード定義のインポート]** オプションを選択しない場合は、後でメタデータをインポートできます。リソースエクスプローラでデータマップを開き、**[ファイル]** - **[コピーブックのインポート]** をクリックします。

2. リソースエクスプローラでデータマップを開き、リレーショナルでない「レコード」またはリレーショナルな「テーブル」のビューをクリックして、メタデータが正常に取得されたことを確認します。
3. リモートの IDMS システム上の PowerExchange Listener にデータマップを送信します。リソースエクスプローラでデータマップを選択し、**[ファイル]** - **[リモートノードへの送信]** をクリックします。この手順により、PowerExchange 抽出ルーチンでランタイム中にデータマップにアクセスすることができます。

【場所】 フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定された PowerExchange Listener ノード名を入力します。必要に応じて、他のフィールドも実行します。

注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。

4. メタデータのテーブルビューに対してデータベース行のテストを実行して、IDMS ソースデータベースからデータを返すことができることをテストします。SECURITY 文を 2 に設定した場合は、Netport ジョブに指定されているノード名を使用します。それ以外の場合は、PowerExchange Listener に指定されているノード名を使用します。
5. PowerCenter Designer で、PowerExchange データマップをインポートします。**【ソース】 - 【PowerExchange からインポート】** をクリックします。次に、以下の必須の属性を入力します。
 - **【場所】** フィールドに、dbmover.cfg ファイルの NODE 文で指定した PowerExchange Listener ノード名を入力します。
 - SECURITY 文の設定に応じて、**【ユーザー名】** フィールドおよび **【パスワード】** フィールドに、z/OS ユーザー ID およびパスワードを入力します。
 - **【ソースタイプ】** リストで、**【IDMS】** を選択します。
必要に応じてオプションの属性に値を設定します。次に、PowerExchange に接続し、データマップをインポートします。
6. PowerCenter Designer でマッピングを作成します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアップリケーション接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

第 9 章

IMS バルクデータ移動

この章では、以下の項目について説明します。

- [IMS バルクデータ移動の概要, 120 ページ](#)
- [IMS バルクデータ移動の構成, 122 ページ](#)
- [IMS バルクデータ移動の実装, 126 ページ](#)
- [バルクデータ移動と IMS アンロードファイル, 130 ページ](#)

IMS バルクデータ移動の概要

PowerExchange は、PowerCenter と連結して IMS データベースとの間でバルクデータを移動できます。

IMS データベースが存在する z/OS システムで PowerExchange Listener を実行する必要があります。この PowerExchange Listener が、PowerExchange Navigator および PowerCenter 統合サービスが動作する PowerExchange システムと通信できることを確認してください。

IMS は非リレーショナルデータベースであるため、PowerExchange Navigator を使用してデータマップを作成する必要があります。PowerExchange はデータマップを次の目的で使用します。

- IMS データとメタデータにアクセスする。
- レコードの行タイプのリレーショナルビューを作成する。

PowerExchange では、SQL 型の文を使用してバルクデータを読み書きするのにリレーショナルビューが必要になります。

IMS データベース記述子 (DBD) が利用できる場合、それを DBDLIB ライブラリからインポートできます。または、PowerExchange はソースオブジェクトの DBD メタデータを IMS カタログから取得できます。データマップの作成時に、DBD メタデータをインポートして次の項目を定義します。

- セグメント
- セグメントの階層
- キーフィールド
- 検索フィールド

DBD メタデータは、すべてのセグメントまたは選択したセグメントに対してインポートできます。DBD メタデータをインポートする IMS セグメントごとに、COBOL または PL/I のコピーブックもインポートできます。コピーブックには、DBD から取得したセグメント情報を増加および上書きする、セグメントレイアウトの詳細情報が含まれます。

必要に応じ、ソースデータを含んだ IMS アンロードデータセットを作成し、続いてそのデータセットをバルクデータ移動操作に使用することができます。PowerExchange および PowerCenter で使用できるアンロードデ

ータセットを作成するには、IMS データベースタイプ用の IBM アンロードユーティリティを使用するか、IMS ソリューション用の BMC データベース管理のアンロード機能を使用します。

IMS データのアクセス方式

IMS データにアクセスするには、次のいずれかの方法を使用します。

- DL/I バッチまたは BMP ジョブとして Netport ジョブを実行して、IMS データベースにアクセスします。
DL/I Batch データマップを作成する必要があります。
- PowerExchange Listener を使用して、IMS データベースまたは IMS アンロードデータセットのいずれかにアクセスします。
 - IMS データベースにアクセスするには、IMS ODBA データマップを作成する必要があります。
PowerExchange は、Open Database Access (ODBA) インタフェースを使用して IMS データにアクセスします。
 - IMS アンロードデータセットにアクセスするには、IMS ODBA または DL/I バッチデータマップのどちらかを作成します。
PowerExchange Listener を使用する場合は、Netport ジョブを作成する必要はありません。

PowerExchange におけるグループソース処理

PowerExchange のグループソース処理では、同一の物理ソース内に保存されているデータが単一のパスで読み取られます。この方法ではソースデータの複数のパスを使用しないため、スループットが向上し、リソース消費が低下します。

IMS アンロードデータソースの場合、Designer を使用して複数のレコードタイプを持つデータマップをインポートし、PowerCenter ソース定義を作成できます。ソース定義で 1 つのレコードタイプだけを示す場合は、データマップからテーブルを 1 つだけインポートします。ソース定義にすべてのレコードタイプを含める場合は、マルチレコードデータマップとしてデータマップをインポートします。

マルチレコードデータマップに含まれる各テーブルのソース定義とのマッピングを含んだセッションを実行すると、PowerExchange によりデータセットまたはファイルがソース定義ごとに 1 度ずつ読み取られます。マルチレコードデータマップに含まれるすべてのレコードに対応する単一のソース定義とのマッピングを含んだセッションを実行すると、PowerExchange によりグループソース処理を使用して、データセットまたはファイル内のすべてのレコードが単一のパスで読み取られます。

詳細については、『*PowerCenter 用の PowerExchange インタフェース*』を参照してください。

IMS カタログの使用

PowerExchange には、ソースオブジェクトのメタデータを取得するために、ソース形式と DBDLIB (DBGEN) 形式の IMS データベース定義 (DBD) へのアクセスが必要です。PowerExchange がこの情報に DBDLIB ライブラリではなく、IMS カタログから直接アクセスするように設定できます。

適切に設定されると、PowerExchange は次の IMS ツールを使用して、IMS ソースオブジェクトの DBD 情報を透過的に IMS カタログから取得できます。

- **IMS カタログ API**。この API は、IMSxx.SDFSRESL.RESLIB ライブラリの DFS3CATQ アセンブリプログラムと、IMSxxx.SDFSMAc ライブラリの DFS3CATQ マクロから構成されています。API は IMS アンロード処理のために DBGEN 形式の DBD 情報を IMS カタログから直接取得します。IMS 制御領域が実行していない場合、または IMS 15 より前の IMS バージョンを使用している場合、ブートストラップデータセットの上位レベル修飾子を提供する必要があります。

- **IMS Catalog Library Builder ユーティリティ、DFS3LU00。**ユーティリティはソース形式の DBD 情報を IMS カタログから取得し、この DBD 情報を事前に割り当てた PDSE に書き込みます。PowerExchange Navigator でデータマップを作成する場合、PowerExchange はソースメタデータを PDSE から取得します。メタデータがデータマップ作成のために Navigator マシンにインポートされると、PowerExchange は PDSE のメンテナンスが不要になるように情報を PDSE から削除します。

IMS カタログの使用は、IMS と PowerExchange でオプションです。ただし、データベースのバージョンニング、実行時アプリケーション制御ブロックの管理など、特定の IMS 機能には IMS カタログが必要です。詳細については、IBM IMS の資料を参照してください。

PowerExchange が IMS カタログを使用するには、いくつかの設定タスクを完了する必要があります。詳細については、[「IMS カタログを使用するための設定タスク」 \(ページ 122\)](#)を参照してください。

IMS バルクデータ移動の構成

PowerExchange 管理者が IMS バルクデータ移動操作のために実行するタスクは、データベースアクセス方式によって部分的に異なる場合があります。

構成に関する一般的な考慮事項

- ロックの競合数を削減するため、ソース IMS データベースごとにキーワード PROCOPT=GO x を指定して IMS プログラム通信ブロック (PCB) が定義されていることを確認します。この設定により、IMS データベースへの読み取り専用アクセスが提供されます。
- IMS ODBA または Netport BMP ジョブを使用して IMS データベースにアクセスする場合は、プログラム指定ブロック (PSB) が IMS SYSGEN 内に定義されていることを確認します。
- PowerExchange では、LANG=PLI キーワードを指定して生成された PSB をサポートしていません。
- IMS HDAM、DEDB、または PHDAM データベースをデータソースまたはルックアップとして使用する場合、PowerExchange でデータベースアクセスを完全に最適化することはできません。PowerExchange では、データをシーケンシャル順に格納してもランダムに格納されていると見なされます。

PowerExchange は、レコードの候補がすべて選択されるアクセスパスを決定します。HDAM、DEDB、および PHDAM データベースの場合、レコードは順次アクセスされ、一致しないレコードはすべて破棄されます。データのアクセス用に PowerExchange で構築される SELECT 文には、KEY=*value* を指定した WHERE 句を含めてください。

IMS カタログを使用するための設定タスク

PowerExchange がソースオブジェクトの DBD メタデータを IMS カタログから取得するように設定できます。

IMS カタログを使用するには、次の設定タスクを実行します。

1. IMS Catalog Library Builder ユーティリティ、DFS3LU00 からの出力を保持するための拡張区分データセット (PDSE) を割り当てます。詳細については、[「IMS Catalog Builder ユーティリティ出力用 PDSE の割り当て」 \(ページ 123\)](#)を参照してください。
2. PowerExchange Listener JCL または PROC をカスタマイズして、IMS カタログの使用に必要な DD 文を指定します。詳細については、[「IMS カタログ使用のための PowerExchange Listener JCL のカスタマイズ」 \(ページ 124\)](#)を参照してください。

3. PowerExchange Listener マシン上の DBMOVER 設定メンバで、次の必須文を定義します。
 - IMSID。IMBSBDS 文の *ims_ssid* に一致する *ims_ssid* 値を指定する必要があります。
 - IMBSBDS。この文の *ims_ssid* が IMSID 文の *ims_ssid* に一致することを確認します。
 - LU00FILE。IMS Catalog Library Builder ユーティリティ、DFS3LU00 からの DBD 出力を格納するために作成した PDSE の名前を入力します。

詳細については、『PowerExchange リファレンスマニュアル』の「DBMOVER 構成ファイル」の章を参照してください。
4. PowerExchange Navigator で IMS ソースオブジェクトのデータマップを追加する場合、次のフィールドを設定してください。
 - **[名前]** ページで、**[レコード定義のインポート]** を選択します。
 - **[DL/1 バッチアクセス方式]** ページで、DBMOVER 構成ファイルの IMBSBDS 文で指定した *ims_ssid* 値に一致する IMS SSID 値を入力します。
 - **[コピーブックのインポート - ソースの詳細]** ページで、**[ソース]** の **[リモート]** を選択して、**[タイプ]** フィールドの **[DBD]** を選択します。
 - **[コピーブックのインポート - リモート DBD 詳細]** ページの **[ファイル名]** フィールドで、PDS の名前と、IMS カタログで見つける DBD メンバ名を入力します。

注: IMS カタログの DBD ソース情報が EXTERNALNAME 値を含むが NAME 値は含まないフィールド（例: FIELD EXTERNALNAME=*external_name*）を定義する場合、PowerExchange Navigator は EXTERNALNAME 値をデータマップレコードのフィールド名として使用します。このフィールドはレコードの **[検索フィールド]** リストに表示されず、PowerExchange が作成するセグメント検索指数（SSA）では使用できません。

IMS Catalog Builder ユーティリティ出力用 PDSE の割り当て

IMS Catalog Builder ユーティリティ、DFS3LU00 を使用して、IMS システムに対しデータベース記述子（DBD）を IMS カタログから作成する場合、ユーティリティ出力用の拡張区分データセット（PDSE）を割り当てます。データマップの作成時に PowerExchange はソースオブジェクトに対し DBD 情報を PDSE から読み取ります。

データセットには次の DCB 特性が含まれている必要があります。

- レコード形式（RECFM）を F（固定長（fixed-length））にします。
- レコード長（LRECL）を 80 にします。
- ブロックサイズ（BLKSIZE）を 80 にします。

また、データセット名が DBMOVER 構成ファイルの LU00FILE 文で指定された PDSE メンバ名に一致する必要があります。

例えば、次の特性を持つデータセットを割り当てることができます。

Data Set Name . . . : USER1.IMS.DFS3LU

General Data	Current Allocation
Management class . . : DEFAULT	Allocated blocks . . : 15,360
Storage class . . . : STANDARD	Allocated extents . . : 1
Volume serial . . . : Z6SMLF	Maximum dir. blocks : NOLIMIT
Device type : 3390	
Data class : DEFAULT	
Organization . . . : PO	Current Utilization
Record format . . . : F	Used pages : 11
Record length . . . : 80	% Utilized : 3
Block size : 80	Number of members . . : 0
1st extent blocks . : 15360	
Secondary blocks . : 13	
Data set name type : LIBRARY	Dates
Data set version . : 1	Creation date . . . : 2019/06/20

Referenced date . . . : 2019/08/30
Expiration date . . . : ***None***

注: データマップが正常に作成されると、PowerExchange は自動的に DBD 情報を PDSE から削除します。PDSE のメンテナンスは不要です。

IMS カタログ使用のための PowerExchange Listener JCL のカスタマイズ

PowerExchange Listener JCL または PROC を編集し、IMS ソースオブジェクトの DBD 情報を IMS カタログから取得するために使用されるデータセットを定義する DD 文を追加する必要があります。PowerExchange は IMS カタログ API、DFS3CATQ を透過的に使用して DBDGEN 形式の DBD 情報を取得し、IMS Catalog Library Builder ユーティリティ、DFS3LU00 を使用してソース形式の DBD 情報を取得します。

次の DD 文を Listener JCL または PROC に追加します。

DD 名	説明
DD DBDSOR	IMS Catalog Library Builder ユーティリティ、DFS3LU00 が IMS ソースオブジェクトの DBD 情報をソース形式で書き込む PDSE を指定します。この PDSE を事前に割り当てる必要があります。DD 文で指定した PDSE 名が DBMOVER 構成ファイルの LU00FILE 文の PDSE 名に一致することを確認します。 データマップの作成時、この PDSE に格納されているソースオブジェクトメタデータが、PowerExchange Navigator または infacmd pwx createdatamaps ユーティリティを実行するローカルマシンに送信されます。 注: データマップが正常に作成されると、PowerExchange は自動的に DBD 情報を PDSE から削除します。PDSE のメンテナンスは不要です。
DD DSN =IMSxxx.SDFSRESL	IMS カタログ API DFS3CATQ プログラムとマクロが含まれている IMSxxx.SDFSRESL ライブラリを定義します。
LUSYPRT DD SYSOUT=*	IMS Catalog Library Builder ユーティリティ、DFS3LU00 のアクティビティからの統計が含まれているデータセットを定義します。
SYSPRINT DD SYSOUT=*	PowerExchange Navigator が IMS Catalog Library Builder ユーティリティ、DFS3LU00 に渡したコントロールカードが含まれているデータセットを定義します。

サンプル JCL:

```
//STEPLIB DD DSN=CEE.SCEERUN,DISP=SHR
// DD DSN=&HLQ..LOADLIB,DISP=SHR
// DD DSN=DSNB10.DSNB.SDSNEXIT,DISP=SHR
// DD DSN=DSNB10.SDSNLOAD,DISP=SHR
// * DD DSN=DSN910.SDSNEXIT,DISP=SHR
// * DD DSN=DSN910.SDSNLOAD,DISP=SHR
// * DD DSN=ICU.PMICU.LOAD,DISP=SHR
// * IF USING IMS THEN CUSTOMIZE BELOW
// DD DSN=IMS1510.SDFSRESL,
// DISP=SHR
//LUSYSPRT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//CEEDUMP DD SYSOUT=*
//DBDSOR DD DISP=OLD,DSN=GRAHAM.IMS.DFS3LU,
// DCB=(RECFM=F,LRECL=80,BLKSIZE=80)
```

DL/1 バッチデータマップを使用したバルクデータ移動用の PowerExchange の構成

以下の手順を使用して、DL/1 バッチデータマップおよび Netport ジョブを使用したバルクデータ移動操作用に PowerExchange を設定します。Netport ジョブは、IMS DL/I バッチまたは BMP ジョブとして実行できます。

DL/1 バッチデータマップを使用したバルクデータ移動用に PowerExchange を構成する手順

1. IMS メタデータにアクセスするため、1 つ以上の Netport ジョブを作成します。RUNLIB ライブラリの IMSJCL メンバにあるサンプルの Netport ジョブをテンプレートとして使用します。このメンバが、使用環境で DL/I バッチまたは BMP ジョブのいずれかとして動作するようにカスタマイズします。
2. IMS データベースが存在する MVS システム上の DBMOVER 構成メンバで、IMSJCL メンバの LISTENER 文および NETPORT 文のコメントを解除してカスタマイズします。以下の構文を使用します。

```
LISTENER=(node_name,TCPIP,port_number)
NETPORT=(node_name,port_number,,,"hlq.RUNLIB(ims_netport_member)",psb_id)
```

説明:

- *node-name* は、IMS ソースまたはターゲットシステムで動作する PowerExchange Listener 用に定義する名前です。LISTENER および NETPORT 文で同じノード名を指定します。また、Windows システムまたは PowerExchange Navigator および PowerCenter Integration Service を実行するシステムの dbmover.cfg ファイルと同じノード名を含む NODE 文も定義します。
- *port_number* は、使用サイトの有効なポート番号です。有効なポート番号を取得するには、ネットワーク管理者に相談してください。LISTENER および NETPORT 文で同じポート番号を指定します。
- *psb_id* は、IMS ソースまたはターゲット用の有効な IMS PSB 識別子です。
- *hlq.RUNLIB.ims_netport_member* は、IMS Netport ジョブが格納されているメンバの完全修飾されたデータセット名およびメンバ名です。サンプルの IMSJCL メンバに基づいて、このメンバを作成します。

制限: 指定した PowerExchange Listener と Netport ジョブは、同じ MVS イメージ上で実行する必要があります。それ以外の場合は、Netport ジョブが失敗します。

複数の PSB に対してバルクデータを移動する場合は、PSB ごとに LISTENER 文と NETPORT 文のペアを入力します。

3. DBMOVER 構成メンバの変更が完了したら、PowerExchange Listener をリスタートします。

IMS ODBA データマップを使用したバルクデータ移動用の PowerExchange の構成

以下の手順を使用して、IMS ODBA データマップを使用したバルクデータ移動操作用に PowerExchange を設定します。

1. IMS ロードモジュールを含む IMS RESLIB データセットが PowerExchange Listener JCL の STEPLIB DD 連結または LNKST ライブラリのいずれかに含まれていることを確認します。PowerExchange Listener JCL は、RUNLIB ライブラリの PWXLSTNR メンバまたは STARTLST メンバにあります。
2. IMS ソースまたはターゲットが存在するシステム上の DBMOVER 構成メンバで、以下の ODBASUPP 文を追加して PowerExchange で ODBA を使用できるようにします。

```
ODBASUPP=YES
```

この文の詳細については、『PowerExchange リファレンスマニュアル』を参照してください。

3. IMS PSB によって PCB 名が指定されていることを確認します。ODBA アクセスには、PCB 名が必要です。

IMS ソースまたはターゲットへの接続の設定とテスト

リモート z/OS システム上の IMS データソースまたはターゲットにアクセスするには、PowerExchange でリモートシステム上の PowerExchange Listener と通信する必要があります。 リモート PowerExchange Listener への接続を構成してテストするには、以下の手順を使用します。

IMS ソースまたはターゲットへの接続を設定およびテストするには：

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用してリモート z/OS システムへのネットワークアクセスをテストします。
2. 各 PowerExchange および PowerCenter システム上の dbmover.cfg ファイルに NODE 文を追加して、IMS ソースまたはターゲットが存在するリモートノードを特定します。 以下の構文を使用します。
`NODE=(node_name,TCP,IP,ipaddress_or_hostname,port_number)`
DL/1 バッチデータマップを使用する場合、この文はバルクデータ移動用のバッチ Netport ジョブが実行される z/OS システムを指定します。
IMS ODBA データマップを使用する場合、この文はリモート PowerExchange Listener が実行される z/OS システムを指定します。
ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。
3. ローカル PowerExchange システムで DTLREXE ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

IMS バルクデータ移動の実装

IMS バルクデータの移動準備をするには、前提情報を収集し、実装に関する考慮事項を検討します。その後、バルクデータの移動手順を実行します。

前提情報

バルクデータ移動操作を実装する前に、以下の情報を収集します。

- IMS サブシステム識別子 (SSID)
- DL/1 バッチデータマップの場合: IMS PCB 番号
- IMS ODBA データマップの場合: IMS PSB の名前および PSB 内の PCB の名前
- IMS DBD ソースライブラリの完全修飾されたデータセット名およびメンバ名
- IMS セグメントのコピーブックの完全修飾されたデータセット名およびメンバ名
- アクセスする IMS セグメント
- z/OS ユーザー ID とパスワード (PowerExchange の SECURITY 文の設定により必要な場合)
- DL/I バッチ Netport ジョブの場合: Netport JCL の z/OS DD 文および DSN 文または STEPLIB DD の IMS 動的割り当てライブラリで指定された各データベースデータセット名

実装に関する考慮事項

バルクデータ移動を実装する前に、以下の考慮事項を確認してください。

一般的な考慮事項

IMS バルクデータ移動の場合、以下の考慮事項に注意する必要があります。

- DBD で定義されているような完全な IMS セグメントがデータマップで表現されていない場合、FILLER キーワードを指定せずに定義されたセグメントで PowerExchange の INSERT 操作を実行すると、セグメントの最後に空白以外のデータが書き込まれる可能性があります。この問題は、セグメントのメタデータを PowerExchange にインポートする前にセグメントの COBOL コピーブックに FILLER を追加することで回避できます。
 - PowerExchange Navigator では、レコードという用語は IMS セグメントを指します。
 - IMS データベースに使用する PCB 値は、選択するアクセス方式によって以下のように異なります。
 - データマップで DL/1 バッチアクセス方法を選択する場合は、PSB 内の PCB オフセットを示す PCB 番号を使用します。CMPAT=YES を指定して PSB を生成した場合は、この PCB 番号に 1 を加えます。
 - データマップで IMS ODBA アクセス方式を選択する場合は、PSB の生成時に PCBNAME パラメータまたは PCB 文の 1~8 カラムで指定した PCB 名を使用します。
 - IMS DBD ソースからメタデータをインポートして、IMS データベースのセグメント、セグメント階層、キーフィールド、および検索フィールドを定義します。また、セグメントごとに COBOL または PL/I コピーブックもインポートして、キーフィールドのオーバーレイ、DBD からのフィールドの検索、および他のすべてのフィールドの定義を行います。PowerExchange では、DBD の IMS 階層構造を保持しながら、コピーブック情報を使用してデータマップが更新されます。
 - キーフィールドが存在しないか、または一意でないキーを持つ 1 つ以上のセグメントが存在する IMS ソースからバルクデータを移動する場合で、IMS セグメントごとに一意の識別子が必要なときは、これらのセグメントごとに RBA を返す式をデータマップ内のユーザー定義フィールド内に定義できます。IMS データマップのセグメントデータにセグメント RBA を追加する場合、Informatica では、GetDatabaseKey 関数ではなく GetIMSRBAByLevel 関数を使用することをお勧めしています。GetIMSRBAByLevel 関数では、キーのない、または一意のキーが存在しない親セグメントの RBA を取得することができます。続いて、この RBA を使用してターゲット内に一意のキーを構築できます。
- 実体化が完了した後で IMS ソース用に変更データのキャプチャを実行する場合、PowerExchange は、IMS と同期が取れた CDC には RBA を使用できますが、IMS ログベースの CDC には RBA を使用できません。IMS ログベースの CDC の場合、PowerExchange は、データマップ内の式に GetDataBaseKey 関数または GETIMSRBAByLevel 関数を使用することはできません。
- IMS データマップでデータベース行のテストを実行する場合、**[場所]** フィールドに指定するノード名は、データマップで選択するアクセス方式および IMS ソースのタイプによって異なります。
 - データマップのテーブルビューでデータベース行のテストを実行すると、そのデータベースタイプに適切な SELECT 文が PowerExchange Navigator によって生成されます。

以下に例を示します。

- データベース行のテストの **[DB タイプ]** リスト内で NRDB2 または IMSUNLD を選択した場合、生成される SQL は、次のようにアンダースコア (_) 文字が付いた状態で **[SQL 文]** ボックス内に表示されます。

```
SELECT * FROM IMSUNLD.DBLOG50F_DB#AASEG
```
- **[DB タイプ]** リスト内で NRDB を選択した場合、生成される SQL は、次のようにピリオド (.) 文字が付いた状態で **[SQL 文]** ボックス内に表示されます。

```
SELECT * FROM IMSUNLD.DBLOG50F.DB#AASEG
```

関連項目：

- [「IMS ソースへのアクセスに影響を与える設定」 \(ページ 128\)](#)

ルックアップトランスフォーメーション

PowerCenter では、マッピングに Lookup トランスフォーメーションを追加することによって、IMS データベース内のデータを検索できます。"GOx" PROCOPT で、PCB の Lookup トランスフォーメーションごとに別々の Netport ジョブを使用する必要があります。

ヒント: Lookup トランスフォーメーションを作成する場合は、IMS セグメントのキーフィールドを指定します。これによって、PowerExchange は、完全修飾されたセグメント検索指数 (SSA) を作成し、IMS データベース検索の効率を高めることができます。

Lookup トランスフォーメーションの使用の詳細については、『*PowerCenter 用の PowerExchange インタフェース*』を参照してください。

IMS ターゲットに関する考慮事項

- セグメントがロックされるリスクを減らすために、PowerExchange からのコミット頻度を減らすことができます。
- 空白が書き込まれるフィールドやデータが一切書き込まれないフィールドが IMS ターゲットに含まれる場合は、それらのフィールドをデータマップでオプションのフィールドとして定義します。このようにしないと、バルクデータの移動時に PowerExchange でエラーが発生します。
- データの書き込みは、IMS SDEP セグメントに対しても、キーなしのセグメントの下にあるセグメントに対しても行えません。
- OCCURS 句によってマッピングされたセグメントにデータを書き込む場合、OCCURS 句内の各フィールドを別々のフィールドとして識別するようにコピーブックを変更し、その後 OCCURS 句を削除します。PowerExchange では、OCCURS 句によって定義される出現ごとに別々の行を生成することはできません。
- IMS データマップを PowerCenter にインポートする場合は、キーフィールドとして使用するすべての CCK フィールドがキーフィールドとして特定されていることを確認します。
- Informatica では、別々の Netport ジョブを使用して IMS データを書き込むことをお勧めします。このように処理すると、書き込み目的で PSB にアクセスし、必要に応じて、IEFRDER ログなどの更新された IMS データのために JCL を変更することができます。

IMS ソースへのアクセスに影響を与える設定

データマップの作成やデータベース行テストの実行など、特定のタスクを PowerExchange から実行する場合、PowerExchange による IMS ソースへのアクセス方法に影響を及ぼすいくつかの設定を入力します。これらの設定は、IMS ソースタイプと相互関係があり、IMS ソースタイプによって異なります。特に以下の設定に注目してください。

- IMS データマップ内で選択する **【アクセス方式】** オプション
- データマップをリモート MVS システムに送信する場合、またはデータベース行のテストを実行する場合に **【場所】** フィールドに指定するノード名
- データベース行のテスト用に選択する **【DB タイプ】** オプション

以下の表に、これらの設定の相互関係および IMS ソースタイプの概要を示します。

IMS ソースタイプ	データマップでのアクセス方式	行のテスト用の DB タイプ	場所
IMS データベース	DL/1 バッチ	NRDB または NRDB2	MVS システム上の Netport ポートのノード名
IMS データベース	IMS ODBA	NRDB	MVS システム上の PowerExchange Listener ポートのノード名
IMS アンロードデータセット	DL/1 バッチ	IMSUNLD	MVS システム上の PowerExchange Listener ポートのノード名
IMS アンロードデータセット	IMS ODBA	IMSUNLD	MVS システム上の PowerExchange Listener ポートのノード名

ルックアップトランスフォーメーション

関連する値をルックアップソースから追加したり、一部のレコードがターゲットにすでに存在するかどうかを判断したりするために、バルクデータ移動のマッピングに Lookup トランスフォーメーションをオプションとして含めることができます。

IMS データベースを検索するために PowerExchange がセグメント検索指数 (SSA) で使用するキーフィールドを、Lookup トランスフォーメーションで指定する必要があります。連結キー (CCK) フィールドを使用すると、パフォーマンスが最適になり、IMS データベースに及ぼす影響を最小限に抑えることができます。

また、ルックアップ処理中の IMS データベースにおける競合を最小限にするため、PROCOPT=GOx パラメータを指定して IMS PCB も作成します。このパラメータは、更新中のセグメントの読み込み中に無効なポインタが検出された場合の IMS の対応を指示します。

Lookup トランスフォーメーションの詳細については、『*PowerCenter 用の PowerExchange インタフェース*』および『*PowerCenter トランスフォーメーションガイド*』を参照してください。

IMS データベースからのターゲットのマテリアライズ

バルクデータを IMS データベースからリレーショナルターゲットまたは非リレーショナルターゲットに移動するには、以下の一般手順を使用します。この手順では、PWXP を使用して PowerExchange を PowerCenter と統合するものとします。

操作を開始する前に、以下のタスクを実行します。

- DL/1 バッチデータマップを作成する場合は、Netport ジョブを作成し、DL/1 バッチデータマップを使用したバルクデータ移動用に PowerExchange を構成するために必要な他の構成タスクを完了します。
- IMS ODBA データマップを作成する場合は、IMS ODBA データマップを使用したバルクデータ移動用に PowerExchange を構成するために必要な構成タスクを完了します。

IMS データベースからターゲットを具体化するには：

- ターゲットデータベースの構造を作成します。
- PowerExchange Navigator から、IMS ソース用のデータマップを作成します。
- リソースエクスプローラでデータマップを開き、リレーショナルでない「レコード」またはリレーショナルな「テーブル」のビューをクリックして、メタデータが正常に取得されたことを確認します。
また、データマップを右クリックして **[IMS 階層の表示]** をクリックすると、DBD ソースからインポートされた IMS セグメント階層を表示できます。

4. データマップをリモート z/OS システム上の PowerExchange Listener に送信します。 リソースエクスプローラでデータマップを選択し、**[ファイル] - [リモートノードへの送信]** をクリックします。
5. IMS ソースまたはターゲットのテーブルビューでデータベース行のテストを実行し、IMS データベースから実際のデータを返すことができることを確認します。
6. PowerCenter Designer で、PowerExchange データマップをインポートし、マッピングを作成します。
データマップをインポートする場合、**[場所]** フィールドに、ローカル dbmover.cfg ファイルの NODE 文で指定した PowerExchange Listener のノード名を入力します。 また、**[ソースタイプ]** フィールドに **[IMS]** を指定します。 このフィールドはターゲットにも表示されます。
ヒント: 関連する値を取得したり、ソースレコードがターゲットに既に存在するかどうかを判断したりするために、Lookup トランスフォーメーションを作成できます。 Lookup トランスフォーメーションをマッピングに追加します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアプリケーション接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

PowerExchange CDC を使用してターゲットを更新する場合は、リスタートトークンを設定する必要があります。

関連項目：

- [「IMS ODBA データマップを使用したバルクデータ移動用の PowerExchange の構成」 \(ページ 125\)](#)

バルクデータ移動と IMS アンロードファイル

IMS アンロードデータセットをデータソースとして使用する PowerCenter バルクデータ移動セッションを作成できます。また、IMS アンロードファイルをソースとして、2 つ目の IMS アンロードファイルをターゲットとして使用する PowerCenter バルクデータ移動セッションも作成できます。

データソースとしての IMS アンロードデータセットの使用

IMS アンロードデータセットを、PowerExchange バルクデータ移動操作のデータソースとして使用することができます。 アンロードデータセットを作成するには、IMS データベースタイプ用の IBM アンロードユーティリティを使用するか、IMS ソリューション用の BMC データベース管理のアンロード機能を使用します。

たとえば、IMS ソースに基づいてターゲットをマテリアライズするには、対象となるソースデータを含む IMS アンロードデータセットを作成します。 次に、ターゲットをマテリアライズする PowerCenter ワークフローに対する入力として、作成したデータセットを使用します。ターゲットの実体化が完了した後、PowerExchange Change Data Capture (CDC) を使用してソースとターゲットの同期をとります。

アンロードデータセットに対して、PowerExchange はグループソース処理を実行できます。つまり、PowerExchange は、1 回の操作でアンロードデータベース全体を読み取ることにより複数の IMS セグメントを処理できます。

以下の表に、異なるタイプの IMS データベースに使用できるアンロードレコードの形式と、キー定義方法を示します。

IMS データベースタイプ	アンロードレコードのフォーマット	CCK の取得	GetDatabaseKey 関数の使用によるキー値の取得	GetIMSRBByLevel 関数の使用によるキー値の取得
高速パス ¹	IBM	はい	使用できません。	使用できません。
高速パス ¹	BMC LONG	はい	はい	はい
HALDB	IBM	はい	はい	はい
HALDB	BMC LONG	はい	はい	はい
HDAM	IBM	はい	はい	はい
HDAM	BMC LONG	はい	はい	はい
HDAM	BMC SHORT	はい	使用できません。	使用できません。
HDAM	BMC XSHORT	はい	使用できません。	使用できません。
HIDAM	IBM	はい	はい	はい
HIDAM	BMC LONG	はい	はい	はい
HIDAM	BMC SHORT	はい	使用できません。	使用できません。
HIDAM	BMC XSHORT	はい	使用できません。	使用できません。
HISAM	IBM	はい	使用できません。	使用できません。
HISAM	BMC LONG	はい	使用できません。	使用できません。
HISAM	BMC SHORT	はい	使用できません。	使用できません。
HISAM	BMC XSHORT	はい	使用できません。	使用できません。
1.PowerExchange は、IMS 高速パスの SDEP セグメントをサポートしていません。				

注: BMC Software のアンロードデータセットの場合、LONG、SHORT、および XSHORT のレコード形式は、BMC アンロード JCL の PLUSIN DD 文の DFSURGU1 キーワードで指定されます。

表では、全タイプのアンロードレコードに対して、PowerExchange で連結キー（CCK）を取得できることを示しています。キーなしのセグメントの場合、IMS データマップ内のユーザー定義フィールドで定義する任意の GetDatabaseKey および GetIMSRBByLevel 関数を PowerExchange で処理して、キーとして使用できる値を返すことができます。GetDatabaseKey および GetIMSRBByLevel 関数は、特定のセグメントに対して 8 バイトの RBA 値を返します。

