



Informatica® PowerExchange
10.4.1

Navigator User Guide

Informatica PowerExchange Navigator User Guide

10.4.1

June 2020

© Copyright Informatica LLC 2005, 2021

This software and documentation are provided only under a separate license agreement containing restrictions on use and disclosure. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica LLC.

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation is subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License.

Informatica, the Informatica logo, PowerCenter, and PowerExchange are trademarks or registered trademarks of Informatica LLC in the United States and many jurisdictions throughout the world. A current list of Informatica trademarks is available on the web at <https://www.informatica.com/trademarks.html>. Other company and product names may be trade names or trademarks of their respective owners.

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

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

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

Publication Date: 2021-04-06

Table of Contents

Preface	13
Informatica Resources.	13
Informatica Network.	13
Informatica Knowledge Base.	13
Informatica Documentation.	13
Informatica Product Availability Matrices.	14
Informatica Velocity.	14
Informatica Marketplace.	14
Informatica Global Customer Support.	14
Chapter 1: PowerExchange Navigator Introduction	15
PowerExchange Navigator Overview.	15
Environment Variable Incompatibilities Between PowerExchange and PowerCenter.	16
Starting the PowerExchange Navigator.	17
PowerExchange Navigator Interface.	17
Resource Explorer, Resources Window, Resource Inspector, and Other Windows.	19
Title Bar.	21
Menus and Commands.	21
Toolbar.	27
Shortcut Keys.	28
Status Bar.	29
PowerExchange Resources.	29
Opening a PowerExchange Resource.	30
Searching for a Resource.	30
Closing a PowerExchange Resource.	32
Deleting a PowerExchange Resource.	32
Resource Configuration.	33
Adding, Editing, or Importing a Resource Configuration.	34
Creating a Shared Resource Configuration.	35
Selecting a Resource Configuration as the Current Configuration.	36
Deleting a Resource Configuration.	36
Locations of the DBMOVER Configuration File and PowerExchange License Key File.	37
Data Map Preferences.	43
Defining Data Map Preferences.	43
Defining IDMS Data Map Preferences.	46
Encrypt Password Utility.	46
Generating an Encrypted Password.	47
Chapter 2: Data Maps	49
Data Maps Overview.	50

Data Map Names.	50
Data Sources That Are Supported for Data Maps.	51
Data Files and File-List Processing.	52
Copybooks.	52
Adabas Field Definition Tables and Adabas Keys.	53
Data Map Records and Fields.	53
Data Map Tables and Columns.	55
Arrays.	56
Code Pages.	58
Data Checking.	60
Exported Data Maps.	60
Datacom Data Map Items.	61
IDMS Data Map Items.	61
IMS Data Map Items.	62
User Access Method Programs.	62
Adding a Data Map.	62
Adding a Data Map Record.	65
Adding a Field to a Record.	66
Adding a User-Defined Field to a Record.	66
Adding Items to a Datacom Record.	67
Adding an Element to a Datacom Record.	67
Adding a Key to a Datacom Record.	68
Adding a Key Segment to a Datacom Record.	69
Adding Items to an IDMS Record.	70
Adding a CALC Element to an IDMS Record.	70
Adding an Owner Record and Set to an IDMS Record.	71
Adding Items to an IMS Record.	73
Adding a CCK Field to an IMS Record.	73
Adding Search Fields to an IMS Record.	74
Adding a Data Map Table.	75
Adding a Column to a Table.	76
Viewing Records and Fields.	77
Viewing Tables and Columns.	78
Viewing a Data File.	79
Sorting Data Map Records and Tables.	81
Viewing or Editing Properties for a Data Map or Items in a Data Map.	82
Viewing or Editing Properties for a Data Map.	82
Viewing or Editing Properties for a Data Map Record.	82
Viewing or Editing Properties for a Field in a Record.	83
Viewing or Editing Properties for an Element in a Datacom Record.	83
Viewing or Editing Properties for a Key in a Datacom Record.	84
Viewing or Editing Properties for a Key Segment in a Datacom Record.	84

Viewing or Editing Properties for a CALC Element in an IDMS Record.	85
Viewing or Editing Properties for an Owner Record and Set in an IDMS Record.	85
Viewing or Editing Properties for a CCK Field in an IMS Record.	86
Viewing or Editing Properties for a Search Field in an IMS Record.	86
Viewing or Editing Properties for a Data Map Table.	86
Viewing or Editing Properties for a Column in a Table.	87
Copying, Cutting, and Pasting Items in a Data Map.	87
Editing Logon Information for Accessing Data Maps and Data.	88
Sending a Data Map to a Remote System.	90
Importing Data Maps Sent to a Remote System.	92
Associating Multiple Data Files with a Data Map.	93
PowerExchange File-List Processing.	93
File-List File Syntax Rules.	94
Creating a File-List File and Associating It with a Data Map.	94
ODBC and File-List Processing.	94
File-List File Example for Linux, UNIX, and Windows.	95
File-List File Example for z/OS.	95
Printing a Data Map or Items in a Data Map.	95
Printing a Data Map.	95
Printing Data Map Records.	96
Printing Fields in a Record.	97
Printing Data Map Tables.	97
Printing Columns in a Table.	97
Deleting Data Maps or Items in a Data Map.	98
Deleting Data Maps.	98
Deleting Data Maps Exported to a Remote System.	98
Deleting Data Map Records.	99
Deleting Fields from a Record.	99
Deleting Data Map Tables.	99
Deleting Columns from a Table.	100
Deleting Elements from a Datacom Record.	100
Deleting Keys from a Datacom Record.	100
Deleting Key Segments from a Datacom Record.	100
Deleting CALC Elements from an IDMS Record.	101
Deleting Owner Records and Sets from an IDMS Record.	101
Deleting CCK Fields from an IMS Record.	101
Deleting Search Fields from an IMS Record.	101
Chapter 3: Data Maps for Specific Data Sources.	102
Data Maps for Specific Data Sources Overview.	102
Adabas Data Maps.	103
Adding an Adabas Data Map.	104
Adding a Key to a Table in an Adabas Data Map.	105

Datacom Data Maps.	105
Adding a Datacom Data Map.	106
DB2 Catalog Data Maps.	107
Use Cases for DB2 Data Maps.	107
Adding a DB2 Catalog Data Map.	108
Editing the Record Layout in a DB2 Data Map.	109
DB2 Unload File Data Maps.	109
Adding a DB2 Unload File Data Map.	110
Accessing DB2 Unload Data from Multiple Tables.	111
Adding a DB2 Unload File Data Map - Example.	111
IDMS Data Maps.	112
Adding an IDMS Data Map.	112
Editing IDMS Record Properties.	113
Adding Expressions to an IDMS Record.	114
IMS Data Maps.	115
Prerequisites for Adding an IMS Data Map.	116
Configuring DL/I Batch or BMP Access to IMS Data.	116
Configuring IMS ODBA Access to IMS Data.	117
Configuring the PowerExchange dbmover.cfg File on Windows.	118
Adding an IMS Data Map.	118
Viewing IMS Data.	119
Viewing an IMS Hierarchy.	119
Adding Expressions to an IMS Record.	120
Getting the RBA for an IMS Record.	120
Editing Properties in an IMS Data Map.	121
Sending an IMS Data Map to a Remote System.	121
Using Lookup Transformations to Look up Data in IMS Databases.	121
Writing Data to an IMS Database.	121
Viewing or Editing IMS Options for a Table.	122
Testing an IMS Data Map.	122
IMS Data Map Examples.	123
SEQ Data Maps.	126
Adding an SEQ Data Map.	126
Viewing an SEQ or VSAM Record Hierarchy.	127
VSAM Data Maps.	127
Adding a VSAM Data Map.	128
Viewing an SEQ or VSAM Record Hierarchy.	129
Improving Bulk Read Performance for VSAM Data Sets.	129
Getting the RRN or RBA for VSAM Data Set Records.	131
Chapter 4: Copybooks.	133
Copybooks Overview.	133
Copybook Types Supported in Data Maps.	134

Importing a Copybook into a Data Map.	135
Importing a Copybook into a Record or a Segment.	137
Importing an FDT into an Adabas Data Map.	139
Displaying and Refreshing the FDT for an Adabas Data Map.	139
Import Copybook - Source Details Dialog Box.	140
Import Copybook - Field Property Details Dialog Box.	143
Import Copybook - Local Details Dialog Box.	143
Import Copybook - Local DB2 Catalog Details Dialog Box.	144
Import Copybook - Remote Details Dialog Box.	145
Import Copybook - Remote Datacom Details Dialog Box.	146
Import Copybook - Remote DB2 Catalog Details Dialog Box.	148
DB2 Column Selection Dialog Box.	149
IDMS Advanced Properties Dialog Box.	150
Import Copybook - Configuration Details Dialog Box.	150
Record Definition and Duplicate Record Definition Dialog Boxes.	152
Chapter 5: Registration Groups and Capture Registrations.	153
Registration Group and Capture Registration Overview.	153
Registration Tag Names.	154
Registration Groups.	157
Adding a Registration Group.	158
Adding a Capture Registration.	163
Status Options.	168
Condense Options.	169
Viewing a Capture Registration.	170
Viewing or Editing Properties for a Registration Group.	171
Viewing or Editing Properties for a Capture Registration.	173
Viewing Elements for a Datacom Capture Registration.	178
Viewing Paths for an IDMS Capture Registration.	178
Changing the DB2 Columns in a Capture Registration.	178
Deleting a Capture Registration.	179
Deleting a Registration Group.	180
Deleting All Capture Registrations in a Registration Group.	180
Chapter 6: Extraction Groups and Extraction Maps.	181
Extraction Groups and Extraction Maps Overview.	181
Extraction Map Names.	182
Adding an Extraction Group.	184
Adding an Extraction Map.	186
Viewing an Extraction Map.	188
Adding or Removing Change Indicator and Before Image Columns for a Data Column.	189
Adding Columns to or Removing Columns from Extraction Processing.	190
Changing the Capture Registration Associated with an Extraction Map.	191

Merging a Data Map with an Extraction Map.	192
Viewing or Editing Properties for an Extraction Group.	192
Viewing Properties for an Extraction Map.	195
Creating a Data Map from an Extraction Map.	195
Deleting an Extraction Map.	196
Deleting an Extraction Group.	197
Deleting All Extraction Maps in an Extraction Group.	197
Extract Definition Window - Map Tab.	198
Extract Definition Window - Extensions Tab.	198
PowerExchange-Generated Columns in Extraction Maps.	199
Chapter 7: Personal Metadata.	205
Personal Metadata Overview.	205
Adding a Personal Metadata Profile.	206
Viewing Metadata for Tables in a Personal Metadata Profile.	209
Viewing Properties for a Table in a Personal Metadata Profile.	210
Viewing or Editing Properties for a Personal Metadata Profile.	210
Refreshing Personal Metadata.	211
Deleting a Personal Metadata Profile.	211
Chapter 8: Database Row Test.	212
Database Row Test Overview.	212
Showing or Hiding the Database Row Test Output Window.	213
Database Row Test and Nonstandard Code Page Data.	213
Testing a Data Map.	213
Testing an Extraction Map.	215
Testing a Personal Metadata Profile.	216
Issuing PowerExchange Listener Commands in a Database Row Test.	217
Generating Restart Tokens for Change Data Capture.	218
Database Row Test Dialog Box.	219
CAPX Advanced Parameters Dialog Box - General Tab.	228
CAPX Advanced Parameters Dialog Box - AS400 Tab.	230
CAPXRT Advanced Parameters Dialog Box - General Tab.	230
CAPXRT Advanced Parameters Dialog Box - Adabas Tab.	234
CAPXRT Advanced Parameters Dialog Box - AS400 Tab.	234
CAPXRT Advanced Parameters Dialog Box - DB2UDB Tab.	235
CAPXRT Advanced Parameters Dialog Box - MSSQL Tab.	235
CAPXRT Advanced Parameters Dialog Box - Oracle Tab.	235
DB2390IMG Advanced Parameters Dialog Box.	236
Advanced Parameters Dialog Box (IMS).	237
Chapter 9: PowerExchange Navigator Examples.	239
PowerExchange Navigator Examples Overview.	239

Adding a Single-Record Data Map - Example.	240
Step 1. Add a Data Map.	240
Step 2. Add a Record to the Data Map.	241
Step 3. Add Fields to the Record in the Data Map.	241
Step 4. Add a Table to the Data Map.	243
Step 5. Test the Single-Record Data Map.	243
Adding a Single-Record Data Map with an Array - Example.	244
Step 1. Associate a Data File with a Data Map.	244
Step 2. Filter the Data in the Data Map.	244
Step 3. Create an Output Row for Each Element in an Array.	245
Step 4. Create One Output Row for All Elements in an Array.	246
Adding a Multiple-Record Data Map with a Complex Table - Example.	246
Step 1. Associate a Data File with a Multiple-Record Data Map.	247
Step 2. View Records in a Multiple-Record Data Map.	247
Step 3. Define a Hierarchy of Records in a Multiple-Record Data Map.	247
Step 4. Test a Multiple-Record Data Map.	248
Importing a COBOL Copybook into a Data Map - Example.	248
Step 1. Add a Data Map and Import a COBOL Copybook into the Data Map.	249
Step 2. Apply a Date Mask to a Field in the Data Map Record.	251
Step 3. Use a WHERE Clause to Filter an Empty Record from the Output.	252
Step 4. Display Multiple Output Rows for Items in an OCCURS DEPENDING ON Clause.	252
Importing a COBOL Copybook with REDEFINES Statements - Example.	253
Step 1. Add a Data Map and Import a COBOL Copybook with REDEFINES Statements into the Data Map.	253
Step 2. Apply a Date Mask to a GROUP Field in the Data Map.	255
Importing a PL/I Copybook - Example.	256
Step 1. Add a Data Map and Import a PL/I Copybook into the Data Map.	256
Step 2. Test the Data Map.	258
Defining Table Properties in a Data Map to Normalize an Array - Example	259
Step 1. Add a Data Map that Contains an Array Defined with Multiple Occurrences.	259
Step 2. Add User-Defined Fields that Demonstrate the Normalization of an Array.	261
Step 3. Define Table Properties to Normalize an Array.	261
Step 4. Test the Normalization of the Array in the Data Map.	262
Calling a User Exit Program by Using the PowerExchange CallProg Function - Example	263
Step 1. Add a Data Map.	264
Step 2. Create and Compile the User Exit Program.	267
Step 3. Add User-Defined Fields.	277
Step 4. Refresh Columns in the MASTER_REC Table	280
Step 5. Test the Results of the User Exit Program.	280
Appendix A: PowerExchange Functions for User-Defined Fields.	282
PowerExchange Functions for User-Defined Fields Overview.	282
PowerExchange Functions.	283

CallProg.	283
Check.	284
CheckNumData.	285
Concat.	286
CopyData.	287
Fragment.	287
GenVRowKey.	289
GetCurrentFileName.	289
GetDatabaseKey.	291
GetDataFlowType.	292
GetDbKey.	293
GetDbKeyOfFirstMember.	293
GetDbKeyOfLastMember.	293
GetDbKeyOfNextMember.	294
GetDbKeyOfOwner.	294
GetDbKeyOfPriorMember.	295
GetFullDbKey.	295
GetFullDbKeyOfFirstMember.	296
GetFullDbKeyOfLastMember.	296
GetFullDbKeyOfNextMember.	297
GetFullDbKeyOfOwner.	297
GetFullDbKeyOfPriorMember.	298
GetIMSRBByLevel.	298
GetPageGroup.	299
GetPageGroupOfOwner.	299
GetPgGrpAndRdx.	300
GetPgGrpAndRdxOfOwner.	300
GetPgGrpOfOwner.	300
GetRadix.	301
GetRadixOfOwner.	301
GetSeqWithinLevel.	301
GetSeqWithinParent.	302
LengthOf.	303
LTrim.	303
RTrim.	303
SetBitA.	304
SetBitN.	304
Split.	305
Strip.	306
ToLower.	307
ToUpper.	307
Calling User-Defined Programs with the CallProg Function.	307

Location of User-Defined Programs.	308
Load Errors for User-Defined Programs Called by the CallProg Function.	308
Parameter List Passed to User-Defined Programs.	309
User-Defined Programs Return Codes.	310
User-Defined Programs - Examples.	311
COBOL Program - Example.	312
Assembler Program - Example.	313
RPG Program - Example.	314
PL/I Program - Example.	314
Testing User-Defined Fields in a Data Map.	316
Appendix B: User Access Method Programs.	317
User Access Method Programs Overview.	317
Parameter Structure Passed to the User Access Method Program.	319
Return Codes from the User Access Method Program.	321
User Access Method Programs - Examples.	321
Appendix C: Application Groups and Applications.	322
Application Groups and Applications Overview.	322
Adding an Application Group.	323
Viewing or Editing Properties for an Application Group.	324
Resetting the Extraction Start Point for an Application.	326
Deleting an Application Group.	326
Appendix D: Data Map Properties.	328
Data Map Properties - ADABAS Access Method.	328
Data Map Properties - DATACOM Access Method.	330
Data Map Properties - DB2 Access Method.	330
Data Map Properties - DB2UNLD Access Method.	331
DB2 Table Filter Dialog Box.	332
DB2 Table Filter - Available Tables Dialog Box.	334
Data Map Properties - DL/I Batch or IMS ODBA Access Method.	334
Data Map Properties - ESDS Access Method.	335
Data Map Properties - IDMS Access Method.	336
Data Map Properties - KSDS Access Method.	337
Data Map Properties - MQSeries Access Method.	338
Data Map Properties - RRDS Access Method.	339
Data Map Properties - SEQ Access Method.	340
Data Map Properties - TAPE Access Method.	342
Data Map Properties - USER Access Method.	343
Data Map Properties - Data Defaults.	345
Data Map Properties - Data Checking.	346
Data Map Properties - Record Level Exit.	348

Data Map Properties - Keys.	348
Appendix E: Record, Field, and Table Properties.	349
Record Properties - Name.	349
Record Properties - Name (Datacom).	350
Record Properties - Name (IDMS).	351
Sort Element Details Dialog Box (IDMS).	353
Record Properties - Import Details	353
Field Properties - Name	354
Field Properties - Array.	360
Non-Adabas Fields.	360
Adabas Fields.	361
Field Properties - Extra Properties	362
Field Properties - Field Checking	362
Field Properties - Code Page.	363
Field Properties - Variable Length.	363
Table Properties - Definition.	364
Table Properties - Filter	367
Table Properties - IMS Options	367
Table Properties - Keys	368
Find Dialog Box.	369
Appendix F: DTL__CAPXTIMESTAMP Time Stamps.	370
Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source.	370
Appendix G: Trace for Creating a Memory Map When Importing a COBOL Copybook.	372
Example Trace Output for a COBOL Memory Map.	373
Index.	375

Preface

Use the *Informatica® PowerExchange® Navigator User Guide* to learn how to use the PowerExchange Navigator user interface to define and manage capture registrations, extraction maps, data maps, and personal metadata profiles.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit <https://network.informatica.com>.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit <https://docs.informatica.com>.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at <https://network.informatica.com/community/informatica-network/product-availability-matrices>.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at <http://velocity.informatica.com>. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at <https://marketplace.informatica.com>.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:

<https://www.informatica.com/services-and-training/customer-success-services/contact-us.html>.