IMS データマップのセグメントデータにセグメント RBA を追加する場合、Informatica では、GetDatabaseKey 関数ではなく GetIMSRBByLevel 関数を使用することをお勧めしています。GetIMSRBByLevel 関数では、キーのない、または一意のキーが存在しない親セグメントの RBA を取得することができます。

バルクデータ移動操作または IMS 同期 CDC に使用される IMS データマップには、GetDatabaseKey 関数および GetIMSRBByLevel 関数を使用します。

次のデータマップタイプには、GetDatabaseKey 関数や GetIMSRBAByLevel 関数が使用できません。

- IMS ログベース CDC に使用される IMS データマップ
- 次のタイプのレコードが含まれる IMS アンロードファイルへのアクセスに使用される IMS データマップ
 - HISAM データセットの任意のフォーマットのアンロードレコード
 - HDAM および HIDAM データセットの BMC SHORT または BMC XSHORT フォーマットのアンロードレコード
 - IMS 高速パスデータセットの標準 IBM フォーマットのアンロードレコード

関連項目：

- [「IMS アンロードデータセットからのターゲットのマテリアライズ」 \(ページ 132\)](#)

IMS アンロードデータセット用の DBMOVER 構成文

IMS アンロードデータセットを使用する場合は、DBMOVER 構成メンバ内に IMSID パラメータを指定する必要があります。

バルクデータ移動の場合、このパラメータは DBDLIB データセットと RECON データセットを定義します。

`IMSID=(ims_ssid,DBD_lib,RECON=(recon,recon,...))`

説明:

- *ims_ssid* は、IMS サブシステム識別子です。1～8 文字の英数文字列を指定します。
IMS データマップ内にこの値と一致する IMS SSID 値を入力して、PowerExchange で DBDLIB データセットが検出できるようにします。
- *DBD_lib* は、DBD ロードモジュールを含む IMS DBDLIB データセットの名前です。
- *recon* は、IMS RECON データセットです。RECON データセットの指定は、IMS ログベース CDC でのみ必要とされます。RECON データセットの区切りにはカンマを使用します。

注: IMS アンロードデータセットの場合、IMS サブシステム識別子オプションおよび IMS DBDLIB データセットオプションのみ必要とされます。

IMS アンロードデータセットからのターゲットのマテリアライズ

IMS ソースデータベースのアンロードデータセットを作成し、それを使用してリレーショナルターゲットまたは非リレーショナルターゲットを具体化するには、以下の一般手順を使用します。この手順では、PWXPC を使用していることが前提になります。

IMS アンロードデータソースの場合、Designer を使用して複数のレコードタイプを持つデータマップをインポートし、PowerCenter ソース定義を作成できます。

IMS アンロードデータセットからターゲットを具体化するには：

1. ターゲットの具体化に使用するソースデータを含む IMS アンロードデータセットを作成します。使用するアンロードユーティリティの詳細については、IBM または BMC のマニュアルを参照してください。
2. ターゲットデータベースの構造を作成します。
3. PowerExchange Navigator で、データのアンロード元となった IMS ソースのデータマップを作成します。
4. PowerCenter Designer で、データマップをインポートし、マッピングを作成します。マッピングする際、ソースとしてのデータのアンロード元となった IMS セグメントを指定し、の手順 2 でターゲットとして作成したデータベース構造を指定します。

データマップに複数のレコードタイプが含まれており、PowerExchange でグループソース処理を行う場合は、**[PowerExchange からインポート]** ダイアログボックス内で **[マルチレコードデータマップ]** を選択します。このオプションを選択すると、PowerExchange によりデータマップがマルチレコードデータマップとしてインポートされます。

5. PowerCenter Workflow Manager で、ワークフローおよびセッションを作成し、PWX NRDB Batch アプリケーション接続を構成します。

アプリケーション接続で、IMS データを含む、z/OS システム上の PowerExchange Listener を指す **[場所]** の値を指定します。セッションプロパティの **[IMS アンロードファイル名]** 属性で、IMS アンロードデータセット名を指定します。

ワークフローを開始して、IMS アンロードデータをターゲットデータベースに移動します。

PowerExchange CDC を使用してターゲットを更新する場合は、リスタートトークンを設定する必要があります。

IMS アンロードデータセットへの複数レコードの書き込み

バルクデータ移動セッション時に、PowerExchange は複数レコードデータマップを使用して、z/OS 上の IMS アンロードデータセットからデータを読み取り、複数レコードを 1 回のパスで IMS アンロードデータセットのターゲットに書き込むことができます。この処理を *複数レコードの書き込み* と呼びます。

PowerExchange が複数レコードの書き込みを実行するときに、ソースシーケンス情報は維持されます。順序付けを行う複数レコードの書き込みを有効にするには、ソース定義およびターゲット定義を作成するときに **[PowerExchange からインポート]** ダイアログボックスで、**[複数レコードのデータマップ]** と **[シーケンスフィールドを使用]** オプションを選択します。

順序付けを行う複数レコードの書き込みを実行するため、PowerExchange は、グループソース処理を使用して 1 回のパスでソースデータを読み取り、グループターゲット処理を使用して 1 回のパスでターゲットにデータを書き込みます。

PowerExchange は、ターゲットに対するソースレコードのリレーションについてのメタデータを渡すためにシーケンスフィールドを生成します。PowerCenter ワークフローに対して順序付けを行う複数レコードの書き込みを有効にすると、ワークフローは複数レコードのソースを読み取り、生成されたシーケンスフィールドを使用して順序付け情報を維持し、データをソースと同じシーケンスにあるターゲットに対して書き込むことができますようになります。

IMS レコードタイプ間のリレーションを決定するため、PowerExchange では DBD から複数レコードのデータマップにインポートされたセグメント情報を使用します。PowerExchange はこのセグメント情報を使用して、ソース定義およびターゲット定義のプライマリキー、外部キー、およびシーケンスメタデータを生成します。

PowerCenter セッションを実行すると、PowerExchange は生成されたキーの値を使用してデータを再構築し、正しいシーケンスでターゲットに書き込みます。PowerExchange は統合サービスノードの順序付けおよびキューキャッシュのデータを維持します。PowerExchange がデータをターゲットに書き込むと、生成したキーフィールドが削除され、データは正しいシーケンスのネットワークを通してターゲットファイルに送信されます。

IMS アンロードデータセットへの複数レコード書き込みのルールとガイドライン

IMS アンロードデータセットターゲットへの複数レコード書き込みを行う場合は、あらかじめこの機能に関連するルールとガイドラインに目を通してください。

詳細については、『PowerCenter 用の PowerExchange インタフェース』を参照してください。

ソースとターゲットのファイル

ソースとターゲットには、以下のルールとガイドラインが適用されます。

- ターゲットが IMS アンロードファイルであれば、ソースも IMS アンロードファイルでなければなりません。
- ソース IMS アンロードファイルは、PowerExchange がデータソースとしてサポートしているどの形式でも構いません。詳細については、「[データソースとしての IMS アンロードデータセットの使用](#)」(ページ 130)を参照してください。
- ターゲットには、標準の IBM アンロード形式のみがサポートされています。BMC またはパーティションされたアンロードファイルはサポートされていません。
- データマップで DEDB データベースを指定すると、アンロードファイルは Fast Path データベース形式で記述されます。この指定をしないかぎり、ファイルは標準の IBM アンロード形式で記述されます。
- IBM アンロードユーティリティを実行して複数のパーティションからデータをアンロードすると、それらのパーティションは出力されたアンロードファイルでインターリーブされます。複数レコード書き込みを行うワークフローでアンロードファイルをデータソースとして使用する前に、そのアンロードファイルをソートしてパーティションを一定の順序に並べる必要があります。あるいは、次の制御カードを追加してジョブのアンロードができます。

TASKCTL=(1)

- Fast Path DEDB の複数領域用のセグメントを単一のアンロードファイルにアンロードする場合は、複数レコード書き込みを行うワークフローでアンロードファイルをデータソースとして使用する前にそのアンロードファイルをソートする必要があります。

データマップ、ソース定義、およびターゲット定義

データマップ、ソース定義、およびターゲット定義には、以下のルールとガイドラインが適用されます。

- 複数レコード書き込みには、新しいソース定義とターゲット定義を作成する必要があります。シーケンス情報のない既存の定義は、複数レコード書き込みでは機能しません。複数レコード書き込みを使用しないワークフローに対しては、既存の定義を引き続き使用できます。
- IMS アンロードファイルのソース定義とターゲット定義は、同じデータマップに基づいたものにする必要があります。
- データマップには、SSID 名と DBD 名を定義する必要があります。これらのフィールドは、既存の DL/1 データマップまたは ODBA データマップではブランクになっていることがあります。SSID フィールドまたは DBD フィールドがブランクの IMS データマップを、複数レコード書き込みを有効にしたソース定義にインポートしようとすると、PowerExchange によりメッセージ PWX-03611 または PWX-03612 が発行されます。
- 階層全体を形成するテーブルすべてがデータマップに含まれていることを検証してください。インポートプロセスでは階層の完全性が検証されません。

データマップで必要なテーブルがすべてソース定義に含まれていることを確認するには、それらがデータマップで単純なテーブルとして定義されていることを検証します。単純なテーブルは単一のレコードに基づきます。

データマップに複雑なテーブルが含まれる場合、PowerExchange はそれらをソースおよびターゲット定義には含めません。複雑なテーブルは 1 つ以上のレコードに基づきます。

- データマップのどのテーブルをインポートするかを選択することはできません。PowerCenter は、データマップの単純なテーブルをすべてインポートします。
- PowerExchange は、インポートされた IMS データマップ内のユーザー定義フィールド、RBA 関連フィールド、または CCK 関連フィールドは、ソース定義から除外します。
- データマップでは、各レコードに対してちょうど 1 つのテーブルが必要です。

マッピングとトランスフォーメーション

PowerCenter マッピングと PowerCenter トランスフォーメーションには、以下のルールとガイドラインが適用されます。

- マッピングトランスフォーメーションでは行が削除される場合があります。この状況が発生する場合には、マッピングによって孤立行が作成されないように対策してください。複数レコード書き込みワークフローを実行すると、孤立行はターゲットに書き込まれず、削除されます。
- マッピングでは生成されたシーケンスフィールドの位置や名前を絶対に変更してはならず、そうでない場合はそれらを削除します。
- マッピングにレコードを挿入するトランスフォーメーションはサポートされていません。
- ソーサ、ジョイナ、およびリンクトランスフォーメーションを含む、行 ID を変更する可能性のあるトランスフォーメーションは、サポートされていません。

接続

接続と接続プロパティには、以下のルールとガイドラインが適用されます。

- 複数レコード書き込みを実行できるのは、PWXPC NRDB バッチ接続を使用する場合のみです。PowerExchange ODBC 接続はサポートされません。
- 【オフロード処理】** のソース接続属性とターゲット接続属性に、**【後でフィルタ】** を選択します。PowerExchange および PWXPC は、z/OS に出力レコードを書き込む前に、Integration Service マシン上でオフロード処理を実行します。
異なる値を選択すると、その値は PowerExchange によって **【後でフィルタ】** に変更されます。
ただし、GetDatabaseKey 関数または GetIMSRBAbLevel 関数を使用して BMC LONG 形式で IMS アンロードファイルから RBA を読み込む場合は、**【オフロード処理】** ソース接続属性に **【いいえ】** を選択する必要があります。この値を選択しないと、メッセージ PWX-03803 でセッションが失敗します。
- 【オフロード処理】** 接続属性を **【後でフィルタ】** に設定し、**【ワーカースレッド】** 接続属性をゼロ以外の値に設定すると、IMS ソース接続でマルチスレッド処理がサポートされます。
- マルチスレッド処理は、IMS ターゲット接続にはサポートされません。**【ワーカースレッド】** 接続属性を設定した場合、その設定は無視されます。
- 【Confirm Write (書き込みの確認)】** のソース接続属性とターゲット接続属性に、**【OFF】** を選択します。他の値を選択すると、PowerExchange によって値が **【Off】** に変更されます。
- 場合によっては、**【CSQ_MEMSIZE】** パラメータの値 (**【PWX オーバーライド】** 接続属性の) を変更する必要があるかもしれません。このパラメータは、キャッシュが複数レコード書き込みに消費できる最大メモリ容量を定義します。

セッションのプロパティ

PowerCenter セッションプロパティには、以下のルールとガイドラインが適用されます。

- 順序付けを有効にして定義されたソースまたはターゲットには、パイプラインのパーティション化は読み取りや書き込みにサポートされません。
- IMS アンロードソースおよびターゲットの **【IMS アンロードファイル名】** プロパティを指定する必要があります。
- ソースには、どのブロックもフラッシュすることなく全グループで発生できる最大のブロックフラッシュ数を **指定したブロックの後にフラッシュ** プロパティで指定します。このプロパティは、すべてのブロックが定期的に必ずフラッシュされるように定義します。

IMS アンロードデータセットへの複数レコード書き込みの実行

複数レコード IMS アンロードデータセットソースから複数レコード IMS アンロードデータセットターゲットに正しい物理シーケンスでデータを転送するには、複数レコード書き込みを実行するセッションを設定します。

1. まだ IMS データソースの複数レコードデータマップを作成していない場合は、作成します。
2. PowerCenter Designer で、[ソース] > [PowerExchange からインポート] の順にクリックします。
3. [PowerExchange からインポート] ダイアログボックスで、必須情報を入力します。

次の表では、必須情報を示します。

属性	説明
場所	ソースファイルが存在するノードの名前。この値は、PowerExchange dbmover.cfg ファイル内の NODE 文の名前と一致する必要があります。
ユーザー名	ターゲットに接続するデータベース権限を持ったユーザーの名前。
パスワード	ユーザー名に関連付けられているパスワード。
複数レコードのデータマップ	選択肢として複数レコードデータマップを表示するかどうか。 このチェックボックスをオンにします。
シーケンスフィールドの使用	複数レコード書き込み操作のシーケンスフィールドを生成するかどうか。 このチェックボックスをオンにします。
ソースタイプ	データソースのタイプです。 [IMS] を選択します。

このダイアログボックスの詳細については、「PowerCenter 用の PowerExchange インタフェース」を参照してください。

4. [接続] をクリックします。
使用可能な複数レコードデータマップが [選択するデータマップ] ボックスに表示されます。
5. インポートするデータマップを選択します。
6. [OK] をクリックします。
ソース定義がワークスペースに表示されます。
7. [ターゲット] > [PowerExchange からのインポート] をクリックします。
8. [PowerExchange からインポート] ダイアログボックスで、必須情報を入力します。

次の表では、必須情報を示します。

属性	説明
場所	ソースファイルが存在するノードの名前。この値は、PowerExchange dbmover.cfg ファイル内の NODE 文の名前と一致する必要があります。
ユーザー名	ソースに接続するデータベース権限を持ったユーザーの名前。
パスワード	ユーザー名に関連付けられているパスワード。

属性	説明
複数レコードのデータマップ	選択肢として複数レコードデータマップを表示するかどうか。 このチェックボックスをオンにします。
シーケンスフィールドの使用	複数レコード書き込み操作のシーケンスフィールドを生成するかどうか。 このチェックボックスをオンにします。
ソースタイプ	データターゲットタイプ。 [IMS] を選択します。

9. **【接続】** をクリックします。
使用可能な複数レコードデータマップが **【選択するデータマップ】** ボックスに表示されます。
10. ソース定義に対して選択したのと同じデータマップを選択します。
11. **【OK】** をクリックします。
ターゲット定義がワークスペースに表示されます。
12. ソース定義、ソース修飾子トランスフォーメーション、およびターゲット定義を含むマッピングを作成します。
ソース修飾子トランスフォーメーションとターゲット定義の間に、[「IMS アンロードデータセットへの複数レコード書き込みのルールとガイドライン」](#) (ページ 133)に記載されている要件を満たす情報を含めることができます。
13. IMS データベースの接続がまだない場合は、定義します。
14. PowerCenter セッションプロパティを定義します。
 - a. Task Developer で、セッションをダブルクリックして編集します。
 - b. **【マッピング】** タブで、**【ソース】** ビューをクリックします。
 - c. **【プロパティ】** で、IMS ソースアンロードファイルの名前に **【IMS アンロードファイル名】** プロパティを設定します。
 - d. 必要に応じ、追加のソースプロパティを定義します。
 - e. **【マッピング】** タブで、**【ターゲット】** ビューをクリックします。
 - f. **【プロパティ】** で、セッションで割り当てる IMS アンロードファイルの名前に **【IMS アンロードファイル名】** プロパティを設定します。
 - g. 必要に応じ、追加のターゲットプロパティを定義します。
注: PowerCenter セッションを起動する前に PowerExchange によって空のターゲットデータセットを作成しておく必要がある場合は、ターゲットの **【Pre SQL】** セッションプロパティに CREATEFILE コマンドを指定します。詳細については、『*PowerExchange Interfaces for PowerCenter*』ガイドの付録「PowerCenter 用の PowerExchange インタフェースのヒント」に記載されている空ファイルの説明を参照してください。
 - h. **【OK】** をクリックします。

第 10 章

Microsoft SQL Server バルクデータ移動

この章では、以下の項目について説明します。

- [Microsoft SQL Server バルクデータ移動の概要, 138 ページ](#)
- [SQL Server バルクデータ移動用にサポートされているデータタイプ, 139 ページ](#)
- [Microsoft SQL Server バルクデータ移動の構成, 140 ページ](#)
- [Microsoft SQL Server バルクデータの移動, 142 ページ](#)
- [SQL Server バルクロードユーティリティを使用したバルクデータのロード, 143 ページ](#)

Microsoft SQL Server バルクデータ移動の概要

オプションとして PowerExchange を PowerCenter と連結して、Windows システム上の Microsoft SQL Server データベースとの間でバルクデータを移動できます。PowerExchange は、データの読み込みまたは書き込みを行うため、SQL Server データベースへの接続を確立します。

PowerCenter は、PowerExchange なしで SQL Server バルクデータの移動に使用できます。ただし、SQL Server バルクデータ移動に PowerExchange を使用することには、以下のような利点があります。

- PowerCenter Integration Service が動作しているプラットフォームでネイティブ接続が使用できない場合、SQL Server への接続が PowerExchange で提供されます。
- ODBC 接続が不要です。
- PowerExchange 接続特有の機能を PowerCenter で活用できます。

SQL Server バルクロードユーティリティを PowerExchange とともに使用して効率的に大量のデータを SQL Server テーブルにロードできます。

SQL Server はリレーショナルデータベースであるため、PowerExchange Navigator でデータマップを作成する必要はありません。PowerCenter Designer および Workflow Manager のみでバルクデータ移動を完全に定義できます。PowerCenter Designer で、SQL Server ソースまたはターゲットのメタデータを PowerExchange からインポートして、ソースまたはターゲット定義を作成できます。

SQL Server のソースまたはターゲットにアクセスするには、Windows システムで PowerExchange Listener を実行して、PowerExchange がそれと通信できていることを確認します。また、SQL Server データベースから見てリモートの Windows システムで PowerExchange リスナを実行する場合は、Microsoft SQL Server 2012 Native Client を PowerExchange リスナのシステムにインストールします。詳細は、[「SQL Service のソースまたはターゲットへの接続の設定およびテスト」 \(ページ 141\)](#)を参照してください。

SQL Server バルクデータ移動用にサポートされているデータタイプ

以下の表に、PowerExchange がバルクデータ移動用にサポートしている SQL Server データタイプとサポートしていない SQL Server データタイプを示します。

データタイプ	バルク移動用にサポートされているか	コメント
bigint	はい	-
binary	はい	-
bit	はい	-
char	はい	-
date	はい	PowerCenter では、ソースメタデータを PowerExchange からインポートしてソース定義を作成する場合に、PowerExchange が日付列を varchar(10) 列に変換します。この変換が行われるのは、PowerCenter のデータ型処理との一貫性を保つためです。
datetime	はい	-
datetime2	はい	-
datetimeoffset	はい	-
decimal	はい	-
float	はい	-
geography	いいえ	-
geometry	いいえ	-
hierarchyid	いいえ	-
image	いいえ	代わりに varbinary(MAX)を使用。
int	はい	-
money	はい	-
nchar	はい	-
ntext	いいえ	代わりに nvarchar(MAX)を使用。
numeric	はい	-
nvarchar	はい	-

データタイプ	バルク移動用にサポートされているか	コメント
real	はい	-
smalldatetime	はい	-
smallint	はい	-
smallmoney	はい	-
sql_variant	いいえ	PowerExchange は sql_variant カラムの変更データをキャプチャしないが、同じテーブル内の他のカラムの変更データはキャプチャする。
text	いいえ	代わりに varchar(MAX)を使用。
time	はい	-
timestamp	はい	-
tinyint	はい	-
uniqueidentifier	はい	PowerCenter は、uniqueidentifier データタイプを 38 文字の varchar データタイプとしてインポートする。
ユーザー定義のデータタイプ (User-Defined Datatype: UDT)	はい	PowerExchange は UDT を UDT の元となるデータタイプと同様に扱う。
varbinary	はい	-
varchar	はい	-
xml	はい	PowerExchange はこのデータタイプを varchar(MAX)として扱う。

Microsoft SQL Server バルクデータ移動の構成

To configure PowerExchange でバルクデータ移動の操作ができるよう構成するには、次のタスクを完了してください。

- Microsoft SQL Server のソーステーブルに対して PowerExchange に読み取りアクセスがあることを確認します。PowerExchange が動作するシステムに SQL Server データベースが存在する場合、PowerExchange は Windows 認証を使用して SQL Server と通信します。リモート Windows システムに SQL Server データベースが存在する場合、PowerExchange は、PowerExchange Listener が動作しているところのユーザー ID を使用します。
- DBMOVER コンフィギュレーションファイルに MSS_ERRORFILE 文を定義することにより、Microsoft SQL Server バルクデータ移動を構成します。
- SQL Server のソースとターゲットへの接続を設定およびテストします。

DBMOVER コンフィギュレーションファイルへの Microsoft SQL Server 文の追加

PowerExchange で Microsoft SQL Server バルクデータ移動操作に使用する、ユーザーカスタマイズの SQL エラーファイルの名前を指定するには、DBMOVER コンフィギュレーションファイルに MSS_ERRORFILE 文を含めます。

ユーザーによってカスタマイズされた SQL エラーファイルの詳細については、「[「カスタマイズされたエラー処理用のエラーアクションファイルの作成」 \(ページ 180\)](#)」を参照してください。

MSS_ERRORFILE 文

MSS_ERRORFILE 文は、PowerExchange が Microsoft SQL Server バルクデータ移動操作に使用する、ユーザーがカスタマイズした SQL エラーファイルの名前を指定します。

オペレーティングシステム: Windows

データソース: Microsoft SQL Server

必須: いいえ

構文:

MSS_ERRORFILE=*file*

値: *file* 値には、PowerExchange でリカバリ可能または致命的エラーとして処理する SQL エラーコードを含むファイルのパスおよび名前を入力します。

PowerExchange では、mssqlerr.act というサンプルのエラーアクションファイルが PowerExchange インストールディレクトリに用意されています。

Windows ネットワークパスを指定する場合、PowerExchange でネットワークパスが正しく解析されるように、先頭にバックスラッシュ (\\) を 3 つ追加してください。

SQL Service のソースまたはターゲットへの接続の設定およびテスト

SQL Server のソースまたはターゲットにアクセスするには、Windows システムで PowerExchange リスナを実行して、PowerExchange がそれと通信できていることを確認します。また、SQL Server データベースから見てリモートの Windows システムで PowerExchange リスナを実行する場合は、Microsoft SQL Server 2012 Native Client を PowerExchange リスナのシステムにインストールします。

1. 分散データベースを含む SQL Server ソースシステムから見てリモートの Windows システムで PowerExchange リスナを実行する場合は、Microsoft SQL Server 2012 Native Client を PowerExchange リスナのシステムにインストールします。
2. PowerExchange Navigator、PowerCenter Client、PowerCenter 統合サービスを実行するシステムで、次のタスクを実行します。
 - リモートの場合は **ping** コマンドを使用して、PowerExchange リスナシステムへのネットワークアクセスをテストします。
 - PowerExchange リスナとそのポートが識別できるように NODE 文を dbmover.cfg ファイルに追加します。

NODE=(*node_name*,TCP/IP,*ipaddress_or_hostname*,*port_number*)

ヒント: ポート番号を決定するには、リモートシステムに PowerExchange リスナをインストールした社内の担当者に問い合わせてください。

- DTLREXE ping ユーティリティを実行して、NODE 文で定義した PowerExchange リスナへの接続状態をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

Microsoft SQL Server バルクデータの移動

Windows 上の Microsoft SQL Server データベースとの間でバルクデータを移動するには、以下の手順を使用します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を取得します。

- SQL Server サーバーの名前
- SQL Server データベースの名前
- SQL Server のソーステーブルまたはターゲットテーブルの名前
- データベースのユーザー ID とパスワード

ヒント: バルクデータ移動操作を実行する前に、PowerExchange Navigator からソーステーブルのメタデータおよびデータをプレビューすることができます。各テーブルカラムのデータを表示するには、そのテーブルのパーソナルメタデータプロファイルに対してデータベース行のテストを実行します。

Microsoft SQL Server バルクデータを移動する手順

1. PowerCenter Designer で、SQL Server データソースの **【ソース】** - **【PowerExchange からインポート】** をクリックするか、SQL Server データターゲットの **【ターゲット】** - **【PowerExchange からインポート】** をクリックします。
2. **【PowerExchange からインポート】** ダイアログボックスで、以下の必須の属性を入力します。
 - PowerCenter が動作するローカルシステムに SQL Server データベースが存在する場合は、**【場所】** フィールドにローカル dbmover.cfg ファイルの LISTENER 文で指定した PowerExchange Listener のノード名を入力します。SQL Server データベースが別のシステムに存在する場合は、ローカル dbmover.cfg の NODE 文で指定したノード名を入力します。
 - **【ユーザー名】** および **【パスワード】** フィールドに、SQL Server のソーステーブルまたはターゲットテーブルにアクセスするためのユーザー ID とパスワードを入力します。
 - **【ソースタイプ】** のリストから、**【MSSQL】** を選択します。
 - **【サーバー名】** フィールドに、SQL Server のインスタンス名を入力します。
 - **【データベース名】** フィールドに、SQL Server のデータベース名を入力します。
必要に応じてオプションの属性に値を設定します。
3. PowerCenter Designer でマッピングを作成します。
4. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX MSSQLServer リレーショナルデータベース接続を構成します。次に、バルクデータ移動を実行するワークフローを開始します。

SQL Server バルクロードユーティリティを使用したバルクデータのロード

SQL Server バルクロードユーティリティを使用すると、PowerExchange がデータソースから読み込んだ SQL Server テーブルに大量のデータを効率よくロードできます。PowerCenter で、PowerExchange Client for PowerCenter (PWXPC) 接続を使用して、バルクデータ移動を実装する PowerExchange と通信します。PowerExchange は SQL Server bcp ユーティリティを起動してバルクロードを実行します。

構成手順の概要

PowerExchange で Microsoft SQL Server バルクロード操作を実行するように設定するには、SQL Server バルクデータを移動する標準的な手順を実行します。

- [「Microsoft SQL Server バルクデータ移動の構成」 \(ページ 140\)](#)
- [「Microsoft SQL Server バルクデータの移動」 \(ページ 142\)](#)

さらに、これらの特定のタスクを実行します。

- 必要に応じて、DBMOVE コンフィギュレーションファイルの APPBUFSIZE の値を調整します。
- SQL Server バルクロードの PWXPC 接続属性を設定します。
- ターゲットロードタイプのセッションプロパティを適切に設定します。
- PowerCenter セッションログを確認し、SQL Server バルクローダーが使用されていることを確認します。

Microsoft SQL Server バルクロードの PWX MSSQLServer 接続属性

以下の表に、Microsoft SQL Server bcp ユーティリティを使用するバルクデータロード操作を制御するために設定できる、各接続属性を示します。

接続属性	説明
配列サイズ	SQL Server バルクロードのストレージ配列のサイズ（単位: レコード数）を指定します。 有効な値は 1~5000 です。デフォルトは 25 です。
拒否ファイル	拒否ファイルに対して PWXR のデフォルトのプレフィックスをオーバーライドします。 書き込みモードが [フォールトトレランスを持つ非同期] の場合、PowerExchange はターゲットマシン上に拒否ファイルを作成します。 注: PWXDISABLE を指定することで、拒否ファイルの作成を防ぐことができます。
書き込みモード	書き込みモードを選択します。 デフォルト値は [書き込み確認オン] です。

PowerCenter における PWX MSSQLServer リレーショナル接続の詳細については、『PowerCenter 用の PowerExchange インタフェース』を参照してください。

ターゲットロードタイプセッションプロパティ

セッションプロパティの [マッピング] タブの [プロパティ] 設定で、[ターゲットロードタイプ] セッションプロパティに [ノーマル] または [バルク] を選択できます。SQL Server bcp ユーティリティを使用してバルクデータをロードするには、[バルク] を選択します。

注: マッピングにアップデイトストラテジトランスフォーメーションが含まれる場合には、[ノーマル] を選択します。

SQL Server バルクロードセッションの PowerCenter セッションログ

SQL Server バルクローダーが使用されていることを確認するには、PowerCenter セッションログを確認します。以下のメッセージは、バルクローダーが使用されていることを示しています。

```
2012-03-27 09:37:44 : INFO : (13240 | WRITER_1_*_1) : (IS | i_opal) : n_opal : WRT_8146 : Writer: Target is
database [@Async], user [], bulk mode [ON]
2012-03-27 09:37:44 : INFO : (13240 | WRITER_1_*_1) : (IS | i_opal) : n_opal : WRT_8106 : Warning! Bulk Mode
session - recovery is not guaranteed.
2012-03-27 09:37:44 : INFO : (13240 | WRITER_1_*_1) : (IS | i_opal) : n_opal : CMN_1021 : Database driver
event...
Connection Information for the PowerExchange Listener: Location[beefy64], Database Type[MSSQL], User
Name[stqa], Compression[N], Encryption[N], Pagesize[0], Interpret As Rows[Y], Confirm Network Write[Y]
Database Type Specific parameters: Server Name[MHV2K8QA6L03], Database Name[Async], Array Size[50], Bulk
load[Yes]
```

バルクロードに関するチューニングの考慮事項

最適なパフォーマンスのために、次の値を増加することが必要な場合があります。

- [配列サイズ] 接続属性。最高のパフォーマンスを得るには、500 以上の値を指定します。
- DBMOVER コンフィギュレーションファイルの APPBUFSIZE 文。

第 11 章

Oracle バルクデータ移動

この章では、以下の項目について説明します。

- [Oracle バルクデータ移動の概要, 145 ページ](#)
- [Oracle バルクデータ移動用にサポートされているデータタイプ, 146 ページ](#)
- [Oracle バルクデータ移動の構成, 147 ページ](#)
- [Oracle バルクデータの移動, 149 ページ](#)

Oracle バルクデータ移動の概要

オプションとして PowerExchange を PowerCenter と連結して、Linux、UNIX、または Windows 上の Oracle データベースとの間でバルクデータを移動できます。

Oracle はリレーショナルデータベースであるため、PowerExchange Navigator でデータマップを作成する必要はありません。PowerCenter Designer および Workflow Manager のみでバルクデータ移動を完全に定義できます。PowerCenter Designer で、Oracle ソースまたはターゲットのメタデータを PowerExchange からインポートして、ソースまたはターゲット定義を作成できます。

PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムから Oracle データベースがリモートシにある場合は、Oracle SQL*Net（Net8 と呼ばれる）または PowerExchange Listener を使用してリモートの Oracle データベースと通信できます。

注: PowerCenter を Oracle バルクデータの移動専用に使使用できます。

PowerExchange のバルクデータ移動で Oracle 12c Multitenant 環境はサポートされません。

Oracle バルクデータ移動用にサポートされているデータタイプ

PowerExchange ではバルクデータ移動用のほとんどの Oracle データタイプをサポートしています。

以下の表に、PowerExchange がバルクデータ移動用にサポートしている Oracle データタイプとサポートしていない Oracle データタイプを示します。

データタイプ	バルクデータ移動のサポートの可否	コメント
BFILE	いいえ	-
BINARY_DOUBLE	はい	Oracle Client 10g以降を使用する場合のみサポート
BINARY_FLOAT	はい	Oracle Client 10g以降を使用する場合のみサポート
CHAR	はい	-
DATE	はい	-
FLOAT	はい	-
LOB	いいえ	-
LONG	いいえ	-
LONG RAW	いいえ	-
NCHAR	はい	-
NUMBER	はい	PowerExchange では NUMBER カラムが次のように処理されます。 <ul style="list-style-type: none">- スケールが 0 で精度の値が 10 未満の数は INTEGER として処理されます。- 精度とスケールが定義された数値は NUMCHAR として処理されます。- 精度とスケールが定義されていない数値は DOUBLE として処理されます。
NVARCHAR2	はい	-
RAW	はい	-
ROWID	はい	-
TIMESTAMP	はい	-
TIMESTAMP WITH TIME ZONE	いいえ	-
TIMESTAMP WITH LOCAL TIME ZONE	いいえ	-
UROWID	いいえ	-

データタイプ	バルクデータ移動のサポートの可否	コメント
VARCHAR2	はい	-
XML タイプ	いいえ	-

Oracle バルクデータ移動の構成

バルクデータ移動操作に PowerExchange を構成するには、PowerExchange 管理者が以下のタスクを実行します。

- Oracle データソースまたはデータターゲットがリモートシステムにあり、PowerExchange Listener を使用して通信する場合は、リモートシステム上の PowerExchange Listener への接続を設定およびテストします。または、SQL*Net を使用できます。
- dbmover.cfg コンフィギュレーションファイルに Oracle 固有の文を定義することにより、Oracle バルクデータ移動を構成します。PowerExchange フォールトトレランス動作をカスタマイズするためのエラーアクションファイルを作成した場合は、ORA_ERRORFILE 文を含めます。

リモートの Oracle ソースまたはターゲットへの接続の設定およびテスト

Oracle データソースまたはターゲットがリモートシステムにある場合は、SQL*Net を使用している場合を除き、PowerExchange および PowerCenter でリモートシステム上の PowerExchange Listener と通信する必要があります。

リモートの Oracle ソースまたはターゲットへの接続を設定およびテストする手順

1. PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムで、**ping** コマンドを使用してリモートシステムへのネットワークアクセスをテストします。
2. 各 PowerExchange および PowerCenter システム上の dbmover.cfg ファイルに次の NODE 文を追加して、リモートの PowerExchange Listener およびそのポートを特定します。
 NODE=(node_name,TCP,IP,ipaddress_or_hostname,port_number)
ヒント: ポート番号を決定するには、リモートシステムに PowerExchange Listener をインストールした社内の担当者に問い合わせてください。
3. ローカルの PowerExchange システムで DTLREXE ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER コンフィギュレーションファイルへの Oracle 固有の文の追加