To find online support resources on the Informatica Network, visit <https://network.informatica.com> and select the eSupport option.

CHAPTER 1

PowerExchange Navigator Introduction

This chapter includes the following topics:

- [PowerExchange Navigator Overview, 15](#)
- [Environment Variable Incompatibilities Between PowerExchange and PowerCenter, 16](#)
- [Starting the PowerExchange Navigator, 17](#)
- [PowerExchange Navigator Interface, 17](#)
- [PowerExchange Resources, 29](#)
- [Resource Configuration, 33](#)
- [Data Map Preferences, 43](#)
- [Encrypt Password Utility, 46](#)

PowerExchange Navigator Overview

The PowerExchange Navigator is a graphical user interface (GUI) that runs on Windows.

Use the PowerExchange Navigator to define and manage PowerExchange objects, including the PowerExchange resources that are required to perform bulk data movement operations and change data capture (CDC) processing.

Run database row tests to verify that PowerExchange can access data when a bulk data movement operation or change data extraction runs. You can also use the **Database Row Test** dialog box to issue PowerExchange Listener LISTTASK and STOPTASK commands and to generate restart tokens for data sources. Use the restart tokens to populate the PowerExchange Client for PowerCenter (PWXPC) restart token file for a PowerCenter® change data capture (CDC) session.

Use the following PowerExchange resources to complete bulk data movement operations or change data capture (CDC) processing:

- Registration groups and capture registrations. Defines the source data for which you want PowerExchange to capture changes. Required to complete CDC processing.
- Extraction groups and extraction maps. Defines the extraction information for data sources registered for capture. Required to complete CDC processing.

- Data maps. Define data maps to access nonrelational data sources, such as IMS databases, VSAM files, and flat files, and to perform column-level processing for DB2 sources. For nonrelational sources, data maps are required to complete bulk data movement operations and to provide metadata for capture registrations.

In a data map, you can manually define the layout of the data. Alternatively, you can import a copybook or an Adabas FDT, for example, to define the layout of the data.

Other PowerExchange objects that you can define are:

- Application groups and applications. If you use ODBC connections to extract change data, defines the extraction information for data source registered for capture. PowerExchange generates an application group when you define a registration group.

Note: Application groups and applications are not used if you use the PowerExchange Client for PowerCenter (PWXPCL) to extract change data, which is recommended.

- Personal metadata profiles. Define a profile to test connectivity to a data source, view metadata, and preview data for relational and nonrelational data sources and targets.

Environment Variable Incompatibilities Between PowerExchange and PowerCenter

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

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

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

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

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

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

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

Starting the PowerExchange Navigator

Use the Start menu to start the PowerExchange Navigator. Before you start the PowerExchange Navigator, you might need to set the PowerExchange environment.

1. If you need to set the PowerExchange environment, issue the following command:

```
pwxsettask setpwxenv
```

2. From the Start menu, start the PowerExchange Navigator.

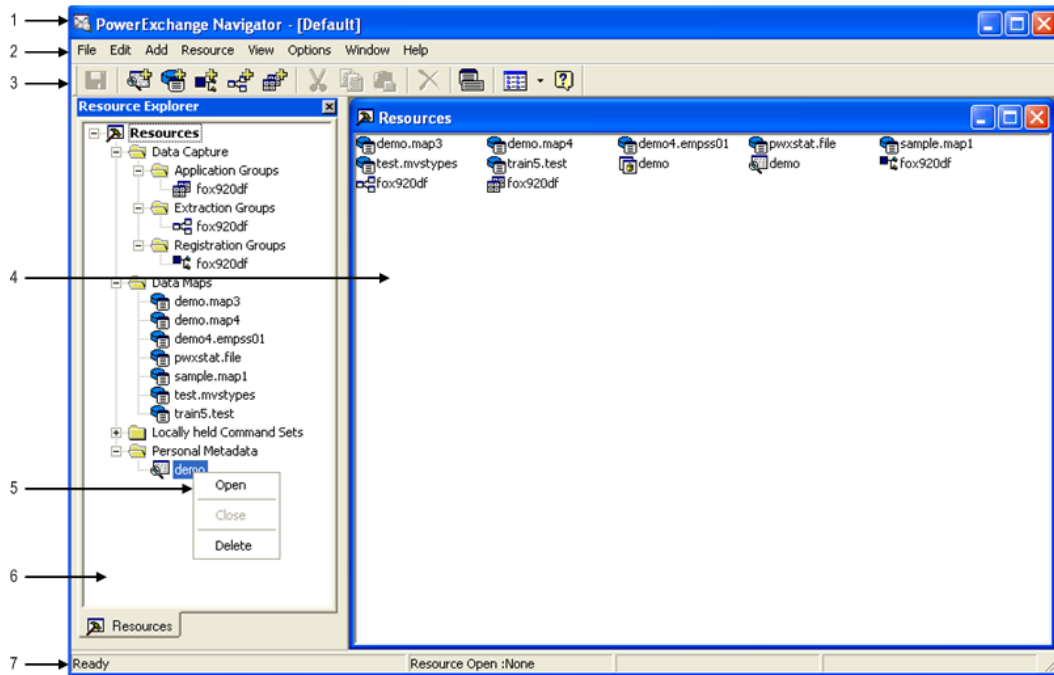
For more information, see [“Environment Variable Incompatibilities Between PowerExchange and PowerCenter” on page 16](#).

PowerExchange Navigator Interface

The PowerExchange Navigator displays PowerExchange resources and resource information in the **Resource Explorer**, **Resources** window, **Resource Inspector**, and, for specific resources, in other windows.

Use the PowerExchange Navigator menu bar, toolbar, and shortcut menus to work with these resources.

The following image shows the **Resources** tab in the **Resource Explorer**, the **Resources** window, and the menus and toolbar that you use to define and manage resources:



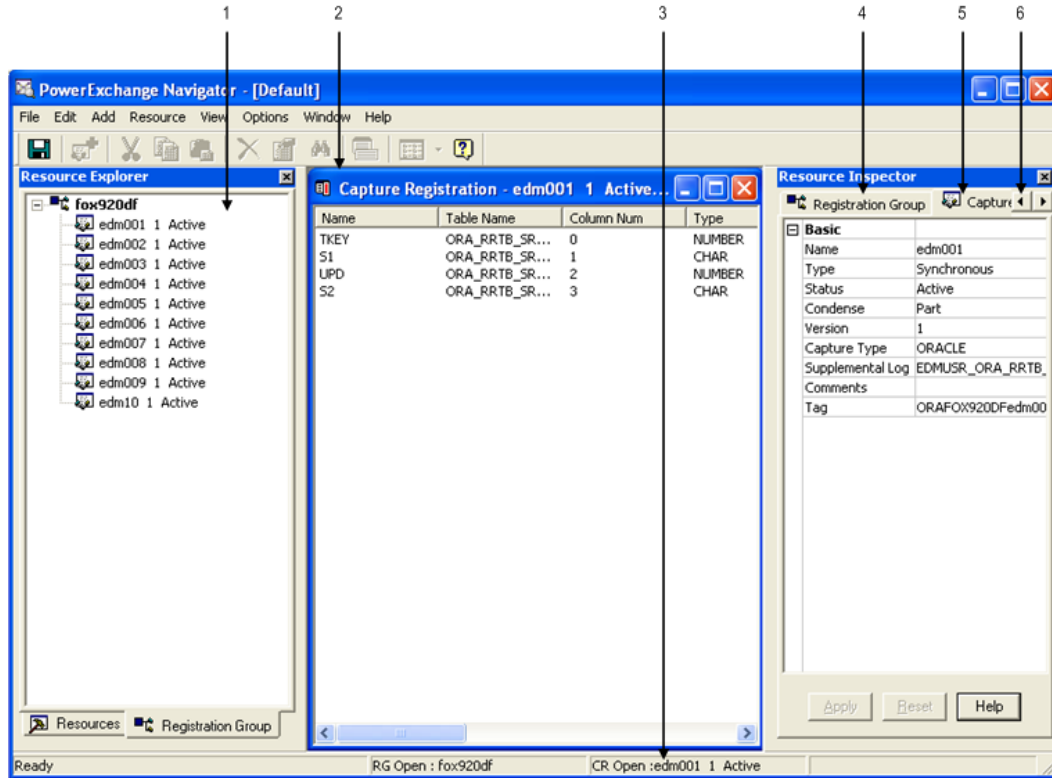
1. Title bar
2. Menu bar
3. Toolbar
4. **Resources** window
5. Shortcut menu
6. **Resources** tab in the **Resource Explorer**
7. Status bar

The PowerExchange Navigator GUI includes the following resource-specific elements:

- The **Registration Group** tab in the **Resource Explorer**.
- The **Capture Registration** window.
- The status bar, which displays information about the open registration group and capture registration.

- The **Registration Group** and **Capture Registration** tabs in the **Resource Inspector**, which display information for an open registration group and capture registration. Use the toggle arrows to toggle between the **Capture Registration** and **Registration Group** tabs in the **Resource Inspector**.

The following image shows these elements:



1. Resource-specific tab in the **Resource Explorer**
2. **Capture Registration** window
3. Status bar
4. **Registration Group** tab in the **Resource Inspector**
5. **Capture Registration** tab in the **Resource Inspector**
6. Toggle arrows

Resource Explorer, Resources Window, Resource Inspector, and Other Windows

The PowerExchange Navigator displays PowerExchange resources and resource information in the **Resource Explorer**, **Resources** window, and **Resource Inspector**, and, for specific resources, in other windows.

The information that appears in the **Resource Explorer** depends on whether a PowerExchange resource is open, and what type of resource is open.

When you start the PowerExchange Navigator and before you open a PowerExchange resource, the following windows appear:

- **Resource Explorer**. Displayed on the left side. On the **Resources** tab, the **Resource Explorer** groups objects in folders in an expandible and collapsible tree structure to help you view PowerExchange resource definitions by category. The folders are:
 - **Data Capture** and its subfolders, **Application Groups**, **Extraction Groups**, and **Registration Groups**

- **Data Maps**
- **Personal Metadata**
- The **Resources** window. Displayed on the right side. Displays the following defined resources in a list or an icon view:
 - Application groups, registration groups, and extraction groups
 - Data maps
 - Personal metadata profiles

You can work with PowerExchange resources from either the **Resource Explorer** or the **Resources** window.

When you open a registration group, extraction group, data map, or personal metadata profile, the **Resource Explorer** displays a new tab that lists the objects in the group, data map, or profile.

Additionally, when you open any of the following PowerExchange resources, the **Resource Inspector** appears on the right side of the interface and displays properties for that resource:

- Application groups and applications
- Registration groups and capture registrations
- Extraction groups and extraction maps

When you open an item in any of these groups, such as a capture registration in a registration group, both the registration and the group remain open at the same time. A window specific to the open item appears, as follows:

- **Application** window. Displays an application.
- **Capture Registration** window. Displays the columns registered by a capture registration.
- **Extract Definition** window. Displays an extraction map.

Additionally, the **Resource Inspector** displays a tab for both the open group and the open item in the group. Use the toggle arrows to toggle between tabs in the **Resource Inspector**. The status bar displays information about the open group and registration.

When you open record in a data map, a **Record** window displays the fields in the record and enables you to add, change, or delete fields in the record.

When you open a table in a data map, a **Table** window displays the columns in the table, and enables you to add, edit, or delete columns in the table.

Showing or Hiding the Resource Explorer or the Resource Inspector

You can show or hide the **Resource Explorer** or the **Resource Inspector** from the **View** menu.

By default, the **Resource Explorer** and the **Resource Inspector** appear.

Changing the View Type for the Resources Window

You can display defined resources in the **Resources** window as large or small icons, or in a list with or without modification details by using the Views toolbar button.

Arranging Windows

You can arrange windows, such as the **Resources** window and the **Capture Registration** window, in an overlapped fashion or side by side from the **Window** menu.

You can also switch to an open window. Arranging windows does not affect the placement of the **Resource Explorer** or **Resource Inspector**.

Title Bar

The PowerExchange Navigator title bar is located across the top of the GUI.

The title bar shows the application name, PowerExchange Navigator, potentially followed by:

- A resource configuration name, in brackets, if you viewed or edited a resource configuration
- A window name, in brackets, if you maximized a window such as the **Resources** window

Menus and Commands

The PowerExchange Navigator provides menus from which you select commands to define and manage PowerExchange resources and complete other tasks.

The following types of menus are available:

- **Menus on the menu bar.** Click a menu name on the menu bar at the top of the **PowerExchange Navigator** window to display a list of commands.
- **Shortcut menus.** Right-click a PowerExchange resource to display a shortcut menu that contains the commands appropriate for that resource.

Menus on the Menu Bar

Each menu on the menu bar displays a list of commands.

The menu bar contains the following menus:

- File
- Edit
- Add
- Resource
- Mode
- View
- Options
- Window
- Help

The commands in each of these menus are described below.

File menu

The following table describes the commands in the **File** menu and lists any available shortcut keys for the commands:

Command	Description	Shortcut Key
Open Resource	Opens a PowerExchange resource.	CTRL+O
Close Resource	Closes a PowerExchange resource.	-
Send to Remote Node	Sends a data map to a remote system.	-
Database Row Test	Opens the Database Row Test dialog box.	-

Command	Description	Shortcut Key
Import Copybook	Imports a copybook into a data map.	-
Exported Data Maps	Displays a list of the exported data maps and enables you to import them.	-
View Data File	Displays the data file associated with a data map.	-
Save	Saves a data map, extraction group, extraction map, registration group, or capture registration.	CTRL+S
Save As	Saves a data map under another name.	-
Print	Prints a data map, record, field, table, or column.	-
Encrypt Password	Encrypts a clear text password.	-
Refresh	Refreshes all locally held resources, including data maps, browser profiles, application groups, extraction groups, and registration groups.	-
Exit	Exits the PowerExchange Navigator.	-

Edit menu

The following table describes the commands in the **Edit** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Cut	Cuts a resource and moves it to the clipboard.	CTRL+X
Copy	Copies a resource to the clipboard.	CTRL+C
Paste	Inserts a copy of the clipboard contents at the insertion point.	CTRL+V
Find	Finds one of the following items: <ul style="list-style-type: none"> - A field in a record in a data map. - A column in a table in a data map. - A table in a personal metadata profile. - A column in a personal metadata table. 	CTRL+F

Add menu

The following table describes the commands in the **Add** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Personal Metadata	Adds a personal metadata profile.	CTRL+T
Data Map	Adds a data map.	CTRL+D
Registration Group	Adds a registration group.	CTRL+G

Command	Description	Shortcut Key
Extraction Group	Adds an extraction group.	CTRL+E
Application Group	Adds an application group.	CTRL+L
Record	Adds a record to the selected data map.	-
Table	Adds a table to the selected data map.	-
Field	Adds a field to the selected record.	-
Column	Adds a column to the selected table.	-
Capture Registration	Adds a capture registration to the selected registration group.	-
Extract Definition	Adds an extraction map to the selected extraction group.	-

Resource menu

The following table describes the commands in the **Resource** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Properties	Displays properties for the selected data map, record, field, table, or column.	CTRL+P
Delete	Deletes a PowerExchange resource.	DELETE
Logons	Enables logon information to be entered to access data maps and data.	-
FDT	For an Adabas data map, displays the imported field definition table (FDT) in the Adabas FDT dialog box. Note: In the Adabas FDT dialog box, click Refresh FDT to refresh the FDT.	-

Mode menu

The following table describes the commands in the **Mode** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Char	Displays records in character format in a data file view.	-
Char/Hex	Displays records in character and hexadecimal format in a data file view.	-
Vertical	Displays records vertically in character and hexadecimal format in a data file view.	-
ASCII	Displays records using the American Standard Code for Information Interchange (ASCII) character encoding in a data file view.	-
EBCDIC	Displays records using the Extended Binary Coded Decimal Interchange Code (EBCDIC) character encoding in a data file view.	-

Note: No shortcut keys are available for these commands.

View menu

The following table describes the commands in the **View** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Toolbar	Shows or hides the toolbar.	-
Status Bar	Shows or hides the status bar.	-
Resource Explorer	Shows or hides the Resource Explorer .	-
Output	Shows or hides the Database Row Test Output window.	-
Resource Inspector	Shows or hides the Resource Inspector .	-

Note: No shortcut keys are available for these commands.

Options menu

The following table describes the commands in the **Options** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Resource Configuration	Enables you to configure directories for PowerExchange resources.	-
Preferences	Enables you to configure preferences for data maps.	-

Note: No shortcut keys are available for these commands.

Window menu

The following table describes the commands in the **Window** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Cascade	Arranges windows in an overlapped fashion.	-
Tile Horizontally	Arranges windows in a non-overlapped fashion.	-
Tile Vertically	Arranges windows side by side.	-
Arrange Icons	Arranges icons of closed windows.	-
Open windows	Displays titles of any open windows. Click a window title to switch to that window.	-

Note: No shortcut keys are available for these commands.

Help menu

The following table describes the commands in the **Help** menu and lists any available shortcut keys:

Command	Description	Shortcut Key
Documentation Contents	Displays the PowerExchange documentation.	-
Help Topics	Displays the Help for the PowerExchange Navigator.	F1
About	Displays version information for the PowerExchange Navigator.	-

Shortcut Menus

A shortcut menu appears when you right-click a folder or resource in the **Resource Explorer**, or right-click an item in a **Record** or **Table** window.

The shortcut menus contain the commands that are appropriate for the selected item.

Shortcut menus are available for the following items:

- Folders on the **Resources** tab in **Resource Explorer**:
 - **Data Capture** and its subfolders, including **Application Groups**, **Extraction Groups**, and **Registration Groups**
 - **Data Maps**
 - **Personal Metadata**
- Resources on the **Resource Explorer** tabs or in a **Record** or **Table** window:
 - Application groups and applications
 - Extraction groups and extraction maps, and the columns in an extraction map
 - Registration groups and capture registrations
 - Data maps, and records, fields, tables, and columns in a data map
 - Personal metadata profiles and tables in personal metadata profiles

Commands are available from the menu bar or a shortcut menu or both. The following tables describe the commands that are available from shortcut menus only. For other commands, see [“Menus on the Menu Bar” on page 21](#).

Application Groups and Applications

The following table describes the commands for application groups and applications that are available from a shortcut menu only:

Right-click	Commands
An open application group	Reset to New Start Point. Reset the extraction start point for an extraction tracked by an application.

Capture Registrations

The following table describes the commands for capture registrations that are available from a shortcut menu only:

Right-click	Commands
An open capture registration	<p>Amend Columns. Add or remove columns from capture registrations for the sources that support selective-column capture.</p> <p>View Elements. For Datacom sources, view elements of a Datacom record such as offset and length.</p> <p>View Paths. For IDMS sources, view the parent hierarchy above the IDMS base record.</p>

Data Maps

The following table describes the commands for data maps that are available from a shortcut menu only:

Right-click	Commands
An open data map	Sort Items. Sort the records and tables in a data map in alphabetical order.
An open IMS data map	Display IMS Hierarchy. Display the IMS hierarchy for an IMS data map.
A field name in a Record window	<p>Add Field Before. Add a field to a record before a selected field.</p> <p>Add Field After. Add a field to a record after a selected field.</p> <p>Add Field as Child. Add a field to a record as a child of a selected field of type GROUP.</p> <p>Add Field at End. Add a field to the end of a record.</p>
A column name in a Table window	<p>Add Column Before. Add a column to a table before a selected column.</p> <p>Add Column After. Add a column to a table after a selected column.</p> <p>Add Column at End. Add a column at the end of a table.</p>
A row on the Expr(n) tab of a Record window	<p>Add Field Before. Add a field and an expression for that field before a selected field.</p> <p>Add Field After. Add a field and an expression for that field after a selected field.</p> <p>Add Field at End. Add a field and an expression for that field after a selected field.</p> <p>Delete Field. Delete a field.</p> <p>Move Field Up. Move a field up on Expr(n) tab in Record window.</p> <p>Move Field Down. Move a field down on Expr(n) tab in Record window.</p> <p>Organize Fields. Move completed fields to the top of the list on Expr(n) tab in Record window.</p> <p>Hide Type Columns. Hide type columns on Expr(n) tab in Record window.</p> <p>Show Type Columns. Show type columns on Expr(n) tab in Record window.</p> <p>Restore Columns. Restore columns on Expr(n) tab in Record window.</p>
Double-click a cell in the Expression column on the Expr(n) tab of a Record window	<p>Insert Unicode control character. Insert Unicode control character into an expression.</p> <p>Right to left Reading order. Enable or disable right-to-left reading order for an expression.</p> <p>Show Unicode control characters. Show or hide Unicode control characters for an expression.</p>

Data Files Associated with a Data Map

The following table describes the commands for data files that are available from a shortcut menu only:

Right-click	Commands
A selected record in a data file view Note: To display a data file view, open a data map for a file source, right-click the data map and select Properties , and then click View File on the Access Method tab.	Show Scale. Show a ruler that marks the number of characters in a record in a data file. Hide Scale. Hide the ruler.

Extraction Groups and Extraction Maps

The following table describes the commands for extraction groups and extraction maps that are available from a shortcut menu only:

Right-click	Commands
An open extraction map	Amend Capture Registrations. Change the capture registration that is associated with the extraction map. Amend Change Indicator/Before Images Extensions. Add or remove the change indicator and before image columns for an extraction map. Create Data Map. Create a data map from an extraction map.
A column in an extraction map	Amend Change Indicator/Before Images Extensions. Add or remove change indicator (CI) and before image (BI) columns for an extraction map. Show Auto Generated Columns. Show the PowerExchange-generated columns in an extraction map.

Personal Metadata

The following table describes the commands for personal metadata profiles and tables that are available from a shortcut menu only:

Right-click	Commands
An open personal metadata profile	Refresh. Refreshes the data that appears in a personal metadata profile.
A table in a personal metadata profile	Explore. Displays metadata for the table in the Table window.

Toolbar

Use the toolbar at the top of the PowerExchange Navigator GUI to quickly initiate common tasks, instead of selecting commands from the menu bar or from shortcut menus.

After you open a PowerExchange resource, the toolbar changes to display the appropriate toolbar buttons for that resource type.

If you point to a toolbar button, its name appears in a pop-up and its description appears in the status bar.

Showing or Hiding the Toolbar

You can show or hide the toolbar.

By default, the toolbar appears.

To show or hide the toolbar:

- ▶ Click **View > Toolbar**.

Shortcut Keys

Use shortcut keys to initiate tasks.

The following table describes shortcut keys:

Shortcut Key	Use
CTRL+L	Add an application group.
CTRL+D	Add a data map.
CTRL+E	Add an extraction group.
CTRL+G	Add a registration group.
CTRL+T	Add a personal metadata profile.
CTRL+F6	Close a Record or Table window. First select a field or column in the window.
ALT+F4	Close an active dialog box or exit the PowerExchange Navigator.
CTRL+C	Copy a resource to the clipboard.
CTRL+X	Cut a resource and moves it to the clipboard.
DELETE	Delete one of the following items: <ul style="list-style-type: none">- PowerExchange resource- Selected field in a Record window, or column in a Table window, and all its cross-references
SHIFT+F1	Display online Help for an item.
CTRL+P	Display properties for the selected data map, record, field, table, or column.
F1	Display the Help for the PowerExchange Navigator.
ESC	Exit from an edit operation, such as entering information in a cell for a user-defined field in a record, without making changes.
CTRL+F	Find one of the following items: <ul style="list-style-type: none">- A field in a record in a data map.- A column in a table in a data map.- A table in a personal metadata profile.- A column in a personal metadata table.
CTRL+V	Insert a copy of the clipboard contents at the insertion point.
INSERT	Insert a field or column above selected field or column in a Record or Table window.
CTRL+Up arrow	Move a user-defined field up on the Expr(n) tab in a Record window.

Shortcut Key	Use
CTRL+Down arrow	Move a user-defined field up on the Expr(n) tab in a Record window.
Arrow keys	Move up or down through fields and user-defined fields in a Record window or columns in a Table window.
CTRL+O	Open a PowerExchange resource.
CTRL+S	Save a data map, extraction group, extraction map, registration group, or capture registration.

Status Bar

The status bar at the bottom of the PowerExchange Navigator GUI displays processing status and information about any open PowerExchange resource.

The status bar displays the following information:

- The processing status or a description of the menu command or toolbar button that you are pointing to
- The type and name of any open PowerExchange resource

If no resource is open, the following text appears:

Resource Open : None

- The node location of an open resource, such as the node of the remote location where a capture registration resides
- For personal metadata, the data source type of the personal metadata profile

Showing or Hiding the Status Bar

You can show or hide the status bar.

By default, the status bar appears.

To show or hide the status bar:

- ▶ Click **View > Status Bar**.

PowerExchange Resources

Use the PowerExchange Navigator to define and manage the PowerExchange resources that are required to complete bulk data movement operations and change data capture (CDC) processing.

These resources include:

- Registration groups and capture registrations
- Extraction groups and extraction maps
- Data maps and the items in data maps including records, tables, and other items
- Application groups and applications

Note: Application groups and applications are not used if you use the PowerExchange Client for PowerCenter (PWXPC) to extract change data, which is recommended.

- Personal metadata profiles

Opening a PowerExchange Resource

To view or edit properties for a PowerExchange resource, you must first open the resource from the **Resource Explorer** or the **Resources** window.

Note: One type of resource can be open at a time with the exception of groups and items in groups. When you open a group, such as a registration group, and you open a capture registration in the group, the registration and the group are open at the same time. For all other types of PowerExchange resources, when you open a resource, PowerExchange closes any other open resource.

You can open data maps, personal metadata profiles, and application, extraction, and registration groups from either the **Resource Explorer** or the **Resources** window. Open applications, extraction maps, and capture registrations from the **Resource Explorer**.

Note: You cannot edit the resource configuration if a PowerExchange resource is open.

To open a PowerExchange resource:

- ▶ In the **Resource Explorer**, right-click the resource and click **Open**.

The **Resource Explorer** displays a tab for the resource. The status bar displays the type and name of the open resource.

If the resource is a data capture resource, the **Resource Inspector** displays the resource properties.

Searching for a Resource

You can use the find feature to enter search criteria to search for a specific resource.

You can search for a table in a personal metadata profile, a field in a data map record, or a column in a data map table, a personal metadata profile table, capture registration, or extraction map.

You can include wildcards for the search. The default wildcards are:

- An asterisk (*) represents one or more matching characters.
- A question mark (?) represents one matching character.

Tip: If a search field contains wildcard characters in its name, precede the wildcard characters in the name with the escape character. By default, the escape character is the tilde (~).

Click the arrow to the right of a search criteria box to insert wildcards or the escape character into the search criteria.

Click **Advanced** to change the wildcard characters or the escape character.

Searching for a Field in a Data Map Record

You can search for a field in a record in a data map.

1. Open the data map and the record.
2. Click anywhere in the **Record** window.
3. Click **Edit > Find**.
4. In the **Find field in record** dialog box, enter search criteria.
5. Click **Find Next**.

The PowerExchange Navigator finds the first field in the record that matches the search criteria.

Searching for a Column in a Data Map Table

You can search for a column in a table in a data map.

1. Open the data map and the table.
2. Click anywhere in the **Table** window.
3. Click **Edit > Find**.
4. In the **Find column in table** dialog box, enter search criteria.
5. Click **Find Next**.

The PowerExchange Navigator finds the first column in the table that matches the search criteria.

Searching for a Column in a Capture Registration

You can search for a column in a capture registration.

1. Open the registration group and the capture registration.
2. Click anywhere in the **Table** window.
3. Click **Edit > Find**.
4. In the **Find Column in Capture Registration** dialog box, enter search criteria.
5. Click **Find Next**.

The PowerExchange Navigator finds the first column in the capture registration that matches the search criteria.

Searching for a Column in an Extraction Map

You can search for a column in an extraction map.

1. Open the extraction group and the extraction map.
2. Click anywhere in the **Table** window.
3. Click **Edit > Find**.
4. In the **Find name in Extract Definition** dialog box, enter search criteria.
5. Click **Find Next**.

The PowerExchange Navigator finds the first column in the extraction map that matches the search criteria.

Searching for a Table in a Personal Metadata Profile

You can search for a table in a personal metadata profile.

1. Open the personal metadata profile.
2. Click **Edit > Find**.
3. In the **Find table** dialog box, enter search criteria.
4. Click **Find Next**.

The PowerExchange Navigator finds the first table in the profile that matches the search criteria.

Searching for a Column in a Table in a Personal Metadata Profile

You can search for a column in a table in a personal metadata profile.

1. Open the personal metadata profile.
2. Open the table in which you want to find a column.
3. Click anywhere in the **Table** window.
4. Click **Edit > Find**.
5. In the **Find column in table** dialog box, enter search criteria.
6. Click **Find Next**.

The PowerExchange Navigator finds the first column in the table that matches the search criteria.

Closing a PowerExchange Resource

To delete a resource or to view or edit the resource configuration, you must close any open PowerExchange resource.

Note: If a PowerExchange resource is open when you open another resource, PowerExchange closes the first resource unless the first resource is a data capture group and the second resource is an item in the group. For example, if you open a registration group and you open a capture registration, both resources remain open at the same time.

To close a PowerExchange resource:

- ▶ In the **Resource Explorer**, right-click the resource and click **Close**.

Deleting a PowerExchange Resource

You can delete PowerExchange resources and items from data maps.

You can delete the following PowerExchange resources:

- Capture registrations
- Extraction maps
- Application groups, extraction groups, and registration groups
- Data maps
- Exported data maps
- Personal metadata profiles

You can also delete the following items from a data map:

- Records
- Fields from a record
- Tables
- Columns from a table

For data maps for the following data sources, you can also delete the following items from records:

- Datacom. Elements, keys, and key segments.
- IDMS. CALC elements, and owner records and sets.
- IMS. CCK fields and search fields.

Before you can delete a data map, personal metadata profile, capture registration, extraction map, or a group, you must close any open resource.

Deleting certain types of resources might affect other resources. For example, when you delete a field or a record, PowerExchange deletes any references to the field or record. When you delete a capture registration, you can no longer extract changes by using any extraction maps associated with the capture registration.

Resource Configuration

A PowerExchange Navigator resource configuration defines the location of data maps, personal metadata profiles, and any locally stored capture registrations and extraction maps. Optionally, a resource configuration also defines the locations of the DBMOVER configuration file and PowerExchange license key file.

PowerExchange maintains a set of user-defined resource configurations in the Windows registry for each PowerExchange version that is installed on the machine.

If you are starting PowerExchange Navigator for the first time after installing or upgrading PowerExchange, the **Resource Configuration** dialog box appears. Click **OK** to create a resource configuration named **Default** that stores PowerExchange resources in the following default directory:

installation_directory/Examples

You can add or edit a resource configuration. If you define multiple resource configurations, select a specific resource configuration as the current configuration. You can delete any configuration except the current configuration. You can also import resource configurations from a previous PowerExchange version.

Caution: Do not import resource configurations from a later PowerExchange version into an earlier version. Resource files that were created or updated with the later version might not work with the earlier version.

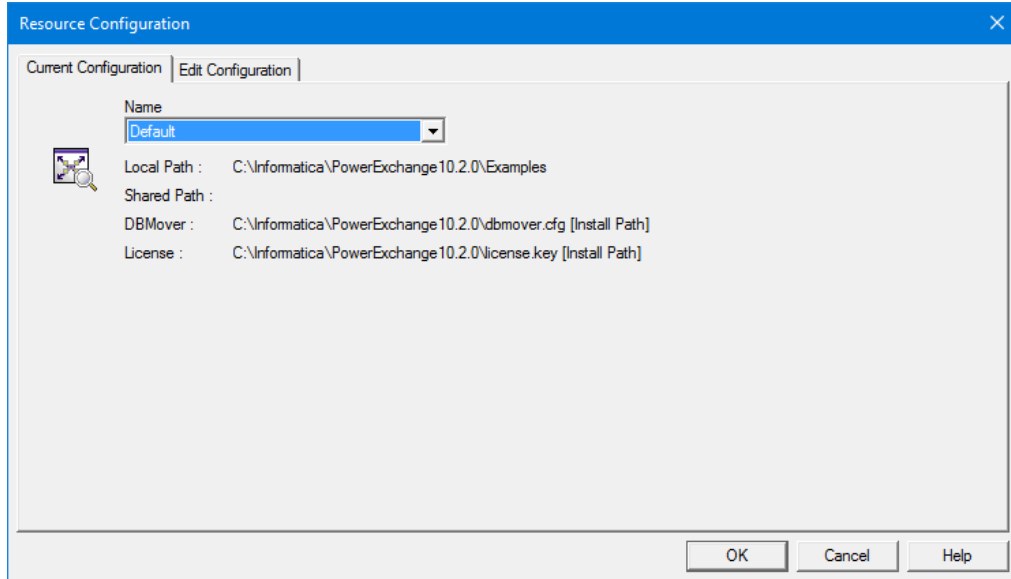
Adding, Editing, or Importing a Resource Configuration

You can add or edit a resource configuration in the PowerExchange Navigator. You can also import resource configurations from an earlier PowerExchange version.

If you previously installed the PowerExchange Navigator and registry entries exist for that installation, complete the following steps to add, edit, or import a resource configuration:

1. Click **Options > Resource Configuration**.

The **Resource Configuration** dialog box appears.



2. Click the **Edit Configuration** tab.
3. Complete one of the following actions:
 - To add a resource configuration, click **Add**.
The **Add Configuration** dialog box appears.
 - To edit a resource configuration, select a configuration in the **Available Configurations** list and click **Edit**.
The **Edit Configuration** dialog box appears.
 - To import resource configurations from a different PowerExchange version, click **Import**.
The **Import Configuration** dialog box appears.
4. If you selected to add or edit a resource configuration, enter information about the resource configuration in the **Edit Configuration** dialog box or **Add Configuration** dialog box.

The following table describes the entry fields:

Field	Description
Name	A user-defined name for the resource configuration.
Local Path	For a local resource configuration, enter the directory location for data maps and other PowerExchange resources, or click the Browse button to browse to the directory.

Field	Description
Shared Path	For a shared resource configuration, enter the directory location of shared data maps, or click the Browse button to browse to the directory. The Shared Path value must be different from the Local Path value. Because the double backslash (\\) is not a valid string in a PowerExchange path name, you must first map a network drive to the shared path.
DBMover	Optionally, enter the path and file name of the DBMOVER configuration file. For more information, see " Locations of the DBMOVER Configuration File and PowerExchange License Key File " on page 37.
License	Optionally, enter the path and file name of the PowerExchange license key file. For more information, see " Locations of the DBMOVER Configuration File and PowerExchange License Key File " on page 37.
Set as current configuration	In the Add Configuration dialog box, sets the resource configuration as the current configuration. By default, this option is selected.

- If you selected to import resource configurations, select the resource configuration or configurations that you want to import in the **Import Configuration** dialog box.

The following table describes the entry fields:

Field	Description
Version	Select the PowerExchange version from which you want to copy resource configurations to the current PowerExchange version.
Configurations	Select the configurations that you want to copy.
Replace available configuration list with the selected items	Select this option to replace the list of available resource configurations to import with the selected items.

- Click **OK**.
- Click **OK**.

The title bar of the GUI displays the active resource configuration. For example, if you edit the Default resource configuration, the title bar displays the application name and resource configuration, as follows:

Informatica PowerExchange Navigator - [Default]

Creating a Shared Resource Configuration

You can store data maps in a shared directory. Multiple users can access shared data maps, but a data map is protected from concurrent updates.