Oracle バルクデータ移動を構成するには、Linux、UNIX、または Windows 上の dbmover.cfg コンフィギュレーションファイルに以下の Oracle 固有の文を含めることができます。

- ORA_ERRORFILE
- ORACLECODEPAGE

DBMOVER 構成メンバを編集した後、変更を有効にするには PowerExchange Listener をリスタートする必要があります。

各 DBMOVER 文の説明については、『*PowerExchange リファレンスマニュアル*』を参照してください。

ORA_ERRORFILE 文

ORA_ERRORFILE 文は、PowerExchange が Oracle バルクデータ移動操作に使用する、ユーザーがカスタマイズした SQL エラーファイルの名前を指定します。

オペレーティングシステム: Linux、UNIX、および Windows

データソース: Oracle

必須: いいえ

構文:

ORA_ERRORFILE=*file*

値: *ile_name* 変数には、PowerExchange でリカバリ可能または致命的エラーとして処理する SQL エラーコードを含むファイルのパスおよび名前を入力します。

PowerExchange では、ora8err.act というサンプルのエラーアクションファイルが PowerExchange インストールディレクトリに用意されています。

Windows ネットワークパスを指定する場合、PowerExchange でネットワークパスが正しく解析されるように、先頭にバックスラッシュ (\\) を 3 つ追加してください。

ORACLECODEPAGE 文

NLS_LANG 環境変数で UTF8 または AL32UTF8 以外の文字セットが指定されている場合、ORACLECODEPAGE 文を定義する必要があります。この文は、バルクデータ移動操作中に特定の Oracle データベースに対して PowerExchange および PowerCenter が使用するコードページを特定します。

この文で指定する 1 つまたは複数のコードページは、NLS_LANG 環境変数で識別される文字セットに一致している必要があります。NLS_LANG 環境変数は、Oracle クライアントライブラリが PowerExchange にデータを渡す方法を特定します。

オペレーティングシステム: Linux、UNIX、および Windows

データソース: Oracle

関連した文: CODEPAGE

必須: NLS_LANG で UTF8 または AL32UTF8 以外の文字セットが指定されている場合

構文:

```
ORACLECODEPAGE=( tnsname_host  
                  ,pwxc_codepage  
                  ,pc_codepage  
                )
```

パラメータ:

tnsname_host

必須。Oracle データベース向けの Oracle tnsnames.ora コンフィギュレーションファイルのエントリ。エントリは、PowerExchange がデータベースへの接続に使用するデータベースアドレスを定義します。

pwxc_codepage

必須。PowerExchange がコードページの識別に使用するコードページ番号またはエイリアス名。有効な PowerExchange コードページ値とエイリアス値を特定するには、ICUCHECK ユーティリティを使用し、レポート 5、「PowerExchange コードページ名とエイリアス」を生成します。

注: PowerExchange は、コードページへのエイリアスとして複数の共通の Oracle 文字セット名をサポートします。

pc_codepage

オプション。PowerCenter バルクデータ移動セッションに代わって、PowerExchange が Oracle に渡す SQL 文の処理を制御する名前。PowerExchange は、一般に適しているデフォルトを指定します。

このパラメータは、デフォルトが機能しない特別の環境を除いて、指定しません。例えば、*pwxc_codepage* パラメータにユーザー定義の ICU コードページを指定する場合は、このパラメータを指定する必要があります。

使用上の注意:

- 別々のデータベースごとに最大 20 個の ORACLECODEPAGE 文を dbmover.cfg コンフィギュレーションファイルに指定できます。
- PowerExchange が同じ NLS_LANG 環境変数を使用して複数の Oracle データベースにアクセスする場合は、各データベースごとに別個の ORACLECODEPAGE 文を指定する必要はありません。代わりに、単一の ORACLECODEPAGE 文を指定して、*tnsname_host* パラメータを空白のままにします。これで、指定されたコードページは、tnsnames.ora ファイルにエントリのある、すべてのデータベースに適用されます。以下の例に、*tnsname_host* パラメータなしの文を示します。

```
ORACLECODEPAGE=(,MS1252)
```

- 正しくない PowerCenter コードページ値を入力した場合、ODLNumResultCols ルーチンは通常、Oracle 戻りコード 911 をレポートします。

例: NLS_LANG 環境変数は、Korean_Korea.KO16MSWIN949 を指定し、以下の ORACLECODEPAGE 文を定義します。

```
ORACLECODEPAGE=(KO102DTL,MS949)
```

Oracle バルクデータの移動

Linux、UNIX、または Windows 上の Oracle データベースとの間でバルクデータを移動するには、以下の手順を使用します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を取得します。

- Oracle SID
- データベースサービス名（Oracle との通信に SQL*Net を使用する場合）
- Oracle ソーステーブルまたはターゲットテーブルのテーブル名
- データベースのユーザー ID とパスワード

ヒント: 移動を開始する前に、PowerExchange Navigator から Oracle ソースデータをプレビューすることができます。パーソナルメタデータプロファイルを作成し、それに対してデータベース行テストを実行します。PowerExchange Navigator には、Oracle カラムごとにデータタイプ、データ形式、CCSID などのメタデータが表示されます。

Oracle バルクデータを移動する手順

1. PowerCenter Designer で、Oracle データソースの **【ソース】 - 【PowerExchange からインポート】** をクリックするか、Oracle データターゲットの **【ターゲット】 - 【PowerExchange からインポート】** をクリックします。
2. **【PowerExchange からインポート】** ダイアログボックスで、以下の必須の属性を入力します。
 - PowerCenter Integration Service が動作するローカルシステムに Oracle データベースが存在する場合は、**【場所】** フィールドに、ローカル dbmover.cfg ファイルの LISTENER 文で指定した PowerExchange Listener のノード名を入力します。Oracle データベースが別のシステムに存在する場合は、ローカル dbmover.cfg の NODE 文で指定したノード名を入力します。
 - **【ユーザー名】** および **【パスワード】** フィールドに、Oracle のソーステーブルまたはターゲットテーブルにアクセスするためのユーザ ID とパスワードを入力します。
 - **【ソースタイプ】** のリストから、**【ORACLE】** を選択します。
 - **【TNS 名】** フィールドに、Oracle インスタンスの SID を入力します。
必要に応じてオプションの属性に値を設定します。
3. PowerCenter Designer で、Oracle ソースまたはターゲットを含むマッピングを作成します。
4. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX Oracle リレーショナルデータベース接続を構成します。次に、バルクデータ移動を実行するワークフローを開始します。

第 12 章

シーケンシャルファイルのバルクデータ移動

この章では、以下の項目について説明します。

- [シーケンシャルファイルのバルクデータ移動の概要, 151 ページ](#)
- [リモートのソースまたはターゲットへの接続の設定およびテスト, 152 ページ](#)
- [シーケンシャルファイルのバルクデータの移動, 152 ページ](#)
- [シーケンシャルファイルのデータマップのプロパティ, 153 ページ](#)
- [複数レコードのシーケンシャルファイルのグループソース処理, 160 ページ](#)
- [シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込み, 161 ページ](#)

シーケンシャルファイルのバルクデータ移動の概要

PowerExchange では、PowerCenter と連結して、z/OS 上のシーケンシャルデータセットや i5/OS、Linux、UNIX、または Windows 上のファイルとの間でバルクデータを移動できます。これらの種類のソースおよびターゲットでは、SEQ アクセス方式を使用します。

以後、「シーケンシャルファイル」という語は、z/OS 上のシーケンシャルデータセットや i5/OS、Linux、UNIX、または Windows システムでのファイルを指して使用されます。

シーケンシャルファイルソースまたはターゲットに対して、PowerExchange Navigator で SEQ アクセス方式を使用するデータマップを作成する必要があります。PowerExchange は、データマップを使用して、データおよびメタデータにアクセスし、レコードのリレーショナルタイプビューを作成します。PowerExchange では、SQL 型の文を使用してバルクデータを読み込むのにリレーショナルビューが必要になります。

また、ソースが存在するシステムで PowerExchange Listener を実行します。この PowerExchange Listener が、PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムと通信できることを確認してください。

シーケンシャルファイルを読み取る際、NRDB データマッププロパティとオペレーティングシステムファイル API で指定された内容に基づいてレコード長が特定され、リレーショナルシステムで処理する準備が整っているカラムにマップされます。

シーケンシャルファイルに書き込む際、データマップフィールドのサイズを使用してレコードの末尾にカラムデータが追加され、NRDB データマッププロパティで指定された内容に基づいて書き込まれます。

リモートのソースまたはターゲットへの接続の設定 およびテスト

リモートシステム上のデータソースまたはターゲットにアクセスするには、PowerExchange でリモートシステム上の PowerExchange Listener と通信する必要があります。

1. PowerExchange Navigator、PowerCenter Client、および PowerCenter 統合サービスを実行するシステムで、ping コマンドを使用してリモートソースまたはターゲットシステムへのネットワークアクセスをテストします。
2. 各 PowerExchange および PowerCenter システム上の dbmover.cfg ファイルに NODE 文を追加して、ソースまたはターゲットが存在するリモートノードを特定します。以下の構文を使用します。
`NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)`
ヒント: ポート番号を決定するには、リモートシステムに PowerExchange リスナをインストールした担当者に問い合わせてください。
3. ローカルの PowerExchange システムで DTLREXE という ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

シーケンシャルファイルのバルクデータの移動

以下の手順に従って、シーケンシャルファイルデータソースのバルクデータ移動操作を実行します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

1. PowerExchange ナビゲータで、ソースシーケンシャルファイルのデータマップを作成します。
2. ローカルまたはリモートの COBOL または PL/I コピーブックをインポートし（使用可能な場合）、すべてのシーケンシャルレコードまたは選択したシーケンシャルレコードのレイアウトを定義します。
3. リソースエクスプローラでデータマップを開き、非リレーショナルレコードビューまたはリレーショナルテーブルビューをクリックして、メタデータが正常に取得されたことを確認します。
4. リモートのシステム上の PowerExchange リスナにデータマップを送信するには、リソースエクスプローラでデータマップを選択し、**[ファイル] - [リモートノードへの送信]** をクリックします。
【場所】 フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定されたリモートノードのノード名を入力します。必要に応じて、他のフィールドも実行します。
注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。
5. メタデータのテーブルビュー上のデータベース行のテストを実行して、シーケンシャルファイルソースから実際のデータを返すことができることを確認します。
ヒント: 目的のデータを取得するためにデータマップを変更する必要がある場合は、変更後のデータマップでデータベース行テストを再度実行します。
6. PowerCenter Designer で、PowerExchange データマップをインポートし、マッピングを作成します。
データマップをインポートするときは、**【場所】** フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定されたリモートノードのノード名を入力します。また、**【ソースタイプ】** フィールドで、**[SEQ]** を選択します。

7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアプリケーション接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

注: PowerExchange 接続を使用してシーケンシャルターゲットファイルに書き出す場合で、PowerCenter セッションを起動する前に PowerExchange によって空のターゲットファイルを作成しておく必要がある場合は、ターゲットの **【Pre SQL】** セッションプロパティに CREATEFILE コマンドを指定します。詳細については、『*PowerExchange Interfaces for PowerCenter*』ガイドの付録「PowerCenter 用の PowerExchange インタフェースのヒント」に記載されている空ファイルの説明を参照してください。

シーケンシャルファイルのデータマップのプロパティ

バルクデータをシーケンシャルファイルに移動またはシーケンシャルファイルから移動するには、PowerExchange Navigator でファイルのデータマップを作成する必要があります。

データマップを定義して **Access Method** に **SEQ** を選択すると、シーケンシャルファイルにいくつかの属性が定義されます。SEQ アクセス方法のすべての属性の説明については、『*PowerExchange Navigator ユーザーガイド*』の「付録 D: データマップのプロパティ」を参照してください。

シーケンシャルファイルの概念

PowerExchange でシーケンシャルファイル进行处理する方法についての理解を深めるために、シーケンシャルファイルの概念の説明を確認します。

レコード形式

【データマップのプロパティ】 ダイアログボックスの **【SEQ アクセス方法】** タブで、次のいずれかのレコード形式を選択します。

- 固定長
- 変数
- デフォルト

【固定長】 または **【変数】** は、Linux、UNIX、Windows ファイルにバイナリデータが含まれ、レコード区切り文字によってレコード境界が特定されない場合にのみ選択します。レコード区切り文字は、Linux および UNIX ではラインフィード (LF) で、Windows ではキャリッジリターン/ラインフィード (CR/LF) です。

すべてのレコードの長さが同じ場合は、レコード形式に **【固定長】** を選択して、各レコードのサイズを指定します。

レコードがファイル内の整数レコードプレフィックスで決定される可変長の場合は、レコード形式に **【変数】** を選択して、レコードプレフィックスのタイプを指定します。例えば、BINARY および QUOTE SITE RDW オプションを指定して FTP でダウンロードされた z/OS レコードで VBP2 タイプを使用できます。

他のすべての状況では、オペレーティングシステムのファイル API でレコード長を特定できるように、**【デフォルト】** レコード形式を選択することをお勧めします。次の種類のファイルには、**【デフォルト】** レコード形式を指定します。

- i5/OS ファイル
- z/OS ファイル

- 区切り文字 LF または CR/LF によってレコード境界が見つかる Linux、UNIX、Windows ファイル
- Linux および UNIX の名前付きパイプ

可変長ストリームデータ

可変長レコードのレコード境界をオペレーティングシステム API から確認できない稀な状況では、レコード形式に **【変数】** および可変長データファイルタイプに **【VS】**（可変長ストリーム）を選択できます。

可変長ストリームデータ処理は、次のように行われます。

1. PowerExchange は、最も大きいレコードを保持するのに十分な大きさのデータブロックを読み取ります。
2. PowerExchange は、データマップでレコードタイプが定義された順序で、データを各レコードタイプにマップしようとします。
3. 一致が見つかったら、そのレコードタイプが使用され、バッファ内のポインタが必要なバイト数だけ移動します。

可変長ストリームデータ処理には次の制限事項があるため、最後の手段として使用することをお勧めします。

- 多くのレコードタイプを使用する場合、可変長ストリームデータ処理は拡張性に問題があります。
- ファイルの読み取りを試行する前に、すべてのレコードタイプを定義する必要があります。その結果、データマップの定義に時間がかかります。
- レコードが正しく順序付けされていない場合、ユーザーエラーが発生する可能性があります。レコード ID の状態はデータマップ内のレコードの順序に基づいており、ユーザーエラーを防ぐために PowerExchange で順序が検証されることはありません。
- ほとんどのレコードタイプは、PowerExchange がこのレコードを選択するかどうか決定する前に完全にマップされるため、かなりの CPU 時間が消費されます。
- 可変長ストリームデータマップを使用した書き込みは許可されていますが、こうしたマップは読み取りのみが想定されています。書き込みを行う場合、PowerExchange はデフォルトのレコード形式を使用します。
- ユーザーシーケンスフィールドを含むマッピングでは、レコード終端文字を追加しない z/OS および i5/OS オペレーティングシステムで、可変長ストリームデータマップを使用して正常に書き込みできることがあります。ただし、Linux、UNIX、Windows では、書き込みによって不要なレコード終端文字が生成される場合があります。

Linux、UNIX、Windows でのレコード境界の特定

Linux および UNIX では、テキストデータレコードは単一のレコード終端文字 LF (X'0A') で区切られます。Windows では、テキストデータレコードは CR/LF (X'0D0A') の 2 文字のシーケンスで区切られます。一般に、テキストファイルには終端 X'0A' 文字と競合する整数は含まれません。代わりに、テキストファイルには、メモ帳や vi などのテキストエディタで表示できる表示形式でデータが含まれます。

テキストファイルの場合、**【デフォルト】** レコード形式を選択します。区切り文字（CSV ファイルのカンマなど）を使用してフィールド境界を識別する場合も、この形式を選択します。テキストファイルが FTP で転送される場合、一般にデフォルトの ASCII オプションが使用されます。

ファイルに整数、パックされた 10 進数、浮動小数などのバイナリデータが含まれる場合、境界が各レコード内のデータの影響を受けることなく、システムでレコード境界を特定できるように、レコード形式として **【固定長】** または **【変数】** を選択します。FTP を使用してファイルを転送する場合、BINARY オプションを使用する必要があります。

i5/OS および z/OS でのレコード境界の特定

i5/OS および z/OS では、オペレーティングシステムファイル API はレコード長を返します。同じファイル API がすべてのファイルに使用されます。そのため、これらのマシンタイプではテキストデータとバイナリデータを区別する必要はありません。

最大レコードサイズ

PowerExchange が処理できる最大レコードサイズは 147,456 バイト（144KB）です。ただし、特定マシンのオペレーティングシステムでは、下限が強制されます。

i5/OS ファイル

i5/OS ネイティブシステムファイルは固定レコード長です。**【固定長】** レコード形式を選択して、レコードサイズを指定することもできますが、**【デフォルト】** レコード形式を選択する方が簡単です。

i5/OS では、変数ファイルタイプのサポートが制限されています。i5/OS 統合ファイルシステム（IFS）のファイルのみが、UNIX 系の可変長レコードを使用できます。

i5/OS ファイルは通常、パーティションに分割されています。ファイル名は *library/filename(member)* で示されます。ファイルに単一メンバしか存在しない場合、メンバコンポーネントを省略できます。省略しない場合、メンバコンポーネントには正確なメンバ名を指定するか、*ALL、*FIRST、*LAST などのオペレーティングシステムキーワードに設定する必要があります。

i5/OS が処理できる最大レコードサイズは 32,768 バイト（32 KB）です。

Linux および UNIX ファイル

PowerExchange では、Linux および UNIX でテキストデータおよびバイナリデータを処理できます。名前付きパイプからテキストデータを読み取ることもできます。

Linux および UNIX テキストデータ

テキストデータのレコード境界は、レコード終端文字 LF（X'0A'）で区切られます。テキストデータファイルには**【デフォルト】** レコード形式を選択します。ファイルは、レコード終端値と競合しない表示フィールド（CHAR、VARCHAR、NUMCHAR、UZONED など）でのみマップする必要があります。

多くの Linux および UNIX プログラムとは異なり、データに埋め込みバイナリゼロ文字が含まれる場合、PowerExchange ではテキストデータレコードを切り詰めません。PowerExchange では、以下によってこれを実現します。

- fgets() C ライブラリ関数を使用して次のレコードを読み取り、レコード終端文字によって区切られたレコードが返されます。
- ftell() C ライブラリ関数を使用して、レコードの開始位置と終了位置を特定します。

Linux および UNIX バイナリデータ

ファイルに整数、パックされた 10 進数、浮動小数フィールドなどのバイナリデータが含まれる場合、境界が各レコード内のデータの影響を受けることなく、システムでレコード境界を特定できるように、**【固定長】** または **【変数】** レコード形式を選択します。

ファイルのすべてのレコードが同じ固定長の場合、**【固定長】** を選択して、各レコードのサイズを指定します。

ファイルに可変長レコードが含まれ、レコードプレフィックスにレコードの残りのデータの長さを示す整数が含まれる場合は、**【変数】** レコード形式を選択します。Linux、UNIX、Windows アプリケーションでバイナリ

可変長レコードを書き込むことはほとんどありませんが、z/OS ファイルを Linux、UNIX、または Windows にダウンロードして処理することが便利なこともあります。よく使用される変数形式は VBP2 であり、BINARY および QUOTE SITE RDW オプションを指定して FTP でダウンロードされたファイルをマップするのに使用できます。詳細については、[「Linux、UNIX、Windows での読み取り用の z/OS ファイルのダウンロード」](#)（ページ 158）を参照してください。

ファイルにレコード長プレフィックスがあるかどうかを特定し、適切な変数形式タイプを選択するには、ファイルを 16 進数モードで表示するのが最適です。この手順では、データベース行セットでのある程度の実験が必要になることがあります。

Linux および UNIX 上の名前付きパイプ

Linux および UNIX では、PowerExchange は通常のディスクファイルと同じように名前付きパイプからテキストファイルを読み取ることができます。ただし、最終レコードにバイナリゼロが含まれる場合、レコードは切り詰められます。

名前付きパイプを処理する際、PowerExchange は `ftell()` API を使用できません。代わりに、PowerExchange は次のいずれかの値をレコード長として使用します。

- 最後のレコードの場合は、バイナリゼロ終端文字の前のバイト数
- 最後のレコード以外のレコードの場合、LF 終端文字の前のバイト数

Linux または UNIX で名前付きパイプを読み取る必要がある場合、次のいずれかの方法で制限事項に対処することができます。

- パイプを閉じる前に、データを書き込むプログラムで追加のトレーラレコードを書き込みます。
- パイプを読み込むユーザーアクセス方法プログラムを記述します。詳細については、『*PowerExchange Navigator ユーザーガイド*』の付録「ユーザーアクセス方法プログラム」を参照してください。

PowerExchange が処理できる最大レコードサイズは 147,456 バイト（144KB）です。ただし、VB12、VBX2、VBP2 など、NUM16 整数を使用する特定の変数タイプでは、32-KB 制限が強制されます。

Windows ファイル

CR/LF (X'0D0A') 文字は、Windows テキストファイルでレコードを区切ります。ただし、Linux または UNIX システムで作成された単一の LF (X'0A') 区切り文字を含むファイルを、Windows で読み取ることもできます。

PowerExchange では、SEQ データマップを使用して、Windows 上の名前付きパイプをファイルのように読み取ることはできません。Windows で名前付きパイプを読み取るには、ユーザーアクセス方法プログラムを記述する必要があります。詳細については、『*PowerExchange Navigator ユーザーガイド*』の付録「ユーザーアクセス方法プログラム」を参照してください。

その他の点では、Windows のファイル処理は Linux および UNIX のファイル処理と類似しており、テキストおよびバイナリデータファイル間での相違は同じです。

PowerExchange が処理できる最大レコードサイズは 147,456 バイト（144KB）です。ただし、VB12、VBX2、VBP2 など、NUM16 整数を使用する特定の変数タイプでは、32-KB 制限が強制されます。

z/OS ファイル

PowerExchange は、z/OS で固定長ファイルと可変長ファイルをサポートします。

z/OS のレコード形式

PowerExchange は、z/OS でほとんどのレコード形式をサポートしています。

PowerExchange では、次のタイプの z/OS ファイルの読み取りおよび書き込みを行うことができます。

- 次のレコード形式の固定長ファイル: F (固定長、非ブロック化)、FB (固定長、ブロック化)、および FS (固定長、非ブロック化、標準)
- 次のレコード形式の可変長ファイル: V (可変長、非ブロック化)、VB (可変長、ブロック化)、および VS (可変長、非ブロック化、スパン)

PowerExchange では、次のレコード形式はサポートしていません。

- VBS (可変長、ブロック化、スパン)
- U (未定義)

可能な場合は、z/OS ファイルの読み取りまたは書き込みを行うデータマップに、[デフォルト] レコード形式を選択することをお勧めします。

PowerExchange で処理できる最大レコードサイズは 147,456 バイト (144KB) で、z/OS は FS、FBS、VS の RECFM 値をサポートしています。

その他の z/OS ファイル属性

PowerExchange では、パーティション分割されたファイルのメンバの読み取りまたは書き込みを行うことができます。

PowerExchange では、USS ファイルシステムで UNIX 系の名前のファイルを読み取ることができますが、書き込むことはできません。

z/OS ファイルの動的割り当て

一般に、シーケンシャルファイルのデータマップを定義するときにファイル名を指定します。この方法により、PowerExchange は PowerExchange リスナまたは Netport ジョブの JCL で事前に定義しなくても、ファイル名を処理できます。PowerExchange は z/OS SVC99() API を使用してファイルに DD 名を動的に割り当てることで、この目的を達成します。

指定したファイル名を使用する代わりに、DD: *JCLNAME* などの名前を使用してファイルを読み取ることもできます。*JCLNAME* は、PowerExchange リスナまたは Netport ジョブの JCL で DD 文の DDNAME を定義します。

z/OS での新規ファイルの作成

PowerCenter ワークフローでは、新しい z/OS データセットを作成できます。z/OS ファイル属性を SEQ ターゲットのセッションプロパティとして定義します。

Disp セッション属性に **NEW** を選択します。また、少なくとも **RECFM**、**LRECL**、**UNIT**、および **Space** 割り当て属性を定義する必要があります。オプションで、**STORCLAS** や **MGMTCLAS** などのクラス属性を定義できます。

z/OS リスナマシンの DBMOVER 構成ファイルで、デフォルトの割り当てパラメータを文として定義できます。例えば、この方法で RECFM、LRECL、UNIT、および SPACE を定義できます。ただし、すべての属性をワークフローで明示的に定義し、PowerExchange リスナのデフォルトを使用しないことをお勧めします。

z/OS での既存のファイルの上書き

シーケンシャルファイルターゲットを持つ PowerCenter ワークフローでは、PowerExchange はディスクまたはテープ上の既存のファイルを上書きできます。接続のセッション属性で指定した値に関係なく、PowerExchange では RECFM、LRECL および BLKSIZE 属性を再利用します。

技術的には、PowerExchange は既存のファイル属性を使用する `fopen()` API を呼び出します。

PowerCenter ワークフローでは、既存のテープファイルを上書きすることもできます。ただし、特定のケースでは、プロセスが仮想テープシステムからの制約の影響を受けることがあります。PowerExchange は `fldata()` API を呼び出して、既存のテープファイルの属性を取得します。PowerExchange は、ファイルを再び開くときにこれらの属性を使用します。

z/OS テープファイル

PowerExchange は、タイプ SEQ または TAPE のデータマップを使用して、テープファイルを読み取りまたは書き込みできます。ただし、データマップで VOLSER を指定する必要がある場合は、タイプ TAPE のデータマップが必要です。

PowerExchange は、大きなブロックサイズのテープファイルに対して読み取りおよび書き込みを行うことができます。

Linux、UNIX、Windows での読み取り用の z/OS ファイルのダウンロード

z/OS 上のシーケンシャルファイルから Linux、UNIX、Windows 上のリレーショナルテーブルにバルクデータを移動する場合、z/OS から直接シーケンシャルファイルを読み取るワークフローを作成し、**【後でフィルタ】** 接続属性を指定して、フィールドからリレーショナルカラムへの CPU 集約型マッピングをオフロードすることをお勧めします。

この処理スタイルでは、ファイルを Linux、UNIX、Windows にダウンロードするデメリットを回避できます。これらのデメリットには、ダウンロードのスケジュール、Linux、UNIX、Windows でのディスク領域コスト、ネットワーク転送時のセキュリティの問題、FTP およびマッピングの問題などがあります。

ただし、特定の状況下では、z/OS ファイルを Linux、UNIX、Windows マシンにダウンロードすると便利な場合があります。次のような状況で、z/OS ファイルをダウンロードする場合があります。

- スケジュールされたプロダクションジョブの一部として、既にファイル転送がある場合。
- 統合サービスマシンからソース z/OS システムにアクセスできない場合。例えば、データが外部サイトから取得される場合です。
- 特定のマッピングおよびテスト状況では、z/OS ファイルを Linux、UNIX、Windows にダウンロードする必要がある場合があります。

次のダウンロードタイプを使用できます。

- FTP (ASCII モード)
- FTP (固定長レコード用の BINARY モード)
- FTP (可変長レコード用の BINARY モード)

FTP (ASCII モード)

次の条件に該当する場合、このダウンロードタイプを使用できます。

- z/OS ファイルに表示フィールドのみが含まれる。表示フィールドには、CHAR、VARCHAR、NUMCHAR、UZONED フィールドなどがあります。
- 文字フィールドが EBCDIC から ASCII に正しく変換される。一般に、文字データがすべてシングルバイトでデフォルトの FTP コードページが適切な場合、文字フィールドは正しく変換されます。

ほとんどの場合、【デフォルト】レコード形式を指定して、ASCII モードでダウンロードされたファイルを開くことができます。レコード区切り文字を使用してレコード長が特定されます。

FTP（固定長レコード用の BINARY モード）

ダウンロードを実行する際、次の FTP コマンドとキーワードを含めます。

- FTP
- ASCII からバイナリモードに切り替えるための BINARY キーワード
- ファイルを転送するための GET または PUT キーワード

データマップを定義する際、次のプロパティを設定します。

- レコード形式を【固定長】に設定して、各レコードのサイズを指定します。
- 数値フィールドが正しく処理されるように、【エンコード】を EBCDIC に設定します。これは、特にローエンドの Linux または Windows マシンで必要となります。
- 【コードページ】を、IBM-037 などの必要な z/OS コードページに設定します。

FTP（可変長レコード用の BINARY モード）

ダウンロードを実行する際、次の FTP コマンドとキーワードを含めます。

- FTP。
- ASCII からバイナリモードに切り替えるための BINARY キーワード。
- QUOTE SITE RDW。z/OS FTP システムが、各レコードのデータより前にレコード長プレフィックスを送信できるようにします。この方法は、RECFM=VB などの可変長ファイルでのみ使用できます。
- ファイルを転送するための GET または PUT キーワード。

データマップを定義する際、次のプロパティを設定します。

- データマッププロパティで、レコード形式を【変数】に設定し、変数タイプを **VBP2** に指定します。
- 数値フィールドが正しく処理されるように、【エンコード】を EBCDIC に設定します。特に、この設定はリトルエンディアンの Linux または Windows マシンで必要になり、レコード長プレフィックスの数値に影響を与えます。
- 【コードページ】を、IBM-037 などの必要な z/OS コードページに設定します。

ユーザーアクセス方法の例

普通でないファイル状況を処理するためのユーザーアクセス方法ライブラリを開発できます。一般的なサンプルユーザーアクセス方法用のコードは製品に付属しています。プログラムをコンパイルして、ユーザーアクセス方法ライブラリをビルドする必要があります。一般に、ユーザーアクセス方法はすべてのオペレーティングシステムで機能するため、C 言語で記述します。アクセス方法タイプ USER のデータマップにより、ファイル名、整数、文字列の各パラメータの仕様で処理を制御できます。

例: ファイルを暗号化されていない状態でディスク上に保持することはできない

外部ソースから暗号化されたファイルを受信します。セキュリティポリシーにより、このファイルを復号化された読み取り可能な形式でディスク上に保持することはできません。

ユーザーアクセス方法プログラムを呼び出すソリューションには、次の要素が含まれます。

- アクセス方法タイプ USER のデータマップは、ファイル名、「`gpg -d %FILENAME%`」などの復号化コマンド、テキストまたはバイナリのファイルタイプを指定します。

- API `popen()`を使用してパイプが作成され、`gpd` コマンドを実行してファイルが復号化され、出力が `stdout` に書き込まれます。
- テキストファイルには APIs `fgets()`、バイナリファイルには `fread()`を使用して、パイプ内の復号化されたレコードが読み取られます。
- パイプは、実行の終了時に閉じられます。

例: IBM Sterling Connect:Direct でダウンロードされた z/OS ファイルの読み取り

IBM Sterling Connect:Direct でダウンロードされた z/OS ファイルには、先頭のレコード長および末尾のレコード終端文字の両方のバイトが含まれますが、次の理由により、PowerExchange でこうしたファイルを読み取るのは困難です。

- 整数やマルチバイト文字フィールドなどのバイナリデータ内で `X'0A'`が発生すると、デフォルトのレコード形式では読み取った値を切り詰める。
- レコード終端文字のバイトをカウントしないため、先頭のレコード長が小さすぎる。最初のレコードは変数タイプ `VBP2` で読み取ることができるが、2 番目以降のレコードのレコード境界が正しくない。

ソリューションには次の要素が含まれます。

- アクセス方法タイプ `USER` のデータマップは、ファイル名、整数長のサイズ、各レコードの末尾から取り除くバイト数のパラメータを指定します。
- ユーザーアクセス方法ライブラリはファイルを読み取り、最初の 2 バイトから長さを取得し、指定したバイト数をレコードの末尾から削除し、変数タイプ `VBP2` に従って、レコードの実際のサイズと一致する長さでレコードを書き込みます。
- 1 つのレコードが使用される場合のみ、データマップは 1 回のパスでデータを読み取ることができます。

複数レコードマッピングでは、2 回のパスが必要です。最初のパスでは、アクセス方法タイプ `USER` のデータマップを使用してレコード長を収集し、2 番目のパスでは、アクセス方式タイプ `SEQ` のデータマップを使用し、変数タイプ `VBP2` を使用してファイルを読み取ります。

複数レコードのシーケンシャルファイルのグループソース処理

PowerExchange ではグループソース処理を使用して、複数のレコードタイプを持つシーケンシャルファイルからバルクデータを読み取ります。

PowerExchange のグループソース処理は、同じ物理ソースに格納されているデータを 1 回のパスで読み取ります。グループソース処理ではソースデータが複数回のパスで処理されることを防ぐことで、バルクデータ移動のパフォーマンスを強化し、ソースシステムにおけるリソース消費を削減します。

グループソース処理を使用するには、以下の作業を完了します。

1. シーケンシャルファイルデータソースに対し、複数レコードのデータマップを作成します。
SEQ アクセス方式を使用します。
2. データマップを PowerCenter にインポートします。

PowerExchange でグループソース処理を使用するには、**[PowerExchange からインポート]** ダイアログボックスで **[複数レコードのデータマップ]** オプションを選択する必要があります。

シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込み

バルクデータ移動セッションの間、PowerExchange は複数レコードデータマップによってシーケンシャルファイルソースからデータの読み取りを行い、それらの複数レコード型を非リレーショナルターゲットに書き込むことができます。この処理を *複数レコードの書き込み* と呼びます。

複数レコードの書き込みが実行されるときに、ソースシーケンス情報は維持されます。順序付けによって複数レコード書き込みを有効にするには、ソースおよびターゲット両方の定義の **「PowerExchange からインポート」** ダイアログボックスで、**「複数レコードのデータマップ」** オプションと **「シーケンスフィールドを使用」** オプションを選択します。

複数レコード書き込みは、次のタイプのターゲットタイプに対して行うことができます。

- z/OS 上の VSAM ESDS、KSDS、または RRDS データセット
- z/OS 上のシーケンシャルデータセット
- i5/OS のファイル
- Files on Linux、UNIX、および Windows

順序付けを行う複数レコードの書き込みを実行するため、PowerExchange は、グループソース処理を使用して 1 回のパスでソースデータを読み取り、グループターゲット処理を使用して 1 回のパスでターゲットにデータを書き込みます。

PowerExchange は、ターゲットに対するソースレコードのリレーションについてのメタデータを渡すためにシーケンスフィールドを生成します。PowerCenter ワークフローに対して順序付けを行う複数レコードの書き込みを有効にすると、ワークフローは複数レコードのソースを読み取り、生成されたシーケンスフィールドを使用して順序付け情報を維持し、データをソースと同じシーケンスにあるターゲットに対して書き込むことができるようになります。

シーケンシャルファイルソースのレコードと VSAM ソースのレコード間の関係を判断してシーケンスフィールドを生成するために、PowerExchange はユーザーによってデータマップに定義されているレコードプロパティを使用します。これらのレコードプロパティには、親レコード名、現在のレコードによって再定義されるベースレコードの名前、およびそのレコードがヘッダーレコードとトレーラレコードのどちらかであるかの記載が含まれます。PowerExchange はこれらのレコードプロパティを使用し、ソース定義とターゲット定義内にプライマリキー、外部キー、およびシーケンスメタデータを生成します。