The first user to access a shared data map has full access. Subsequent concurrent users cannot complete the following actions:

- Rename, edit, or delete the data map.
- Import an exported data map or send the data map to a remote node.
- Test the data map.

If you try to open a shared data map that is already opened by another user, a warning message appears.

1. Right-click **This PC**, and click **Map Network Drive**.
2. In the **Map Network Drive** dialog box, enter the following information:
 - In the **Drive** list, select an available drive letter for the connection.
 - In the **Folder** box, enter or browse to the directory that contains the shared data maps.
 - Select **Reconnect at sign-in**.
3. Click **Finish**.
4. In the PowerExchange Navigator, click **Options > Resource Configuration**.
5. In the **Resource Configuration** dialog box, click the **Edit Configuration** tab.
6. Perform one of the following actions:
 - To add a resource configuration, click **Add**.
 - To edit a resource configuration, select a configuration in the **Available Configurations** list and click **Edit**.
7. In the **Add Configuration** or **Edit Configuration** dialog box, enter the following information:
 - In the **Shared Path** field, browse to the network drive that you mapped for the shared resources. If you specify both a shared path and a local path, verify that these paths differ. Otherwise, an error message appears.
 - For a new resource configuration, select the **Set as current configuration** option.
8. Click **OK**.

The PowerExchange Navigator title bar displays the name of the resource configuration and indicates that it is shared. For example:

```
Informatica PowerExchange Navigator - [Default (Shared)]
```

Selecting a Resource Configuration as the Current Configuration

If you define multiple resource configurations, select one to use as the current configuration.

The current configuration remains in effect until you select another configuration.

By default, when you add a resource configuration, it becomes the current configuration.

1. Click **Options > Resource Configuration**.
2. On the **Resource Configuration** dialog box, click the **Current Configuration** tab.
3. In the **Name** list, select the resource configuration that you want to use as the current resource configuration.
4. Click **OK**.

The PowerExchange Navigator title bar displays the active resource configuration.

Deleting a Resource Configuration

You can delete a resource configuration other than the current resource configuration.

1. Click **Options > Resource Configuration**.
2. In the **Resource Configuration** dialog box, click the **Edit Configuration** tab.
3. In the **Available Configurations** list, select a configuration and click **Delete**.

Note: The **Delete** button is unavailable for the current resource configuration.

Locations of the DBMOVER Configuration File and PowerExchange License Key File

By default, the PowerExchange Navigator uses the DBMOVER configuration file and PowerExchange license key file in the installation directory. You can define different paths and file names by using environment variables, the command line, or the **Resource Configuration** dialog box of the PowerExchange Navigator.

The **Current Configuration** tab of the **Resource Configuration** dialog box displays the paths and file names of the DBMOVER configuration file and PowerExchange license key file. After each path and file name, a string in brackets indicates the source that the Navigator uses to determine the path and file name.

The following table identifies the sources that the Navigator uses to find the DBMOVER configuration file and PowerExchange license key file, in increasing order of precedence:

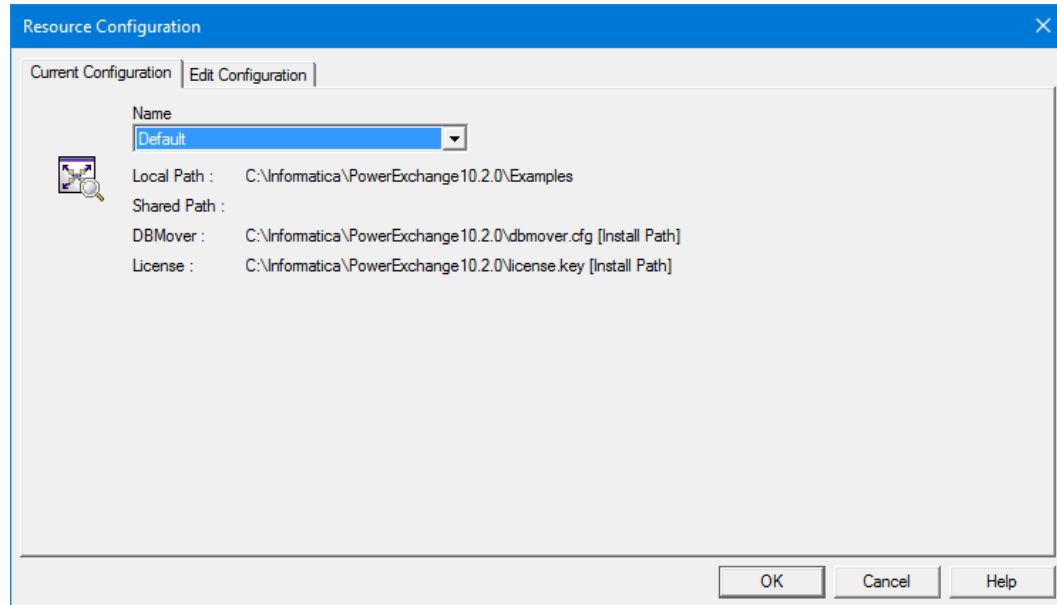
Source	Designation in GUI for DBMOVER Source	Designation in GUI for License Key Source	Description
System defaults	[Install Path]	[Install Path]	If you specify no override, the Navigator loads the dbmover.cfg and license.key files that are located in the PowerExchange root installation directory. This is the path that contains the Navigator executable file, dtlui.exe.
Environment variables	[\$PWX_CONFIG]	[\$PWX_LICENSE]	You can define the following environment variables before you start the Navigator: <ul style="list-style-type: none"> - PWX_CONFIG. Enter the full path to the dbmover.cfg. - PWX_LICENSE. Enter the full path to the license.key.
Command line arguments	[Command Line]	[Command Line]	When you start the Navigator from the command line or from a Windows shortcut, you can specify the following arguments: <ul style="list-style-type: none"> - config. Enter the full path to the DBMOVER configuration file. - license. Enter the full path to the license key file.
PowerExchange Navigator overrides	[NAVIGATOR]	[NAVIGATOR]	You can specify Navigator overrides in the Resource Configuration dialog box for a particular resource configuration name.

Default Locations

If you specify no overrides for the locations of the DBMOVER configuration file and PowerExchange license key file, the PowerExchange Navigator loads the DBMOVER file and license.key files that are in the PowerExchange root installation directory.

The **Current Configuration** tab of the **Resource Configuration** dialog box displays the paths and file names of the DBMOVER configuration file and PowerExchange license key file. When the Navigator loads these files from the installation directory, after each path and file name the following string appears: **[Install Path]**.

The following image shows the **Resource Configuration** dialog box when the Navigator loads the files from the default location:



Environment Variable Overrides

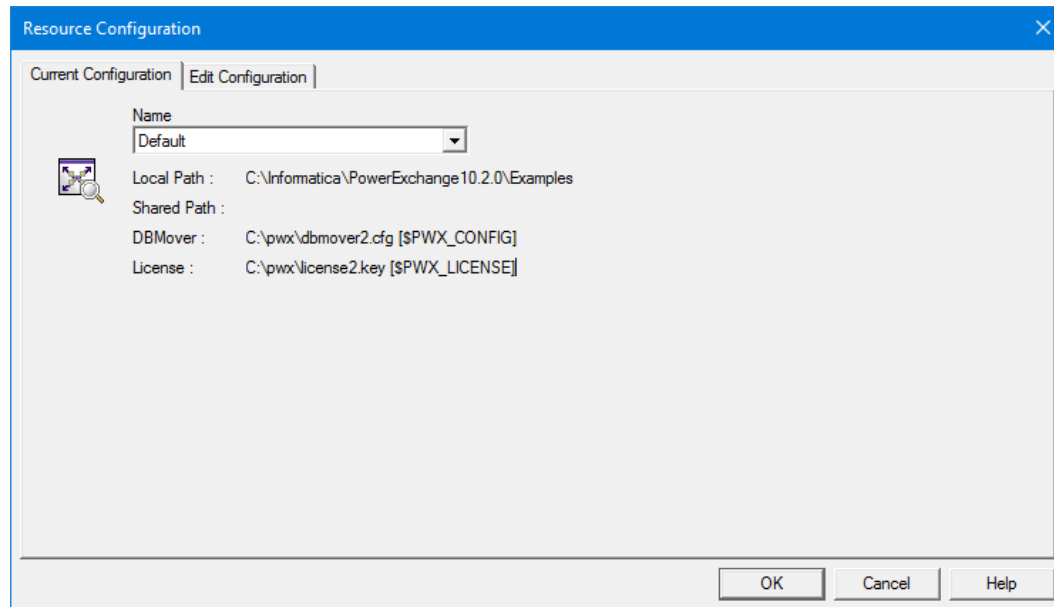
Before you start the PowerExchange Navigator, you can define environment variables to specify the locations of the DBMOVER configuration file and PowerExchange license key file.

You can define the following environment variables:

- **PWX_CONFIG**. Enter the full path to the DBMOVER configuration file. For example:
`PWX_CONFIG=C:\pwx\dbmover2.cfg`
- **PWX_LICENSE**. Enter the full path to the PowerExchange license key file. For example:
`PWX_LICENSE=C:\pwx\license2.key`

The **Current Configuration** tab of the **Resource Configuration** dialog box displays the paths and file names of the DBMOVER configuration file and PowerExchange license key file. When the Navigator uses the environment variable overrides, the following strings appear after each of these paths, respectively: **[\$PWX_CONFIG]** and **[\$PWX_LICENSE]**.

The following image shows the **Current Configuration** tab of the **Resource Configuration** dialog box when the PowerExchange Navigator uses environment variable overrides:



Command Line Overrides

To specify the location of the DBMOVER configuration file and the PowerExchange license key file from the command line, use the config and license command line arguments.

Enter the config and license arguments at the command line as follows:

- config. Enter the full path to the DBMOVER configuration file. For example:

```
dtlui.exe config=C:\pwx\dbmover1.cfg
```

- license. Enter the full path to the license key file. For example:

```
dtlui.exe license=C:\pwx\license1.key
```

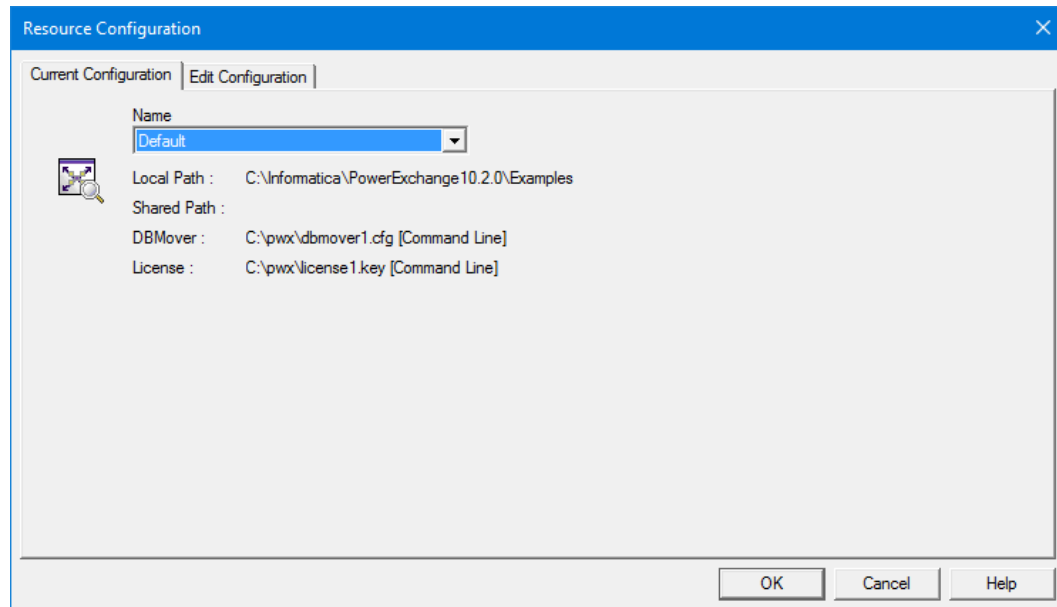
You can specify both arguments together. For example:

```
dtlui.exe config=C:\pwx\dbmover1.cfg license=C:\pwx\license1.key
```

You can also include the config and license arguments in the **Target** property of the Windows shortcut tab for starting the PowerExchange Navigator. For more information, see ["Defining Command Line Overrides in the Target Property" on page 40](#).

The **Current Configuration** tab of the **Resource Configuration** dialog box displays the paths and file names of the DBMOVER configuration file and PowerExchange license key file. When the Navigator uses the command line overrides, after each path and file name the following string appears: **[Command Line]**.

The following image shows the **Current Configuration** tab of the **Resource Configuration** dialog box when the Navigator uses command line overrides:

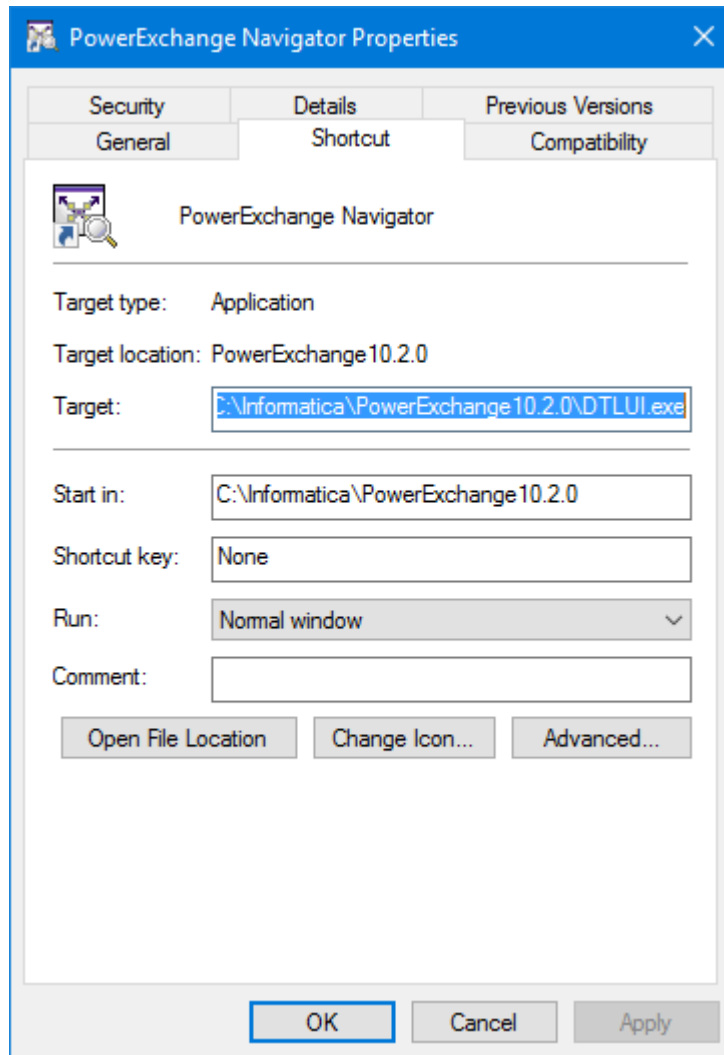


Defining Command Line Overrides in the Target Property

You can include the config and license arguments in the **Target** property of the Windows shortcut tab for starting the PowerExchange Navigator.

1. To open the **PowerExchange Navigator Properties** dialog box, perform one of the following actions:
 - On Windows 10, right-click the **PowerExchange Navigator** shortcut in the **Start** menu, and select **More > Open file location**. Then right-click **PowerExchange Navigator** and select **Properties**.
 - On Windows versions earlier than 10, right-click the **PowerExchange Navigator** shortcut in the Start menu and select **Properties**.

The **PowerExchange Navigator Properties** dialog box appears:



2. Add the config or license argument to the end of the command line that appears in the **Target** field of the dialog box.
3. Click **OK**.

PowerExchange Navigator Overrides

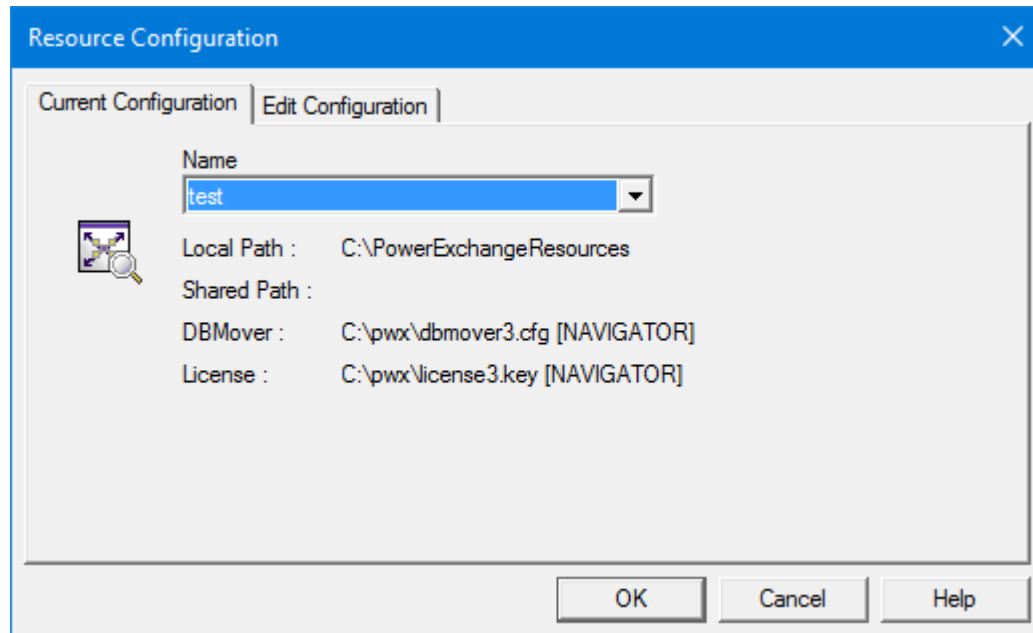
When you add or edit a resource configuration, you can specify the locations of the DBMOVER configuration file and PowerExchange license key file.

In the **Add Configuration** or **Edit Configuration** dialog box, enter the following information:

- At **DBMover**, enter the path to the DBMOVER configuration file.
- At **License**, enter the path to the PowerExchange license key file.