PowerCenter セッションを実行すると、PowerExchange は生成されたキーの値を使用してデータを再構築し、正しいシーケンスでターゲットに書き込みます。PowerExchange は統合サービスノードの順序付けおよびキューキャッシュのデータを維持します。PowerExchange がデータをターゲットに書き込むと、生成したキーフィールドが削除され、データは正しいシーケンスのネットワークを通過してターゲットファイルに送信されます。

シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みのルールとガイドライン

シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みを行う場合は、あらかじめこの機能に関連するルールとガイドラインに目を通してください。

詳細については、『*PowerCenter 用の PowerExchange インタフェース*』を参照してください。

ソースとターゲットのファイル

ソースとターゲットには、以下のルールとガイドラインが適用されます。

- PowerExchange は VSAM ESDS、KSDS、RRDS ターゲットへの複数レコード書き込みをサポートします。PowerExchange は複数レコードの VSAM VRRDS ターゲットへの書き込みをサポートしません。
- PowerExchange では、ヘッダーレコードまたはトレーラレコード内の値の変更はサポートされません。
- ソースデータでは、子よりも先に親が出現していなければなりません。

データマップ、ソース定義、およびターゲット定義

データマップ、ソース定義、およびターゲット定義には、以下のルールとガイドラインが適用されます。

- PowerCenter では、複数レコード書き込みには新しいソース定義とターゲット定義を作成する必要があります。シーケンス情報のない既存の定義は、複数レコード書き込みでは機能しません。複数レコード書き込みを使用しないワークフローに対しては、既存の定義を引き続き使用できます。
- シーケンシャルソースまたはシーケンシャルターゲットのデータマップは、SEQ アクセス方式を使用します。VSAM ターゲットのデータマップは、アクセス方式として ESDS、KSDS、または RRDS のいずれかを使用します。
- 入力ファイル内の各レコードが 1 つのテーブルにのみ配布されるようにするため、ソースデータマップでは一意のレコード ID 条件を指定する必要があります。同一のレコードが複数のテーブルに配布されると、処理は失敗します。この状況は、データマップでレコード ID 条件を重複させるかレコード ID 条件を定義しない場合に発生する可能性があります。
- 階層全体を形成するテーブルすべてがデータマップに含まれていることを検証してください。インポートプロセスでは階層の完全性が検証されません。
データマップに必要なテーブルがすべてソース定義に含まれるようにするには、データマップ内でそれらを単純なテーブルとして定義します。単純なテーブルは単一のレコードに基づきます。
データマップに複雑なテーブルが含まれる場合、PowerExchange はそれらをソースおよびターゲット定義には含めません。複雑なテーブルは 1 つ以上のレコードに基づきます。
- データマップのどのテーブルをインポートするかを選択することはできません。PowerCenter は、データマップの単純なテーブルをすべてインポートします。
- データマップでは、各レコードに対してちょうど 1 つのテーブルが必要です。

マッピングとトランスフォーメーション

PowerCenter マッピングと PowerCenter トランスフォーメーションには、以下のルールとガイドラインが適用されます。

- マッピングトランスフォーメーションでは行が削除される場合があります。この状況が発生する場合には、マッピングによって孤立行が作成されないように対策してください。複数レコード書き込みワークフローを実行すると、孤立行はターゲットに書き込まれず、削除されます。
- マッピングでは生成されたシーケンスフィールドの位置や名前を絶対に変更してはならず、そうでない場合はそれらを削除します。
- マッピングにレコードを挿入するトランスフォーメーションはサポートされていません。
- ソーサ、ジョイナ、およびリンクトランスフォーメーションを含む、行 ID を変更する可能性のあるトランスフォーメーションは、サポートされていません。

接続

接続と接続プロパティには、以下のルールとガイドラインが適用されます。

- 複数レコード書き込みを実行できるのは、PWXPC NRDB バッチ接続を使用する場合のみです。PowerExchange ODBC 接続はサポートされません。

- **【オフロード処理】** のソース接続属性とターゲット接続属性に、**【後でフィルタ】** を選択します。異なる値を選択すると、その値は PowerExchange によって **【後でフィルタ】** に変更されます。

PowerExchange および PWXPC は、z/OS に出力レコードを書き込む前に、Integration Service マシン上でオフロード処理を実行します。

- マルチスレッド処理は、シーケンシャルまたは VSAM ソース接続またはターゲット接続にはサポートされません。

【ワーカースレッド】 接続属性を設定した場合、その設定は無視されます。

- **【Confirm Write（書き込みの確認）】** のソース接続属性とターゲット接続属性に、**【OFF】** を選択します。他の値を選択すると、PowerExchange によって値が **【Off】** に変更されます。
- 場合によっては、**【CSQ_MEMSIZE】** パラメータの値（**【PWX オーバーライド】** 接続属性の）を変更する必要があるかもしれません。このパラメータは、キャッシュが複数レコード書き込みに消費できる最大メモリ容量を定義します。

セッションのプロパティ

PowerCenter セッションプロパティには、以下のルールとガイドラインが適用されます。

- パイプラインのパーティション化は、シーケンス有効で定義されたソースやターゲットの読み取りや書き込みではサポートされていません。
- シーケンシャルまたは VSAM ソースおよびターゲットには、**【ファイル名のオーバーライド】** プロパティを指定できます。
- ソースには、どのブロックもフラッシュすることなく全グループで発生できる最大のブロックフラッシュ数を**指定したブロックの後にフラッシュ**プロパティで指定します。このプロパティは、すべてのブロックが定期的に必ずフラッシュされるように定義します。
- z/OS 上の SEQ ターゲットの場合、PowerExchange ではセッションプロパティで必要に応じて定義する z/OS データセット割り当てプロパティに基づいてターゲットデータセットを割り当てることができます。

シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みの実行

複数レコードシーケンシャルファイルソースから複数レコードシーケンシャルファイルターゲットまたは VSAM ターゲットに正しい物理シーケンスでデータを転送するには、複数レコード書き込みを実行するセッションを設定します。

1. シーケンシャルファイルデータソースの複数レコードデータマップをまだ作成していない場合は、作成します。

次の情報を入力します。

- 各レコードタイプを特定するには、フィールドの **【Record ID Values（レコード ID 値）】** ボックスにレコード ID 値を割り当てます。
- 各レコードに単純なテーブルを割り当てます。複雑なテーブルは作成しないでください。
- 必要に応じ、各レコードに **【Parent Record Name（親レコード名）】**、**【Redefines Record Name（レコード名の再定義）】**、または **【Header/Trailer Record（ヘッダー/トレーラレコード）】** オプションプロパティを割り当てます。これらのプロパティは、ソース定義とターゲット定義内にシーケンス情報を生成するために使用されます。

詳細については、『PowerExchange ナビゲータユーザーガイド』を参照してください。

2. ソースとターゲットの **【ソースタイプ】** 値が異なる場合は、複数レコードデータマップを作成します。ソースとターゲットのデータソースタイプが同じであれば、ソースとターゲットの両方に同じデータマップを使用できます。

3. PowerCenter Designer で、**【ソース】 > 【PowerExchange からインポート】** の順にクリックします。
4. **【PowerExchange からインポート】** ダイアログボックスで、必須情報を入力します。

次の表では、必須情報を示します。

属性	説明
場所	ソースファイルが存在するノードの名前。この値は、PowerExchange dbmover.cfg ファイル内の NODE 文の名前と一致する必要があります。
ユーザー名	ソースに接続するデータベース権限を持ったユーザーの名前。
パスワード	ユーザー名に関連付けられているパスワード。
複数レコードのデータマップ	選択肢として複数レコードデータマップを表示するかどうか。 このチェックボックスをオンにします。
シーケンスフィールドの使用	複数レコード書き込み操作のシーケンスフィールドを生成するかどうか。 このチェックボックスをオンにします。
ソースタイプ	データソースタイプ。ソースとターゲットのソースタイプは同じである必要があります。 次のいずれかの値を選択します。 - シーケンシャルファイル場合、SEQ - VSAM データセットの場合、ESDS、KSDS、または RRDS

このダイアログボックスの詳細については、『PowerCenter 用の PowerExchange インタフェース』を参照してください。

5. **【接続】** をクリックします。
使用可能な複数レコードデータマップが **【選択するデータマップ】** ボックスに表示されます。
6. インポートするデータマップを選択します。
7. **【OK】** をクリックします。
ソース定義がワークスペースに表示されます。
8. **【ターゲット】 > 【PowerExchange からのインポート】** をクリックします。
9. **【PowerExchange からインポート】** ダイアログボックスで、必須情報を入力します。

次の表では、必須情報を示します。

属性	説明
場所	ターゲットファイルが存在するノードの名前。この値は、PowerExchange dbmover.cfg ファイル内の NODE 文の名前と一致する必要があります。
ユーザー名	ターゲットに接続するデータベース権限を持ったユーザーの名前。
パスワード	ユーザー名に関連付けられているパスワード。
複数レコードのデータマップ	選択肢として複数レコードデータマップを表示するかどうか。 このチェックボックスをオンにします。

属性	説明
シーケンスフィールドの使用	複数レコード書き込み操作のシーケンスフィールドを生成するかどうか。 このチェックボックスをオンにします。
ソースタイプ	データターゲットタイプ。ソースとターゲットのソースタイプは同じである必要があります。 次のいずれかの値を選択します。 - シーケンシャルファイルの SEQ。 - VSAM データセットの ESDS、KSDS、または RRDS

注: 複数レコード書き込みのシーケンシャルファイルターゲット定義は、ソース定義をドラッグしてドロップする方法では作成できません。

10. **【接続】** をクリックします。

使用可能な複数レコードデータマップが **【選択するデータマップ】** ボックスに表示されます。

11. ソースとターゲットの **【ソースタイプ】** 値が同じであれば、ソース定義に選択した同じデータマップをターゲット定義にも選択してください。同じでない場合は、ターゲット用にデータマップを選択してください。

12. **【OK】** をクリックします。

ターゲット定義がワークスペースに表示されます。

13. ソース定義、ソース修飾子トランスフォーメーション、およびターゲット定義を含むマッピングを作成します。

ソース修飾子トランスフォーメーションとターゲット定義の間に、[「シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みのルールとガイドライン」 \(ページ 161\)](#)に記載されている要件を満たす情報を含めることができます。

14. 必要に応じ、ソースおよびターゲットの PWX NRDB バッチ接続を定義します。

15. PowerCenter セッションプロパティを定義します。

- a. Task Developer で、セッションをダブルクリックして編集します。
- b. **【マッピング】** タブで、**【ソース】** ビューをクリックします。
- c. 必要に応じて、**【プロパティ】** で、ソースファイルの名前に **【ファイル名のオーバーライド】** プロパティを設定します。
- d. 必要に応じ、追加のソースプロパティを定義します。
- e. **【マッピング】** タブで、**【ターゲット】** ビューをクリックします。
- f. 必要に応じて、**【プロパティ】** で、セッションで割り当てるファイルの名前に **【ファイル名のオーバーライド】** プロパティを設定します。
- g. 必要に応じ、追加のターゲットプロパティを定義します。

注: PowerCenter セッションを起動する前に PowerExchange で空のターゲットファイルが作成されるようにするには、ターゲットの **【Pre SQL】** セッションプロパティに CREATEFILE コマンドを指定します。詳細については、『*PowerExchange Interfaces for PowerCenter*』ガイドの付録「PowerCenter 用の PowerExchange インタフェースのヒント」に記載されている空ファイルの説明を参照してください。

- h. **【OK】** をクリックします。

第 13 章

VSAM バルクデータ移動

この章では、以下の項目について説明します。

- [VSAM バルクデータ移動の概要, 166 ページ](#)
- [VSAM バルクデータ移動の構成, 167 ページ](#)
- [VSAM バルクデータの移動, 169 ページ](#)
- [複数レコードの VSAM データセットのグループソース処理, 171 ページ](#)
- [シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込み, 171 ページ](#)

VSAM バルクデータ移動の概要

PowerExchange では、PowerCenter と連結して次のタイプの VSAM データセットとの間でバルクデータを移動できます。

入力順のデータセット (ESDS)

作成順にレコードを格納するデータセットで、最新のレコードが最後にきます。ESDS にインデックスはありません。相対バイトアドレス (RBA) を使用するか入力順に基づいた順序で ESDS レコードに直接アクセスできます。PowerExchange は、ESDS データセットに順次アクセスします。

キー順のデータセット (KSDS)

キー順にロードされインデックスで制御されるレコードを含むデータセットです。キーまたは RBA を使用するかキー順に基づいた順序で KSDS に直接アクセスできます。

デフォルトで PowerExchange は、WHERE 句が指定されていない場合に、KSDS レコードに順次アクセスします。

以下の条件を満たす場合、PowerExchange はキーを使用してレコードにアクセスします。

- WHERE 句が指定されている。
- KSDS キーのプライマリフィールドが、複数ある WHERE 句条件のいずれかで使用されている。

例えば、実行時に置き換えられる疑問符がルックアップ条件に含まれている場合、PowerExchange は KSDS ルックアップソースを使用する PowerCenter ルックアップトランスフォーメーションを実行するために、キー済みアクセスを使用します。WHERE 句を処理するときは、可能な限り必ずキーを使用して KSDS ファイルにアクセスします。

PowerExchange ではファイル名のオーバーライドを使用して指定する代替インデックスを介して KSDS ファイルにアクセスできます。実行時にキー済みアクセスと順次アクセスのどちらを使用するのは、ファイルが開かれてキーの開始位置と長さが決定された後で決まります。

データマップを定義するときに順次アクセスのデータマップを最適化するには、CI アクセスを有効化し、データバッファ数を高い値に設定します。キー済みアクセスのデータマップを最適化するには、インデックスバッファ数を高い値に設定し、データバッファ数を低い値（2 など）に設定します。

相対レコードデータセット（RRDS）

固定長レコードを含むデータセットです。RRDS は、相対レコード番号（RRN）を使用して順次または直接アクセスされます。PowerExchange は、RRN に基づいて RRDS データセットに順次アクセスします。

可変長関連レコードのデータセット（VRRDS）

可変長レコードを含むデータセットです。VRRDS は、相対レコード番号（RRN）を使用して順次または直接アクセスできます。PowerExchange は、RRN に基づいて VRRDS レコードに順次アクセスします。

PowerExchange は、基本フォーマットまたは拡張フォーマットの VSAM データセットを処理できます。また、圧縮データを含むデータセットも処理できます。

VSAM はリレーショナルデータベースではないため、PowerExchange Navigator でデータマップを作成する必要があります。PowerExchange は、データマップを使用して、VSAM データおよびメタデータにアクセスし、レコードの行タイプのリレーショナルビューを作成します。PowerExchange では、SQL 型の文を使用してバルクデータを読み書きするのにリレーショナルビューが必要になります。

VSAM データマップを作成するとき、VSAM データセットタイプに対応するアクセス方法を選択する必要があります。以下の VSAM アクセス方法を使用できます。

- ESDS
- KSDS
- RRDS（固定長および可変長の相対レコードデータセット用）

VSAM データセットのにコピーブックを使用できる場合は、データマップ用にインポートして、レコードごとの詳細レイアウトを取得できます。コピーブックをインポートしない場合は、VSAM レコードのフィールドを手動で定義する必要があります。

また、VSAM ソースまたはターゲットが存在する MVS システムで PowerExchange Listener を実行する必要があります。この PowerExchange Listener が、PowerExchange Navigator、PowerCenter クライアント、および PowerCenter Integration Service が動作するシステムと通信できることを確認してください。

VSAM バルクデータ移動の構成

バルクデータ移動操作用に PowerExchange を構成するには、PowerExchange 管理者がリモート z/OS システムで PowerExchange Listener への接続を設定およびテストし、オプションとして、DBMOVER 構成メンバに VSAM 文および SHOW_THREAD_PERF 文を追加します。

制限: VSAM ESDS、RRDS、または VRRDS データセットへデータを書き込む場合、PowerExchange では挿入操作しか実行できません。KSDS データセットへデータを書き込む場合、挿入、更新、削除を実行できます。

リモートのソースまたはターゲットへの接続の設定およびテスト

リモートシステム上のデータソースまたはターゲットにアクセスするには、PowerExchange でリモートシステム上の PowerExchange Listener と通信する必要があります。

1. PowerExchange Navigator、PowerCenter Client、および PowerCenter 統合サービスを実行するシステムで、ping コマンドを使用してリモートソースまたはターゲットシステムへのネットワークアクセスをテストします。

2. 各 PowerExchange および PowerCenter システム上の dbmover.cfg ファイルに NODE 文を追加して、ソースまたはターゲットが存在するリモートノードを特定します。以下の構文を使用します。

NODE=(node_name,TCPIP,ipaddress_or_hostname,port_number)

ヒント: ポート番号を決定するには、リモートシステムに PowerExchange リスナをインストールした担当者に問い合わせてください。

3. ローカルの PowerExchange システムで DTLREXE という ping ユーティリティを実行して、NODE 文で定義したリモートの PowerExchange Listener への接続をテストします。

関連項目：

- [「PowerExchange Listener の管理」 \(ページ 47\)](#)

DBMOVER 構成メンバへの VSAM 固有の文の追加

VSAM バルクデータ移動を構成する場合、VSAM ソースまたはターゲットシステムの DBMOVER 構成メンバにオプションの VSAM 文を含めることができます。

ヒント: 通常の方法では、このグローバルパラメータを使用するのではなく、個々のバルクデータ移動に対してデータマップでこれらの値を設定します。

DBMOVER 構成メンバを編集した後、変更を有効にするには PowerExchange Listener をリスタートする必要があります。

各 DBMOVER 文の説明については、『*PowerExchange* リファレンスマニュアル』を参照してください。

VSAM 文

VSAM 文は、PowerExchange が VSAM データセットの処理中にデータおよびインデックスコントロールインターバルのために使用するバッファの数を指定します。

パフォーマンスの最適化とシステムによって管理されるバッファリングの詳細は、DFSMS のマニュアルに記載されています。

オペレーティングシステム: z/OS

データソース: VSAM データセット

必須: 不要

構文:

```
VSAM=( [BUFNI=index_buffers]
       [,BUFND=data_buffers]
)
```

パラメータ:

BUFNI=index_buffers

オプション。PowerExchange が VSAM インデックスコントロールインターバルのために使用する I/O バッファの数。バッファとは、インデックス内のコントロールインターバルのサイズです。

index_buffers 変数には、以下のいずれかの値を入力します。

- 0: オペレーティングシステムのデフォルトを使用する場合
- 1 から 255 までの数値

BUFND=data_buffers

オプション。PowerExchange が VSAM データコントロールインターバルのために使用する I/O バッファの数。バッファとは、データコンポーネント内のコントロールインターバルのサイズです。

`data_buffers` 変数には、以下のいずれかの値を入力します。

- 0: オペレーティングシステムのデフォルトを使用する場合
- 2 から 255 までの数値

注: BUFND には 1 は指定できません。

使用上の注意:

- インデックスバッファを追加すると、高レベルインデックスの一部または全部が常駐することにより、パフォーマンスが向上します。バッファを追加すると、キーダイレクト処理のために高レベルインデックスレコードが DASD から取得される回数が最小限に抑えられます。

ヒント: 通常は、PowerExchange がデータおよびインデックスコントロールインターバルのために使用するバッファの数は、グローバル VSAM 文を使って設定するのではなく、データマップで個々のバルクデータ移動操作について設定します。

- データバッファの数を増やすほど、制御領域が分割されているときのダイレクト挿入または更新、およびスパンレコードへのアクセスのパフォーマンスが向上します。
- 指定可能なバッファの最大数は、255 (254 個のデータバッファと 1 個の挿入バッファ) です。

VSAM バルクデータの移動

VSAM バルクデータの移動操作を行うには、以下の手順を使用します。この手順では、PWXPC を使用して PowerExchange を PowerCenter と統合するものとします。

始める前に、以下の情報を取得します。

- 完全修飾された VSAM データセット名
- ソースまたはターゲットレコードのレイアウト定義に使用する COBOL または PL/I のコピーブックの完全修飾されたデータセット名およびメンバ名
- z/OS ユーザー ID とパスワード (PowerExchange の SECURITY 文の設定により必要な場合)

VSAM バルクデータを移動する手順

1. PowerExchange Navigator でデータマップを作成します。以下の情報を含めます。

- ソースまたはターゲットとして使用している VSAM データセットのタイプに応じて、**[アクセス方法]** リストで **ESDS**、**KSDS**、または **RRDS** を選択します。
- **[レコード定義のインポート]** オプションを選択して、VSAM レコードのメタデータを COBOL または PL/I のコピーブックからインポートするようプロンプト表示します。このオプションを選択しない場合は、**[ファイル] - [コピーブックのインポート]** をクリックすることでコピーブックを後でインポートできます。
- PowerExchange で 8 バイトの XRBA または RRN 値をプレフィックスとして各レコードに追加する場合、ESDS データセットには **[XRBA 値のプレフィックスレコード]** オプションを、RRDS データセットには **[RRN 値のプレフィックスレコード]** オプションをそれぞれ選択します。いずれかのオプションを選択する場合、XRBA または RRN の長さに合わせてデータマップのレコードレイアウトに十分な領域があることを確認してください。
- VSAM データセットの読み込み時に個々のデータレコードではなくコントロールインターバル全体の内容に PowerExchange からアクセスする場合は、**[CI ACCESS]** オプションを選択します。制御間隔からすべてのデータを抽出する場合、このオプションによって、シーケンシャル読み取りのパフォーマンス

スを向上させることができることがあります。ただし、WHERE 句を指定した場合には、このオプションではパフォーマンスは向上しません。

- KSDS データセットでは、**【オフロード処理】** オプションを設定して、z/OS システム上の PowerExchange Listener からのデータレコードを変換および統合するための処理を、抽出処理を行うシステムに PowerExchange でオフロードするかどうかを制御します。通常、PowerCenter Client および PowerCenter Integration Service が動作するシステムが抽出システムになります。
- **【データバッファ数】** フィールドに、PowerExchange が VSAM に割り当てを要求する、データコントロールインターバルに使用する I/O バッファ数を入力します。順次アクセスのパフォーマンスを高めるには、バッファ数を増やします。KSDS、RRDS、および VRDS データセットの場合にのみ、**【インデックスバッファ数】** フィールドでインデックスコントロールインターバルに使用する I/O バッファ数を指定することもできます。これらのバッファ値により、z/OS システム上にある DBMOVER 構成メンバの VSAM 文（指定されている場合）がオーバーライドされます。

必要に応じて他のフィールドを入力します。操作が完了したら、**【完了】** をクリックします。

制限: PowerExchange では、圧縮データまたはスパンされたレコードに対して CI ACCESS オプションをサポートしていません。

【レコード定義のインポート】 オプションを選択しなかった場合は、手順 [「VSAM バルクデータの移動」](#) (ページ 169)に進みます。

【レコード定義のインポート】 オプションを選択すると、**【VSAM コピーブック-ソースの詳細】** ページが表示されます。

2. ローカルまたはリモートの COBOL または PL/I コピーブックをインポートし（使用可能な場合）、すべての VSAM レコードまたは選択した VSAM レコードのレイアウトを定義します。

コピーブックをインポートするための構成オプションの設定方法に応じて、レコード定義をインポートするかどうか、および代替定義がコピーブックに存在する場合に使用するレコード定義またはフィールド定義を指定するかどうかをレコードごとに指示するため、PowerExchange Navigator からプロンプト表示することができます。

3. リソースエクスプローラでデータマップを開き、非リレーショナルレコードビューまたはリレーショナルテーブルビューをクリックして、メタデータが正常に取得されたことを確認します。
4. リモートの z/OS システム上の PowerExchange Listener にデータマップを送信するには、リソースエクスプローラでデータマップを選択し、**【ファイル】 - 【リモートノードへの送信】** をクリックします。

【場所】 フィールドに、ローカルの PowerExchange システムにある dbmover.cfg ファイルの NODE 文で指定された PowerExchange Listener ノード名を入力します。必要に応じて、他のフィールドも実行します。

注: この手順を実行しないと、データベース行のテストを実行するときに、データマップをリモートノードに送信するように求められます。

5. メタデータのテーブルビュー上のデータベース行のテストを実行して、VSAM ソースデータベースから実際のデータを返すことができることを確認します。

ヒント: 目的のデータを取得するためにデータマップを変更する必要がある場合は、変更後のデータマップでデータベース行テストを再度実行します。

6. PowerCenter Designer で、PowerExchange データマップをインポートし、マッピングを作成します。
データマップをインポートする場合、**【場所】** フィールドに、dbmover.cfg ファイルの NODE 文で指定した PowerExchange Listener ノード名を入力します。また、**【ソースタイプ】** フィールド（ターゲットにも表示される）で **【VSAM】** も指定します。
7. PowerCenter Workflow Manager で、ワークフローおよびセッションを定義し、PWX NRDB バッチアプリケーション接続を設定します。次に、バルクデータ移動を実行するワークフローを開始します。

複数レコードの VSAM データセットのグループソース処理

PowerExchange ではグループソース処理を使用して、複数のレコードタイプを持つ ESDS、KSDS、または RRDS データセットからバルクデータを読み取ります。

PowerExchange のグループソース処理は、同じ物理ソースに格納されているデータを 1 回のパスで読み取ります。グループソース処理ではソースデータが複数回のパスで処理されることを防ぐことで、バルクデータ移動のパフォーマンスを強化し、ソースシステムにおけるリソース消費を削減します。

グループソース処理を使用するには、以下の作業を完了します。

1. VSAM データソースに対し、複数レコードのデータマップを作成します。
ESDS、KSDS、または RRDS アクセス方式を使用します。
2. データマップを PowerCenter にインポートします。
PowerExchange でグループソース処理を使用するには、**[PowerExchange からインポート]** ダイアログボックスで **[複数レコードのデータマップ]** オプションを選択する必要があります。

シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込み

バルクデータ移動セッションの間、PowerExchange は複数レコードデータマップによって z/OS 上の VSAM データセットからデータの読み取りを行い、単一のパスに入ったそれらの複数レコード型を非リレーショナルターゲットに書き込むことができます。この処理を *複数レコードの書き込み* と呼びます。

複数レコードの書き込みが実行されるときに、ソースシーケンス情報は維持されます。順序付けによって複数レコード書き込みを有効にするには、ソースおよびターゲット両方の定義の **[PowerExchange からインポート]** ダイアログボックスで、**[複数レコードのデータマップ]** オプションと **[シーケンスフィールドを使用]** オプションを選択します。

複数レコード書き込みは、VSAM ESDS、KSDS、または RRDS ソースから次のターゲットタイプに対して行うことができます。

- z/OS 上の VSAM ESDS、KSDS、または RRDS データセット
- z/OS 上のシーケンシャルデータセット
- i5/OS のファイル
- Files on Linux、UNIX、および Windows

順序付けを行う複数レコードの書き込みを実行するため、PowerExchange は、グループソース処理を使用して 1 回のパスでソースデータを読み取り、グループターゲット処理を使用して 1 回のパスでターゲットにデータを書き込みます。

PowerExchange は、ターゲットに対するソースレコードのリレーションについてのメタデータを渡すためにシーケンスフィールドを生成します。PowerCenter ワークフローに対して順序付けを行う複数レコードの書き込みを有効にすると、ワークフローは複数レコードのソースを読み取り、生成されたシーケンスフィールドを使用して順序付け情報を維持し、データをソースと同じシーケンスにあるターゲットに対して書き込むことができますようになります。

シーケンシャルファイルソースのレコードと VSAM ソースのレコード間の関係を判断してシーケンスフィールドを生成するために、PowerExchange はユーザーによってデータマップに定義されているレコードプロパティ

ィを使用します。これらのレコードプロパティには、親レコード名、現在のレコードによって再定義されるベースレコードの名前、およびそのレコードがヘッダーレコードとトレーラレコードのどちらかであるかの記載が含まれます。PowerExchange はこれらのレコードプロパティを使用し、ソース定義とターゲット定義内にプライマリキー、外部キー、およびシーケンスメタデータを生成します。

PowerCenter セッションを実行すると、PowerExchange は生成されたキーの値を使用してデータを再構築し、正しいシーケンスでターゲットに書き込みます。PowerExchange は統合サービスノードの順序付けおよびキューキャッシュのデータを維持します。PowerExchange がデータをターゲットに書き込むと、生成したキーフィールドが削除され、データは正しいシーケンスのネットワークを通してターゲットファイルに送信されます。

VSAM ターゲットまたはシーケンシャルファイルターゲットへの複数レコード書き込みのルールとガイドライン

シーケンシャルファイルターゲットまたは VSAM ターゲットへの複数レコード書き込みを行う場合は、あらかじめこの機能に関連するルールとガイドラインに目を通してください。

詳細については、『*PowerCenter 用の PowerExchange インタフェース*』を参照してください。

ソースとターゲットのファイル

ソースとターゲットには、以下のルールとガイドラインが適用されます。

- PowerExchange は VSAM ESDS、KSDS、RRDS ターゲットへの複数レコード書き込みをサポートします。PowerExchange は複数レコードの VSAM VRRDS ターゲットへの書き込みをサポートしません。
- PowerExchange では、ヘッダーレコードまたはトレーラレコード内の値の変更はサポートされません。
- ソースデータでは、子よりも先に親が出現していなければなりません。

データマップ、ソース定義、およびターゲット定義

データマップ、ソース定義、およびターゲット定義には、以下のルールとガイドラインが適用されます。

- PowerCenter では、複数レコード書き込みには新しいソース定義とターゲット定義を作成する必要があります。シーケンス情報のない既存の定義は、複数レコード書き込みでは機能しません。複数レコード書き込みを使用しないワークフローに対しては、既存の定義を引き続き使用できます。
- VSAM ソースまたは VSAM ターゲットのデータマップは、アクセス方式として ESDS、KSDS、または RRDS のいずれかを使用します。シーケンシャルファイルソースまたはシーケンシャルファイルターゲットのデータマップは、SEQ アクセス方式を使用します。
- 入力ファイル内の各レコードが 1 つのテーブルにのみ配布されるようにするため、ソースデータマップでは一意のレコード ID 条件を指定する必要があります。同一のレコードが複数のテーブルに配布されると、処理は失敗します。この状況は、データマップでレコード ID 条件を重複させるかレコード ID 条件を定義しない場合に発生する可能性があります。
- 階層全体を形成するテーブルすべてがデータマップに含まれていることを検証してください。インポートプロセスでは階層の完全性が検証されません。

データマップで必要なテーブルがすべてソース定義に含まれるようにするには、データマップ内でそれらを単純なテーブルとして定義します。単純なテーブルは単一のレコードに基づきます。

データマップに複雑なテーブルが含まれる場合、PowerExchange はそれらをソースおよびターゲット定義には含めません。複雑なテーブルは 1 つ以上のレコードに基づきます。

- データマップのどのテーブルをインポートするかを選択することはできません。PowerCenter は、データマップの単純なテーブルをすべてインポートします。
- データマップでは、各レコードに対してちょうど 1 つのテーブルが必要です。

マッピングとトランスフォーメーション

PowerCenter マッピングと PowerCenter トランスフォーメーションには、以下のルールとガイドラインが適用されます。

- マッピングトランスフォーメーションでは行が削除される場合があります。この状況が発生する場合には、マッピングによって孤立行が作成されないように対策してください。複数レコード書き込みワークフローを実行すると、孤立行はターゲットに書き込まれず、削除されます。
- マッピングでは生成されたシーケンスフィールドの位置や名前を絶対に変更してはならず、そうでない場合はそれらを削除します。
- マッピングにレコードを挿入するトランスフォーメーションはサポートされていません。
- ソーサ、ジョイナ、およびリンクトランスフォーメーションを含む、行 ID を変更する可能性のあるトランスフォーメーションは、サポートされていません。

接続

接続と接続プロパティには、以下のルールとガイドラインが適用されます。

- 複数レコード書き込みを実行できるのは、PWXPC NRDB バッチ接続を使用する場合のみです。PowerExchange ODBC 接続はサポートされません。
- [オフロード処理]** のソース接続属性とターゲット接続属性に、**[後でフィルタ]** を選択します。異なる値を選択すると、その値は PowerExchange によって **[後でフィルタ]** に変更されます。
PowerExchange および PWXPC は、z/OS に出力レコードを書き込む前に、Integration Service マシン上でオフロード処理を実行します。
- マルチスレッド処理は、シーケンシャルまたは VSAM ソース接続またはターゲット接続にはサポートされません。
[ワーカースレッド] 接続属性を設定した場合、その設定は無視されます。
- [Confirm Write (書き込みの確認)]** のソース接続属性とターゲット接続属性に、**[OFF]** を選択します。他の値を選択すると、PowerExchange によって値が **[Off]** に変更されます。
- 場合によっては、**[CSQ_MEMSIZE]** パラメータの値 (**[PWX オーバーライド]** 接続属性の) を変更する必要があるかもしれません。このパラメータは、キャッシュが複数レコード書き込みに消費できる最大メモリ容量を定義します。

セッションのプロパティ

PowerCenter セッションプロパティには、以下のルールとガイドラインが適用されます。

- パイプラインのパーティション化は、シーケンス有効で定義されたソースやターゲットの読み取りや書き込みではサポートされません。
- シーケンシャルまたは VSAM ソースおよびターゲットには、**[ファイル名のオーバーライド]** プロパティを指定する必要があります。
- ソースには、どのブロックもフラッシュすることなく全グループで発生できる最大のブロックフラッシュ数を **指定したブロックの後にフラッシュ** プロパティで指定します。このプロパティは、すべてのブロックが定期的に必ずフラッシュされるように定義します。

シーケンシャルターゲットまたは VSAM ターゲットへの複数レコード書き込みの実行

マルチレコード VSAM ソースからマルチレコードシーケンシャルターゲットまたは VSAM ターゲットに正しい物理シーケンスでデータを転送するには、複数レコード書き込みを実行するセッションを設定します。

- まだ VSAM データソースの複数レコードデータマップを作成していない場合は、作成します。

次の情報を入力します。

- 各レコードタイプを特定するには、フィールドの **[Record ID Values (レコード ID 値)]** ボックスにレコード ID 値を割り当てます。
- 各レコードに単純なテーブルを割り当てます。複雑なテーブルは作成しないでください。
- 必要に応じ、各レコードに **[Parent Record Name (親レコード名)]**、**[Redefines Record Name (レコード名の再定義)]**、または **[Header/Trailer Record (ヘッダー/トレーラレコード)]** オプションプロパティを割り当てます。これらのプロパティは、ソース定義とターゲット定義内にシーケンス情報を生成するために使用されます。

詳細については、『PowerExchange ナビゲータユーザーガイド』を参照してください。

2. ソースとターゲットの **[ソースタイプ]** 値が異なる場合は、複数レコードデータマップを作成します。
ソースとターゲットのデータソースタイプが同じであれば、ソースとターゲットの両方に同じデータマップを使用できます。
3. PowerCenter Designer で、**[ソース] > [PowerExchange からインポート]** の順にクリックします。
4. **[PowerExchange からインポート]** ダイアログボックスで、必須情報を入力します。

次の表では、必須情報を示します。

属性	説明
場所	ソースファイルが存在するノードの名前。この値は、PowerExchange dbmover.cfg ファイル内の NODE 文の名前と一致する必要があります。
ユーザー名	ソースに接続するデータベース権限を持ったユーザーの名前。
パスワード	ユーザー名に関連付けられているパスワード。
Multi-Record Datamaps(複数レコードのデータマップ)	選択肢として複数レコードデータマップを表示するかどうか。 このチェックボックスをオンにします。
シーケンスフィールドの使用	複数レコード書き込み操作のシーケンスフィールドを生成するかどうか。 このチェックボックスをオンにします。
ソースタイプ	データソースのタイプです。 VSAM データソースの場合は、ソースタイプとして ESDS、KSDS、または RRDS のいずれかを選択します。

このダイアログボックスの詳細については、「PowerCenter 用の PowerExchange インタフェース」を参照してください。

5. **[接続]** をクリックします。
使用可能な複数レコードデータマップが **[選択するデータマップ]** ボックスに表示されます。
6. インポートするデータマップを選択します。
7. **[OK]** をクリックします。
ソース定義がワークスペースに表示されます。
8. **[ターゲット] > [PowerExchange からのインポート]** をクリックします。
9. **[PowerExchange からインポート]** ダイアログボックスで、必須情報を入力します。

次の表では、必須情報を示します。

属性	説明
場所	ソースファイルが存在するノードの名前。この値は、PowerExchange dbmover.cfg ファイル内の NODE 文の名前と一致する必要があります。
ユーザー名	ソースに接続するデータベース権限を持ったユーザーの名前。
パスワード	ユーザー名に関連付けられているパスワード。
Multi-Record Datamaps(複数レコードのデータマップ)	選択肢として複数レコードデータマップを表示するかどうか。 このチェックボックスをオンにします。
シーケンスフィールドの使用	複数レコード書き込み操作のシーケンスフィールドを生成するかどうか。 このチェックボックスをオンにします。
ソースタイプ	データターゲットタイプ。 次のうち 1 つのソースタイプを選択します。 - シーケンシャルファイルの SEQ。 - VSAM データセットの ESDS、KSDS、または RRDS

注: 複数レコード書き込みのシーケンシャルファイルターゲット定義は、ソース定義をドラッグしてドロップする方法では作成できません。

10. **【接続】** をクリックします。

使用可能な複数レコードデータマップが **【選択するデータマップ】** ボックスに表示されます。

11. ソースとターゲットの **【ソースタイプ】** 値が同じであれば、ソース定義に選択した同じデータマップをターゲット定義にも選択してください。同じでない場合は、ターゲット用にデータマップを選択してください。

12. **【OK】** をクリックします。

ターゲット定義がワークスペースに表示されます。

13. ソース定義、ソース修飾子トランスフォーメーション、およびターゲット定義を含むマッピングを作成します。

ソース修飾子トランスフォーメーションとターゲット定義の間に、[「VSAM ターゲットまたはシーケンシャルファイルターゲットへの複数レコード書き込みのルールとガイドライン」](#) (ページ 172)に記載されている要件を満たす情報を含めることができます。

14. 必要に応じ、ソースおよびターゲットの PWX NRDB バッチ接続を定義します。

15. PowerCenter セッションプロパティを定義します。

- a. Task Developer で、セッションをダブルクリックして編集します。
- b. **【マッピング】** タブで、**【ソース】** ビューをクリックします。
- c. **【プロパティ】** で、ソースファイルの名前に **【ファイル名のオーバーライド】** プロパティを設定します。
- d. 必要に応じ、追加のソースプロパティを定義します。
- e. **【マッピング】** タブで、**【ターゲット】** ビューをクリックします。
- f. **【プロパティ】** で、セッションで割り当てるファイルの名前に **【ファイル名のオーバーライド】** プロパティを設定します。

- g. 必要に応じ、追加のターゲットプロパティを定義します。
- h. **[OK]** をクリックします。

第 14 章

フォールトトレランスを持つデータ書き込み

この章では、以下の項目について説明します。

- [PowerExchange ターゲットへのデータ書き込みのモード, 177 ページ](#)
- [フォールトトレランスの概要, 178 ページ](#)
- [フォールトトレランスを持つエラー処理, 179 ページ](#)
- [フォールトトレランスを持つ拒否ファイル, 181 ページ](#)

PowerExchange ターゲットへのデータ書き込みのモード

PowerExchange では、以下のモードでデータベースにデータを書き込むことができます。

- **同期。** PowerExchange は、データブロックを PowerExchange Listener に送信し、応答を待ってから別のブロックを送信します。このモードでは、データの整合性および回復性は確保されますが、パフォーマンスは低下します。
- **非同期。** PowerExchange は、データブロックを PowerExchange Listener に送信し、応答を待たずに別のブロックを送信します。このモードでは、パフォーマンスは最適化されますが、回復性は確保されません。
- **フォールトトレランスを持つ非同期。** PowerExchange は、データブロックを PowerExchange Listener に非同期で送信します。PowerExchange Listener は、エラー処理、更新方式、エラーしきい値、拒否ファイルなど各種項目の設定方法に基づいて、ログファイルにエラーを書き込みます。このモードは、パフォーマンスを最適化し、サイトのフォールトトレランス要件に基づいてエラーをログに記録する場合に使用します。

書き込みモードは、PowerExchange クライアントで PowerCenter (PWXPC) データベース接続属性に指定するか、または Windows ODBC Data Source Administrator から指定します。

以下の表に、各書き込みモードに対する、PWXPC の書き込みモード接続属性、または ODBC の書き込みモード設定の設定方法を示します。

書き込みモード	PWXPC の書き込みモード設定	ODBC の書き込みモード設定
同期	書き込み確認オン	CONFIRMWRITE = Y Windows ODBC Data Source Administrator の [PowerExchange データソース] タブで、[書き込みモード] リストから [書き込み確認オン] を選択します。
非同期	書き込み確認オフ	CONFIRMWRITE = N Windows ODBC Data Source Administrator の [PowerExchange データソース] タブで、[書き込みモード] リストから [書き込み確認オフ] を選択します。
フォールトトレランスを持つ非同期	フォールトトレランスを持つ非同期 注: PWX DB2zOS、PWX DB2i5OS、PWX DB2LUW、PWX MSSQL Server、および PWX Oracle のリレーショナル接続でのみ使用可能です。	CONFIRMWRITE = T Windows ODBC Data Source Administrator の [PowerExchange データソース] タブで、[書き込みモード] リストから [フォールトトレランスを持つ非同期] を選択します。

フォールトトレランスの概要

フォールトトレランスを実現すると、データベースへの書き込み操作中に生成されるエラーを PowerExchange でどのように処理するかを制御できます。フォールトトレランスを持つ非同期書き込みを設定するには、以下の作業を実行する必要があります。

- 書き込みモードを [フォールトトレランスを持つ非同期] に設定して、フォールトトレランスを有効にします。
- PowerExchange でのエラーの処理方法をカスタマイズします。エラー処理動作と、書き込み処理を停止するエラーしきい値を設定できます。
- 拒否ファイルの場所および命名方式を指定します。拒否ファイルには、Writer でターゲットに書き込まれなかったデータ行が含まれています。

フォールトトレランスを持つ非同期書き込みモードは、以下のデータベースで使用できます。

- DB2 for Linux、UNIX、および Windows
- DB2 for i5/OS
- DB2 for z/OS
- Microsoft SQL Server
- Oracle (Linux、UNIX、または Windows)

関連項目：

- [「フォールトトレランスを持つエラー処理」 \(ページ 179\)](#)
- [「フォールトトレランスを持つ拒否ファイル」 \(ページ 181\)](#)

フォールトトレランスを持つエラー処理

エラー処理動作と、処理停止のタイミングを制御するエラーしきい値を設定することによって、PowerExchange で書き込みエラーを処理する方法をカスタマイズできます。

エラー処理の構成

存在しない行に対する UPDATE 操作および DELETE 操作のエラー処理を定義するには、ターゲットシステムの DBMOVER コンフィギュレーションファイルに ERRROWNOTFOUND 文を含めます。

ERRROWNOTFOUND 文

ERRROWNOTFOUND 文は、PowerExchange が、存在しない行に対する UPDATE 操作または DELETE 操作に対して、エラーを生成するか生成しないかを指定します。

ターゲットシステム上の DBMOVER コンフィギュレーションファイルに ERRROWNOTFOUND 文を含めます。

データソース: すべて

必須: No

構文:

ERRROWNOTFOUND={N|Y}

有効な値:

- N。PowerExchange はエラーを生成しません。
- Y。PowerExchange は、エラーを生成し、エラーカウントを 1 つ増やし、エラーが発生したレコードを拒否ファイルに書き込みます。

デフォルトは N です。

使用上の注意: この文は、フォールトトレランスを持つ非同期書き込みモードでのみ有効です。このモードを使用するには、PWXP 接続属性で **【書き込みモード】** 値を **【フォールトトレランスを持つ非同期】** に設定します。

エラーしきい値の構成

エラーしきい値では、発生を許可する致命的でないエラーの最大数を指定します。このしきい値に達すると、PowerExchange がセッションでの書き込み処理を停止します。

以下の表では、PWXPC インタフェースおよび ODBC インタフェースからエラーしきい値を指定する方法について説明します。

インタフェース	構成方法
PowerExchange Client for PowerCenter (PWXPC)	セッションプロパティで、 【ストップするまでのエラー数】 属性の値を変更します。詳細については、『 <i>PowerCenter 用の PowerExchange インタフェース</i> 』を参照してください。
Windows ODBC Data Source Administrator の [PowerExchange データソース] タブ	[Windows ODBC Data Source Administrator] ダイアログボックスの [PowerExchange データソース] タブで、 【ストップするまでのエラー数】 フィールドに値を入力します。
UNIX ODBC.ini ファイル	ODBC.ini ファイルに以下のパラメータを追加します。 <code>STOPONERRORS=number</code>

カスタマイズされたエラー処理用のエラーアクションファイルの作成

特定のエラー処理方法が必要である場合は、フォールトトレランス動作をカスタマイズできます。ただし、PowerExchange にはサポートするデータベースごとにリカバリ可能なエラーのコードおよび致命的なエラーのコードをまとめたリストがデフォルトで用意されているため、カスタマイズするかどうかはオプションです。

特定のデータベースエラーコードに対する PowerExchange の動作をカスタマイズするには、以下の手順を実行します。

エラー処理のカスタマイズ用のエラーアクションファイルを作成するには：

1. PowerExchange エラー処理のカスタマイズ対象となるデータベースエラーコードをすべて定義するエラーアクションファイルを作成します。PowerExchange にテンプレートとして用意されているサンプルのエラーアクションファイルのいずれかを使用できます。エラーコードのフォールトトレランスオプションを指定するには、以下の構文を使用します。

error_code,option

説明:

- *error_code* は、DB2 SQLSTATE コードや Oracle *ORA-nnnnn* エラーコードなど、データベース固有のコードです。
- *option* は、リカバリ可能である場合は **R**、致命的である場合は **F** になります。

以下に例を示します。

```
00017,R
00018,F
```

2. ターゲットシステムの DBMOVER コンフィギュレーションファイルで、カスタマイズされたエラーアクションファイルのフルパスおよびファイル名を、指定します。たとえば、データベース用の文を以下のように使用します。

```
DB2_ERRORFILE=path_filename
ORA_ERRORFILE=path_filename
MSS_ERRORFILE=path_filename
```

エラーアクションファイルを定義すると、PowerExchange はデフォルトのエラー処理ではなくカスタマイズしたエラーコード指定を使用します。その結果、ユーザーがエラーアクションファイルにリカバリ可能なエラーとして指定していないエラーコードを処理するようになります。

PowerExchange エラーアクションファイルの例

以下の表に、PowerExchange に用意されているサンプルのエラーアクションファイルを一覧表示します。

データベースおよびプラットフォーム	エラーファイルの例
DB2 for z/OS	RUNLIB での DB2ERR メンバ
DB2 for i5/OS	<i>datalib</i> /CFG での DB2ERR メンバ
DB2 for Linux、UNIX、および Windows	PowerExchange インストールディレクトリ内の db2err.act
Microsoft SQL Server	PowerExchange インストールディレクトリの mssqlerr.act
Oracle	PowerExchange インストールディレクトリにある ora8err.act

フォールトトレランスを持つ拒否ファイル

PowerExchange は、ターゲットプラットフォームに拒否ファイルを作成します。拒否ファイルには、ターゲットに書き込まれなかったデータ行が記載されています。

データベース制約に違反する行の他、切り捨てやオーバーフローが発生しているデータを拒否するようにターゲットデータベースが設定されている場合には、フィールドにそのようなデータが含まれている行も拒否されます。

拒否ファイルを読み込むには、その拒否ファイルのコードページをサポートするテキストエディタを使用します。

拒否ファイルでは、行インジケータおよびカラムインジケータによって注目すべき情報が示されます。例えば、重複するキーを挿入しようとする、以下の拒否ファイルエントリが生成されます。

```
23,D,FAILNAME,D,1,D
```

行が拒否された理由を特定するには、カラムインジケータを確認し、セッションログを調べます。

注: PowerCenter 拒否ファイルは、PowerExchange 非同期書き込みを使用するセッションでも作成されます。ただし、その PowerCenter 拒否ファイルは空です。拒否されたデータは、PowerExchange の拒否ファイルに書き込まれます。

拒否ファイルの構造

PowerExchange 拒否ファイルは、PowerCenter 拒否ファイルとは構造が異なります。PowerExchange は、挿入 (I)、更新 (U)、削除 (D)、変化型更新 (M) の操作ごとに別の拒否ファイルを作成します。

PowerCenter は、全操作に対して 1 つの拒否ファイルを作成します。

拒否ファイルのカラムインジケータ

PowerExchange 拒否ファイルでは、どのデータカラムの後にも、そのデータを定義するカラムインジケータが表示されます。

次の表では PowerExchange が使用するカラムインジケータについて説明します。

カラムインジケータ	データの種類	Writer のアクション
D	有効なデータ	writer はデータをターゲットデータベースに渡します。ターゲットは、重複キーなどのデータベースエラーが発生しない限りデータを受け入れます。
N	NULL 値	writer は、NULL 値をターゲットに渡します。ターゲットデータベースが NULL 値を受け入れない場合、NULL 値は拒否されます。

拒否ファイルでは、NULL 値は以下のように「N」で示されます。

```
1,D,,N,35,D
```

拒否ファイルでは、バイナリ値は以下のようにテキスト文字列「BINARY DATA」で示されます。

```
0,D,1,D,PAUL,D,BINARY DATA,D
```

拒否ファイルの区切り文字

拒否ファイルのデフォルトの区切り文字はコンマ (,) です。このデフォルトの区切り文字をオーバーライドする場合は、DBMOVE コンフィギュレーションファイルの REJECT_FILE_DELIMITER 文で別の区切り文字を指定できます。データベースに書き込まれるデータにコンマが含まれている場合は、区切り文字を上書きすることをお勧めします。

区切り文字としてセミコロン (;) を使用するには、以下のようにセミコロンを二重引用符で囲んで入力します。

```
REJECT_FILE_DELIMITER=";"
```

データの一部に含まれる区切り文字をエスケープするには、その区切り文字の前にバックスラッシュ (\) を追加します。たとえば、デフォルトのコンマ (,) 区切り文字を使用している場合、データに含まれるコンマは以下ようになります。

```
\,
```

また、セミコロン (;) をカラム区切り文字として使用する場合、データに表示されるセミコロンは以下のようになります。

```
\;
```

REJECT_FILE_DELIMITER 文の詳細については、『PowerExchange リファレンスマニュアル』を参照してください。

拒否ファイルの例

以下に、拒否ファイルの例を示します。

```
/* Insert reject file
/* Table: USER1.EMPLOYEE
/* Columns: EMPNO,ENAME,JOB,MGR,HIREDATE,SAL,COMM,DEPTNO
/* Column indicator key: D=Valid, N=NULL
93,D,SURNAME,D,ANALYST,D,7902,D,20060705163431000000,D,9001.00,D,0.00,D,10,D
8,D,SURNAME,D,ANALYST,D,7902,D,20060705163431000000,D,9001.00,D,0.00,D,10,D
58,D,SURNAME,D,MANAGER,D,7902,D,20060705163431000000,D,9001.00,D,0.00,D,10,D
108,D,SURNAME,D,ANALYST,D,7902,D,20060705163431000000,D,1002.00,D,0.00,D,10,D
```

90,D,SURNAME,D,SALESMAN,D,7902,D,20060705163430000000,D,7001.00,D,0.00,D,10,D
 92,D,SURNAME,D,ANALYST,D,7902,D,20060705163431000000,D,9001.00,D,0.00,D,10,D
 82,D,SURNAME,D,MANAGER,D,7902,D,20060705163431000000,D,10001.00,D,0.00,D,10,D
 132,D,SURNAME,D,SALESMAN,D,7902,D,20060705163432000000,D,2001.00,D,0.00,D,10,D
 87,D,SURNAME,D,PRESIDENT,D,7902,D,20060705163431000000,D,4001.00,D,0.00,D,10,D
 82,D,SURNAME,D,ANALYST,D,7902,D,20060705163432000000,D,10002.00,D,0.00,D,10,D
 52,D,SURNAME,D,CLERK,D,7902,D,20060705163431000000,D,5001.00,D,0.00,D,10,D

拒否ファイルの命名規則

拒否ファイルの場所および名前の指定方法は、使用しているプラットフォームと、PWXPC 接続と ODBC 接続のどちらを使用しているかによって異なります。

以下の表に、構成方法を示します。

インタフェース	構成方法
PWXPC リレーショナル接続	[接続オブジェクト定義] ダイアログボックスで 【拒否ファイル】 属性に必要なパスとファイル名に変更します。
Windows ODBC Data Source Administrator の [PowerExchange データソース] タブ	[PowerExchange データソース] タブで、 【リジェクトファイル】 フィールドにパスおよびファイル名を入力します。
UNIX ODBC.ini ファイル	REJECTFILE= <i>path_filename</i>

PowerExchange は、プラットフォーム固有の規則に従って拒否ファイルを作成します。

Linux、UNIX、および Windows における拒否ファイル命名規則

Linux、UNIX、および Windows では、以下の拒否ファイル命名規則が使用されます。

- 拒否ファイルのパスおよびファイル名を指定しないと、DETAIL.LOG ファイルが保存されている PowerExchange インストールディレクトリに拒否ファイルが作成されます。拒否ファイル名のデフォルトのフォーマットは以下のとおりです。

pwxr.schema_name.table_name.strategy.timestamp.bad

- 別のディレクトリで拒否ファイルを作成するには、スラッシュ (/) またはバックスラッシュ (\\uff09 で終わる拒否ファイルパスを入力します。プラットフォームではどちらも有効です。たとえば、Windows では「C:\PWX\Log\201d」と指定できます。PowerExchange は、デフォルトフォーマットに基づいてファイル名を生成します。

- デフォルトのファイル名にプレフィックスを追加し、特定のディレクトリにそのファイルを作成するには、スラッシュ (/) またはバックスラッシュ (\\uff09 で終わる拒否ファイルパスを入力し、その後にプレフィックスを続けます。例えば、「C:\PWX\Log\prefix」のようになります。PowerExchange は、指定のディレクトリにファイルを作成し、以下のフォーマットでファイル名を生成します。

prefix.schema_name.table_name.strategy.timestamp.bad

注: Linux および UNIX では、拒否ファイルパスにスラッシュ (/) を使用します。Windows では、バックスラッシュ (\\uff09 を使用します。

z/OS 上の拒否ファイル命名規則

MVS では、以下の拒否ファイル命名規則が使用されます。

- 拒否ファイル値を指定しない場合は、以下のデータセットに拒否データが書き込まれます。

userid.jname.JOBnnnnn.PWXR.strategy.Tnnnnnn

説明:

- *userid* は、ジョブを実行するユーザー ID です。
- *jname* はジョブ名です。
- *JOBnnnnn* はジョブ番号です。
- *strategy* は、I（挿入）、U（更新）、D（削除）、M（変化型更新）のいずれかです。
- *Tnnnnnnn* は、最大 8 文字の長さの一意の名前です。

PowerExchange では、PowerExchange Listener ジョブまたは Netport ジョブをバッチジョブとして実行する場合には *JOBnnnnn* を使用し、PowerExchange Listener ジョブまたは Netport ジョブを開始されたタスクとして実行する場合には *STCnnnnn* を使用します。

- 任意で、データセットプレフィックスを入力できます。PowerExchange は、以下のフォーマットで一意のデータセット名を生成します。

prefix.Jnnnn.PWXR.strategy.Tnnnnnnn

prefix 修飾子は、最大 19 文字の長さになることがあります。

i5/OS における拒否ファイル命名規則

i5/OS では、PowerExchange で以下の拒否ファイル命名規則が使用されます。

- 拒否ファイル値を指定しなかった場合は、1 つのセッションで作成される拒否ファイルメンバが 1 つの FILENAME にまとめられます。PowerExchange では次のデフォルトが使用されます。

LIBRARYNAME=DATALIB

この値は、CRTPWXENV インストール手順からのものです。

- FILENAME=PWXR*nnnnnn*

このとき、*nnnnnn* は 6 桁の i5/OS ジョブ番号です。

以下に例を示します。

FILENAME=PWXR713895

- MEMBERNAME=*strategy_i5OSjobnumber_uniquepwref*

次の表では、[MEMBERNAME] の変数を示します。

変数	説明
<i>strategy</i>	I（挿入）、U（更新）、D（削除）、または M（変化型更新）
<i>i5OSjobnumber</i>	6 桁のジョブ番号（ <i>nnnnnn</i> ）
<i>uniquepwref</i>	一意の 3 桁の数値（ <i>nnn</i> ）

以下に例を示します。

MEMBERNAME=I713895001

任意で、LIBRARYNAME と FILENAME をスラッシュ (/) で区切って入力できます。

- LIBRARYNAME のみを指定した場合、PowerExchange はデフォルトの FILENAME フォーマットに基づいて FILENAME を作成します。

PowerExchange は、セッションごとの拒否ファイルメンバをそれぞれ別の i5/OS ファイルに書き込みます。

- スラッシュ (/) で終わる FILENAME か、または 5 文字以上でスラッシュ (/) で終わらない FILENAME を指定した場合、PowerExchange は指定のとおりその値を使用します。

PowerExchange は、複数のセッションの拒否ファイルメンバをまとめて 1 つの i5/OS ファイルに書き込みます。

例えば、以下の拒否ファイル値を入力するものとします。

```
MYLIBNAME/MYFILENAME/
```

PowerExchange では、以下の形式の名前を持つ拒否ファイルメンバを作成します。

```
MYLIBNAME/MYFILENAME/member_name
```

最初のセッションでは、PowerExchange により i5/OS ファイルに以下の拒否ファイルメンバが書き込まれます。

```
LIBRARY/MYFILENAME/I713895001  
LIBRARY/MYFILENAME/U713895002  
LIBRARY/MYFILENAME/D713895003
```

後続のセッションでは、PowerExchange により同じファイルに以下の拒否ファイルメンバが追加されます。

```
LIBRARY/MYFILENAME/I713895001  
LIBRARY/MYFILENAME/U713895002  
LIBRARY/MYFILENAME/D713895003  
LIBRARY/MYFILENAME/I839406001  
LIBRARY/MYFILENAME/U839406002  
LIBRARY/MYFILENAME/D839406003
```

- 1~4 文字でスラッシュ (/) で終わらない FILENAME 値を入力した場合、PowerExchange は 6 桁のジョブ番号をファイル名に追加します。

たとえば、以下の FILENAME 値を指定するものとします。

```
MYLIBNAME/PREF
```

PowerExchange では、以下の形式の名前を持つ拒否ファイルメンバを作成します。

```
MYLIBNAME/PREFnnnnnn/member_name
```

PowerExchange は、セッションごとの拒否ファイルメンバをそれぞれ別の i5/OS ファイルに書き込みます。

たとえば、最初のセッションの場合、PowerExchange は以下のメンバを 1 つの i5/OS ファイルに書き込みます。

```
LIBRARY/PREF713895/I713895001  
LIBRARY/PREF713895/U713895002  
LIBRARY/PREF713895/D713895003
```

後続のセッションでは、PowerExchange により新しい i5/OS ファイルに以下のメンバが書き込まれます。

```
LIBRARY/PREF839406/I839406001  
LIBRARY/PREF839406/U839406002  
LIBRARY/PREF839406/D839406003
```

拒否ファイルの作成の無効化

REJECTFILE 文を PWXDISABLE に設定することによって、拒否ファイルの作成を無効にすることができます。

以下の表に、使用可能な各インターフェースからこの文を設定する方法を示します。

インターフェース	構成方法
PWXPC リレーショナル接続	[接続オブジェクト定義] ダイアログボックスで、 拒否ファイル 属性を、PWXDISABLE に変更します。
Windows ODBC Data Source Administrator の [PowerExchange データソース] タブ	[PowerExchange データソース] タブで、 [拒否ファイル] フィールドに PWXDISABLE と入力します。
UNIX ODBC.ini ファイル	REJECTFILE=PWXDISABLE

第 15 章

監視およびチューニングオプション

この章では、以下の項目について説明します。

- [バルクデータ移動セッションの監視, 187 ページ](#)
- [バルクデータ移動セッションのチューニング概要, 193 ページ](#)
- [PowerExchange DBMOVER 文によるバルクデータ移動セッションのチューニング, 194 ページ](#)
- [接続属性によるバルクデータ移動セッションのチューニング, 196 ページ](#)
- [接続プール, 197 ページ](#)
- [データマップのキャッシュ, 198 ページ](#)
- [バルクデータオフロードとマルチスレッド処理, 201 ページ](#)
- [バルクデータ移動セッション内のパイプラインパーティション, 205 ページ](#)
- [PowerExchange zIIP Exploitation, 216 ページ](#)
- [PowerExchange リスナの WLM サービスクラスへの割り当て, 219 ページ](#)

バルクデータ移動セッションの監視

PowerExchange では、バルクデータ移動セッションの進捗状況の監視に使用できるメッセージが発行されます。PWXPC では、PowerCenter Workflow Monitor にバルクデータ移動セッションについての進捗状況と統計情報を表示することもできます。

PowerExchange におけるバルクデータ移動セッションの監視

PowerExchange では、以下の方法で、バルクデータ移動セッションによるデータ抽出を監視できます。

- **読み取り進捗メッセージ。**バルクデータ移動セッションで読み取られたデータレコードの数を示すメッセージの書き込みを PowerExchange に対して要求できます。
- **バルクデータ移動統計メッセージ。**バルクデータ移動セッションが終了した時点で、処理されたデータレコードの量に関する統計情報を含むメッセージが PowerExchange によって書き込まれます。
- **マルチスレッド処理統計メッセージ。**マルチスレッド処理を使用したバルクデータ移動セッションに関する統計情報の書き込みを PowerExchange に対して要求できます。
- **LISTTASK コマンド出力。**LISTTASK コマンドを使用し、アクティブなバルクデータ移動セッションを表示できます。

- **DISPLAYSTATS コマンド出力。** DISPLAYSTATS コマンドを使用し、Linux、zLinux、UNIX、Windows、または z/OS 上で実行される PowerExchange リスナの監視統計を発行できます。

PowerExchange により、読み取り進捗メッセージ、バルクデータ移動統計メッセージ、およびマルチスレッド処理統計メッセージが PowerExchange ログファイルに書き込まれます。PWXPC を設定し、これらの PowerExchange 情報メッセージと他の PowerExchange 情報メッセージを PowerCenter セッションログに書き込むこともできます。

デフォルトでは、PWXPC はセッションログにエラーメッセージと警告メッセージを書き込みますが、情報メッセージは書き込みません。PowerExchange 情報メッセージをセッションログに書き込むように PWXPC を設定するには、以下の接続属性または接続オプションのうちの 1 つを定義します。

- **[PWX ログエントリの取得]** 接続属性を選択する。(PWXPC アプリケーション接続タイプのみ)
- **[PWX 上書き]** 接続属性に RETLOGINFOMSG=Y オプションを追加する。(すべての PWXPC 接続タイプ)

[PWX ログエントリの取得] 接続属性と、**[PWX 上書き]** 接続属性の RETLOGINFOMSG=Y オプションは同じです。

読み取り進捗メッセージ

読み取り進捗を報告する PWX-04587 メッセージを発行するように PowerExchange を設定できます。

読み取り進捗メッセージを書き込むように PowerExchange に指示するには、DBMOVER コンフィギュレーションファイルに以下のパラメータを含めます。

- **PRGIND。** バルクデータ移動セッションのために読み取られたレコードの数を示す PWX-04587 メッセージを PowerExchange に書き込ませるには、Y を指定します。デフォルトは N です。
- **PRGINT。** PowerExchange が PWX-04587 メッセージを PowerExchange ログファイルに書き込む前に読み取るレコードの数を指定します。デフォルトは 250 レコードです。

PWX-04587 メッセージの形式は以下のとおりです。

```
PWX-04587 int_server/workflow_name/session_name: Records read=num_records
```

説明:

- *int_server* は PowerCenter 統合サービスの名前です。
- *workflow_name* は、バルクデータ移動セッションを含むワークフローの名前です。
- *session_name* はバルクデータ移動セッションの名前です。
- *num_records* は、バルクデータ移動セッションの開始後に読み取られたレコードの累積数です。

たとえば、100 件のレコードを読み取った後に読み取り進捗メッセージを書き込むように PowerExchange に指示するには、DBMOVER コンフィギュレーションファイルに以下の文を含めます。

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

セッション名が s_DB2_SQL_stats であるバルクデータ移動セッションが実行されると、PowerExchange は以下のメッセージを PowerExchange ログファイルに書き込みます。

```
PWX-04587 intserv/wf_blk_mon_stats/s_DB2_SQL_stats: Records read=100
PWX-04587 intserv/wf_blk_mon_stats/s_DB2_SQL_stats: Records read=200
PWX-04587 intserv/wf_blk_mon_stats/s_DB2_SQL_stats: Records read=300
```

PowerExchange は、このバルクデータ移動セッションが完了するまで、このセッションの PWX-04587 メッセージを書き込み続けます。PowerExchange ログファイルでは、これらのメッセージのそれぞれに日時スタンプが付いています。この情報を使用して、PowerExchange がソースシステム上のバルクデータを処理する速度を判断できます。

注意事項:

- 非リレーショナルバルクデータ移動ソースの場合、PowerExchange リスナマシンで DBMOVE 構成ファイルに PRGIND 文と PRGINT 文を含めます。リレーショナルバルクデータ移動ソースの場合、PowerCenter 統合サービスマシンで DBMOVE 構成ファイルに PRGIND 文と PRGINT 文を含めます。
- DB2 リレーショナル接続の場合、PowerExchange は次の数のレコードを読み取った後、PWX-04587 メッセージを書き込みます。
PRGINT の値以上である、**【配列サイズ】** 接続属性の最小の倍数
- PRGIND 文と PRGINT 文は、統合サービスマシンで実行され、PowerExchange リスナに接続するすべてのワークフローに適用されるため、これらの文を適切に使用することをお勧めします。

バルクデータ移動統計メッセージ

バルクデータ移動セッションが終了した時点で、このセッションに関する統計情報を含む以下のメッセージが PowerExchange によって書き込まれます。

- **PWX-00408.** PowerExchange により、バルクデータ移動セッション内の各ソースのソースシステム上にこのメッセージが書き込まれます。このメッセージには、データソースから読み取られたレコードとバイトの数が含まれています。
- **PWX-00409.** PowerExchange により、バルクデータ移動セッション内の各ターゲットのターゲットシステム上にこのメッセージが書き込まれます。このメッセージには、そのバルクデータ移動セッションのために読み取られたレコードとバイトの数が含まれています。

マルチスレッド処理の統計

バルクデータオフロード処理を使用する場合は、マルチスレッド処理を使用し、オフロード処理が実行される PowerCenter 統合サービスマシンのスループットの向上を図ることもできます。

これらのメッセージを発行するには、PowerCenter 統合サービスマシンの DBMOVE コンフィギュレーションファイルに SHOW_THREAD_PERF 文を指定する必要があります。

SHOW_THREAD_PERF=*number_of_records*

この文により、マルチスレッド抽出処理に関する統計メッセージを PowerExchange メッセージログファイルに書き込む前に PowerExchange が処理する必要のあるレコードの数を指定します。この文の詳細については、『*PowerExchange リファレンスマニュアル*』を参照してください。

また、統計を生成できるようにマルチスレッド処理を実装するには、アプリケーション接続の**【ワーカースレッド】**属性に 1 以上を指定する必要があります。

PowerExchange により、各統計インターバルで以下のメッセージの一部またはすべてが書き込まれます。

- **PWX-31255.** サイクル時間です。これは、Integration Service マシン上の PowerExchange が PWXPC に渡す前にバルクデータの処理に費やした時間の合計です。このメッセージには、時間と平均の割合の合計と、最小時間、最大時間がマイクロ秒単位で記載されます。
- **PWX-31256.** I/O 時間です。これは、Integration Service マシン上の PowerExchange がソースシステム上の PowerExchange リスナからバルクデータを読み取るのに費やした時間です。ローカルフラットファイルを読み取る場合、I/O 時間は PowerExchange によってそのファイルの読み取りに費やされた時間です。このメッセージには、合計時間における I/O の割合と、平均時間、最小時間、最大時間がマイクロ秒単位で記載されます。
- **PWX-31257.** 解析時間です。これは、Integration Service マシン上の PowerExchange によってすべてのスレッドのバルクデータレコードをカラムレベルで処理するために費やされた時間です。このメッセージには、合計時間と平均の解析の割合と、最小時間、最大時間がマイクロ秒単位で記載されます。

- **PWX-31258.** 外部時間です。これは、Integration Service マシン上の PowerExchange によって、すべてのスレッドから取得したバルクデータレコードを 1 つのストリームにまとめて PWXPC に渡し、PWXPC がデータを PowerCenter にフラッシュするために費やされた時間です。このメッセージには、合計時間における外部の割合と、平均時間、最小時間、最大時間がマイクロ秒単位で記載されます。
- **PWX-31259.** 遅延時間です。これは、Integration Service マシン上の PowerExchange がソースシステム上の PowerExchange リスナから処理する新しいバルクデータレコードを受け取るために待機した時間です。このメッセージには、合計時間と平均の遅延の割合と、最小時間、最大時間がマイクロ秒単位で記載されます。
- **PWX-31261.** PowerExchange の Joiner スレッド I/O 時間です。このメッセージは、パーティション化を使用する場合に含まれます。このメッセージには、マルチスレッドセッション中に I/O を実行するために Joiner によって費やされた時間が示されます。I/O を実行していない場合、Joiner はパーティションが処理を完了するのを待機します。
- **PWX-31262.** スレッド I/O 遅延です。このメッセージは、パーティション化を使用する場合に含まれます。このメッセージには、Reader による解析データの提供を待機するために各パーティションによって費やされた時間の平均が示されます。