The **Current Configuration** tab of the **Resource Configuration** dialog box displays the paths and file names of the DBMOVER configuration file and PowerExchange license key file. When the Navigator uses Navigator overrides, after each path and file name the following string appears: **[NAVIGATOR]**.

The following image shows the **Current Configuration** tab of the **Resource Configuration** dialog box when the Navigator uses Navigator overrides:



RELATED TOPICS:

- [“Adding, Editing, or Importing a Resource Configuration” on page 34](#)

Error Handling for the DBMOVER Configuration File and PowerExchange License Key File

The PowerExchange Navigator might encounter an error during startup that is caused by an invalid DBMOVER configuration file or PowerExchange license key file.

When this situation occurs, the Navigator displays error messages, and the **Resource Configuration** dialog box appears. You can enter Navigator overrides in this dialog box to override the paths for the DBMOVER configuration file or PowerExchange license key file.

Example: DBMOVER configuration file is not present

If the PowerExchange Navigator does not find the DBMOVER configuration file in its expected location, the Navigator displays error messages such as the following ones:

```
PWX-00268 DBAPI Error. DB_OPEN failed for file C:\pwx\dbmover.cfg.  
PWX-07406 Permanent error set by Open call on file "C:\pwx\dbmover.cfg" because API  
wfoopen() failed. Mode rb.
```

Example: DBMOVER configuration file contains errors

If the DBMOVER configuration file contains errors, the PowerExchange Navigator might display error messages such as the following ones:

```
PWX-00961 Input error. File "##DTLCFG". Statement 1. Line 19.  
PWX-00968 Error "NODEa=" sublist limit exceeded. Position 1.
```

Example: PowerExchange license key file is not present

If the PowerExchange Navigator does not find the license key file in its expected location, the Navigator displays error messages such as the following ones:

```
PWX-00113 KEY Error - failed to open license key file.  
PWX-00268 DBAPI Error. DB_OPEN failed for file C:\pwx\license.key.  
PWX-23532File name was converted from ANSI to Unicode.  
PWX-00150DBAPI SEQ Filename "C:\pwx\license.key" Open error reason 2 <No such file  
or directory>.  
PWX-07406 Permanent error set by Open call on file "C:\pwx\license.key" because API  
w fopen() failed. Mode rb.
```

Example: PowerExchange license key has expired

If the PowerExchange license key has expired, the Navigator displays error messages such as the following ones:

```
PWX-00115 KEY Error - license has expired. Key=2016-01-31, current=2017-03-02.  
PWX-00115 KEY Error - license has expired. Key=2016-01-31, current=2017-03-02.
```

Example: PowerExchange license key file is illegal

If the PowerExchange license key is illegal, the Navigator displays error messages such as the following ones:

```
PWX-00121 KEY Error - Not licensed for navigator feature.  
PWX-00111KEY Error - key length must be 44 or 64 license_key, len=63
```

Data Map Preferences

You can define preferences for creating and testing data maps.

You can define the following preferences for all types of data maps, except where noted:

- Whether to invoke the **Import Copybook** wizard when you add a data map
- Whether to invoke the **Adabas FDT Import** wizard when you add an Adabas data map
- The naming convention used in the SQL that is generated for a table when you run a database row test on a data map
- A default insertion point for adding a field to a record or a column to a table
- Display settings for data in the **Record** window, **Table** window, and **Database Row Test Output** window

For IDMS data maps, you can define Distributed Database System (DDS) dictionary information.

Defining Data Map Preferences

You can define preferences for adding and testing all types of data maps.

1. Click **Options > Preferences**.
2. In the **Preferences** dialog box, click the **Data Map** tab.

The following table describes the options on this tab:

Option	Description
Import Record Definitions on a new Data Map	Select this option to run the Import Copybook wizard when you add a data map. Clear this option to not run the Import Copybook wizard. Alternatively, clear the Import Record Definitions option in the Name dialog box in the Add Data Map wizard. Default is selected.
Import Key Fields/FDT (Adabas Only)	Select this option to run the Adabas FDT Import wizard when you add an Adabas data map. Clear this option to not run the Adabas FDT Import wizard. Alternatively, clear the Import Key Fields/FDT option on the Name dialog box in the Add Data Map wizard. Default is selected.
Use 2-tier names	Indicates the naming convention to use for identifying a table in the SQL that PowerExchange generates when you run a database row test for a data map. Select this option to use a two-tier naming convention. A two-tier name contains a schema name and map name with an appended table name, in the following format: <code>select * from schema.map_table</code> Clear this option to use a three-tier naming convention. A three-tier name contains a schema name, map name, and table name, in the following format: <code>select * from schema.map.table</code> Note: This option does not affect the naming conventions that the NRDB and NRDB2 data source types use: <ul style="list-style-type: none"> - NRDB uses the three-tier naming convention. - NRDB2 uses the two-tier naming convention.
Network Operation Timeout	Enter the number of seconds that must elapse during a network send or receive operation before PowerExchange ends the connection and issues a timeout error message. Default is 180.
Connection Pool Size	If you use connection pooling, enter the maximum number of connections to PowerExchange Listeners to keep open. The PowerExchange Navigator uses up to this number of open connections for faster processing. If you enter a value greater than zero, verify that the value of MAXTASKS in the DBMOVER file on the PowerExchange Listener machine is large enough to accommodate the maximum number of connections in the pool for the Listener task.
Retain	If you use connection pooling, enter the number of seconds to retain an inactive connection in the connection pool before closing it.
Insertion Point	Select one of the following options to specify the insertion point for adding a field to a record or adding a column to a table: <ul style="list-style-type: none"> - Before current item - After current item - At End The specified insertion point is used when you complete one of the following actions: <ul style="list-style-type: none"> - To add a field to a record, click Add > Field. - To add a column to a table, click Add > Column.

Option	Description
Show Field Offsets	Select one or both of the following options to indicate the formats in which to display the offset for records in the Record window: <ul style="list-style-type: none"> - Decimal. Decimal format. - Hex. Hexadecimal format.
Show Column SQL Type	Select this option to show the SQL datatypes of the columns that appear for a table in the Table window. Clear this option to hide the column datatypes.
Filter Column Names	Select an option to filter columns in data maps based on column name. Options are: <ul style="list-style-type: none"> - Include. Include columns that match the name pattern that you specify. - Exclude. Exclude columns that match the name pattern. By default, Exclude is selected.
Column Name	Enter a column name to use as filter criteria. You can use the following wildcard characters: <ul style="list-style-type: none"> - Asterisk (*) to represent one or more matching characters. - Question mark (?) to represent a single matching character. By default, PowerExchange excludes all columns with a name that begins with FILLER.
Number of Records	Enter the maximum number of records that can be displayed at a time in the Data File window when you view a data file that is associated with a data map. Valid values are 1 to 99999. Default is 10. A very large value can increase PowerExchange Navigator response time and CPU usage. Note: If a data file is open in a Data File window, close the data file and associated data map before setting this option. Otherwise, the new option setting does not take effect until the next time you open the window.
Display Format	Select the display format for records in the data file view. Options are: <ul style="list-style-type: none"> - Char/Hex. Display records in character and hexadecimal format. - Char. Display records in character format. - Vertical. Display records vertically in character and hexadecimal format. Default is Vertical . Note: If a data file is open in a Data File window, close the data file and associated data map before setting this option. Otherwise, the new option setting does not take effect until the next time you open the window.
Row Test Connection Timeout (Minutes)	Enter the number of minutes after which a connection to a data source for an active database row test times out. An active database row test is one that is open in the Database Row Test dialog box and that has not reached the end-of-file (EOF). When the row test is active, the bottom right corner of the PowerExchange Navigator window displays the time remaining before the timeout is reached. After the timeout is reached, the PowerExchange Navigator disconnects from the data source. Valid values are 0 to 1440 (24 hours). If you specify 0, no row test timeout is in effect. Default is 15 minutes.

3. Click **OK**.

RELATED TOPICS:

- [“Viewing a Data File” on page 79](#)
- [“Importing a Copybook into a Data Map” on page 135](#)
- [“Importing an FDT into an Adabas Data Map” on page 139](#)

Defining IDMS Data Map Preferences

For IDMS data maps, you can define Distributed Database System (DDS) dictionary information.

This information is used for all IDMS data maps, but you can override these values for individual IDMS data maps.

1. Click **Options > Preferences**.
2. Click the **IDMS Properties** tab.

The following table describes the information to enter:

Option	Description
Dictionary Name	DDS dictionary name from which to retrieve subschema metadata.
DBName	IDMS database that contains the dictionary.
Dictionary UserID	If required, a user ID that has authority to access the specified dictionary.
Dictionary Password	If a dictionary user ID is specified, enter a password for that user ID.
Node Name	DDS node name.
Dictionary Node	DDS dictionary node name.

3. Click **OK**.

RELATED TOPICS:

- [“IDMS Data Maps” on page 112](#)

Encrypt Password Utility

To improve security, you can use the encrypt password utility to encrypt a clear text password or a valid PowerExchange passphrase. PowerExchange supports passphrases for access to i5/OS and z/OS only.

You can use the encrypted password or passphrase to control access to a source, execute `pwxcmd` commands, run some PowerExchange utilities, or configure the PowerExchange Logger for Linux, UNIX, and Window. Copy and paste the encrypted password or passphrase into the relevant command parameter, utility control statement, or configuration file.

The following table describes the parameters for which an encrypted password or passphrase can be specified:

Location	Parameter or Field	Documentation Reference
pwxcmd commands	-epassword	<i>PowerExchange Command Reference</i>
DBMOVER configuration file	EPWD in the UDB CAPI_CONNECTION statement <i>search_user_password</i> in the LDAP_BIND_EPWD statement EPASS in the LDAP_TLS statement EPASS in the SSL statement	<i>PowerExchange Reference Manual</i>
pwxcl.cfg configuration file for the PowerExchange Logger	CAPTURE_NODE_EPWD	<i>PowerExchange CDC Guide for Linux, UNIX, and Windows</i>
DTLREXE utility	-epwd	<i>PowerExchange Utilities Guide</i>
DTLUAPPL utility	EPWD	<i>PowerExchange Utilities Guide</i>
DTLUCBRG utility	EPWD	<i>PowerExchange Utilities Guide</i>
DTLUCUDB utility	EPWD	<i>PowerExchange Utilities Guide</i>
DTLURDMO utility	EPWD	<i>PowerExchange Utilities Guide</i>
DTLUTSK utility	EPWD	<i>PowerExchange Utilities Guide</i>

Generating an Encrypted Password

You can specify an encrypted password in some PowerExchange configuration files, pwxcmd commands, and utility statements. For access to z/OS, you can encrypt a valid PowerExchange passphrase.

1. Click **File > Encrypt Password**.

The **Generate Encrypted Password** dialog box appears.

2. In the **Password** box, type a clear text password. Alternatively, for access to i5/OS or z/OS, enter a valid PowerExchange passphrase.

A passphrase for i5/OS access can be from 9 to 31 characters in length. A passphrase for z/OS access can be from 9 to 128 characters in length (non-ODBC) or from 9 to 79 characters in length (ODBC). A passphrase can contain the following characters:

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

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

The first character is an apostrophe.

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

3. Click **Generate Encrypted Password**.

The encrypted password or passphrase appears in the **Encrypted Password** box.

4. Copy and paste the encrypted password or passphrase into the appropriate parameter in a configuration file, utility statement, or command option.

For example, the following encrypted password is specified in the CAPTURE_NODE_EPWD parameter in the pwxcl.cfg configuration file:

```
CAPTURE_NODE_EPWD=F40B3B2742FB3A1F
```


CHAPTER 2

Data Maps

This chapter includes the following topics:

- [Data Maps Overview, 50](#)
- [Adding a Data Map, 62](#)
- [Adding a Data Map Record, 65](#)
- [Adding a Field to a Record, 66](#)
- [Adding a User-Defined Field to a Record, 66](#)
- [Adding Items to a Datacom Record, 67](#)
- [Adding Items to an IDMS Record, 70](#)
- [Adding Items to an IMS Record, 73](#)
- [Adding a Data Map Table, 75](#)
- [Adding a Column to a Table, 76](#)
- [Viewing Records and Fields, 77](#)
- [Viewing Tables and Columns, 78](#)
- [Viewing a Data File, 79](#)
- [Sorting Data Map Records and Tables, 81](#)
- [Viewing or Editing Properties for a Data Map or Items in a Data Map, 82](#)
- [Copying, Cutting, and Pasting Items in a Data Map, 87](#)
- [Editing Logon Information for Accessing Data Maps and Data, 88](#)
- [Sending a Data Map to a Remote System, 90](#)
- [Importing Data Maps Sent to a Remote System, 92](#)
- [Associating Multiple Data Files with a Data Map, 93](#)
- [Printing a Data Map or Items in a Data Map, 95](#)
- [Deleting Data Maps or Items in a Data Map, 98](#)

Data Maps Overview

You can define data maps to map nonrelational records to relational tables so that PowerExchange can use the SQL language to access the data.

For nonrelational sources, such as IMS databases, VSAM files, sequential data sets, and flat files, data maps are required to perform bulk data movement operations, and to provide the metadata used to define capture registrations and extraction maps for CDC.

For DB2 sources, you can also define data maps so that you can split the data in a column that contains multiple fields into separate fields.

A data map for a flat file or sequential data set must be associated with a data file that defines the record types and the data in the data map.

When you add a data map, optionally import a copybook to define the layout of the data in a data source and, for an Adabas data map, import an Adabas field definition table (FDT) and Adabas keys. Alternatively, you can define the layout of the data by manually defining one or more records and tables in a data map.

Records in data maps define the field layout of the records in a source or target. Tables in data maps provide a relational view of the data. PowerExchange requires a relational view to access data sources by using the SQL language, as follows:

- For nonrelational sources and targets, PowerExchange supports a limited subset of the SQL language. PowerExchange interprets the SQL statements internally to convert them, as needed, for access to the data source type.

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

- For relational sources and targets, PowerExchange also supports a larger set of the SQL language. PowerExchange passes the SQL statements to the RDBMS for processing.

PowerExchange maps fields in data maps to the byte level. Some copybooks, such as PL/I copybooks, have bit-level mappings. If you import a copybook that has bit-level mappings into a data map, PowerExchange concatenates the bit fields into a byte field, naming the field by using the field name of the first bit field. In a data map, you can define a user-defined field that uses one or more PowerExchange functions in an expression to separate this field back into bits. Then, in PowerCenter, use an Expression transformation to calculate the numeric value of the combined bits in the bit string.

You can run a database row test on a data map to verify that it can retrieve source data.

For information about opening, closing, and deleting data maps, and searching for fields in records and columns in tables in data maps, see [“PowerExchange Resources” on page 29](#).

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)

Data Map Names

PowerExchange provides a naming convention for data maps.

When you add a data map, you specify the following values:

- A schema name, such as `demo`
- A map name, such as `map1`

PowerExchange uses these values to construct the data map name in the format `schema.map`, for example, `demo.map1`.

You can use the naming convention to group data maps by project, operating system, or other category.

When you run a database row test on a table, PowerExchange generates an SQL statement that contains the table name prefixed with the schema name and map name. The naming format depends on the **DB Type** you select in the **Database Row Test** dialog box and on the setting for the **Use 2-tier names** option on the **Data Maps** tab in the **Preferences** dialog box.

You can use either a two- or three-tier naming convention:

- A two-tier name has the following statement:

```
select * from schema.map_table
```

- A three-tier name has the following format:

```
select * from schema.map.table
```

To use a two-tier naming convention, click **Options > Preferences**. On the **Data Maps** tab in the **Preferences** dialog box, select the **Use 2-tier names** option to use a two-tier name. Clear this option to use a three-tier name.

RELATED TOPICS:

- [“Defining Data Map Preferences” on page 43](#)

Data Sources That Are Supported for Data Maps

You can define data maps for various data sources on the i5/OS, z/OS, Linux, UNIX, and Windows operating systems.

When you define a data map, you must select the appropriate PowerExchange access method for the data source and operating system.

The following table indicates which access method to use for each data source and operating system:

Data Source	i5/OS	Linux, UNIX, and Windows	z/OS	PowerExchange Access Method
Adabas file	-	X	X	ADABAS
CA Datacom/DB file	-	-	X	DATAKOM
CA IDMS/DB database	-	-	X	IDMS
DB2 database	X	X	X	DB2
DB2 unload file	X	X	X	DB2UNLD
Flat files and sequential data sets	X	X	X	SEQ
IBM MQSeries message queue	X	X	X	MQSERIES
IMS database	-	-	X	DL/1 BATCH
IMS database accessed through a PowerExchange Listener	-	-	X	IMS ODBA
Tape data set	-	-	X	TAPE
User-defined user access method program	X	X	X	USER
VSAM entry-sequenced data set (ESDS)	-	-	X	ESDS

Data Source	i5/OS	Linux, UNIX, and Windows	z/OS	PowerExchange Access Method
VSAM key-sequenced data set (KSDS)	-	-	X	KSDS
VSAM relative record data set (RRDS)	-	-	X	RRDS

Data Files and File-List Processing

You must associate data maps for flat files or sequential data sets with a data file that defines the record types and the data in the data map.

To associate multiple data files with a data map, you can associate a data map with a file-list file. In a file-list file, you list multiple data files. PowerExchange concatenates and processes the data in the data files in the order in which you list the files.

Note: File-list processing is supported for data maps defined with the ESDS, SEQ, and TAPE access methods only.

You must associate a data file with a data map for the following data sources:

- DB2 unload files
- Flat files
- Sequential data sets
- Tape data sets
- User access method programs
- VSAM ESDS, KSDS, and RRDS data sets

To determine the data file or files associated with a record, you can use the `GetCurrentFileName` function in an expression in a user-defined field in the data map.

Note: PowerExchange cannot read large format sequential data sets. If you specify `DSNTYPE=LARGE` in the DD statement when you create the data set, PowerExchange cannot read it. Instead, you can specify `DSNTYPE=EXTREQ` in the DD statement.

RELATED TOPICS:

- [“Associating Multiple Data Files with a Data Map” on page 93](#)
- [“Step 1. Associate a Data File with a Data Map” on page 244](#)
- [“Step 1. Associate a Data File with a Multiple-Record Data Map” on page 247](#)
- [“GetCurrentFileName” on page 289](#)

Copybooks

In the PowerExchange Navigator, the term *copybook* refers generically to data definitions for records, segments, fields, keys, and other items from a data source that you import into a data map to define the layout of the data.