たとえば、DBMOVER コンフィギュレーションファイルに SHOW_THREAD_PERF=10000 と指定すると、10,000 件のバルクデータレコードが読み取られた後、PowerExchange によって以下のサンプルメッセージが書き込まれます。

```
PWX-31254 PowerExchange threading stats for last 10000 rows. Cycle (array) size is 25 rows. 0 out of array
occured.
PWX-31255 Cycle time: 100% (avg:      5709 min:      4741 max:      7996 usecs)
PWX-31256 IO time: 4% (avg:        235 min:         51 max:      1021 usecs)
PWX-31257 Parse time: 79% (avg:     4551 min:     4102 max:     5495 usecs)
PWX-31258 Extern time: 20% (avg:     1145 min:      618 max:     3287 usecs)
PWX-31259 Delay time: 0% (avg:         7 min:         4 max:       165 usecs)
PWX-31254 PowerExchange threading stats for last 100000 rows. Cycle (array) size is 25 rows. 0 out of array
occured.
PWX-31255 Cycle time: 99% (avg:      5706 min:      4735 max:      7790 usecs)
PWX-31256 IO time: 4% (avg:        234 min:         51 max:       950 usecs)
PWX-31257 Parse time: 79% (avg:     4549 min:     4108 max:     5425 usecs)
PWX-31258 Extern time: 20% (avg:     1144 min:      616 max:     3242 usecs)
PWX-31259 Delay time: 0% (avg:         7 min:         4 max:       115 usecs)
```

解析および外部処理の時間が I/O 時間より長い場合、バルクデータ移動セッションのスレッドの数を増やし、スループットの向上を図れる場合があります。

PowerExchange リスナの LISTTASK コマンド

PowerExchange リスナの LISTTASK コマンドを使用し、PowerExchange リスナでアクティブになっているバルクデータ移動セッションを表示できます。

あるいは、Linux、UNIX、または Windows システムからローカルシステムまたはリモートシステム上で実行されている PowerExchange リスナに対して、`pwxcmd listtask` コマンドを発行します。

コマンド出力には、**PwrCntrSess** フィールドが含まれます。このフィールドには、以下の形式で PowerCenter セッション名が示されます。

integration_server_name/workflow_name/session_name

たとえば、2 つのバルクデータ移動セッションがアクティブな場合、コマンドには以下の出力が表示されます。

```
PWX-00711 Active tasks:
PWX-00712 TaskId=1, Partner=10.10.10.01, Port=2480 ,
PwrCntrSess=intserv1/workflow1/bulk_sess1,
Application=appl_name1, Status=Active, AM=DB2, Mode=Append, Process=, SessId=
PWX-00729 Userid=userid, Client=clientid, File=capture_registration_file_name, Table=table_name,
DB=database_name
PWX-00712 TaskId=2, Partner=10.10.10.02, Port=2480, PwrCntrSess=intserv2/workflow2/bulk_sess2,
Application=appl_name2, Status=Active, AM=DB2, Mode=Append, Process=, SessId=
```

```
PWX-00729 Userid=userid, Client=clientid, File=capture_registration_file_name, Table=table_name,  
DB=database_name  
PWX-00713 2 active tasks  
PWX-00709 0 Dormant TCBS
```

PowerExchange リスナの DISPLAYSTATS コマンド

PowerExchange リスナの DISPLAYSTATS または `pwxcmd displalstats` コマンドを使用して、i5/OS、Linux、zLinux、UNIX、Windows、または z/OS で実行される PowerExchange リスナの監視統計を発行できます。

コマンドを実行する前に、DBMOVER 構成ファイルで以下の文を構成します。

- DBMOVER 構成ファイル内の STATS 文の MONITOR パラメータを指定することにより、PowerExchange がこれらの統計情報を収集できるようにします。 *interval* サブパラメータを含めると、統計情報をオンデマンドだけでなく定期的に発行できます。
- z/OS でモニタリング出力を正しく表示するためには、LOG_LINE_LIMIT 文を 132 に設定します。この設定を行わないと、行が不自然に重なり、読みづらい状態になります。

このコマンドは、次に示す方法のいずれかで実行できます。

- PowerExchange リスナが実行されている Linux、UNIX、Windows、または zLinux システムのコマンドラインから。
- PowerExchange リスナが実行されている z/OS で MVS MODIFY (F)コマンドを使用する。
- リモートの Linux、UNIX、および Windows システムからサポートされている任意のオペレーティングシステムのリスナに `pwxcmd` プログラムを使用する。

注: i5/OS 上の PowerExchange リスナの監視統計は、この方法を使用してオンデマンドで発行する必要があります。

コマンド構文は、オペレーティングシステムの種類と、`pwxcmd` を使用するかどうかによって異なります。詳細については、『*PowerExchange コマンドリファレンス*』を参照してください。

どのコマンドパラメータを使用するかに基づき、以下のレポートタイプの 1 つを発行できます。

- **Listener.** メモリの使用状況、CPU の処理時間、クライアント要求のために行われる活動などについての PowerExchange リスナサマリ統計がレポートされます。これらの統計には、クライアントタスクの数、接続数、送受信されたメッセージの数、送受信されたデータのバイト数、レポートジョブ数 (z/OS のみ) などが含まれます。これらの統計には、バルクデータ移動と CDC タスクが含まれます。

注: Informatica ドメインで PowerExchange リスナサービスを実行する場合は、`infacmd pwxcmd displayStatsListener` コマンドを使用してこれらの統計を発行できます。詳細については、『*Informatica コマンドリファレンス*』を参照してください。

- **Accessmethods.** PowerExchange リスナメッセージとデータ転送活動の統計が、クライアントタスクとアクセス方式別にレポートされます。これらの統計には、アクティブタスクとアクセス方式の組み合わせごとに、読み取られた行および書き込まれた行の数、読み取られたデータおよび書き込まれたデータのバイト数、ソースまたはターゲットのファイル名またはデータマップファイル名、および CPU 処理時間が含まれます。アクセス方式として CAPX または CAPXRT を使用する CDC 要求の場合、レポートには SQL 挿入の数、更新回数、タスクが処理した削除の数も含まれます。
- **Clients.** PowerExchange リスナ下で実行されているアクティブクライアントタスクについての情報が報告されます。これらの統計には、ステータス、アクセス方式、読み取りまたは書き込みモード、プロセス名、セッション ID (存在する場合)、CPU の処理時間、開始日時などの情報の一部またはすべてがタスクごとに示されます。これらの統計には、クライアントのポート番号と IP アドレスも含まれます。クライアントが PowerCenter の場合、統計には PowerCenter セッション ID および CDC のアプリケーション名が含まれます。

デフォルトでは、Listener レポートが発行されます。

z/OS 上の PowerExchange リスナのレポートは、i5/OS、Linux、zLinux、UNIX、または Windows 上の PowerExchange リスナのレポートと同様です。

次の Listener レポート例は、z/OS 上の PowerExchange リスナの場合を示しています。

```
PWX-00723 Command <displaystats Listener> succeeded
PWX-37101 Listener <PWXLST > ASID=375 (x'177') UserID=AUSRID
PWX-37102 Memory
PWX-37103   Region below 16-MB line: In Use      108 KB Limit Value      9192 KB Free      9084 KB
PWX-37104   Region above 16-MB line: In Use    53912 KB Limit Value   1675264 KB Free   1621352 KB
PWX-37117 CPU Time
PWX-37118   TCB Time = 0 SRB Time = 0 zIIP-NTime = 0
PWX-37119   Listener = 0 hrs, 0 mins, 1 secs, 275762 mcrs
PWX-37106 Cumulative Requests
PWX-37107   Total Tasks= 11 Active Tasks = 3 HWM Tasks = 3 Maxtasks = 50
PWX-37108   Connections= 11 Accepted = 11 Active = 0
PWX-37109   Msgs Sent = 0 Msgs Received= 22
PWX-37110   Data Sent = 0 Data Received= 7304
PWX-37111   NetportJobs= 0
```

Memory、**TCB Time**、**SRB Time**、および **NetportJobs** 値は、z/OS 上の PowerExchange リスナ固有です。i5/OS、Linux、UNIX、または Windows 上の PowerExchange リスナの場合、レポートには総メモリ使用量が表示されます。

このレポートを使用し、クライアントタスクの数が、DBMOVER 構成ファイルの MAXTASKS 文に設定されている上限に達しているかどうかを確認できます。**HWM Tasks** 値を **Maxtasks** 値と比較します。HWM Task 値が MAXTASKS 限界に達すると、PowerExchange リスナの処理が遅れ、スループットの低下と接続タイムアウトが発生することがあります。

次の accessmethods レポートの例は z/OS 上の PowerExchange リスナの場合を示していますが、i5/OS、Linux、UNIX、Windows、または zLinux 上の PowerExchange リスナでも同じフィールドが表示されます。

```
PWX-00723 Command <displaystats AccessMethods> succeeded
PWX-37201 Active Access Methods
PWX-37202   Task ID = 42412 AM = CAPXRT
PWX-37203   Rows read = 1029 Rows written = 0
PWX-37204   Bytes read = 116277 Bytes written = 0
PWX-37205   File = d2ivd0.d002root_ROOT
PWX-37206   Table = <Capture Extract Realtime>
PWX-37208   Inserts = 564 Updates = 0 Deletes = 465
PWX-37121   CPU time = 0 hrs, 0 mins, 0 secs, 299809 mcrs
PWX-37202   Task ID = 42414 AM = NRDB
PWX-37203   Rows read = 10 Rows written = 0
PWX-37204   Bytes read = 570 Bytes written = 0
PWX-37205   File = ABC.VSAM.MASTER_REC
PWX-37206   Table = <Non-relational source>
PWX-37202   Task ID = 42414 AM = KSDS
PWX-37203   Rows read = 10 Rows written = 0
PWX-37204   Bytes read = 800 Bytes written = 0
PWX-37205   File = XYQ.TEST.V1.KSDS
PWX-37206   Table = XYQ.TEST.V1.KSDS
PWX-37121   CPU time = 0 hrs, 0 mins, 0 secs, 76151 mcrs
```

アクセス方式 CAPXRT と CAPX の場合、レポートには SQL 挿入の数、更新回数、CDC 要求のためにタスクで処理された削除の数が含まれます。

1 つのクライアントタスクで複数のアクセス方式が使用されることがあります。例えば、ソースデータの読み取りに 1 つの方式を使用し、リレーショナル形式ではないソースデータをリレーショナル形式にマップするために 1 つの方式を使用するということが考えられます。この出力例では、タスク 42414 はアクセス方式として NRDB を使用しており、**File** フィールドに指定されているデータマップファイルはリレーショナル形式でないデータをリレーショナル形式にマップします。この同じタスクが、アクセス方式 KSDS を使用して、**File** フィールドに指定されている KSDS データからデータを取得します。

次の clients レポートの例は Windows 上の PowerExchange リスナの場合を示していますが、i5/OS、Linux、zLinux、UNIX、または z/OS 上の PowerExchange リスナでも同じフィールドが表示されます。

```
PWX-00723 Command <displaystats Clients> succeeded
PWX-37112 Active Tasks
PWX-37113   Task ID = 41942 Status = Active
PWX-37114   Port = 2480 Partner = 127.0.0.1
PWX-37115   PwrCtrSess = N/A
PWX-37207   Application = N/A
PWX-37116   AM = NRDB Mode = Read Process = DTLLST3 SessionId =
PWX-37121   CPU time = 0 hrs, 0 mins, 0 secs, 62400 mcrs
PWX-37122   Start time = 2014-05-01 14:21:37
PWX-37113   Task ID = 41943 Status = Active
PWX-37114   Port = 2480 Partner = 127.0.0.1
```

PWX-37115	PwrCntrSess = N/A
PWX-37207	Application = N/A
PWX-37116	AM = NRDB Mode = Read Process = DTLLST3 SessionId =
PWX-37121	CPU time = 0 hrs, 0 mins, 0 secs, 124800 mcrcs
PWX-37122	Start time = 2014-05-01 14:22:01

Partner フィールドには、PowerExchange リスナによるタスク作成の原因となった要求を発行したクライアントの IP アドレスが示されます。この値は、IPv6 アドレスの場合、「::ffff」から始まります。

これらの各レポートのフィールドについての詳細は、『PowerExchange コマンドリファレンス』を参照してください。

バルクデータ移動セッションのチューニング概要

バルクデータ移動セッションのチューニングには、PowerExchange DBMOVER 構成文と PWXPC 接続属性を使用できます。また、接続プール、データマップのキャッシュ、オフロード処理、およびマルチスレッド処理も使用できます。また、z/OS では、PowerExchange リスナ開始済みタスクを適切な Workload Manager サービスクラスに割り当てることができます。

以下に示すチューニング方法のいずれかを使用します。

- 構成文と接続属性。特定の DBMOVER 文と PWX Batch 接続属性を定義します。
- 接続プール。ネットワーク接続情報がキャッシュされて再利用されるため、後続の接続の接続時間が短縮されます。例えば、z/OS で PowerExchange リスナへの接続を開くには、2 秒かかる場合があります。接続プールを使用すると、PowerExchange リスナへの後続の接続が瞬時に確立されるようになります。
- データマップのキャッシュ。PowerExchange は、z/OS 非リレーショナルデータソースに対して定義されたデータマップを、データマップファイルにアクセスする代わりにジョブレベルメモリから取得します。データマップファイルにアクセスしないため、データマップのキャッシュによりパフォーマンスが向上します。
- オフロード処理。DB2 for z/OS テーブルとイメージコピー、IMS アップロードデータセット、VSAM データセット、およびシーケンシャルファイルの場合、オフロード処理を使用して、PowerExchange バルクデータのカラムレベル処理とフィルタリングをバルクデータ移動セッションを実行する PowerCenter 統合サービスマシンに分散させます。処理を他のマシンに分散させることにより、ソースシステム上における PowerExchange バルクデータ移動のオーバーヘッドを削減することができます。
- マルチスレッド処理。DB2 for z/OS テーブル、VSAM データセット、またはシーケンシャルファイルにバルクデータオフロード処理を使用する場合は、マルチスレッド処理を使用してスループットの向上を図ることもできます。マルチスレッド処理では、PowerCenter Integration Service マシン上の複数のスレッドを使用して、オフロードされた PowerExchange 処理が実行されます。
- パイプラインのパーティション化。バルクデータ移動セッション用のパイプライン内で複数のパーティションを使用することで、セッションパフォーマンスを向上させます。Reader パーティション化、Writer パーティション化、またはこの両方を使用できます。パーティション化を行うと、複数のパーティションにわたるデータをセッションで同時に処理できます。この同時処理は、PowerCenter Integration Service マシンにオフロードされた、大量のリソースを消費するカラムレベルの処理にとりわけ有効です。
- Workload Manager (WLM) サービスクラス。適切な z/OS WLM サービスクラスを PowerExchange リスナに割り当てて、PowerExchange 経由で z/OS にアクセスするプロセスがパフォーマンス目標を満たすようにします。

PowerExchange DBMOVE 文によるバルクデータ移動セッションのチューニング

PowerExchange バルクデータ移動セッションをチューニングするために、DBMOVE コンフィギュレーションファイルに以下の文を含めることができます。

APPBUFSIZE=*bytes*

データの読み込みまたは書き込みに PowerExchange で使用するバッファの最大サイズをバイト単位で定義します。このデータバッファは、ソースシステムまたはターゲットシステムに存在できます。

PowerExchange はバッファがいっぱいになるまでソースシステムのアプリケーションデータバッファにバルクデータを書き込みます。その後、PowerExchange はソースシステム上の送信 TCP/IP バッファにデータを送信します。TCP/IP は、ターゲットシステム上の受信 TCP/IP バッファにバルクデータを転送します。ターゲットシステムの PowerExchange は、TCP/IP バッファからアプリケーションデータバッファにバルクデータを読み取ります。その後、PWXP がバルクデータを読み取り、PowerCenter に渡します。PowerCenter はデータを処理し、ターゲットに適用します。

送信する単一データ行の最大サイズより大きい APPBUFSIZE 値を入力します。

有効な値は、34816～8388608 です。デフォルトは 256000 です。

ターゲットシステムがリモートの場合は、ソースシステムおよびターゲットシステムにある DBMOVE コンフィギュレーションファイルに同じ APPBUFSIZE 値を入力します。

APPBUFSIZE 値が最適ではない場合、PowerExchange はソースシステムの PowerExchange ログファイルに PWX-01295 メッセージを書き込みます。このメッセージには推奨値が含まれます。

動的アプリケーションバッファのサイズ決定が有効な状態では、PowerExchange リスナが実行される間に行われるすべての接続のアプリケーションデータバッファの初期サイズが APPBUFSIZE 文で定義されます。PowerExchange は、必要に応じて個々の接続のアプリケーションデータバッファのサイズを動的に変更します。動的アプリケーションバッファのサイズ決定はデフォルトで有効になっています。この決定は、DBMOVE 構成ファイルの APPBUFSIZEDYN 文で「Y」を指定することによって明示的に有効にすることができます。

APPBUFSIZEDYN={N|Y}

動的アプリケーションバッファのサイズ決定を有効にするかどうかを指定します。

DBMOVE APPBUFSIZE 文は、PowerExchange リスナの実行中に行われるすべての接続の、アプリケーションバッファの初期サイズを定義します。APPBUFSIZEDYN=Y の場合、必要に応じて個々の接続のアプリケーションバッファのサイズが PowerExchange によって変更されます。

APPBUFSIZEDYN 文は、固定長または可変長のレコードを含むデータソースへの PowerExchange 接続に適用されます。可変長レコードは、1 つ以上の可変長フィールドが含まれるレコードです。可変長フィールドのデータ型は、VARCHAR または VARBIN です。

可変長レコードを含むデータソースへの接続ごとに、大きすぎてバッファに収まりきれないレコードが発生した場合は、PowerExchange によってアプリケーションバッファのサイズが変更されます。

PowerExchange によって、アプリケーションバッファのサイズは、オーバーフローしたレコードのサイズの 10 倍の値に増やされます（最大 8MB まで）。新しいサイズは、リスナが実行されている期間またはアプリケーションバッファのサイズが再度変更されるまで有効のまま維持されます。リスナの実行が開始された後で、接続のためのアプリケーションバッファのサイズが PowerExchange によって減らされることはありません。

固定長レコードを含むデータソースへの接続ごとに、接続が開かれる時点で PowerExchange によってレコード長が確認されます。この際、必要に応じて、最大 8MB のバッファサイズとなるように、アプリケーションバッファのサイズが一度変更されます。

COMPRESS={Y|N}

リモートシステムに転送する TCP/IP に PowerExchange がデータを送信する前に、データを圧縮するために、PowerExchange が独自の圧縮アルゴリズムを使用するかどうかを定義します。

デフォルトは Y です。

PowerExchange は、PowerExchange リスナにアクセスするリモートシステム上の DBMOVER コンフィギュレーションファイルの COMPRESS 設定を使用します。PWX NRDB Batch アプリケーション接続および PWX リレーショナル接続では、DBMOVER コンフィギュレーションファイルの圧縮設定をオーバーライドできます。圧縮を有効にすると、ソースシステム上の PowerExchange リスナの CPU 使用率が高まることがあります。

不要な CPU の使用を防ぐには、PowerCenter 統合サービスマシンの PowerExchange DBMOVER コンフィギュレーションファイルの [COMPRESS] を [N] に設定します。

LISTENER=(*node_name*,TCPIP,*port*,*send_bufsize*,*receive_bufsize*,*send_size*,*receive_size*,...)

指定された PowerExchange リスナプロセスが作業要求をリスンする TCP/IP ポート。

send_bufsize および *receive_bufsize* 位置パラメータは、PowerExchange が使用する TCP/IP 送受信バッファサイズのデータ部分を定義します。これらの値を指定しないと、PowerExchange によってオペレーティングシステムのデフォルト設定が使用されます。

スループットを向上するには、ソースシステムの DBMOVER コンフィギュレーションファイルにある LISTENER 文で *send_bufsize* および *receive_bufsize* の値を大きくします。使用する最適な値を決定するために支援が必要な場合は、ネットワーク管理者に問い合わせてください。

NODE=(*node_name*,TCPIP,*host_name*,*port*,*send_bufsize*,*receive_bufsize*,*send_size*,*receive_size*,...)

PowerExchange が PowerExchange リスナプロセスに連絡するために使用する TCP/IP ホスト名およびポート。

send_bufsize および *receive_bufsize* 位置パラメータは、PowerExchange が使用する送受信バッファサイズのデータ部分を定義します。これらの値を指定しないと、PowerExchange によってオペレーティングシステムのデフォルト設定が使用されます。

スループットを向上するには、ターゲットシステムの DBMOVER コンフィギュレーションファイルにある NODE 文で *send_bufsize* および *receive_bufsize* の値を大きくします。使用する最適な値を決定するために支援が必要な場合は、ネットワーク管理者に問い合わせてください。

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

Informatica グローバルカスタマサポートが PowerExchange コードの問題を解決するために使用する PowerExchange 診断トレースを有効化します。

TRACE 文は PowerExchange のパフォーマンスに重大な影響を及ぼすことがあります。これらの文は、Informatica グローバルカスタマサポートから指示があったときにのみ使用します。

Informatica グローバルカスタマサポートが問題を診断した後、すべてのシステムの DBMOVER コンフィギュレーションファイルから TRACE 文を削除またはコメントアウトします。

接続属性によるバルクデータ移動セッションのチューニング

PowerCenter では、PWX NRDB Batch アプリケーション接続と PWX バルクリレーショナル接続に関するオプションをカスタマイズし、バルクデータ移動セッションをチューニングできます。

以下の表に、バルクデータ移動セッションのチューニングのために使用できる接続属性を示します。

接続属性	説明	チューニングの提案
圧縮	このオプションは、PowerCenter セッション中にソースデータを圧縮する場合に選択します。 デフォルトでは無効になっています。	圧縮は使用しません。
暗号化タイプ	PowerExchange が使用するデータ暗号化タイプです。 デフォルトは [なし] です。	暗号化は使用しません。
オフロード処理	このオプションは、バルクデータオフロード処理を要求する場合に選択します。 デフォルトは [いいえ] です。	詳細については、「 バルクデータオフロードとマルチスレッド処理 」(ページ 201)を参照してください。

接続属性	説明	チューニングの提案
ワーカースレッド	<p>ワーカースレッド対応のデータタイプのオフロード処理を有効にする場合は、このオプションを設定して、統合サービスマシン上でオフロードされた処理に複数のスレッドを使用します。PowerExchange で使用するスレッドの数を入力します。</p> <p>有効な値は 1～64 です。</p> <p>デフォルトは 0 です。マルチスレッド処理は無効になります。</p>	<p>詳細については、「バルクデータオフロードとマルチスレッド処理」(ページ 201)を参照してください。</p>
配列サイズ	<p>VSAM データセットまたはシーケンシャルファイルで、【ワーカースレッド】 の値が 0 より大きい場合は、【配列サイズ】 にスレッドのストレージ配列のサイズ（単位: レコード数）を指定します。</p> <p>DB2 for z/OS テーブルの場合、【配列サイズ】 には DB2 フェッチ配列サイズを指定します。</p> <p>DB2 for z/OS イメージコピーの場合、【配列サイズ】 で割り当てる必要があるメモリバッファの最大数を決定します。zIIP 処理を有効にした場合、【配列サイズ】 は、zIIP プロセッサにディスパッチする圧縮された行数も表しています。</p> <p>有効な値は 25～5000 です。デフォルトは 25 です。</p>	<p>より大きな配列サイズに割り当てられた追加のメモリが有益で、サーバーのパフォーマンスが低下していないかどうかをテストおよび判断できる場合を除き、デフォルト値（25）を使用することをお勧めします。これらの判断ができ、かつ次のような場合は、配列サイズを 500～1000 にすることをお勧めします。</p> <ul style="list-style-type: none"> - オフロード処理を有効にし、【ワーカースレッド】 に 0 以外の値を選択します。 - SQL Server バルクロードユーティリティを使用して、SQL Server テーブルにデータをロードします。 - PowerExchange バルクデータ移動のデータソースとして圧縮イメージコピーを使用し、z/OS で zIIP 処理を有効にします。 <p>次の場合は配列サイズを増やすことをお勧めしません。</p> <ul style="list-style-type: none"> - NRDB 処理を実行していて、オフロード処理を使用していない。 - 多数のカラムが含まれているため、非常に大きい LOB カラムまたは行を移動している。

関連項目：

- [「バルクデータオフロードとマルチスレッド処理」](#) ([ページ 201](#))

接続プール

接続プールは、キャッシュされた接続情報を再利用して、ネットワーク接続を開いたり新しいタスクを作成したりするためのオーバーヘッドをなくす方法です。

接続プールが有効である場合、PowerExchange は PowerExchange Listener 接続のプールが保持されます。プールされた接続のために、ファイルまたはデータベースを開いたままの状態を維持することはありません。

接続プールは以下のクライアントによりサポートされます。

- Data Integration Service。Informatica Administrator を使用して、Data Integration Service のための接続プールのプロパティを定義します。接続プールを有効にする前に、PowerExchange Listener システム上の DBMOVER ファイル内に存在する MASTASKS 文の値が、Listener タスク用プール内の接続最大数に十分に対応できる大きさであることを確認します。詳細については、『*Informatica 管理者ガイド*』を参照してください。
- PowerExchange Navigator。【設定】ダイアログボックスで接続プールのプロパティを定義して、PowerExchange Navigator のための接続プールを設定します。詳細については、『*PowerExchange ナビゲータユーザーガイド*』を参照してください。
- DTLUCBRG ユーティリティ。接続プールのサイズが接続数 6 で保持期間が 30 秒の接続プールが DTLUCBRG によって自動的に有効になります。これらの値は変更できません。
- DTLURDMO ユーティリティ。接続プールのサイズが接続数 2 で保持期間が 30 秒の接続プールが DTLURDMO によって自動的に有効になります。これらの値は変更できません。
- z/OS 上の PowerExchange エージェント。PowerExchange エージェントは、場合によっては自動的に接続プールを使用します。接続プールは PowerExchange エージェント用に設定できません。
- PowerCenter Integration Service。PowerCenter Integration Service は、場合によっては自動的に接続プールを使用します。PowerCenter Integration Service 用に接続プールは設定できません。
- Informatica Developer ツール。デベロッパツールは、場合によっては自動的に接続プールを使用します。接続プールはデベロッパツール用に設定できません。

データマップのキャッシュ

データマップが単一の PowerExchange リスナジョブによって、または複数の PowerExchange リスナと Netport ジョブによって使用されるシステムでは、非リレーショナル z/OS データソースに対して定義されたデータマップをキャッシュするように PowerExchange を設定できます。

注: データマップのキャッシュを IMS Netport ジョブと一緒に使用することはできません。

データマップのキャッシュを有効にすると、PowerExchange は、データマップファイルにアクセスする代わりに、ジョブレベルメモリからデータマップを取得します。データマップファイルにアクセスしないため、データマップのキャッシュによりパフォーマンスが向上します。

データマップのキャッシュを有効にすると、接続プールによってパフォーマンスが向上するだけでなく、初期化時間を 0.1 秒短縮することができます。データマップのキャッシュは、多数の短いトランザクションが 1 秒ごとに完了する場合に役に立ちます。ただし、実行時間の長いバルクデータ移動セッションを実行する場合に、初期化時間が処理時間全体のごく一部にすぎないときは、データマップのキャッシュを使用してもパフォーマンスが向上しない可能性があります。

データマップのキャッシュを有効にして設定するには、PowerExchange リスナジョブまたは Netport ジョブが実行されるシステムで DBMOVER 文を定義します。単一ジョブモードまたは複数ジョブモードでデータマップのキャッシュを実行するように、PowerExchange を設定できます。

データマップのキャッシュの有効化

データマップのキャッシュを有効にするには、DBMOVER コンフィギュレーションファイルの DMXCACHE_MAX_MEMORY_MB 文を、0 よりも大きな値に設定します。

この値は、データマップキャッシュの最大サイズを MB 単位で設定します。

DMXCACHE_MAX_MEMORY_MB 文の詳細については、『*PowerExchange リファレンスマニュアル*』を参照してください。

注: JCL の DTLLST の手順で指定した REGION のサイズが PowerExchange Listener ジョブを実行できる十分な大きさであることを確認します。例えば、DMXCACHE_MAX_MEMORY_MB=20 と定義する場合は、REGION のサイズを 20 MB 増やすことが必要になる可能性があります。

単一ジョブモードまたは複数ジョブモードで実行するためのデータマップのキャッシュの設定

単一ジョブモードまたは複数ジョブモードでデータマップのキャッシュを実行するように、PowerExchange を設定できます。

1 つの PowerExchange Listener ジョブを使用し、Netport ジョブを使用しない場合は、単一ジョブモードでデータマップのキャッシュを実行するように PowerExchange を設定します。

複数の PowerExchange Listener または Netport ジョブを使用して同じデータマップファイルにアクセスする場合は、複数ジョブモードでデータマップのキャッシュを実行するように PowerExchange を設定します。

単一ジョブモード

単一ジョブモードでは、以前に読み取ったデータマップのコピーがジョブレベルメモリに保持されます。PowerExchange では、データマップに関する情報の保持に ESCA メモリは使用されません。

その後のデータマップの読み取り時に、データマップがジョブレベルメモリから取得されるため、データマップ KSDS ファイルをキューに入れて開き、ポイントして読み取ってから閉じる操作のオーバーヘッドがなくなります。

データマップが変更または削除されると、キャッシュ内のそのデータマップのコピーが削除されます。データマップが変更された場合、データマップは実際に使用されるまでキャッシュに再度追加されません。

単一ジョブモードでは、PowerExchange Listener ジョブがデータマップファイルを更新する唯一のジョブであり、データマップのキャッシュは古くならないため、単一ジョブモードは複数ジョブモードよりも効率的です。したがって、複数ジョブモードのようにキャッシュを完全に消去する必要はありません。

単一ジョブモードでデータマップのキャッシュを実行するように PowerExchange を設定するには、次の DBMOVER 設定文を定義します。

```
DMXCACHE_MULTIPLEJOBS=N
```

DMXCACHE_MULTIPLEJOBS 文の詳細については、『*PowerExchange リファレンスマニュアル*』を参照してください。

複数ジョブモード

複数ジョブモードでは、以前に読み取ったデータマップのコピーがジョブレベルメモリに保持されます。また、PowerExchange Listener の起動時に、PowerExchange では、データマップ KSDS ファイルの名前とその最終更新のタイムスタンプを保持するための 4096 バイトの ECSA メモリが動的に割り当てられます。PowerExchange は、この情報を使用して、データマップのキャッシュが古いために消去する必要があるかどうかを判断します。

その後のデータマップの読み取り時に、データマップがジョブレベルメモリから取得されるため、データマップ KSDS ファイルをキューに入れて開き、ポイントして読み取ってから閉じる操作のオーバーヘッドがなくなります。

データマップが変更または削除されると、次の処理が実行されます。

- ECSA メモリで、データマップ KSDS ファイルに対する最終更新のタイムスタンプが変更されます。

- 別のタスクがデータマップにアクセスしようとしたときに、PowerExchange は、ECSA メモリでのタイムスタンプとデータマップのキャッシュのタイムスタンプを比較して、キャッシュが古いかどうかを判断します。キャッシュが古い場合は消去され、必要なデータマップがディスクから読み取られてキャッシュに追加されます。

複数ジョブモードでデータマップのキャッシュを実行するように PowerExchange を設定するには、次の DBMOVER 設定文を定義します。

```
DMXCACHE_MULTIPLEJOBS=Y
```

また、PowerExchange Listener を閉じた後に PowerExchange で ECSA メモリが解放されるようにする場合は、次の DBMOVER 設定文を定義します。

```
DMXCACHE_DELETEECSA=Y
```

ただし、共有メモリの割り当ては複雑なタスクで、4096 バイトのメモリは少量なので、デフォルト値の N を受け入れて ECSA メモリを永続的に保持することもできます。

DBMOVER のこれらの文の詳細については、『*PowerExchange リファレンスマニュアル*』を参照してください。

データマップのキャッシュの設定 - 例

この例では、データマップのキャッシュを有効にして、データマップのキャッシュの最大サイズを 20 MB に設定します。このサイズは、20 個の大規模な IDMS データマップまたは何千もの小規模なデータマップを保持できる十分な大きさです。

JCL の DTLLST の手順で指定した REGION のサイズが PowerExchange Listener ジョブを実行できる十分な大きさであることを確認します。例えば、DMXCACHE_MAX_MEMORY_MB=20 と定義する場合は、REGION のサイズを 20 MB 増やすことが必要になる可能性があります。

1. データマップのキャッシュを有効にして、データマップのキャッシュサイズを 20 MB に設定するには、DBMOVER 構成ファイルで次の文を定義します。

```
DMXCACHE_MAX_MEMORY_MB=20
```
2. 環境に応じて、データマップのキャッシュを単一ジョブモードまたは複数ジョブモードで実行するように PowerExchange を設定します。

次の表では各モードの環境および設定手順について説明します。

環境	モード	設定手順
1 つの PowerExchange Listener ジョブを使用し、Netport ジョブを使用しない場合。	単一ジョブモード	次の DBMOVE 設定文を定義します。 DMXCACHE_MULTIPLEJOBS=N この場合は、デフォルトのキャッシュサイズで十分です。 デフォルトは N です。 単一ジョブモードの詳細については、「 単一ジョブモード 」(ページ 199)を参照してください。
複数の PowerExchange Listener または Netport ジョブを使用して同じデータマップファイルにアクセスする場合。	複数ジョブモード	次の DBMOVE 設定文を定義します。 DMXCACHE_MULTIPLEJOBS=Y また、PowerExchange Listener を閉じた後に PowerExchange で ECSA メモリが解放されるようにする場合は、次の DBMOVE 設定文を定義します。 DMXCACHE_DELETEECSA=Y ただし、共有メモリの割り当ては複雑なタスクで、4096 バイトのメモリは少量なので、デフォルト値の N を受け入れて ECSA メモリを永続的に保持することもできます。 複数ジョブモードの詳細については、「 複数ジョブモード 」(ページ 199)を参照してください。

- データマップのキャッシュを複数ジョブモードで実行する場合は、必要に応じて、PWXUDMX ユーティリティを実行して ECSA メモリを表示できます。
PWXUDMX ユーティリティの詳細については、『PowerExchange ユーティリティガイド』を参照してください。
- データマップのキャッシュによってパフォーマンスが向上したかどうかを判断するには、PowerExchange リスナを閉じて、PowerExchange ログのメッセージを確認します。

バルクデータオフロードとマルチスレッド処理

バルクデータオフロード処理を使用すると、PowerExchange によってカラムレベルの処理がバルクデータ移動セッションを実行する PowerCenter Integration Service マシンに移行されます。PowerExchange には、大半のデータソースについて、ソースデータのフィルタリングを PowerCenter 統合サービスマシンに移動するオプションが用意されています。