Import a copybook to eliminate the need to manually define the layout of these items in a data map.

You can import a copybook when you add a data map or after you add it. You can also import one or more copybooks into a record or segment in a data map to add data definitions to the record or segment.

After you import a copybook, it is stored on the local system. You can then re-import that copybook from the local system.

Select the type of copybook to import in the **Type** list in the **Import Copybook - Source Details** dialog box. The types of copybooks that you can import depend on the data source type of the data map.

RELATED TOPICS:

- [“Copybooks” on page 133](#)

Adabas Field Definition Tables and Adabas Keys

You can import an Adabas field definition table (FDT) and Adabas keys into an Adabas data map when you add a data map or after you add it.

Each database file has a corresponding FDT, which defines the record structure and the content of each field in the physical file. For each field in the record, the FDT defines the level, name, length, format, options, and special field and descriptor attributes.

If you import an FDT, you can open the Adabas data map later without connecting to the remote system because the data definitions in the data map are stored on the local system.

You can display an FDT for an Adabas data map from the PowerExchange Navigator. When you display the FDT, you can optionally refresh the locally saved FDT with data from the FDT on the remote system.

RELATED TOPICS:

- [“Importing an FDT into an Adabas Data Map” on page 139](#)
- [“Displaying and Refreshing the FDT for an Adabas Data Map” on page 139](#)

Data Map Records and Fields

A data map includes records and fields that you define or import from a copybook to describe the layout of a source or target. The fields can include user-defined fields and fields that are defined as arrays.

Note: The term *record* refers to both records and segments.

In the relational view of a data map, the table columns are based on the fields defined in data map records.

RELATED TOPICS:

- [“Defining Array Properties in the Table Properties Dialog Box” on page 57](#)

Single-Record and Multiple-Record Data Maps

You can create a data map that defines one or more records for a source or target database or data file. In the simplest case, a data map contains one record and one table that is derived from the record. The table provides a relational view of the data.

You can create a data map with multiple records by importing a copybook that defines multiple records or by specifying Record ID Values for a field. A data map with multiple records is referred to as a multiple-record data map.

Use multiple-record data maps in the following cases:

- Each segment in an IMS source has a different record definition.
- For a sequential, flat file, or VSAM data source, you define different records based on the record ID values that you enter for a field in the Field Properties dialog box.

- You need to distinguish redefined records that store field data sometimes as numeric data and sometimes as alphabetic data, depending on other information in the record.

When you create a data map for a sequential file or VSAM data source, you can define record properties that identify hierarchical relationships among the records or redefined records. PowerExchange and PowerCenter use this information to write multiple record types to VSAM or sequential targets while preserving the sequencing of the source data. For IMS unload data set sources, PowerExchange uses the segment information that you import from an IMS DBD into the multiple-record data map to write multiple record types to an IMS unload data set target.

For examples of creating single-record and multiple-record data maps, see [Chapter 9, “PowerExchange Navigator Examples” on page 239](#).

For information about the hierarchical linkage information that you can define in an SEQ or VSAM data map, see [“Record Properties - Name” on page 349](#).

User-Defined Fields

In data maps, you can define fields in records that use one or more PowerExchange functions in an expression to process source data.

These functions provide operations such as field splitting and concatenation. You can also use the CallProg function in an expression to invoke a user-defined program to process source data.

RELATED TOPICS:

- [“Adding a User-Defined Field to a Record” on page 66](#)

Record-Level Exit Programs

Create a record-level exit program in Assembler, C, COBOL, or PL/I to call a user-defined program to process data in a record in a data map before data retrieval.

For example, set up a record-level exit to call a program that decompresses compressed data in a record before data retrieval.

You can define a record-level exit program for a data map that uses any of the following access methods:

- DATACOM
- DL/1 BATCH
- ESDS
- IDMS
- IMS ODBA
- KSDS
- MQSERIES
- SEQ
- TAPE

To define a record-level exit program, specify the attributes of the record-level exit program on the **Record Level Exit** tab in the **Data Map Properties** dialog box. Include the name of the compiled DLL or shared library for the record-level exit program, the calling convention of the program, and parameters to pass to the program. PowerExchange calls the record-level exit program repeatedly to process the data in the record until all of the data is processed.

PowerExchange provides example record-level exit programs with headers files and JCL. These files, named `urle*. *`, are located in the examples directory in the PowerExchange Navigator installation directory. Customize and compile these example programs for your environment.

RELATED TOPICS:

- [“Data Map Properties - Record Level Exit” on page 348](#)
- [“User Access Method Programs” on page 317](#)

Data Map Tables and Columns

In data maps, tables provide the relational view of the data.

If you import a copybook into a data map to define the layout of data in the source, PowerExchange creates tables and columns in the data map that are based on the imported records and fields. You can also manually add tables and columns to a data map. Columns in tables are based on fields in records.

RELATED TOPICS:

- [“Adding a Data Map Table” on page 75](#)

Simple and Complex Tables

A simple table is based on a single record. A complex table is based on more than one record.

In a complex table, use options in the **Table Properties** dialog box to define a hierarchical relationship among the records on which the table is based. In a data map, a complex table mirrors the hierarchy of records or segments in the associated data source, such as an IMS database or sequential data set.

A complex table combines information into one table that, in a more normalized form, would reside in two or more tables.

You can define complex tables based on multiple related records, such as parent and child segments in an IMS database or parent and child records in a sequential data set. However, to retain the hierarchical linkage information for an IMS unload file, sequential file, or VSAM source or target, use simple single-record tables and sequencing fields that PowerExchange can generate when you import the data source into PowerCenter. File processing and data extraction are then faster, because column data in parent tables is not repeated.

The following table lists the data sources, with the associated access methods, that support complex tables:

Data Source	Access Methods	Comments
CA IDMS/DB databases	IDMS	For CA IDMS/DB, a record can have multiple parents.
IMS databases	IMS ODBA - Uses the ODBA interface DL/1 BATCH - Uses BMP and DL/1 batch jobs	For IMS, click the Display IMS Hierarchy command to view the hierarchy for an IMS data map.
Sequential data sets or flat files	SEQ	-
Tape data sets on z/OS	TAPE	-
VSAM ESDS data sets	ESDS	-
VSAM KSDS data sets	KSDS	-

Arrays

In a data map, a record can contain a field defined as an array, or in the case of some legacy data sources with OCCURS clauses, multiple fields defined as arrays.

To define a field in a record as an array, define options on the **Array** tab in the **Field Properties** dialog box. Then, to define how the elements in an array are displayed in a database row test or mapped to rows and columns in a target table, define options in the **Table Properties** dialog box.

Defining Array Properties in the Field Properties Dialog Box

To define a field in a record as an array, define options on the **Array** tab in the **Field Properties** dialog box.

The following table describes the options that you can use to define arrays:

Option	Description
Array	Defines the field as an array.
Fixed	Defines the field as a fixed array.
Variable	Defines the field as a variable array.
Array Size	The number of elements in the array. Valid values are from 1 to 32768. For a record definition imported through a COBOL copybook, this value corresponds to the x value specified in the OCCURS x clause that defines the array.
Count Field	For a variable array, the name of the field that determines the actual number of elements in this instance of the array field. For a record definition imported through a COBOL copybook, this value corresponds to the field-y value specified in the OCCURS x DEPENDING ON field-y clause that defines the array.

For example, suppose you import the following OCCURS clause in a COBOL copybook into a data map:

```
ITEM OCCURS 3 DEPENDING ON ITEMCT PIC X(15).
```

The following table lists the options and values that the **Array** tab in the **Field Properties** dialog box shows for this example:

Option	Value
Array	This option is selected.
Fixed	This option is cleared.
Variable	This option is selected.
Array Size	3
Count Field	ITEMCT

Defining Array Properties in the Table Properties Dialog Box

To define how the elements in an array are displayed in a database row test or mapped to rows and columns in a target table, define options in the **Table Properties** dialog box.

The following table describes the options in the **Table Properties** dialog box that you can use to define arrays:

Option	Description
Column Generation	<p>When you first define a table, PowerExchange names columns by using the field names in the record on which the table is based. However, a special naming convention is used for records that contain fields defined as arrays.</p> <p>If you rename columns after you define the table and you do not want column names to be regenerated, select one of the following options:</p> <ul style="list-style-type: none"> - Apply array format changes. Only changes made to array elements are reflected in the table. - Refresh with missing columns. Only new columns are added to the table. - Reset to defaults. A new set of columns are generated with the default names derived from the field names. Any changes that you have made are discarded. - Remove Hidden Columns. DB2UNLD only. A new set of columns are generated by using the default values of the Hide from Table option for each field.
Fields	<p>Controls how elements in an array or group field are displayed:</p> <ul style="list-style-type: none"> - To display each element in an array or group field in a single row, select the field. - To display each element in an array or group field in a separate row, clear the field. <p>Note: To display only fields defined as a group field or as an array in the Fields list, select the Groups and Arrays only option. To display all fields in the record in the Fields list, clear this option. This option controls the fields that are displayed in the Fields list and in the Table Properties dialog box, but not the fields that are displayed in the Database Row Test Output window.</p>
How do you want to handle multiple instances of selected records?	<p>Select one of the following options:</p> <ul style="list-style-type: none"> - New Row. For every element in the array, a new row is displayed or written to the target. - Ignore. Second and subsequent elements in the array are not displayed or written to the target. - Array. The number of elements specified in the Array list are displayed or written to the target in a single row of output. PowerExchange populates the output row until it is full, and then either ignores subsequent elements or displays a new row with the overflow elements if you select the New Row on Overflow option.
Multiple Arrays in a Single Input Row	<p>Generates multiple output rows from a single record that contains multiple arrays, or OCCURS clauses. PowerExchange sets the output fields to NULL when the data in the record is exhausted. For more information about this option, see "Multiple Arrays in a Single Input Row" on page 58.</p> <p>This option is enabled only for a table with an imported COPYLIB with multiple OCCURS clauses.</p>

For an example of how to use these options to define how the elements in an array are displayed or mapped, see ["Adding a Single-Record Data Map with an Array - Example" on page 244](#).

RELATED TOPICS:

- ["Adding a Single-Record Data Map with an Array - Example" on page 244](#)
- ["Viewing or Editing Properties for a Data Map Table" on page 86](#)

Multiple Arrays in a Single Input Row

Some legacy data sources contain multiple arrays, or OCCURS clauses, on a single record.

PowerExchange can generate multiple output rows from a single record, and set output fields to NULL when the data in the record is exhausted.

The following table shows some example input:

Input Row	fld1 - OCCURS DEPENDING	fld2 - OCCURS DEPENDING
1	3 values 10, 20, 30	2 values AA, BB
2	1 value 55	4 values DD1, DD2, DD3, DD4

The following table shows the example output:

Output Row	Fld1	Fld2
1	10	AA
2	20	BB
3	30	NULL
4	55	DD1
5	NULL	DD2
6	NULL	DD3
7	NULL	DD4

You can also use this single input-many output process with other types of fields, such as account numbers, so that every output row is complete.

Code Pages

PowerExchange provides a set of standard code pages that you can associate with data maps or with character fields in data maps. You can also associate user-defined code pages with data maps or fields.

You can associate a specific code page with a data map or a field, or use the default code page for the operating system.

If you associate a code page with a field, the field-level code page overrides the data map-level code page.

PowerExchange Code Pages

PowerExchange provides a set of standard code pages.

PowerExchange includes ICU software to provide extended single-byte and multibyte code page support for the following data sources:

- Datacom
- DB2 for i5/OS

- DB2 for z/OS
- Flat files
- Sequential data sets
- VSAM data sets

On i5/OS and z/OS systems, the IBM-037 code page is the default code page. On Linux, UNIX, and Windows systems, the ISO-8859 code page is the default code page.

For a complete list of all of the code pages and aliases that are defined for a PowerExchange installation, use the ICUCHECK utility. For more information about this utility, see the *PowerExchange Reference Manual*.

The following table describes some of the code pages that PowerExchange supports:

Code Page	Description
IBM-037*	USA, Canada, Brazil
IBM-1047*	Latin 1/Open Systems
IBM-273*	Germany, Austria
IBM-277*	Denmark, Norway
IBM-278*	Finland, Sweden
IBM-280*	Italy
IBM-284*	Spain, Latin America
IBM-285*	United Kingdom
IBM-297*	France
IBM-424*	Modern Hebrew
IBM-500*	International
ISO-8859	UNIX and Windows

Note: Code pages prefixed with IBM- can handle data from i5/OS and z/OS systems.

User-Defined Code Pages

Define up to ten user-defined code pages.

User-defined code pages enable you to display or print specified hexadecimal values.

You must name user-defined code pages USRCPnn, where nn is a value from 00 to 09.

For more information about PowerExchange code pages, see the *PowerExchange Reference Manual*.

Data Checking

Define data checking to specify how PowerExchange handles record mapping errors and field-level data errors during an extraction.

The data checking options are:

- Record mapping errors. You can specify that PowerExchange stops the extraction or that it continues after skipping the row and writing a message to the log file. Default is to stop the extraction.

If you do not want to take the default action of stopping the extraction when a mapping error occurs, you can select **skip row and write log file msg**. Informatica recommends that you put any necessary effort into mapping the data by using tools such as record IDs, so that PowerExchange skips rows only as the last resort.

- Data errors in fields. You can specify that PowerExchange stops the extraction or that it continues after performing one of the following actions:
 - Skipping the row and writing a message to the log file.
 - Setting the field to null.
 - Replacing the bad data with values according to the **Data Defaults** tab of the **Data Map Properties** dialog box. This option is available for DATE, TIME, CHAR, VARCHAR, and CLOB field types.

Default is to stop the extraction.

Notes:

- Selecting field types on the **Data Defaults** tab in the **Data Map Properties** dialog box provides a default setting if you select the **Use Data Map rules** option on the **Field Checking** tab of the **Field Properties** dialog box. Selecting field types also defines whether the fields that result from calls to user-defined functions are checked. Because it is seldom a good idea to perform data checking on user-defined fields, Informatica makes the following recommendations:
 - On the **Data Defaults** tab of the **Data Map Properties** dialog box, do not select field types.
 - On the **Field Checking** tab of the **Field Properties** dialog box, avoid selecting **Use data map rules** and define rules on a field-by-field basis.
- Because performing data checking increases the CPU cost on the machine where the mapping runs, Informatica recommends that you perform data checking only when absolutely necessary.
- Data checking is required only when fields contain data that cannot be loaded onto the target.
- Invalid data in DATE or TIME columns often causes problems for target systems and might need to be checked.
- Most modern relational databases are tolerant of CHAR, VARCHAR and CLOB columns that contain control characters with hexadecimal values less than the value of the space character. However, some legacy databases which cannot tolerate these control characters.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)
- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)

Exported Data Maps

If the data source for a data map is on a remote system, such as on an i5/OS or z/OS system, send the data map to the remote system to make it available to PowerExchange for a bulk data movement operations.

When you send the data map to the remote system, PowerExchange converts the data map into a platform-independent file on the remote system, which PowerExchange uses for data extraction during a bulk data movement operation. The original data map remains on the PowerExchange Navigator system, but if you change the local data map, the changes are not reflected in the remote data map until you send the data map to the remote system again.

You can import one or more data maps that were sent, or exported, to a remote system to the PowerExchange Navigator system. For each data map that you import, the data map remains on the remote system and a local copy is saved on the PowerExchange Navigator system.

RELATED TOPICS:

- [“Sending a Data Map to a Remote System” on page 90](#)
- [“Importing Data Maps Sent to a Remote System” on page 92](#)

Datacom Data Map Items

In a Datacom data map, define elements and keys to optimize CA Datacom/DB data access.

An element is the unit of transfer used between CA Datacom/DB and applications issuing CA Datacom/DB commands. An element consists of one or more contiguous columns, or FIELD entity occurrences. An element should contain columns that an application program uses at run time. Each table has at least one and no more than 255 elements.

Keys and key segments set the sort order for data retrieval. A key is composed of columns, which can be non-contiguous and in any sequence. Each column in a key can be either ascending or descending in value. Each key can be up to 180 characters long. Define a maximum of 99 keys for a table and 999 keys for a database. Define any key as unique, if each row in the table specifies a unique value for the key.

You must define a master key and a native key for each table in a Datacom data map. The master key behaves like other keys, but you can enable or disable changes to the master key. The native key dictates the physical sequence in which the data is stored. The native key can be the same as the master key.

RELATED TOPICS:

- [“Adding an Element to a Datacom Record” on page 67](#)
- [“Datacom Data Maps” on page 105](#)
- [“Adding a Key to a Datacom Record” on page 68](#)
- [“Adding a Key Segment to a Datacom Record” on page 69](#)

IDMS Data Map Items

In an IDMS data map, define owner record and sets and CALC elements to optimize CA IDMS/DB database access.

RELATED TOPICS:

- [“Adding a CALC Element to an IDMS Record” on page 70](#)
- [“IDMS Data Maps” on page 112](#)
- [“Adding an Owner Record and Set to an IDMS Record” on page 71](#)

IMS Data Map Items

In an IMS DL/1 batch or IMS ODBA data map record, define concatenated key (CCK) fields and search fields to optimize IMS database access.

The CCK field and search field definitions, including the name, position, and length values, must match the definitions in the IMS DBD segment. Although you can include any field that is defined in the IMS DBD segment in the search, include only the defined SEQ fields with the most important fields at the top of the list.

RELATED TOPICS:

- [“Adding a CCK Field to an IMS Record” on page 73](#)
- [“IMS Data Maps” on page 115](#)
- [“Adding Search Fields to an IMS Record” on page 74](#)

User Access Method Programs

Create a user access method program written in Assembler, C, or PL/I to process data in a data map.

Call the user access method program by defining a data map with the USER access method and specifying the attributes of the user access method program, including the name of the compiled user access method DLL or shared library, parameters to pass to the program, and the full path and name of the associated data file. PowerExchange calls the user access method program repeatedly to process the data in the associated data file until all the data is processed.

PowerExchange provides example user access method programs with headers files and JCL. These files, named `uam*. *`, are located in the examples directory in the PowerExchange Navigator installation directory. Edit and compile these example programs to customize them for your environment.

RELATED TOPICS:

- [“User Access Method Programs” on page 317](#)
- [“Adding a Data Map” on page 62](#)

Adding a Data Map

When you add a data map, specify a schema name, a data map name, and an access method.

If the data map is for a flat file or sequential data set, associate it with a data file.

Optionally, import a copybook to define the layout of the data in a data source. For Adabas data maps, you can import an Adabas field definition table (FDT) and Adabas keys.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
The **Name** dialog box appears.
2. Enter information about the data map.

The following table describes the information to enter:

Option	Description
Schema Name	<p>Enter a user-defined schema name.</p> <p>PowerExchange uses this name as the first part of the data map name, as follows: <i>schema_name.data_map_name</i></p> <p>Valid values are alphanumeric characters.</p> <p>Must begin with an alphabetic character.</p> <p>Maximum length is 10 characters.</p>
Data Map Name	<p>Enter a user-defined map name.</p> <p>PowerExchange uses this name as the second part of the data map name, as follows: <i>schema_name.data_map_name</i></p> <p>Valid values contain only alphanumeric characters and begin with a letter.</p> <p>Maximum length is 10 characters.</p>
Access Method	<p>Select one of the following access methods:</p> <ul style="list-style-type: none"> - ADABAS. Adabas file. - DATACOM. CA Datacom/DB file. - DB2. DB2 for i5/OS, DB2 for Linux, UNIX, and Windows, or DB2 for z/OS database. - DB2UNLD. DB2 unload file. - DL/1 BATCH. DL/I batch or BMP access to an IMS database on z/OS. - ESDS. VSAM Entry Sequenced Data Set (ESDS) on z/OS. - IDMS. CA IDMS/DB database on z/OS. - IMS ODBA. IMS ODBA access to an IMS database on z/OS. The ODBA queries run as subtasks of the PowerExchange Listener on z/OS. - KSDS. VSAM Key Sequenced Data Set (KSDS) on z/OS. - MQSERIES. IBM MQSeries message queue. - RRDS. VSAM Relative Record Data Set (RRDS) on z/OS. - SEQ. Flat file. - TAPE. Data set on tape. - USER. User-defined user access method program. <p>Note: You can override the DL/1 BATCH or IMS ODBA access method with the other available method for a PowerCenter session or database row test.</p>
Import Record Definitions	<p>Select this option to import a copybook into the data map.</p> <p>Clear this option to add only the data map.</p> <p>Default is selected.</p>
Import Key Fields/FDT	<p>Adabas. Select this option to import an Adabas FDT and Adabas keys into the data map.</p> <p>Clear this option to add only the data map.</p> <p>Default is selected.</p>

3. Enter information in the dialog box for the access method that you choose.

The following table lists the dialog box that appears for each access method:

Access Method	Dialog Box	Reference
ADABAS	ADABAS Access Method	"Data Map Properties - ADABAS Access Method" on page 328
DATACOM	DATACOM Access Method	"Data Map Properties - DATACOM Access Method" on page 330
DB2	DB2 Access Method	"Data Map Properties - DB2 Access Method" on page 330
DB2UNLD	DB2UNLD Access Method	"Data Map Properties - DB2UNLD Access Method" on page 331
DL/1 BATCH	DL/1 Batch Access Method	"Data Map Properties - DL/1 Batch or IMS ODBA Access Method" on page 334
ESDS	ESDS Access Method	"Data Map Properties - ESDS Access Method" on page 335
IDMS	IDMS Access Method	"Data Map Properties - IDMS Access Method" on page 336
IMS ODBA	IMS ODBA Access Method	"Data Map Properties - DL/1 Batch or IMS ODBA Access Method" on page 334
KSDS	KSDS Access Method	"Data Map Properties - KSDS Access Method" on page 337
MQSERIES	MQSeries Access Method	"Data Map Properties - MQSeries Access Method" on page 338
RRDS	RRDS Access Method	"Data Map Properties - RRDS Access Method" on page 339
SEQ	SEQ Access Method	"Data Map Properties - SEQ Access Method" on page 340
TAPE	TAPE Access Method	"Data Map Properties - TAPE Access Method" on page 342
USER	USER Access Method	"Data Map Properties - USER Access Method" on page 343

4. For an Adabas data map, if you selected the **Import Key Fields/FDT** option, click **Next**. Then, enter information in the **Adabas FDT Import** dialog box.

The following table describes the information to enter:

Option	Description
UserID	Enter an Adabas user ID that has the authority to access the location.
Password	Enter a password for the specified user ID. If you specify a user ID and omit the password, the PowerExchange Navigator prompts you to enter the password in the Missing Password dialog box.
Location	Select the location of the FDT. Options are: <ul style="list-style-type: none"> - Local. The FDT is located on the PowerExchange Navigator system. - Remote. The FDT is located on a remote system. The locations displayed in the Location list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is Local .

Option	Description
File Password	If the FDT file is password-protected, enter the password to access the file.
Save User ID and Password(s) for session	Select this option to save the user ID and password, and the file password if specified, for the remainder of the session. Default is selected.

- If the **Import Records Definitions** option is selected, click **Next**.
If you cleared this option, click **Finish**.

The data map appears on the **Data Map** tab in the **Resource Explorer**.

If you imported an FDT into an Adabas data map, edit table properties in the data map to add the imported keys to the table.

RELATED TOPICS:

- [“Adding a Key to a Table in an Adabas Data Map” on page 105](#)
- [“Importing a Copybook into a Data Map” on page 135](#)

Adding a Data Map Record

Add one or more records to a data map.

For data maps of all data source types, enter a record name to add a record to a data map.

For Datacom, IDMS, IMS DL/1 batch, and IMS ODBA data maps, you must also enter additional information.

- Open the data map.
- On the **Data Map** tab, right-click the data map and click **Add Record**.
The **Add Record** dialog box appears.
- On the **Name** tab, define record properties. The properties that you can define depend on the data source type.
- Click **OK**.

The record appears on the **Data Map** tab in the **Resource Explorer**.

The **Record** window displays the record.

For a Datacom data map record, optionally add elements, keys, and key segments.

For an IDMS data map record, optionally add CALC elements and owner records and sets.

For an IMS DL/1 batch or IMS ODBA data map record, optionally add CCK fields and search fields.

RELATED TOPICS:

- [“Adding an Element to a Datacom Record” on page 67](#)
- [“Adding a Key to a Datacom Record” on page 68](#)
- [“Adding a Key Segment to a Datacom Record” on page 69](#)
- [“Adding a CALC Element to an IDMS Record” on page 70](#)
- [“Adding an Owner Record and Set to an IDMS Record” on page 71](#)

- [“Adding a CCK Field to an IMS Record” on page 73](#)
- [“Adding Search Fields to an IMS Record” on page 74](#)
- [“Record, Field, and Table Properties” on page 349](#)

Adding a Field to a Record

Add one or more fields to a record in a data map.

1. Open the data map and the record.
2. To add a field to the end of the record, right-click the record name in the **Record** window and click **Add Field at End**.

To add a field adjacent to a field, right-click a field and select to add the field before or after the selected field, or as a child of a GROUP field.

3. On the **Name** tab in the **Add Field** dialog box, define the field properties.
4. Click **OK**.

The **Record** window displays the field.

RELATED TOPICS:

- [“Field Properties - Name ” on page 354](#)

Adding a User-Defined Field to a Record

In data maps, you can define fields in records that use one or more PowerExchange functions in an expression to process source data.

These functions provide operations such as field splitting and concatenation. You can also use the CallProg function in an expression to run a user-defined program to process source data.

1. Open the data map and the record.
2. In the **Record** window, click the **Expr(n)** tab.
3. To add a field to the end of the record, right-click anywhere on the **Expr(n)** tab and click **Add Field at End**.
To add a field adjacent to a field, right-click a field and select to add the field before or after the selected field.
4. To define the field properties, click in the cells in the **Name**, **Type**, **Precision**, **Scale**, **Length**, and **Phase** columns.
5. To enter an expression for the field, click in the cell in the **Expression** column, or click the Browse button to open the **Expression Editor** dialog box.

The following table describes the information to enter in the **Expression Editor** dialog box:

Option	Description
Function List	Double-click one or more PowerExchange functions to build an expression. The functions appear in the Expression box.
Expression	Enter an expression to populate or process the data in the field. An expression can contain one or more of the following items: <ul style="list-style-type: none"> - A text or numeric constant such as <i>My text</i> or <i>1234</i> - A PowerExchange function such as <i>Concat</i> or <i>Split</i> - The PowerExchange CallProg function that specifies a user-defined Assembler, C, COBOL, or PL/I program and its parameters
Validate	Validates the expression in the Expression box. If PowerExchange detects an error, an error message displays in the Validation box.
Validation	Displays error messages for errors detected during validation of an expression.

6. Click **OK**.
7. To test the user-defined field, complete the following actions:
 - On the **Data Map** tab in the **Resource Explorer**, right-click a table and click **Properties**.
 - In the **Table Properties** dialog box, select **Refresh with missing columns** in the **Column Generation** list.
 - Click **OK**.
 - Run a database row test on the table.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)
- [“PowerExchange Functions for User-Defined Fields Overview” on page 282](#)

Adding Items to a Datacom Record

Edit a record in a Datacom data map to add Datacom-specific items to the record.

You can add one or more elements, keys, and key segments to a Datacom record.

Adding an Element to a Datacom Record

Add one or more elements to a record in a Datacom data map.

To add an element to a Datacom record:

1. Open the Datacom data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**. The **Record Properties** dialog box appears.
3. On the **Elements** tab, click the **Add New** toolbar button.

The **Add New Element** dialog box appears.

4. In the **Add New Element** dialog box, define element properties.

The following table describes the information to enter:

Option	Description
Name	Enter the name of the record element.
Position	Enter the position offset of the element from the start of the record. Default is 1.
Length	Enter the length of the element. Default is 1.

5. Click **OK**.

The element appears in the **Elements** list on the **Elements** tab.

RELATED TOPICS:

- [“Datacom Data Map Items” on page 61](#)
- [“Viewing or Editing Properties for an Element in a Datacom Record” on page 83](#)

Adding a Key to a Datacom Record

Add one or more keys to a record in a Datacom data map.

To add a key to a Datacom record:

1. Open the Datacom data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.

The **Record Properties** dialog box appears.

The following table describes the information that appears for any defined keys on the **Keys** tab in the **Record Properties** dialog box:

Option	Description
Long Key Name	The long name of the key. Maximum length is 32 characters.
Short Key Name	The short name of the key. Maximum length is five characters.

3. On the **Keys** tab, click the **Add New** toolbar button.

The **Key Properties** dialog box appears.

4. On the **Key Details** tab, define the key.

The following table describes the information to enter:

Option	Description
Long Key Name	Enter the long name of the key. Maximum length is 32 characters.
Short Key Name	Enter the short name of the key. Maximum length is five characters.
Master Key	Enter the master key for the table. Every table requires a master key. The master key can be the same key as the native key.
Change Allowed	Select this option to allow the master key to be changed. Clear this option to prevent the master key from being changed. If you clear this option, CA Datacom/DB prohibits any change to the value of the master key for a row in the table. Default is cleared.
Unique	Select this option to prohibit duplicate master keys. Default is cleared.
Duplicate Allowed	Select this option to allow duplicate master keys. Default is cleared.
Native	Select this option to define the key as the native sequence key for this table. Every table requires one, and only one, native key, which designates the order in which data is loaded and maintained. The native key represents the order most often used in sequential processing. Default is cleared.
Nullable	Select this option to allow the key to have nil values placed in the index. Nil represents low values or blanks. Default is cleared.
Compound Boolean Selection Allowed	Select this option to allow compound Boolean selection. Default is cleared.

5. Click **OK**.

The key appears in the **Keys** list on the **Keys** tab.

RELATED TOPICS:

- [“Datacom Data Map Items” on page 61](#)
- [“Viewing or Editing Properties for a Key in a Datacom Record” on page 84](#)

Adding a Key Segment to a Datacom Record

Add one or more key segments to a record in a Datacom data map.

To add a key segment to a Datacom record:

1. Open the Datacom data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.

The **Record Properties** dialog box appears.

3. On the **Keys** tab, select the key in the **Keys** list and click the **Properties** toolbar button.

The **Key Properties** dialog box appears.

4. On the **Segment Details** tab, click the **Add New** toolbar button.

The **New Segment** dialog box appears.

5. In the **New Segment** dialog box, define the key segment.

The following table describes the information to enter:

Option	Description
Name	Enter the name of the key segment. Note: This field is read-only on the Segment Details tab in the Key Properties dialog box.
Position	Enter the position of the key segment. Default is 1.
Length	Enter the length of the key segment. Default is 1.
Order	Select one of the following options to specify the order of the key segment: - ASC . Ascending order. - DESC . Descending order.
Sensitivity	Select one of the following options to specify the sensitivity of the key segment: - ON . Sensitivity is enabled. - OFF . Sensitivity is disabled.

6. Click **OK**.

The key segment appears in the **Segment Details** list on the **Segment Details** tab.

RELATED TOPICS:

- [“Datacom Data Map Items” on page 61](#)
- [“Viewing or Editing Properties for a Key Segment in a Datacom Record” on page 84](#)

Adding Items to an IDMS Record

Edit a record in an IDMS data map to add IDMS-specific items to the record.

You can add one or more CALC elements and owner record and sets to an IDMS record.

Adding a CALC Element to an IDMS Record

Add one or more CALC elements to a record in an IDMS data map.

To add a CALC element to an IDMS record:

1. Open the IDMS data map.

- On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
The **Record Properties** dialog box appears.
- On the **CALC Element Names** tab, click the **Add New** toolbar button.
The **New CALC Element** dialog box appears.
- In the **New CALC Element** dialog box, define the CALC element.
The following table shows the information to enter:

Option	Description
Name	Enter the name of the CALC element.
Position	Enter the position offset of the CALC element from the start of the record. Default is 1.
Length	Enter the length of the CALC element. Default is 1.

- Click **OK**.
The CALC element appears in the **CALC Element Names** list on the **CALC Element Names** tab. For information about the properties displayed for a CALC element, see [“Adding an Owner Record and Set to an IDMS Record” on page 71](#).

RELATED TOPICS:

- [“Viewing or Editing Properties for a CALC Element in an IDMS Record” on page 85](#)

Adding an Owner Record and Set to an IDMS Record

Add one or more owner records and sets to a record in an IDMS data map.

To add an owner record and set to an IDMS record:

- Open the IDMS data map.
- On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
The **Record Properties** dialog box appears.
- On the **Owner Record and Set Names** tab, click the **Add New** toolbar button.
The **Owner Record and Set Properties** dialog box appears.
- On the **Owner Record and Set Details** tab, define the owner record and set properties.
The following table describes the information to enter:

Option	Description
Owner Name	Enter the name of the parent record. Enter SYSTEM to identify the set as a system-owned index.
Set Name	Enter a description of the relationship between the record and owner.

Option	Description
Area Name	Enter the name of the area in which the owner record and index structure are located. Enabled when you edit an existing owner record and set with an owner name of SYSTEM.
Order	<p>Select the logical order for adding new member record occurrences to a set occurrence at run time:</p> <ul style="list-style-type: none"> - None - First - Last - Next - Prior - Sorted <p>Note: If you select Sorted, the Sort Element Name tab appears on the Owner Record and Set Names dialog box. Click the Sort Element Name tab to set the sort order for elements.</p> <p>Default is none.</p>
Mode	<p>Select the set characteristic that determines how CA IDMS/DB pointers are maintained at run time:</p> <ul style="list-style-type: none"> - None - Chain - Chain Linked to Prior Index - VSAM Index <p>Default is none.</p>
Duplicates	<p>Select how CA IDMS/DB handles a record occurrence with a sort key that duplicates the sort key of an existing occurrence:</p> <ul style="list-style-type: none"> - None - By DBKey - First - Last - Not Allowed <p>Default is none.</p>
Set Options	<p>Select one of the following set options:</p> <ul style="list-style-type: none"> - None - Mandatory Automatic - Mandatory Manual - Optional Automatic - Optional Manual <p>Default is none.</p>

Option	Description
Natural Sequence	Select this option to use the natural sort sequence, which sorts and evaluates the key fields using negative values before positive values. Clear this option to use the default sort sequence. By default, CA IDMS/DB sorts and evaluates the key fields using a standard collating sequence, which sorts data by its hexadecimal value. Default is cleared.
Linked to Owner	Select this option to link each member record of the specified type in the set to the owner record. You cannot change this option if the relationship is already being used by a defined table in the data map.
Owner Key Position	Enabled if the Linked to Owner option is selected. Enter the position within the record of the owner key. This value is the owner database key position or the index value, relative to 1, defined in a schema set statement. If you do not enter a position value, a value is automatically assigned. Default is 0.
Owner Page Group	Enter the IDMS page group of the owner record. Maximum value is 32767. Default is 0.
Owner Radix	Enter the IDMS radix for the owner record. Valid values are from 0 through 12. Default is 8.

5. Click **OK**.

The owner record and set appears in the **Owner Record and Set Names** list on the **Owner Record and Set Details** tab.

RELATED TOPICS:

- [“IDMS Data Map Items” on page 61](#)
- [“Viewing or Editing Properties for an Owner Record and Set in an IDMS Record” on page 85](#)

Adding Items to an IMS Record

Edit a record in an IMS DL/1 batch or IMS ODBA data map to add IMS-specific items.

You can add one or more CCK fields and search fields to an IMS data map record.

Adding a CCK Field to an IMS Record

Add one or more CCK fields to a record in an IMS DL/1 batch or IMS ODBA data map.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
The **Record Properties** dialog box appears.
3. On the **CCK Fields** tab, click the **Add New** toolbar button.

The **New CCK Field** dialog box appears.

4. In the **New CCK Field** dialog box, define the CCK field properties.

The following table describes the information to enter:

Option	Description
Name	Enter a name for the CCK field. The name must be a field name in the IMS DBD, usually the name of a SEQUENCE field. It can be any field name defined within a SEGM. Maximum length is eight characters. Valid characters are ASCII characters and the pound and yen sign.
Field Type	Select the datatype of the field.
Precision	Enter the precision of the field.
Scale	Enter the scale of the field.
Length	Enter the length of the field.
Data Codepage	Select a code page for the contents of the CCK field. The code page describes the character set for the character data in the database.

5. Click **OK**.

The CCK field appears in the **CCK Fields** list on the **CCK Fields** tab.