バルクデータオフロード処理は、以下のデータソースに使用できます。

- DB2 for z/OS テーブル
- DB2 for z/OS イメージコピー
- シーケンシャルファイルとフラットファイル
- VSAM データセット
- IMS アンロードファイル

バルクデータオフロード処理は、以下のデータターゲットに使用できます。

- z/OS シーケンシャルファイル

- VSAM データセット
- IMS アンロードファイル

大半のデータソースでは、オフロード処理を使用するバルクデータ移動セッションで、マルチスレッド処理を使用できます。デフォルトでは、PowerExchange は単一のスレッドを使用して PowerCenter Integration Service マシン上のバルクデータを処理します。マルチスレッド処理を設定すると、PowerExchange は複数のスレッドを使用してバルクデータレコードを処理します。

バルクデータオフロード処理に関するルールおよびガイドライン

オフロード処理を使用するバルクデータ移動操作には、以下の制限が適用されます。

- バルクデータオフロード処理は、以下のオプションのいずれかを使用しているデータマップには実行できません。
 - ユーザーアクセス方法
 - CALLPROG 関数を使用してプログラムを起動するユーザー定義フィールド
 - レコードレベルの終了
- [アクセス方式]ダイアログボックスで[デフォルト]以外の[エンコード]設定があるソースデータマップの場合は、PWX NRDB 接続の[オフロード処理]オプションで[事後フィルタ]を指定します。これらのデータマップの場合は、[オフロード処理]オプションで[事前フィルタ]を指定しないでください。
- IMS アンロードファイルおよび DB2 for z/OS イメージコピーデータソースに【事前フィルタ】を選択した場合は、PowerExchange によって【事後フィルタ】に変更されます。
- 同一の VSAM*ファイルを指す複数のデータターゲットをマッピング内に含めることはできません。
- データターゲットへの挿入のみがサポートされています。削除と更新はサポートされていません。
- データターゲット用のデータマップには、レコードレベルの出口を使用することはできません。

マルチスレッド処理に関するルールおよびガイドライン

マルチスレッド処理は、特定の状況において、バルクデータ移動セッションのパフォーマンス向上に役立つことがあります。

マルチスレッド処理を設定する前に、以下のルールとガイドラインを確認します。

- PowerCenter Integration Service マシンの複数 CPU サーバーでバルクデータ移動セッションの PWX reader スレッドによって 1 つの CPU が 100%使用されている場合、マルチスレッド処理を使用します。この場合、マルチスレッド処理によってバルクデータ処理が複数のスレッドに分散されることで、セッションスループットを向上させることができます。しかし、その 1 つの CPU が 100%使用されていない場合には、スレッドが増えてもスループットは向上しません。
- ソースシステムと PowerCenter 統合サービスマシン間のネットワーク処理が遅い場合には、【ワーカーズレッド】接続属性に 1 を指定してスループットを向上させます。1 つ以上のワーカーズレッドを使用すると、PowerExchange ではネットワーク処理と PowerCenter 統合サービスマシンマシン上のバルクデータ処理が重複されます。
- 最適なパフォーマンスを得るためには、PowerCenter 統合サービスマシンにインストールされているプロセッサまたはこのマシンで使用可能なプロセッサの数を超えないように【ワーカーズレッド】属性の値を設定します。
- [SEQ アクセス方式] ダイアログボックス内の【変数】プロパティに【VS】を指定するシーケンシャルファイルとフラットファイルのデータマップには、マルチスレッド処理を使用できません。
- 可変長レコードが入った z/OS シーケンシャルファイルのために Linux、UNIX、または Windows でマルチスレッド処理を使用する場合は、ファイル転送メカニズムによってレコード記述子が削除されないことを確認します。各レコードにはレコード長のプレフィックスが付けられている必要があります。

- 複数のワーカースレッドを使用してデータをターゲットに書き込むことはできません。代わりに、データに Insert だけが含まれる場合は、Writer パーティション化を使用することを検討します。
- バルクデータ移動セッションに Reader パーティション化または Writer パーティション化を使用した場合、**【ワーカースレッド】** 設定は無視されます。
- 一部の NRDB Batch 接続タイプはワーカースレッドをサポートしていません。これらの接続のいずれかの **【ワーカースレッド】** 接続属性がゼロ以外の値に設定されている場合は、スレッドなしで処理が続行されます。
- メッセージ PWX-31505 に接続パフォーマンス設定（ワーカースレッドの数を含む）がレポートされます。ワーカースレッドが使用されない場合は、メッセージ PWX-31505 にワーカースレッド数としてゼロがレポートされます。

バルクデータ移動セッションのためのオフロード処理とマルチスレッド処理の有効化

バルクデータオフロード処理とマルチスレッド処理を使用するには、バルクデータ移動セッションのための接続属性をいくつか設定する必要があります。

シーケンシャルファイルとフラットファイル、VSAM データセット、IMS アンロードデータセット、DB2 イメージコピーの場合は、PWX NRDB バッチアプリケーション接続にバルクデータ移動セッションのための属性を設定します。

以下の表に、PWX NRDB Batch アプリケーション接続で設定する属性を示します。

接続属性	説明
オフロード処理	<p>ソースシステムまたはターゲットシステムから PowerCenter 統合サービスへの PowerExchange バルクデータ処理の移行にオフロード処理を使用するかどうかを指定します。</p> <p>データソースの場合、次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> - いいえ。オフロード処理は使用しません。 - 自動。オフロード処理を使用するかどうか PowerExchange によって決定されます。 - 事前フィルタ。カラムレベルの処理を Integration Service マシンにオフロードしますが、データのフィルタリングはソースシステム上で継続します。 - 後でフィルタ。カラムレベルの処理とデータのフィルタリングを統合サービスマシンにオフロードします。 <p>データターゲットの場合、次のいずれかの値を選択します。</p> <ul style="list-style-type: none"> - いいえ。オフロード処理は使用しません。 - 自動。オフロード処理は使用しません。 - 事前フィルタ。オフロード処理を使用します。フィルタリングは、データターゲットのオフロード処理に適用されません。 - 後でフィルタ。オフロード処理を使用する。フィルタリングは、データターゲットのオフロード処理に適用されません。 <p>デフォルトは [いいえ] です。</p>
ワーカーレッド	<p>オフロード処理を有効にする場合は、バルクデータの処理のために PowerExchange が統合サービスマシン上で使用するスレッドの数を指定します。</p> <p>最適なパフォーマンスを得るためには、PowerCenter 統合サービスマシンにインストールされているプロセッサまたはこのマシンで使用可能なプロセッサの数を超えないように値を割り当てます。</p> <p>有効な値は 1~64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。</p>
配列サイズ	<p>VSAM データセットまたはシーケンシャルファイルで、[ワーカーレッド] の値が 0 より大きい場合は、スレッドのストレージ配列のサイズ（単位: レコード数）です。</p> <p>DB2 for z/OS イメージコピーの場合は、割り当てる必要があるメモリバッファの最大数を決定します。zIIP 処理を有効にした場合、[配列サイズ] は、zIIP プロセッサにディスパッチする圧縮された行数も表しています。</p> <p>有効な値は 25~5000 です。デフォルトは 25 です。</p> <p>より大きな配列サイズに割り当てられた追加のメモリが有益で、サーバーのパフォーマンスが低下していないかどうかをテストおよび定量化できる場合を除き、デフォルト値を使用することをお勧めします。これらの判断ができる場合は、マルチスレッド処理のオフロードを有効にするときに配列サイズを 500~1000 にすることをお勧めします。</p> <p>警告: 大きな値を指定したり、レコード数を大きくしたり、マルチスレッド処理を使用するセッションを多数実行したりすると、統合サービスマシンでメモリ不足が発生することがあります。</p>

DB2 for z/OS テーブルの場合、バルクデータ移動セッション用の PWX DB2zOS リレーショナル接続に属性を設定します。

以下の表に、PWX DB2zOS リレーショナル接続で設定する属性を示します。

接続属性	説明
オフロード処理	バルクデータオフロード処理を使用して、DB2 データ用のカラムレベル処理を、ソースシステムから PowerCenter 統合サービスマシンへ移動するかどうかを指定します。 次のいずれかの値を選択します。 <ul style="list-style-type: none">- いいえ。オフロード処理は使用しません。- はい。オフロード処理を使用します。- 自動。オフロード処理を使用するかどうか PowerExchange によって決定されます。デフォルトは [いいえ] です。
ワーカースレッド	オフロード処理を有効にする場合は、バルクデータの処理のために PowerExchange が統合サービスマシン上で使用するスレッドの数を指定します。 最適なパフォーマンスを得るためには、統合サービスマシンにインストールされているプロセッサまたはこのマシンで使用可能なプロセッサの数を超えないようにこの値を設定します。 有効な値は 1～64 です。デフォルトは 0 です。マルチスレッド処理は無効になります。
配列サイズ	DB2 フェッチ配列サイズを指定します。DB2 フェッチ配列サイズの詳細については、IBM DB2 のマニュアルを参照してください。 有効な値は 1～100000 です。デフォルトは 25 です。 より大きな配列サイズに割り当てられた追加のメモリが有益で、サーバーのパフォーマンスが低下していないかどうかをテストおよび定量化できる場合を除き、デフォルト値を使用することをお勧めします。これらの判断ができる場合は、オフロードとマルチスレッド処理を有効にするときに配列サイズを 500～1000 にすることをお勧めします。 警告: 大きな値を指定したり、レコード数を大きくしたり、マルチスレッド処理を使用するセッションを多数実行したりすると、統合サービスマシンでメモリ不足が発生することがあります。

バルクデータ移動セッション内のパイプラインパーティション

パフォーマンスを向上させるために、バルクデータ移動セッション内の Reader パーティションポイントと Writer パーティションポイントでパイプラインパーティション化を使用することができます。パーティション化を有効にすると、セッションにおいて複数のパーティションにわたり同時にデータを処理できます。

Reader パーティション、Writer パーティション、またはこの両方を設定できます。Reader パーティション化の場合、データソースの一部または全体に対応できる代替パーティション化スキームを使用できます。Writer パーティション化の場合、VSAM またはシーケンシャルファイルターゲットを使用したパススルーパーティション化を使用する必要があります。

Reader および Writer パーティション化スキームは、標準の PowerCenter パイプラインパーティション化動作に準じています。Reader パーティション化または Writer パーティション化を有効にすると、PowerCenter によりパイプライン全体を通してパーティションが使用されます。パイプラインにおける 1 つのステージでパーティションカウントを変更した場合、PowerCenter は他のすべてのステージでパーティションの数を更新して一致させます。パイプラインパーティション化の詳細については、『PowerCenter 上級ワークフローガイド』を参照してください。

Reader パーティション化

パススルーパーティションまたはキー範囲パーティションをソース修飾子または Reader のパーティションポイントで定義することによって、バルクデータ移動セッションのパフォーマンスを向上させることができます。Reader パーティション化を設定すると、セッションで 1 つ以上のパーティションを使用し、ソースからのデータの読み取りと処理が行えるようになります。

同時処理の程度は、使用するデータソースとパーティション化スキームによって異なります。

- オフロードされた DB2 for z/OS アンロードデータセット、VSAM データセット、およびシーケンシャルファイルの場合、PowerExchange はデータソースに対して単一の接続を開き、複数のパーティションを使用してソースデータの読み取りと処理を行います。
- 他のデータソースの場合、PowerExchange はデータソースに対して複数の接続を開くか、あるいはソースデータを単一のパーティションに読み込みます。PowerExchange が単一のパーティションにデータを読み込んだ場合は、再パーティションポイントでデータを再配分することによって、後続のパイプラインステップでパーティション化を使用できます。

データ処理速度はパーティションによって異なるため、セッションはシーケンスに関係なくデータを処理する可能性があります。

以下の表に、さまざまなタイプのバルクデータソースに使用できる Reader パーティション化スキームの概要を示します。

Reader パーティション化スキーム	サポートされるデータソース	説明
キー範囲	すべてのリレーショナルバルクデータソース	データソースの行は、キー範囲値に基づいてパーティション化されます。 このパーティション化スキームは、リレーショナルデータソースにお勧めします。
SQL オーバーライドを行わないパススルーパーティション化	オフロードされた DB2 アンロードデータセット、VSAM データセット、およびシーケンシャルデータセット	PowerExchange によってソースデータが 1 度読み取られ、行が複数のパーティションにわたって自動的に分散されます。 SQL オーバーライドを行わないパススルーパーティション化を使用した場合、 【ワークスレッド】 接続属性は PowerExchange によって無視されます。 このパーティション化スキームは、指定された z/OS データソースにお勧めします。
SQL オーバーライドを行わないパススルーパーティション化	その他の非リレーショナルバルクデータソースすべて	データは最初のパーティションにだけ読み込まれます。後続のパーティションポイントでラウンドロビンパーティション化を使用し、データを再配分できます。
SQL オーバーライドを行うパススルーパーティション化	すべて	各行のデータは、SQL オーバーライドに基づいてパーティションに読み込まれます。 パーティションは、他のパーティションとは無関係に動作し、各クエリを独立した PowerExchange 要求として扱います。

データソースへの SQL オーバーライドを行わないパススルーパーティション化の設定

オフロード処理に次のデータソースを使用する場合は、以下の手順を実行して SQL オーバーライドを行わないパススルーパーティション化を設定します。DB2 for z/OS アンロードデータセット、VSAM データセット、およびシーケンシャルファイル。

PowerExchange によってソースデータが 1 度読み取られ、行が複数のパーティションにわたって自動的に分散されます。

1. Workflow Manager で以下の手順を実行し、PWX アプリケーション接続を開きます。
 - a. **【接続】 - 【アプリケーション】** > をクリックします。
【アプリケーション接続ブラウザ】 ダイアログボックスが表示されます。
 - b. **【タイプの選択】** フィールドで、**【PWX NRDB Batch】** を選択します。
 - c. **【オブジェクト】** リストで接続オブジェクトを選択し、**【編集】** をクリックします。または、**【新規作成】** をクリックし、接続を作成します。
2. **【接続オブジェクト定義】** ダイアログボックスで、**【オフロード処理】** リストから次のいずれかのオプションを選択します。
 - **事前フィルタ**。PowerExchange によってカラムレベル処理が PowerCenter Integration Service マシンにオフロードされますが、データのフィルタリングはソースシステム上で継続されます。
 - **事後フィルタ**。PowerExchange により、カラムレベル処理とデータフィルタリングが PowerCenter Integration Service マシンにオフロードされます。
3. **【OK】** をクリックします。
4. Task Developer で、セッションをダブルクリックして **【タスクの編集】** ダイアログボックスを開きます。
5. **【プロパティ】** タブで、**【PWX パーティション方式】** 属性に次のいずれかのオプションを選択します。
 - **【単一接続】**。PowerExchange により、データソースに対して単一の接続が作成されます。最初のパーティションに指定するオーバーライドはどれも、すべてのパーティションに使用されます。最初のパーティションのオーバーライドとは異なるオーバーライドを他のパーティションに指定した場合、セッションは失敗し、エラーメッセージが表示されます。
 - **オーバーライドドリブン**。指定したオーバーライドがすべてのパーティションで同一の場合、PowerExchange はデータソースに対して単一の接続を作成します。すべてのパーティションにわたってオーバーライドが同一でない場合、PowerExchange は複数の接続を作成します。
6. **【マッピング】** タブで、**【パーティション】** ビューをクリックします。
7. ソース修飾子トランスフォーメーションを選択し、**【パーティションポイントの編集】** をクリックします。
【パーティションポイントの編集】 ダイアログボックスが表示されます。
8. 追加するパーティションごとに、**【追加】** をクリックします。
9. **【パーティションタイプ】** が **【パススルー】** であることを確認します。
10. **【OK】** をクリックします。
11. **【マッピング】** タブで、**【トランスフォーメーション】** ビューをクリックします。
【プロパティ】 領域には、パーティションごとに **【SQL クエリ】** 属性が表示されます。
12. **【SQL クエリ】** 属性が空であるか、あるいは各パーティションに同じクエリが入った状態であることを確認します。
13. **【SQL クエリ】** 属性が空ではなく、同一のクエリが入っていない場合は、パーティションごとに以下の手順を実行します。
 - a. **【SQL クエリ】** 属性内の **【参照】** ボタンをクリックします。

- b. **【SQL エディタ】** ダイアログボックス内のクエリを消去するか、あるいは各パーティションに同じクエリを入力します。
 - c. **【OK】** をクリックします。
14. **【OK】** をクリックします。

非リレーショナルデータソースへの SQL オーバーライドを行うパススルーパーティション化の設定

オフロード処理を使用しない非リレーショナルデータソースに SQL オーバーライドを行うパススルーパーティション化を設定するには、以下の手順を実行します。

パーティションは、他のパーティションとは無関係に動作し、各クエリを独立した PowerExchange 要求として扱います。

1. Task Developer で、セッションをダブルクリックしてセッションプロパティを開きます。
2. **【マッピング】** タブで、**【パーティション】** ビューをクリックします。
3. ソース修飾子トランスフォーメーションを選択し、**【パーティションポイントの編集】** をクリックします。
【パーティションポイントの編集】 ダイアログボックスが表示されます。
4. 追加するパーティションごとに、**【追加】** をクリックします。
5. **【パーティションタイプ】** が **【パススルー】** であることを確認します。
6. **【OK】** をクリックします。
7. **【マッピング】** タブで、**【トランスフォーメーション】** ビューをクリックします。
【プロパティ】 領域には、パーティションごとに **【SQL クエリ】** 属性が表示されます。
8. パーティションごとに、**【SQL クエリ】** フィールドにある **【参照】** ボタンをクリックします。次に、**【SQL エディタ】** ダイアログボックスで、クエリを入力して **【OK】** をクリックします。
ヒント: ソース修飾子トランスフォーメーションの設定時に Designer でクエリを入力した場合は、各パーティションの **【SQL クエリ】** フィールドにそのクエリが表示されます。このクエリをオーバーライドするには、**【SQL エディタ】** ダイアログボックスに別のクエリを入力します。

最初の Reader パーティションのみへのデータの読み取り

最初の Reader パーティションにだけデータを読み込む場合は、以下の手順を実行します。このパーティション化スキームは、オフロード処理を使用しない非リレーショナルデータソースに適用されます。Reader パーティション内の行を後続の複数のパーティションに再配分することで、セッション処理の速度を高めることができます。

SQL オーバーライドを行わないパススルーパーティション化を使用する必要があります。最初の Reader パーティションの行を後続のパーティションに再配分するには、ラウンドロビンパーティションポイントを Reader パーティションの直後に追加します。

1. Task Developer で、セッションをダブルクリックして **【タスクの編集】** ダイアログボックスを開きます。
2. **【マッピング】** タブで、**【パーティション】** ビューをクリックします。
3. ソース修飾子トランスフォーメーションを選択し、**【パーティションポイントの編集】** をクリックします。
【パーティションポイントの編集】 ダイアログボックスが表示されます。
4. 追加するパーティションごとに、**【追加】** をクリックします。
5. **【パーティションタイプ】** が **【パススルー】** であることを確認します。
6. **【OK】** をクリックします。
7. **【マッピング】** タブで、**【トランスフォーメーション】** ビューをクリックします。

- 【プロパティ】領域には、パーティションごとに【SQL クエリ】属性が表示されます。
8. 【SQL クエリ】属性が空であるか、あるいは各パーティションに同じクエリが入った状態であるかを確認します。
 9. 【SQL クエリ】属性が空ではなく、各パーティションに同一のクエリが入っていない場合は、パーティションごとに以下の手順を実行します。
 - a. 【SQL クエリ】フィールドで、[参照] ボタンをクリックします。
 - b. 【SQL エディタ】ダイアログボックス内のクエリを消去するか、あるいは各パーティションに同じクエリを入力します。
 - c. 【OK】をクリックします。
 10. 【マッピング】タブで、[パーティション] ビューをクリックします。
 11. 最初の Reader パーティションの行を後続のパーティションに再配分するには、以下の手順を実行します。
 - a. 【パーティションポイントの追加】アイコンをクリックし、ソース修飾子トランスフォーメーションの後にパーティションポイントを追加します。
 - b. 【パーティションポイントの編集】をクリックします。
 - c. 【パーティションタイプ】リストで【ラウンドロビン】を選択し、【OK】をクリックします。
 12. 【OK】をクリックします。

リレーショナルデータソースのキー範囲パーティション化の設定

リレーショナルデータソースのキー範囲パーティション化を設定するには、以下の手順を実行します。

1. Task Developer で、セッションをダブルクリックしてセッションプロパティを開きます。
2. 【マッピング】タブで、[パーティション] ビューをクリックします。
3. ソース修飾子トランスフォーメーションを選択し、【パーティションポイントの編集】をクリックします。
【パーティションポイントの編集】ダイアログボックスが表示されます。
4. 追加するパーティションごとに、[追加] をクリックします。
5. 【パーティションタイプ】リストで、【キー範囲】を選択します。
6. 【パーティションキーの編集】ダイアログボックスで、キーに対し 1 つ以上のポートを選択し、【OK】をクリックします。
7. パーティションごとに、【範囲の開始】ボックスと【範囲の終了】ボックスに値を入力します。
8. 【OK】をクリックします。

Writer パーティション化

ターゲットインスタンスまたは Writer のパーティションポイントでパススルーパーティションを定義し、VSAM またはシーケンシャルファイルターゲットを持つバルクデータ移動セッションのパフォーマンスを向上させることができます。Writer パーティション化を使用すると、セッションで複数のパーティションを使用して SQL Insert を処理したり、SQL Insert をターゲットに同時に書き込んだりできます。

Writer パーティション化を使用して、バルクデータ移動セッションのパフォーマンスを向上させます。Writer パーティション化は、セッションに負荷の大きいトランスフォーメーション処理が含まれる場合に特に有効です。

必要に応じ、オフロード処理を使用し、ターゲットシステムから、セッションが実行される PowerCenter Integration Service マシンにフィールドレベルの処理を移行することもできます。ターゲット形式へのデータ変換のようなフィールドレベルの処理は、大量のリソースを消費しがちであり、ターゲット z/OS システムのチャージバックコストを高める可能性があります。この処理を PowerCenter Integration Service マシンにオ

フロードすると、処理は複数の Writer パーティションにわたって同時に実行されます。ターゲットへのデータ書き込みの効率を最適化するには、Writer パーティション化とターゲット側のオフロード処理の両方を使用します。

Writer パーティション化では、パススルーパーティションが使用され、SQL Insert 操作のみが処理されます。パーティションが SQL Update または SQL Delete を受け取ると、セッションは失敗し、エラーメッセージ PWXPC_12183 が表示されます。データに多数の更新または削除が含まれる場合には、Writer パーティション化を使用しないでください。

各 Writer パーティションはスレッドを 1 つ使用します。マルチスレッド処理を有効にすると、セッションの実行は継続しますが、**【ワーカスレッド】** 設定は無視されます。

パーティションレベルでセッションプロパティをいくつか指定できます。**【PWX パーティション方式】** 設定と、複数のパーティションにわたって同じセッションプロパティを指定するか異なるセッションプロパティを指定するかに基づき、PowerExchange では単一のターゲット接続または複数のターゲット接続（パーティションごとに 1 つ）が使用されます。

単一の接続を使用すると、ネットワーク上でターゲットにデータを送信する前に、複数パーティションからデータが Joiner スレッドによって統合されます。Joiner は、パーティションからデータを受け取る順序を維持しません。この結果、ターゲット上の PowerExchange Listener は、パーティションがデータを処理した順序とは異なる順序でデータを取得します。

ターゲットに対して、PWX NRDB Batch アプリケーション接続を使用する必要があります。**【書き込みモード】** 接続属性を使用して、Joiner による PowerExchange Listener へのデータ送信を同期をとって行うか同期をとらずに行うかを制御します。

複数のデータセットまたはファイルにデータを書き込む場合は、Writer パーティションに**【ファイル名のオーバーライド】** セッションプロパティを指定します。このプロパティを設定して、各パーティションに適したターゲットファイルを指定します。複数のパーティションが同じターゲットファイルに同時に書き込むと、競合が発生する可能性があります。

Writer パーティションに関するルールおよびガイドライン

Writer パーティション化には、以下のルールとガイドラインが適用されます。

- Writer パーティションは SQL Insert のみを処理します。Writer パーティションは、SQL Update や SQL Delete は処理しません。
- シーケンシャルファイルターゲットと VSAM ターゲット（VSAM ESDS、KSDS、RRDS データセットを含む）には、Writer パーティション化を使用できます。
- Writer パーティションは、パススルーパーティション化を使用する必要があります。
- ターゲットに対して、PWX NRDB Batch アプリケーション接続を使用する必要があります。
- ターゲットシステム上のリソースに制約がある場合は、Writer パーティション化と組み合わせてオフロード処理の使用を検討し、セッションパフォーマンスの向上を図ります。
- パーティション化されたセッションのパフォーマンスは、PowerCenter Integration Service マシン、ネットワーク速度、ターゲットシステムなどによって異なります。セッションの CPU 使用率が高い場合は、CPU が複数搭載された PowerCenter Integration Service マシンを使用します。また、ターゲットシステムとネットワークの速度が、パーティション化されたセッション処理の対応に十分であることを確認します。速度が十分ではないと、ボトルネックが発生する可能性があります。
- 単一のターゲット接続および同一のセッションオーバーライドをすべてのパーティションに使用する場合は、**【PWX パーティション方式】** セッションプロパティを **【単一接続】** に設定します。この設定より、最初のパーティションにセッションオーバーライドを指定するだけですみ、PowerCenter Integration Service は他のすべての Writer パーティションについてもこれらのオーバーライドを使用するようになります。

- PWX NRDB Batch 接続と Writer パーティション化を使用するバルクデータ移動セッションの **[Pre SQL]** プロパティ内に SQL 文を指定すると、セッションは各パーティションにこの SQL 文を実行します。セッションで 1 度しか実行できない SQL 文（CREATEFILE など）を指定すると、セッションは失敗します。最初のパーティションでその文を実行した後、他のパーティションで再びその文の実行を試みるときにセッションは失敗します。各パーティションで 1 度実行できる SQL 文を入力するか、Writer パーティションに **[Pre SQL]** 属性を指定しないようにします。

Writer パーティションのセッションプロパティ

Writer パーティションを使用したバルクデータ移動セッションの場合、セッションレベルとパーティションレベルで特定のセッションプロパティを指定できます。これらのプロパティはオプションです。

セッションレベルでは、VSAM データセット用の以下のセッションプロパティを入力することができます。

- 挿入のみ
- テーブルの切り詰めオプション

個々のパーティションごとに、VSAM ターゲットまたはシーケンシャルファイルターゲットのいずれかについて、オーバーライドを含む以下のセッションプロパティを入力できます。

- ファイル名のオーバーライド
- Insert SQL のオーバーライド
- Post SQL
- Pre SQL

注: **[Pre SQL]** セッションプロパティの場合、各パーティションで実行できる SQL 文を入力します。セッションで 1 度しか実行できない文（CREATEFILE など）を入力すると、セッションは失敗します。

これらのプロパティの詳細については、『*PowerCenter 用の PowerExchange インタフェース*』を参照してください。

[PWX パーティション方式] セッションプロパティと、複数のパーティションにわたって同じパーティションレベルセッションプロパティを指定するか、別のパーティションレベルセッションプロパティを指定するかに基づき、Joiner では単一のターゲット接続と複数のターゲット接続のどちらを使用するかが決定されます。

- **[PWX パーティション方式]** プロパティに **[単一接続]** を選択する場合、セッションは単一の接続を使用します。最初のパーティションだけにセッションプロパティを入力すると、PowerCenter Integration Service によってそのプロパティが他のパーティションすべてに適用されます。複数のパーティションに異なるセッションプロパティを指定すると、セッションは失敗します。
- **[オーバーライドドリブン]** を選択し、すべてのパーティションに同じセッションプロパティを入力すると、セッションはターゲットに対して単一の接続を使用します。パーティションに異なるセッションプロパティを入力すると、セッションは複数の接続（パーティションごとに 1 つ）を使用します。

パススルー Writer パーティション化の設定

VSAM ターゲットまたはシーケンシャルファイルターゲットを持つバルクデータ移動セッションのパススルー Writer パーティション化を設定するには、以下の手順を実行します。これらの手順では、Writer パーティションに対しての接続属性のカスタマイズ、パーティションの追加、セッションプロパティの設定を行います。

1. Workflow Manager で、次のように PWX NRDB Batch アプリケーション接続を作成するか、または開きます。
 - a. **[接続] - [アプリケーション]** > をクリックします。
[アプリケーション接続ブラウザ] ダイアログボックスが表示されます。

- b. **【タイプの選択】** リストで、**【PWX NRDB Batch】** を選択します。
 - c. **【オブジェクト】** リストで接続オブジェクトを選択し、**【編集】** をクリックします。または、**【新規作成】** をクリックし、接続を作成します。
2. **【接続オブジェクト定義】** ダイアログボックスで、接続属性を設定します。
- 以下の表に、Writer パーティション化に影響を与える属性を示します。

属性	説明
オフロード処理	<p>オフロード処理が使用されるかどうかを制御します。 次のいずれかのオプションを入力します。</p> <ul style="list-style-type: none"> - いいえ。オフロード処理は使用しません。 - 自動。オフロード処理を使用するかどうかを PowerExchange に決定させます。 - 事後フィルタ。レコードレベルの処理を PowerCenter Integration Service マシンにオフロードしますが、データのフィルタリングはターゲットシステム上で継続します。 - 事前フィルタ。レコードレベルの処理とフィルタリングを PowerCenter Integration Service マシンにオフロードします。 <p>デフォルトは 【いいえ】 です。</p> <p>パーティション化されたセッションの場合、【事前フィルタ】 を選択し、パフォーマンスを最適化します。</p>
ワーカースレッド	<p>マルチスレッド処理が使用されるかどうかを制御します。 マルチスレッド処理を無効にするには「0」を入力します。</p> <p>Reader パーティションまたは Writer パーティションを使用した複数のワーカースレッドを使用することはできません。</p>
配列サイズ	<p>複数のパーティションにわたって共有されるストレージ配列サイズ（単位: レコード数）です。 1～5000 の有効な値を入力します。 必要に応じて配列サイズを調整し、パーティション化されたバルクデータ移動セッションをチューニングします。</p> <p>デフォルトは 25 です。</p>

属性	説明
書き込みモード	<p>Joiner による Writer パーティションからターゲットへのデータ送信を、同期をとって行うか同期をとらずに行うかを制御します。 次のいずれかのオプションを選択します。</p> <ul style="list-style-type: none"> - 書き込み確認オン。Joiner は、ネットワークを介し、ターゲット上の PowerExchange Listener へと同期をとってデータを送信します。Joiner は、PowerExchange Listener からの確認メッセージを待機し、メッセージが到着してから次のデータを送信します。PowerExchange Listener によりエラーが検出されると、パーティションにエラーメッセージが返されます。このオプションはエラーから回復するのに便利ですが、データ転送の速度が遅くなる場合があります。 - 書き込み確認オフ。Joiner は、ファイルの末尾 (EOF) に達するか、または古いデータブロックを原因とするエラーが発生するまで、ターゲットに対してネットワークを介し同期をとらずにデータを送信します。エラーが発生すると、レコード番号、レコードデータ、処理されたレコードの数などの不正なレコード情報を報告するエラーメッセージが表示され、セッションが失敗する可能性があります。このオプションによりデータ転送の速度が上がりますが、エラーからは回復しにくくなります。このオプションは、エラーが発生した場合にターゲットファイルにデータを再びロードできる場合にのみ選択します。 <p>デフォルト値は 【書き込み確認オン】 です。</p>
PWX オーバーライド	<p>以下のオーバーライド値を入力し、PowerExchange Listener から返されるエラーに対して代替エラー処理方法を使用します。</p> <p>WRT_ERROR_HANDLING=Y</p> <p>この代替方法は、書き込みモード接続属性が 【書き込み確認オン】 に設定され、Writer パーティションに対する入力データに多数のエラーが含まれる場合に、より効果的な場合があります。このような状況でないかぎり、このオーバーライドを指定しません。</p> <p>PowerCenter 統合サービスマシン上の dbmover.cfg ファイル内に WRT_ERROR_HANDLING 文も指定できます。</p>

3. **【OK】** をクリックします。次に、**【閉じる】** をクリックします。
4. Task Developer で、セッションをダブルクリックします。
【タスクの編集】 ダイアログボックスが表示されます。
5. **【マッピング】** タブで、**【パーティション】** ビューをクリックし、パーティションビューを表示します。
6. VSAM ターゲットまたはシーケンシャルファイルターゲットを選択し、**【パーティションポイントの編集】** をクリックします。
【パーティションポイントの編集】 ダイアログボックスが表示されます。
7. パーティションを追加するには、**【追加】** をクリックし、パーティションタイプとして **【パススルー】** を選択します。また、説明も追加できます。
8. パーティションをすべて追加した後、**【OK】** をクリックします。
9. **【マッピング】** タブで、**【トランスフォーメーション】** タブをクリックし、トランスフォーメーションビューを表示します。
ターゲットが選択されていることを確認します。
10. **【接続値】** フィールドで、PWX NRDB Batch アプリケーション接続を選択します。

11. **【プロパティ】** で、**【PWX パーティション方式】** リストから次のいずれかのオプションを選択します。

- **【単一接続】**。PowerExchange により、データターゲットに対して単一の接続が作成されます。最初のパーティションに指定するオーバーライドはどれも、すべてのパーティションに使用されます。最初のパーティションのオーバーライドとは異なるオーバーライドを他のパーティションに指定した場合、セッションは失敗し、エラーメッセージが表示されます。
- **オーバーライドドリブン**。指定したオーバーライドがすべてのパーティションで同一の場合、PowerExchange はデータターゲットに対して単一の接続を作成します。すべてのパーティションにわたってオーバーライドが同一でない場合、PowerExchange は複数の接続を作成します。

12. ターゲットのタイプとプロパティに応じ、セッションレベルまたはパーティションレベルで、オプションのオーバーライドプロパティを設定します。

【PWX パーティション方式】 プロパティに **【単一接続】** を選択した場合は、最初のパーティションについてのみオーバーライドを入力できます。その後、PowerCenter Integration Service によって、セッション内のすべてのパーティションに、このオーバーライドが使用されます。

以下の表に、Writer パーティションに関連するセッションレベルとパーティションレベルのセッションプロパティを示します。

プロパティ	説明	ターゲットタイプ	レベル
ファイル名のオーバーライド	PowerExchange データマップ内のデータセット名をオーバーライドする完全修飾データセット名を指定します。	シーケンシャルファイルおよび VSAM	パーティションレベル
挿入のみ	SQL Update と SQL Delete を Insert として処理します。 ターゲットにキーが存在しない場合、このオプションを選択する必要があります。	VSAM	セッションレベル
Insert SQL オーバーライド	PowerExchange に送信されたデフォルトの Insert SQL をオーバーライドする Insert SQL を指定します。	シーケンシャルファイルおよび VSAM	パーティションレベル
Pre SQL	ターゲットデータベース接続を使用したセッションの前に実行される SQL 文を指定します。	シーケンシャルファイルおよび VSAM	パーティションレベル
Post SQL	ターゲットデータベース接続を使用したセッションの後に実行される SQL 文を指定します。	シーケンシャルファイルおよび VSAM	パーティションレベル
テーブルの切り詰めオプション	バルクデータをロードする前にターゲットコンテンツを切り詰めます。 注: 切り詰めを正常に機能させるには、REUSE オプションを使用して VSAM データセットを定義する必要があります。	VSAM	セッションレベル

これらすべてのプロパティの詳細については、『PowerCenter ガイド用の PowerExchange インタフェース』を参照してください。

13. **[OK]** をクリックします。

パーティション化されたバルクデータ移動セッションのパフォーマンスの評価

パーティション化されたバルクデータ移動セッションのパフォーマンスを評価する統計を含むメッセージを、ユーザーが設定した間隔で出力するには、dbmover.cfg ファイル内の SHOW_THREAD_PERF 文を使用します。このメッセージは、Reader パーティションまたは Writer パーティションを使用したセッションに関連したものです。

PowerCenter Integration Service マシン上の dbmover.cfg コンフィギュレーションファイルに、SHOW_THREAD_PERF 文を入力します。この文により、統計メッセージ PWX-31261 と PWX-31262 を PowerExchange メッセージログに書き込む前に PowerExchange が処理する必要のあるレコードの数が定義されます。接続で **[PWX ログエントリの取得]** 属性を指定した場合、PWXPC によってこれらのメッセージが PowerCenter セッションログにも書き込まれます。

このメッセージにより以下の情報が提供されます。

- PWX-31261。一定数のレコードの I/O を処理するのに Joiner スレッドが費やした時間の平均、最小、および最大をミリ秒単位で示したものです。
- PWX-31262。パーティションにおける特定の数のレコードの I/O 遅延時間の平均、最小、および最大をミリ秒単位で示したものです。遅延時間は、パーティションがデータの受け取りを待機する時間です。このメッセージは、セッション内のパーティションごとに発行されます。

パーティション化されたセッションのチューニング

パーティション化されたバルクデータ移動セッションが予期されたとおりに効率よく実行されない場合、チューニングパラメータを調整してパフォーマンスの向上を図ることができます。

パーティション化されたセッションのチューニングを行うには、DBMOVER 構成ファイルに以下の文を設定します。

APPBUFSIZE 文

PowerCenter 統合サービスマシン上の DBMOVER 構成ファイルに APPBUFSIZE 文を設定します。この文は、PowerExchange がデータの読み取りまたは書き込みに使用する最大バッファサイズ（バイト単位）を指定するものです。バッファがこの最大サイズに達すると、PowerExchange は別のバッファの使用を開始する前に、要求しているシステムに対してネットワークを介してバッファデータを送信します。セッションパフォーマンスの向上を図るには、バッファサイズを増やします。

動的アプリケーションバッファのサイズ決定が有効な状態では、PowerExchange リスナが実行される間に行われるすべての接続のアプリケーションデータバッファの初期サイズが APPBUFSIZE 文で定義されます。PowerExchange は、必要に応じて個々の接続のアプリケーションデータバッファのサイズを動的に変更します。

APPBUFSIZEDYN 文

PowerCenter 統合サービスマシンにおける動的アプリケーションバッファのサイズ決定を有効にします。動的アプリケーションバッファのサイズ決定が有効な状態では、データソースへの接続ごとに、大きすぎてバッファに収まりきれないレコードがデータソースに含まれる場合は PowerExchange によってアプリケーションバッファのサイズが変更されます。動的アプリケーションバッファのサイズ決定はデフォルトで有効になっています。このサイズ決定は、DBMOVER 構成ファイルで「APPBUFSIZEDYN=Y」を指定することによって明示的に有効にすることができます。

MAXTASKS 文

Reader パーティション化または Writer パーティション化を使用するバルクデータ移動セッションで処理速度が低下するかハングする場合、MAXTASKS 値を大きくして、パフォーマンスを向上させます。

パーティション化されたセッションのチューニングを行うには、以下の PowerCenter パラメータを使用します。

【配列サイズ】 接続属性

この属性は、PowerCenter の PWX Batch NRDB アプリケーション接続に入力します。この属性は、パーティション化されたバルクデータ移動セッションで複数のパーティションにわたって共有されるストレージ配列サイズ（単位: レコード数）を指定します。この属性は、マルチスレッド処理のワークスレッドにも関係しています。セッションパフォーマンスの向上を図るには、配列サイズを増やします。有効な値は 1~5000 です。

注: 【書き込みモード】 接続属性を 【書き込み確認オン】 に設定し、配列サイズが APPBUFSIZE を超えている場合、ストレージ配列内のレコード数が以下の状態になるたびに PowerExchange はネットワーク上でバッファを送信します。

$records = array_size / partition_count$

【デフォルトのバッファブロックサイズ】 属性

このセッション設定属性は、パーティション化されたバルクデータ移動セッションに適応させる場合に設定します。この属性は、PowerCenter 統合サービスがデータとインデックスキャッシュをソースからターゲットに移行するために使用するデフォルトのバッファブロックサイズを指定します。ブロックサイズを小さくすると、APPBUFSIZE または 【配列サイズ】 値の増加によるパフォーマンス向上を抑えることができます。例えば、ブロックサイズに 10 レコードしか入らない場合、各パーティションの配列サイズは 10 レコードに制限されます。【書き込みモード】 接続属性を 【書き込み確認オン】 に設定すると、APPBUFSIZE も 10 レコードに制限されます。

ブロックサイズを設定する場合は、以下の基準を使用します。

- 【書き込み確認オン】 を使用する場合は、 $array_size / partition_count$ で計算されるレコード数に対応できるブロックサイズを入力します。さらに、同じ数のレコードを保持するように APPBUFSIZE を設定します。
- 【書き込み確認オフ】 を使用する場合は、 $array_size / partition_count$ で計算されるレコード数に対応できるブロックサイズを入力します。

また、オフロード処理を使用せず、ソースシステムまたはターゲットシステム上のリソースに制約がある場合は、オフロード処理を有効にします。この機能により、バルクデータのフィールドレベルの処理が、ソースシステムまたはターゲットシステムから PowerCenter 統合サービスマシンに移行されます。

PowerExchange zIIP Exploitation

IBM System z Integrated Information Processor (zIIP) は、通常の計算容量を確保し、一定のデータ計算および z/OS 上のトランザクション処理の負荷に関する全体的な総合コストを下げることを支援するように設計されています。ソフトウェアが zIIP 上の負荷を実行するように設計されている場合は、zIIP が適切な負荷を実行できます。

zIIP が 1 つ以上インストールされている場合、作業の一部が zIIP にオフロードされるように PowerExchange Listener を z/OS 上に設定できます。複数の PowerExchange Listener を実行している場合、作業を zIIP にオフロードするように各 PowerExchange Listener を設定できます。

zIIP 上で実行できるようにするには、作業は以下の要件を満たす必要があります。

- zIIP (System Utility プロセッサ (SUP) と呼ばれる) にオフロード可能として分類されている WorkLoad Manager 領域で実行する
- システム要求ブロック (SRB) 領域で実行する

SRB で実行するプログラムは以下の要件を満たす必要があります。

- スーパバイザ状態 (KEY=ZERO) で実行する。
- SVC を発行しない (SVC 13 (ABEND)は除く)
- サブルーチンでは、別のサブルーチンを呼び出さない。

PowerExchange zIIP Exploitation の DBMOVER 文

次の DBMOVER 文は zIIP 設定を制御します。

SUP_SSNAME=*subsystem_name*

作業を zIIP にオフロードするため、PowerExchange Listener の開始済みタスクを識別するサブシステム名を IBM Workload Manager に定義します。システムに複数の Listener が含まれる場合は、Listener ごとに異なる名前を定義できます。最大 8 文字の名前を入力します。

デフォルトは PWXLSTNR です。

SUP_SSTYPE=*subsystem_type*

作業がディスパッチされる zIIP 上の SRB 領域のサブシステムタイプとして、IBM Workload Manager が使用する名前を定義します。最大 4 文字の名前を入力します。

デフォルトは、PWX です。

USESUP={N|Y}

PowerExchange が zIIP が有効な PowerExchange Listener 関数を zIIP にオフロードするかどうかを制御します。USESUP=Y と指定すると zIIP へのオフロードが有効になります。

WORKCLASS

Workload Manager の分類用にトランザクション名を定義します。最大 8 文字の名前を入力します。

デフォルトは PWXWORK です。

PowerExchange zIIP Exploitation の z/OS システムログメッセージ

PowerExchange は、z/OS システムログにメッセージを発行して PowerExchange zIIP 操作のステータスを報告します。

メッセージ ID の形式は以下のとおりです。

PWXmmm34xxs

文字列 *mmm* は呼び出しルーチンを表し、表示されたメッセージがエラーメッセージの場合は、Informatica グローバルカスタマサポートが使用する場合があります。

xx はメッセージ番号の末尾 2 桁です。

コード *s* は、情報メッセージの I またはエラーメッセージの E のいずれかになります。

これらのメッセージを使用して、以下のように zIIP 設定が成功したかどうかを判断します。

- 情報メッセージは設定の成功を示します。これらのメッセージが表示されない場合は、zIIP のオフロードの前提条件が満たされなかったことを示す可能性があります。詳細については、[「作業を zIIP にオフロードするように PowerExchange を設定」 \(ページ 218\)](#)を参照してください。
- エラーメッセージはエラー状態を示します。ほとんどの場合、Informatica グローバルカスタマサポートにお問い合わせいただく必要があります。

- メッセージ `PWXmmm3412E` および `PWXmmm3414E` はエラー状態の可能性を示しますが、`rc = 4` の場合は Informatica グローバルカスタマサポートにお問い合わせいただく必要はありません。

詳細については、『*PowerExchange メッセージリファレンスボリューム 1*』を参照してください。

作業を zIIP にオフロードするように PowerExchange を設定

作業を zIIP にオフロードするように PowerExchange を設定する前に、以下の前提条件が満たされていることを確認します。

- システム呼び出し可能なサービライブラリ `SYS1.CSSLIB` が、`LNKLST` 連結または `LPALIB` データセットから利用できます。
- システム `PARMLIB` の `IEAOPTxx` メンバの投影使用関数 (`PROJECTCPU`) が有効になっています。zIIP なしでシステム上での zIIP の使用を有効にして、`PROJECTCPU` を `FALSE` に設定すると、zIIP が存在するようにシステムで CPU 使用率が投影されず、PowerExchange は `IWM4EOCT` から `RC=4` をレポートします。PowerExchange は `SRB` モードで zIIP 対応機能を実行し続けます。
- PowerExchange リスナ `STEPLIB` 連結内のすべてのライブラリが、`APF` 許可されています。
- `DBMOVER` 構成メンバに `TRACE` 文が含まれていません。

1. z/OS で `DBMOVER` コンフィギュレーションファイルに `USESUP=Y` 文を含めて、オプションで次の文を含めます。

- `SUP_SSNAME`
- `SUP_SSTYPE`
- `WORKCLASS`

2. `PWX` を IBM Workload Manager for z/OS (WLM) に追加します。

- a. WLM ISPF アプリケーションのメインメニューから **PWX** をサブシステムタイプとして追加するか、`DBMOVER` 構成メンバの `SUB_SSTYPE` 文に指定した値を指定します。
- b. PowerExchange リスナごとに、`SI` (システムインスタンス) タイプの作業修飾子をリストに追加します。名前は `DBMOVER SUP_SSNAME` 文の値と一致する必要があります (デフォルト `PWXLSTNR`)。
- c. オプションで、`TN` 修飾子タイプを使用してデフォルトのトランザクション名を変更します。この値は `DBMOVER WORKCLASS` 文の値と一致する必要があります (デフォルト `PWXWORK`)。
- d. ジョブログをチェックして、zIIP の有効化に成功したことを確認します。

zIIP の有効化に成功すると、z/OS システムログに次のような情報メッセージが表示されます。

```
PWXDSP3400I Checking number_of_entries entries in PCCA vector table...
PWXDSP3401I Cpu 00 Serial FF04EEC52098 Type CP Rel. Speed 1
PWXDSP3401I Cpu 01 Serial FF04EEC52098 Type CP Rel. Speed 1
PWXDSP3401I Cpu 06 Serial FF04EEC52098 Type zIIP Rel. Speed 1
PWXDSP3403I 1 Processor available for zIIP offload
PWXWC03405I Connect to WLM Sub = PWX Subn = GRAPWX token = 140C2118
PWXWCF3409I Classify work to WLM Service class = 00238000
PWXWCE3411I WLM Create Enclave function = PWXFUNC enclave token = 0000003C00000033
PWXWSE3415I WLM Set Rules tok = PWXR id = IWMOCT ver = 00 cnt = 01 Dur = 1000000 Pct = 100
DTL-00607 Listener NODE1 VRM 9.5.0 Build DEV_BUILD started.
```

ジョブログに zIIP の有効化に成功したことを示すメッセージが表示されない場合、zIIP 有効化の前提条件が満たされていることを確認します。PowerExchange リスナ `STEPLIB` 連結内の一部のライブラリが `APF` 許可されていない場合、または `DBMOVER` 構成メンバに `TRACE` 文が含まれる場合、zIIP Exploitation は無効化されます。

PowerExchange リスナの WLM サービスクラスへの割り当て

Workload Manager (WLM) は、ストレージ、CPU、I/O デバイスなど、1 つ以上の z/OS イメージにまたがる z/OS 共有リソースの要求の優先順位を、定義したサービスクラスに基づいて動的に管理する z/OS コンポーネントです。PowerExchange リスナの開始済みタスクを適切な WLM サービスクラスに割り当てることにより、PowerExchange を介して z/OS データにアクセスするプロセスで、必要なパフォーマンスを達成することができます。

サービスクラスを定義するには、IBM z/OS WLM ツール (WLM ISPF アプリケーションなど) を使用します。各サービスクラスには、目標と重要度レベルが設定されます。このレベルにより、ビジネス要件に基づいて作業の優先順位が設定されます。PowerExchange リスナタスクの性質のため、PowerExchange リスナは、Velocity 実行目標が設定されたサービスクラスに割り当てることをお勧めします。

PowerExchange リスナのサブタスク

PowerExchange は、z/OS 上のデータベース、シーケンシャルおよび VSAM ファイルに格納されているデータに対する読み書きアクセス権を提供します。このアクセス権は、大容量のデータを読み書きする場合、または単一の行を取得するために使用できます。各アクセス要求は、PowerExchange リスナによってアタッチされる個別のサブタスクによって処理されます。

同じリスナを CDC データアクセス要求にも使用できます。これらの要求の詳細については、『*PowerExchange CDC ガイド (z/OS 版)*』の「CDC セッションのチューニング」の章を参照してください。

PowerExchange リスナのリソースの使用量

バルクデータ移動に使用される PowerExchange リスナのリソース使用量には以下に示す特徴があります。

- CPU 使用量は中程度。リスナが読み書きするデータの量と複雑さによって異なります。
ヒント: オフロード処理を使用すると、z/OS 上でバルクデータを処理するために必要な CPU サイクル数を減らすことができます。
- I/O アクティビティは中程度。リスナが読み書きするデータ量によって異なります。
- 仮想メモリの使用量は、同時にアクティブなリスナサブタスク数によって異なります。データを読み取るまたは書き込むための各要求は、個別のサブタスクによって処理されます。

PowerExchange リスナに必要なパフォーマンスプロファイルは、PowerExchange リスナを使用してデータにアクセスするプロセスの、z/OS 上で実行中の他のプロセスに対する相対的な重要度によって異なります。PowerExchange リスナに、z/OS データにアクセスする他のバッチジョブのサービスクラスと同じサービスクラスを割り当ててください。

索引

A

ADA_L3_ALLOW 文
 DBMOVER 構成ファイル [52](#)
Adabas
 暗号コード [57, 59](#)
ADABAS_DEFAULT 文
 DBMOVER 構成ファイル [53](#)
ADABAS_MU_SEARCH 文
 DBMOVER 構成ファイル [53](#)
ADABAS_PREFIX 文
 DBMOVER 構成ファイル [54](#)
ADABASCODEPAGE 文
 DBMOVER 構成ファイル [55](#)
Adabas データベースの読み取り処理を最適化
 ADA_L3_ALLOW 文 [52](#)
 ADAOPT 文 [52](#)
Adabas バルクデータ移動
 Adabas LOADLIB へのアクセス [58](#)
 ADABAS_MU_SEARCH 文 [52](#)
 ADARUN 用のデフォルト SVC 番号のオーバーライド [58](#)
 DBMOVER 構成文 [52](#)
 設定タスク [51](#)
 バルクデータの移動 [60](#)
 リモートの Adabas ソースまたはターゲットへの接続 [51](#)
ADAOPT 文
 DBMOVER 構成ファイル [55](#)
ADAPREFETCH 文
 DBMOVER 構成ファイル [56](#)
ADASTATS 文
 DBMOVER 構成ファイル [57](#)
APF 許可
 :Adabas LOAD ライブラリ要件 [58](#)
 IDMS ロードライブラリのコピー [116](#)
 PowerExchange リスナ要件 [17, 43](#)
 z/OS プログラム呼び出しサービスによる IDMS ロードライブラリ [117](#)
APPBUSIZEDYN 文
 DBMOVER 構成ファイル [18](#)
APPBUSIZE 文
 DBMOVER 構成ファイル [18](#)

C

CA TCPAccess
 PowerExchange 用の設定 [43](#)
close (pwxcmd) [47](#)
closeforce (pwxcmd) [47](#)
CMDNODE 文
 DBMOVER 構成ファイル [19](#)
COBOL コピーブック
 IMS バルクデータ移動のためのインポート [120](#)
CONFIG 文
 DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [42](#)

copybooks
 IMS バルクデータ移動のためのインポート [120](#)
CREDENTIALS_CASE 文
 DBMOVER 構成ファイル [20](#)

D

Datacom バルクデータ移動
 概要 [62](#)
 構成タスク [62](#)
 バルクデータの移動 [63](#)
 リモート Datacom ソースへの接続 [62](#)
DATAMAP_SERVER 文
 DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [21](#)
DB2 for Linux、UNIX、および Windows バルクデータ移動
 DBMOVER 構成文 [77](#)
 構成タスク [77](#)
 サポートされるデータタイプ [76](#)
 リモートの DB2 ソースまたはターゲットへの接続 [77](#)
DB2 for z/OS
 DB2 リソースへのアクセスに必要な権限 [93](#)
 TIMESTAMP データ型 [82](#)
 バルクデータ移動用にサポートされているデータタイプ [81](#)
DB2 for z/OS アンロードファイル
 アンロードファイルを作成するためのツール [99](#)
DB2 LOAD ユーティリティ
 JCL テンプレートおよび制御カードテンプレート [101](#)
 バルクデータのロードを使用 [101](#)
DB2_BIN_AS_CHAR 文
 DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [69](#)
DB2_BIN_CODEPAGE 文
 DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [69](#)
DB2_ERRORFILE 文
 DBMOVER 構成ファイル [70, 78, 84](#)
DB2CODEPAGE 文
 DBMOVER 構成ファイル [85](#)
DB2DEF_ENCODING 文
 DBMOVER 構成ファイル [89](#)
DB2 for i5/OS
 バルクデータ移動用にサポートされているデータタイプ [67](#)
DB2 for i5/OS のバルクデータ移動
 DB2 アクセス方式によるバルクデータの移動 [71](#)
 DBMOVER 構成文 [68](#)
 SEQ アクセス方式によるバルクデータの移動 [72](#)
 概要 [65](#)
 構成タスク [68](#)
 リモートの DB2 ソースまたはターゲットへの接続 [68](#)
DB2 for LUW のバルクデータ移動
 概要 [75](#)
 バルクデータの移動 [79](#)
DB2 for z/OS のバルクデータ移動
 DB2 LOAD ユーティリティの使用 [101](#)
 DB2 LOAD 用の JCL テンプレートおよび制御カードテンプレート [101](#)
 DB2 リソースにアクセスするための権限の付与 [94](#)

DB2 for z/OS のバルクデータ移動 (続く)

DBMOVER 構成文 [84](#)

概要 [80](#)

構成タスク [83](#)

ソースとしての DB2 イメージコピー [96, 98](#)

ソースとしての圧縮イメージコピー [97](#)

非リレーショナル定義によるバルクデータの移動 [99](#)

複数行の FETCH 文および INSERT 文の使用 [94](#)

リモートの DB2 ソースまたはターゲットへの接続 [83](#)

リレーショナル定義によるバルクデータの移動 [96](#)

DB2ID 文

DBMOVER 構成ファイル [90](#)

DB2LDCTL

DB2 LOAD 用の制御カードテンプレート [101](#)

DB2LDCTP

DB2 LOAD 用の制御カードテンプレート [101](#)

DB2LDJCL

DB2 LOAD 用の JCL テンプレート [101](#)

DB2LDJCP

DB2 LOAD 用の JCL テンプレート [101](#)

DB2PLAN 文

DBMOVER 構成ファイル [91](#)

DB2 イメージコピー

ターゲットの実体化に使用 [96, 98](#)

DBDs

IMS バルクデータ移動のためのインポート [120](#)

DBMOVER 構成文

Adabas バルクデータ移動 [52](#)

CONFIG [42](#)

DB2 for i5/OS のバルクデータ移動 [68](#)

DB2 for Linux、UNIX、および Windows バルクデータ移動 [77](#)

DB2 for z/OS のバルクデータ移動 [84](#)

DB2 LOAD 文 [103](#)

IDMS バルクデータ移動 [114](#)

IMSID [132](#)

LICENSE [42](#)

Oracle バルクデータ移動 [148](#)

PowerExchange リスナ [17](#)

VSAM バルクデータ移動 [168](#)

zIIP Exploitation [217](#)

バルクデータ移動セッション [194](#)

DBMOVER 文

ADA_L3_ALLOW [52](#)

ADABAS_DEFAULT [53](#)

ADABAS_MU_SEARCH [53](#)

ADABAS_PREFIX [54](#)

ADABASCODEPAGE [55](#)

ADAOPT [55](#)

ADAPREFETCH [56](#)

ADASTATS [57](#)

APPBUFSIZE [18](#)

APPBUFSIZEDYN [18](#)

CMDNODE [19](#)

CREDENTIALS_CASE [20](#)

DATAMAP_SERVER [21](#)

DB2_BIN_AS_CHAR [69](#)

DB2_BIN_CODEPAGE [69](#)

DB2_ERRORFILE [70, 78, 84](#)

DB2CODEPAGE [85](#)

DB2DEF_ENCODING [89](#)

DB2ID [90](#)

DB2PLAN [91](#)

DM_SUBTASK [22](#)

DMX_DIR [23](#)

ERRROWNOTFOUND [179](#)

LISTENER [23](#)

LOADJOBFILE [114](#)

LOGPATH [26](#)

DBMOVER 文 (続く)

MAXTASKS [26](#)

MSS_ERRORFILE [141](#)

MVSDB2AF [91](#)

NODE [27](#)

ORA_ERRORFILE [148](#)

ORACLECODEPAGE [148](#)

PC_AUTH [115](#)

RDBMSINSRTDFLT [70, 78, 92](#)

SECURITY [29](#)

SESSID [92](#)

START_UP_USER_EXIT [57](#)

SUBMITTIMEOUT [32](#)

SVCNODE [33](#)

TCPIPVER [34](#)

TEMPHLQ [115](#)

TRACING [35](#)

VSAM [168](#)

DM_SUBTASK 文

DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [22](#)

DMX_DIR 文

DBMOVER 構成ファイル [23](#)

DTLREXE ユーティリティ

PowerExchange Listener のテスト [47](#)

E

ERRROWNOTFOUND 文

DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [179](#)

F

FILLER キーワード

IMS セグメントをコピーブックに追加 [127](#)

G

GetDatabaseKey 関数

IMS アンロードデータセット [130](#)

GetIMSRBByLevel 関数

IMS アンロードデータセット [130](#)

I

IDMS バルクデータ移動

DBMOVER 構成文 [114](#)

IDMS ロードライブラリコピーの APF 許可 [116](#)

Netport ジョブ設定 [117](#)

z/OS プログラム呼び出しサービスによるロードライブラリの APF 許可 [117](#)

構成タスク [112](#)

セキュリティ考慮事項 [113](#)

バルクデータの移動 [118](#)

リモート IDMS ソースへの接続 [113](#)

概要 [112](#)

IMS

アンロードファイルへの書き込み [133](#)

IMS Catalog Builder ユーティリティ

ユーティリティ出力用の PDSE の割り当て [123](#)

IMSID 文

DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [132](#)

IMS アンロードデータセット

CCK の取得 [130](#)

キーの RBA を取得する PowerExchange 関数 [130](#)

IMS アンロードデータセット (続く)

ターゲットとして [133](#), [136](#)

ターゲットのマテリアライズ [132](#)

データソース [130](#)

レコード形式 [130](#)

IMS カタログ

PowerExchange がカタログを使用するように設定 [122](#)

カタログ使用のための Listener JCL のカスタマイズ [124](#)

概要 [121](#)

IMS バルクデータ移動

DL/1 バッチデータマップを使用した構成 [125](#)

import DBD [120](#)

IMS アクセスに影響を与える設定 [128](#)

IMS アンロードデータセットを使用したターゲットのマテリアライズ [132](#)

IMS データベースからのターゲットのマテリアライズ [129](#)

PCB 値 [127](#)

構成に関する考慮事項 [122](#)

構成文 [125](#)

コピーブックのインポート [120](#)

実装に関する考慮事項 [127](#)

前提情報 [126](#)

ソースとしての IMS アンロードデータセット [130](#)

リモートの IMS ソースまたはターゲットへの接続 [126](#)

ルックアップトランスフォーメーション [129](#)

概要 [120](#)

infacmd pwx コマンド

コマンド処理サービスの設定 [33](#)

ユーザーによる実行の許可 [29](#)

L

LICENSE 文

DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [42](#)

LISTENER 文

DBMOVER 構成ファイル [23](#)

listtask (pwxcmd) [48](#)

LOADCTLFILE 文

DB2 バルクロード文 [103](#)

LOADJOBFILE 文

DBMOVER 構成ファイル [114](#)

LOGPATH 文

DBMOVER 構成ファイル [26](#)

M

MAXTASKS 文

DBMOVER 構成ファイル [26](#)

MSS_ERRORFILE 文

DBMOVER 構成ファイル [141](#)

MVSD2AF 文

DBMOVER 構成ファイル [91](#)

N

Netport ジョブ

IDMS バルクデータ移動に関する設定 [117](#)

NODE 文

DBMOVER 構成ファイル [27](#)

O

ODBASUPP 文

IMS バルクデータ移動 [125](#)

ORA_ERRORFILE 文

DBMOVER 構成ファイル [148](#)

Oracle

バルクデータ移動用にサポートされているデータタイプ [146](#)

ORACLECODEPAGE 文

DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [148](#)

Oracle バルクデータ移動

DBMOVER 構成文 [148](#)

概要 [145](#)

構成タスク [147](#)

バルクデータの移動 [149](#)

リモートの Oracle ソースまたはターゲットへの接続 [147](#)

P

PC_AUTH 文

DBMOVER 構成ファイル [115](#)

PCB、IMS

PROCOPT キーワードの設定 [122](#)

PCB (IMS)

識別子タイプ [127](#)

PowerExchange

オペレーティングシステムの認証の設定 [29](#)

PowerExchange リスナ

TCP/IP ポート番号 [41](#)

概要 [16](#)

起動コマンド [45](#)

構成タスク [17](#)

接続および状態のテスト [47](#)

停止コマンド [47](#)

PowerExchange リスナ

Listener JCL の構成 [43](#)

アクティブなタスクの停止 [48](#)

アクティブなタスクの表示 [48](#)

構成文 [17](#)

複数のメッセージおよびトレースログファイル [42](#)

PROCOPT キーワード

IMS PCB [122](#)

PSB、IMS

LANG キーワードの制限 [122](#)

pwxcmd

closeforce コマンド [47](#)

close コマンド [47](#)

PowerExchange リスナ、アクティブなタスクの一覧表示 [48](#)

PowerExchange リスナ、タスクの停止 [48](#)

pwxcmd コマンド

コマンド処理サービスの設定 [33](#)

ユーザーによる実行の許可 [29](#)

PWXLSTNR メンバ

PowerExchange リスナ JCL の開始されたタスク [43](#)

R

RBA

IMS セグメントのキー作成のための生成 [127](#)

RDBMSINSRTDFLT 文

DBMOVER 構成ファイル [70](#), [78](#), [92](#)

RTVSQLSTMT コマンド

トラブルシューティング用にオブジェクトを再作成する SQL 文の作成 [72](#)

S

SECURITY 文

DBMOVER 構成ファイル [29](#)

SESSID 文
DBMOVER 構成ファイル [92](#)
SQL Server バルクデータ移動
概要 [138](#)
構成タスク [140](#)
サポートしているデータタイプ [139](#)
ソースとターゲットへの接続状態 [141](#)
バルクデータの移動 [142](#)
バルクロードユーティリティの使用 [143](#)
START_UP_USER_EXIT 文
DBMOVER 構成ファイル [57, 59](#)
STARTLST
PowerExchange リスナ JCL のバッチ [43](#)
stoptask (pwxcmd) [48](#)
SUBMITTIMEOUT 文
DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [32](#)
SVCNODE 文
DBMOVER 構成ファイル [33](#)

T

TCPAccess
PowerExchange 用の設定 [43](#)
TCP/IPVER 文
DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [34](#)
TCP/IP ポート
PowerExchange Listener 用の指定 [41](#)
TEMPHLQ 文
DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [115](#)
TRACING 文
DBMOVER 構成ファイル [35](#)

V

VSAM インデックスバッファ [168](#)
VSAM 制御間隔のバッファ [168](#)
VSAM バルクデータ移動
DBMOVER 構成文 [168](#)
概要 [166](#)
バルクデータの移動 [169](#)
VSAM 文
DBMOVER 構成ファイル[DBMOVER こうせいふぁいる] [168](#)

Z

z/OS プログラム呼び出しサービス
IDMS ロードライブラリの APF 許可 [117](#)
zIIP Exploitation
DBMOVER 文 [217](#)
zIIP、作業のオフロード [216](#)

あ

暗号コード
Adabas [59](#)
アンロードファイル
IMS アンロードデータセットを使用したターゲットのマテリアライズ [132](#)
ソースとしての IMS アンロードデータセット [130](#)

い

イメージコピー、DB2
ターゲットの実体化に使用 [96, 98](#)

え

エラーアクションファイル
カスタムエラー処理設定 [180](#)
エラーしきい値
フォールトトレランス [179](#)
エラー処理
フォールトトレランス [179](#)

お

オペレーティングシステムの認証
PowerExchange に関する [29](#)

か

書き込みモード
フォールトトレランスを持つ非同期 [177](#)
可変長関連レコードのデータセット (VRRDS)
説明 [166](#)

き

キー順のデータセット (KSDS)
説明 [166](#)
許可
ユーザーによる infacmd pwx コマンドの実行 [29](#)
ユーザーによる pwxcmd コマンドの実行 [29](#)
拒否ファイル
フォールトトレランス [181](#)

く

グループソース
説明 [121](#)
グループターゲット処理 [133, 161, 171](#)

こ

構成
データマップのキャッシュのジョブモード [199](#)
複数ジョブモードで実行するデータマップのキャッシュ [199](#)
コンフィギュレーションファイル、DBMOVER
Adabas バルクデータ移動 [52](#)
DB2 for Linux、UNIX、および Windows 文 [77](#)
DB2 for i5/OS 文 [68](#)
DB2 for z/OS のバルクデータ移動 [84](#)
DB2 LOAD 文 [103](#)
IDMS バルクデータ移動 [114](#)
Oracle バルクデータ移動 [148](#)
PowerExchange リスナ文 [17](#)
VSAM バルクデータ移動 [168](#)

し

シーケンシャルファイルのバルクデータ移動
i5/OS ファイル [155](#)
Linux および UNIX 上の名前付きパイプ [156](#)
Linux および UNIX テキストデータ [155](#)
Linux および UNIX バイナリデータ [155](#)
Linux および UNIX ファイル [155](#)
Windows ファイル [156](#)
z/OS ファイル [156](#)
z/OS ファイルのダウンロード [158](#)
概要 [151](#)
可変長ストリームデータ [154](#)
最大レコードサイズ [155](#)
データマッププロパティ [153](#)
バルクデータの移動 [152](#)
ファイルの概念 [153](#)
ユーザーアクセス方法の例 [159](#)
レコード境界 [154](#), [155](#)
レコード形式 [153](#)
シーケンスフィールド [133](#), [161](#), [171](#)

そ

相対レコードデータセット (RRDS)
説明 [166](#)

た

代替ロギング [42](#)
タスクフロー
バルクデータ移動の実装 [14](#)
単一ジョブモード
データマップのキャッシュ [199](#)

ち

チューニング
バルクデータ移動セッション [194](#)

て

データマップのキャッシュ
概要 [198](#)
単一ジョブモード [199](#)
複数ジョブモード [199](#)
有効化 [198](#)
データ型
DB2 for Linux、UNIX、および Windows [76](#)
SQL Server [139](#)

と

トラブルシューティング
RTVSQSTMT コマンドを使用したソースまたはターゲット環境の
再作成 [72](#)

に

入力順のデータセット (ESDS)
説明 [166](#)

は

パーティション化
Reader パーティション化 [206](#)
SQL オーバーライドを行うパススルー Reader パーティション化の
構成 [208](#)
SQL オーバーライドを行わないパススルー Reader パーティション
化の構成 [207](#)
Writer パーティション化 [209](#)
Writer パーティションに関するルールおよびガイドライン [210](#)
Writer パーティションのセッションプロパティ [211](#)
概要 [205](#)
キー範囲 Reader パーティション化の設定 [209](#)
最初の Reader パーティションのみへのデータの読み取り [208](#)
パーティション化されたセッションのパフォーマンス [215](#)
パーティション化されたバルクデータ移動セッション [215](#)
パススルー Writer パーティション化の設定 [211](#)
バッファ
VSAM インデックスおよび制御間隔 [168](#)
バルクデータ移動セッション
チューニング [194](#)
バルクデータ移動のチューニング
チューニングオプション [193](#)

ひ

非キーセグメント (IMS)
キー作成のための RBA 生成 [127](#)

ふ

フォールトトレランス
エラーアクションファイル [180](#)
エラーしきい値 [179](#)
エラー処理 [179](#)
概要 [178](#)
拒否ファイル [181](#)
フォールトトレランスを持つ非同期
書き込みモード [177](#)
複数ジョブモード
データマップのキャッシュ [199](#)
複数のログファイル
設定 [42](#)
複数レコードデータマップ [133](#), [161](#), [171](#)
複数レコードの書き込み
ルールおよびガイドライン [133](#), [161](#), [172](#)

ま

マルチスレッド処理
ルールおよびガイドライン [202](#)

め

メッセージおよびトレース出力用のログファイル
設定 [42](#)

ゆ

有効化
データマップのキャッシュ [198](#)

る

ルックアップトランスフォーメーション
IMS バルクデータ移動での使用 [129](#)