To view CCK columns, select the **Generate CCK Columns** option in the properties for the IMS data map table that is based on the record that contains the CCK fields.

RELATED TOPICS:

- [“IMS Data Map Items” on page 62](#)
- [“Viewing or Editing Properties for a CCK Field in an IMS Record” on page 86](#)
- [“Table Properties - Definition” on page 364](#)

Adding Search Fields to an IMS Record

Add one or more search fields to an IMS DL/1 batch or IMS ODBA data map record.

1. Open the data map.

Note: If message PWX-01574 is displayed, the DBD that you imported for the data map supplied the key type value of "S" for a record search field. The "S" key type, for sequence fields, is deprecated. Edit record properties to change the **Key** value for the search field to **U**.

2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.

The **Record Properties** dialog box appears.

3. On the **Search Fields** tab, click the **Add New** toolbar button.
4. In the **New Search Field** dialog box, define the search field.

The following table describes the information to enter:

Option	Description
Name	Enter a name for the search field. The name must be a field name in the IMS DBD, usually the name of a SEQUENCE field. It can be any field name defined within a SEGM. Maximum length is eight characters. Valid characters are ASCII characters and the pound and yen sign.
Position	Enter the position of the search field. Valid values are from 1 through 32767. Default is 1.
Length	Enter the length of the search field. Valid values are from 1 through 32767. Default is 1.
Key	Select one of the following key types: - Blank. - U . Unique key type. - M . Many, the non-unique key type. Default is U.

5. Click **OK**.

The search field appears in the **Search Fields** list on the **Search Fields** tab.

RELATED TOPICS:

- [“IMS Data Map Items” on page 62](#)
- [“Viewing or Editing Properties for a Search Field in an IMS Record” on page 86](#)

Adding a Data Map Table

If you import a copybook into a data map to define the layout of data in the source, PowerExchange creates tables and columns in the data map that are based on the imported records and fields.

You can also manually add tables and columns to a data map. Columns in tables are based on fields in records.

You can define a complex table that is based on multiple records. In a data map, a complex table mirrors the hierarchy of records or segments in the associated data source, such as an IMS or CA IDMS/DB database, data set, or sequential data set. When you define a complex table, define a hierarchical relationship among the records on which the table is based. You can also set options that control how records on which columns in the complex table appear in a database row test, and are mapped to rows and columns in a target table.

1. Open the data map.
2. On the **Data Map** tab, right-click the data map and click **Add Table**.

3. Depending on the data source for the data map, view or define the following properties on some of all of the following tabs in the **Table Properties** dialog box:
 - **Definition.** Properties for a complex table, including settings for how records in a complex table appear in a database row test or are mapped in a target table, and how changes in records on which the table is based are reflected in the corresponding columns in the table.
 - **Filter.** Filter columns by name to either include them in or exclude them from a table.
 - **Keys.** For Adabas, the primary key for the table.
 - **IMS Options.** For IMS DL/1 batch and IMS ODBA, properties for segments in a complex table.
4. Click **OK**.
 The table appears on the **Data Map** tab in the **Resource Explorer**.
 The **Table** window displays the table.

RELATED TOPICS:

- [“Adding a Key to a Table in an Adabas Data Map” on page 105](#)
- [“Adding a Multiple-Record Data Map with a Complex Table - Example” on page 246](#)
- [“Table Properties - Definition” on page 364](#)
- [“Table Properties - Filter ” on page 367](#)
- [“Table Properties - Keys ” on page 368](#)
- [“Table Properties - IMS Options ” on page 367](#)

Adding a Column to a Table

Add one or more columns to a table in a data map.

1. Open the data map and the table.
2. To add a column to the end of the table, right-click the table name in the **Table** window and click **Add Column at End**.
 To add a column adjacent to a column, right-click the column and select to add the column before or after the selected column.
 The **Add Column** dialog box appears.
3. On the **Name** tab, enter column properties.
 The following table describes these properties:

Option	Description
Name	Enter a user-defined name for the column. The column name must be unique within the table. Valid characters are ASCII alphanumeric characters and the pound and yen sign. The first character must be alphabetic. Maximum length is 32 characters.
Base Field	Select a field in the record on which to base the column.

4. Click **OK**.

The **Table** window displays the column.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Column in a Table” on page 87](#)

Viewing Records and Fields

You can view one or more records in a data map, and fields in each record.

Each record appears in its own **Record** window. A **Record** window displays the record name and the fields in the record.

To view records and fields:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, complete one of the following actions:
 - To view a single record, double-click the record.
 - To view multiple records, each in a separate **Record** window, right-click each record and click **Explore**.

The **Fields** tab in a **Record** window displays the record name and the following information for each field:

- Field name
- Datatype
- Precision, scale, and length values, in parentheses
- If the field is defined as an array, the value **ARRAY** followed by the number of elements in the array
- The decimal offset of the field within the record
- The hexadecimal offset of the field within the record

The **Expr(n)** tab in the **Record** window displays any user-defined fields. The *n* variable indicates the number of user-defined fields.

Each row on the **Expr(n)** tab contains either a field definition or the assignment of a value to a defined field.

The following table describes the columns that the **Expr(n)** tab displays for each field:

Column	Description
Field status indicator	Indicates the status of the field. The options are: <ul style="list-style-type: none">- Check mark. The field is valid.- X. The field is not valid. The comment in the Validation column describes the type of error that the field contains..- Arrow. A new field is selected, after selecting one of the Add Field menu commands.
Name	The user-defined name for the field.
Type	The datatype of the field.

Column	Description
Precision	The precision of the field.
Scale	The scale of the field.
Length	The length of the field.
Phase	Indicates the type of database communication for the user-defined field. For example, if a field is defined with a phase of R, its expression is not recalculated when the database is being written to. The options are: - R . Read. - W . Write. - RW . Read and write.
Expression	Any expression defined for the field.
Validation	If you click Validate in the Expression Editor dialog box, the Validation column displays the validation message.

3. To sort or delete fields on the **Expr(n)** tab, perform the following actions:
 - To move a field up or down in the window, right-click a field and click **Move Field Up** or **Move Field Down**.
 - To move the completed fields to the top of the list, right-click anywhere on the **Expr(n)** tab in the **Record** window and click **Organize Fields**.
 - To delete a field, right-click a field and click **Delete Field**.
4. To show or hide columns that define the datatype attributes for fields on the **Expr(n)** tab, right-click anywhere on the **Expr(n)** tab in the **Record** window and click one of the following commands:
 - To show or hide the **Type**, **Precision**, **Scale**, and **Length** columns, click **Hide Type Columns** or **Show Type Columns**.
 - To show all columns and restore all columns to the default width, click **Restore Columns**.

RELATED TOPICS:

- [“Adding a User-Defined Field to a Record” on page 66](#)

Viewing Tables and Columns

You can view one or more tables in a data map, and columns in each table.

Each table appears in its own **Table** window. A **Table** window displays the table name and the columns in the table.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, complete one of the following actions:
 - To view a single table, double-click the table.
 - To view multiple tables, each in a separate **Table** window, right-click each table and click **Explore**.

When you open a table, the **Table** window displays the table name and the following information for each column in the table:

- Column name
- Record and field on which the column is based, in the following format:

record:field

- Datatype of the field on which the column is based
- Length of the field on which the column is based, in parentheses

Viewing a Data File

If a data map has an associated data file, you can view the data file in the **Data File** window.

Define the default display format and the number of records to initially display in the **Data Map Preferences** dialog box. When you view the data file, you can display additional records, until all of the records in the data file have been displayed. You can also display a ruler above a record to show the number of characters in the record.

1. Open the data map.
2. On the **Resources** tab in the **Resource Explorer**, click **File > View Data File**.

Note: If the data map does not have an associated data file, this command is not available.

The first time you open a data file for a data map in a session, the **Data File Remote Node** dialog box appears. Thereafter, the **Data File** window appears.

3. If the **Data File Remote Node** dialog box appears, enter connection information for the system on which the data file is located, and click **OK**.

The following table describes the connection information to enter:

Option	Description
User ID	<p>If a user ID is required to access the system where the file is located, enter a user ID.</p> <p>For files on supported Linux, UNIX, or Windows systems, if you have enabled PowerExchange LDAP user authentication, the user name is the enterprise user name. For more information, see the <i>PowerExchange Reference Manual</i>.</p>
Password	<p>If you specify a user ID, enter a password for that user ID.</p> <p>For access to a file on i5/OS or z/OS, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:</p> <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p>
Location	<p>Select the node name of the system where the data file is located.</p> <p>This field lists the locations that are defined in the NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.</p> <p>Default is local.</p>
File Password	<p>If a password is required to access the data file, enter a password.</p>
Save User ID and Password(s) for session	<p>Select this option to save the information you enter for the duration of the session.</p> <p>Default is selected.</p>

The following table describes the columns that the **Data File** window displays for each record in the data file:

Column	Description
Rec No	The record number.
Offset	The data offset of the hexadecimal data. Used for the vertical display format.
Data	The record data, which is displayed in one of the following display formats: <ul style="list-style-type: none">- Char/Hex. Displays records in character and hexadecimal format.- Char. Displays records in character format.- Vertical. Displays records vertically in character and hexadecimal format. Default is Vertical.

By default, ten records are displayed.

4. To display additional records in the data file, enter the number of records to display in the box next to the **Get Records** toolbar button. Then click the **Get Records** toolbar button. After all of the records have been retrieved, a message box indicates that the end of the file is reached.
5. To change the display format or to show or hide the scale for a record, perform these actions:
 - To change the display format, click **Mode** and click a display format.
 - To show or hide the scale for a record, right-click a record and click **Show Scale** or **Hide Scale**.

RELATED TOPICS:

- [“Defining Data Map Preferences” on page 43](#)

Sorting Data Map Records and Tables

You can sort the records and tables in a data map in alphabetical order.

By default, records and tables appear in a data map in the order in which you added them. Sorting records and tables in alphabetical order is a temporary operation. The next time you open a data map, records and tables appear in their original sort order.

To sort data records and tables in a data map:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the data map and click **Sort Items**.

Viewing or Editing Properties for a Data Map or Items in a Data Map

You can view or edit properties for records, fields, including user-defined fields and fields defined as arrays, tables, and columns in a data map.

Note: The term *record* refers to both records and segments.

Based on the data source type, you can also view or edit properties for the following items in data map records:

- **Datacom.** Elements, keys, and key segments.
- **IDMS.** CALC elements, and owner records and sets.
- **IMS DL/1 batch.** CCK and search fields.

Viewing or Editing Properties for a Data Map

You can view or edit properties that pertain to an entire data map.

To view or edit properties for a data map:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the data map and click **Properties**.
The **Data Map Properties** dialog box appears.
3. Based on the data source type of the data map, view or edit data map properties on some or all tabs:
 - **Name.** The schema name, data map name, and access method.
 - **Access Method.** Access method information.
 - **Data Defaults.** Default values for dates and replacement characters for fields that fail data checking.
 - **Data Checking.** The type of data checking that PowerExchange performs for record mapping and field data errors.
 - **Record Level Exit Program.** A record-level exit program.
 - **Keys.** Read-only. For Adabas, information about keys.

Viewing or Editing Properties for a Data Map Record

You can view or edit properties for a record or segment in a data map.

To view or edit properties for a data map record:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click a record and click **Properties**.
The **Record Properties** dialog box appears.
3. Based on the data source type of the data map, view or edit record properties on some or all tabs:
 - **Name.** For all data source types except Datacom and IDMS, the user-defined record name.
For Datacom, the long table name, short table name, and other information.
For IDMS, the record name, the corresponding IDMS record name, and other information.
 - **Import Details.** Read-only. Information about the last imported copybook.
 - **Keys.** For Datacom, the long key name and short key name for a key.

- **Elements.** For Datacom, the name, position, and length for an element.
- **CALC Element Names.** For IDMS, the name, position, and length for a CALC element.
- **Owner Records and Set Names.** For IDMS, owner record and set properties.
- **CCK Fields.** For IMS, CCK field properties.
- **Search Fields.** For IMS, search field properties.

Viewing or Editing Properties for a Field in a Record

You can view or edit properties for a field in a record in a data map.

You might edit properties for a field to change its name or its datatype, define the field as an array, or change the code page for the field.

To view or edit properties for a field in a record:

1. Open the data map and the record.
2. In the **Record** window, right-click a field and click **Properties**.
The **Field Properties** dialog box appears.
3. Based on the data source type of the data map and the datatype of the field, view or edit field properties on some or all tabs:
 - **Name.** The name for the field.
 - **Array.** Properties to define the field as an array.
 - **Extra Properties.** Read-only. The extended metadata properties for the field.
 - **Field Checking.** For CHAR, CLOB, NUMCHAR, PACKED, SPACKED, UNPACKED, UZONED, VARCHAR, and ZONED datatypes, the type of data checking that PowerExchange completes at the field level, which overrides the data checking behavior defined for the data map.
 - **Code Page.** For CHAR and NUMCHAR datatypes for some data source types, the code page for the field, which overrides the code page defined for the data map.
 - **Variable Length.** For VARBIN and VARCHAR datatypes, properties for a variable-length field.

RELATED TOPICS:

- [“Field Properties - Name ” on page 354](#)
- [“Field Properties - Array” on page 360](#)
- [“Field Properties - Variable Length” on page 363](#)
- [“Field Properties - Extra Properties ” on page 362](#)
- [“Field Properties - Field Checking ” on page 362](#)
- [“Field Properties - Code Page” on page 363](#)

Viewing or Editing Properties for an Element in a Datacom Record

You can view or edit properties for an element in a record in a Datacom data map.

1. Open the Datacom data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
The **Elements** tab in the **Record Properties** dialog box displays properties for defined elements.

3. To edit properties for an element, select the element in the **Elements** list and click the **Properties** toolbar button.
4. In the **New Element** dialog box, view or edit the element properties.

RELATED TOPICS:

- [“Adding an Element to a Datacom Record” on page 67](#)
- [“Datacom Data Map Items” on page 61](#)

Viewing or Editing Properties for a Key in a Datacom Record

You can view or edit properties for a key in a Datacom record.

To view or edit properties for a key in a Datacom record:

1. Open the Datacom data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
3. On the **Keys** tab in the **Record Properties** dialog box, select the key in the **Keys** list and click the **Properties** toolbar button.

The **Key Details** tab in the **Key Properties** dialog box displays properties for the key.

4. On the **Key Details** tab, view or edit the key properties.

RELATED TOPICS:

- [“Adding a Key to a Datacom Record” on page 68](#)
- [“Datacom Data Map Items” on page 61](#)

Viewing or Editing Properties for a Key Segment in a Datacom Record

You can view or edit properties for a key segment in a Datacom record.

To view or edit properties for a key segment in a Datacom record:

1. Open the Datacom data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
3. On the **Keys** tab in the **Record Properties** dialog box, select a key in the **Keys** list and click the **Properties** toolbar button.

The **Segment Details** tab in the **Key Properties** dialog box displays properties for defined key segments.

4. To edit properties for the key segment, select it in the **Segment Details** list and click the **Properties** toolbar button.
5. In the **New Segment** dialog box, view or edit the key segment properties.

RELATED TOPICS:

- [“Adding a Key Segment to a Datacom Record” on page 69](#)
- [“Datacom Data Map Items” on page 61](#)

Viewing or Editing Properties for a CALC Element in an IDMS Record

You can view or edit properties for a CALC element in an IDMS record.

To view or edit properties for a CALC element in an IDMS record:

1. Open the IDMS data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
3. The **CALC Element Names** tab in the **Record Properties** dialog box displays properties for defined CALC elements.
4. To edit properties for a CALC element, select it in the **CALC Elements** list and click the **Properties** toolbar button.
5. In the **New CALC Element** dialog box, view or edit the CALC element properties.

RELATED TOPICS:

- [“Adding a CALC Element to an IDMS Record” on page 70](#)
- [“IDMS Data Map Items” on page 61](#)

Viewing or Editing Properties for an Owner Record and Set in an IDMS Record

You can view or edit properties for an owner records and set in an IDMS record.

To view or edit properties for an owner record and set in an IDMS record:

1. Open the IDMS data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.

The **Owner Record and Set Properties** dialog box appears.

The following table describes the information that the **Owner Record and Set** tab in the **Owner Record and Set Properties** dialog box displays for any defined owner record and sets:

Option	Description
Owner Name	Read-only. The name of the parent record. The name SYSTEM, which is a reserved name, identifies the set as a system-owned index.
Set Name	Specify the relationship between the record and owner.

3. To edit properties for an owner record and set, select the owner record and set in the **Owner Record and Set Names** list and click the **Properties** toolbar button.

The **Owner Record and Set Properties** dialog box appears.

4. On the **Owner Record and Set Details** tab, view or edit the owner record and set properties.

RELATED TOPICS:

- [“Adding an Owner Record and Set to an IDMS Record” on page 71](#)
- [“IDMS Data Map Items” on page 61](#)

Viewing or Editing Properties for a CCK Field in an IMS Record

You can view or edit properties for a CCK field in a record in an IMS DL/1 batch or ODBA data map.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record and click **Properties**.
The **CCK Fields** tab in the **Record Properties** dialog box displays properties for defined CCK fields.
3. To edit properties for a CCK field, select it in the **CCK Fields** list and click the **Properties** toolbar button.
4. In the **New CCK Field** dialog box, view or edit the CCK field properties.

RELATED TOPICS:

- [“Adding a CCK Field to an IMS Record” on page 73](#)
- [“IMS Data Map Items” on page 62](#)

Viewing or Editing Properties for a Search Field in an IMS Record

You can view or edit properties for a search field in a record in an IMS DL/1 batch or IMS ODBA data map.

1. Open the data map.
Note: If message PWX-01574 is displayed, the DBD that you imported for the data map supplied the key type value of "S" for a record search field. The "S" key type, for sequence fields, is deprecated. Edit record properties to change the **Key** value for the search field to **U**.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the record for which you want to view or edit search fields and click **Properties**.
3. In **Record Properties** dialog box, click the **Search Fields** tab.
4. To edit properties for a search field, click in the column cell that you want to edit and enter another value.
Alternatively, select the field in the list and click the Properties button. Then, edit the search field properties in a separate dialog box.

RELATED TOPICS:

- [“Adding Search Fields to an IMS Record” on page 74](#)
- [“IMS Data Map Items” on page 62](#)

Viewing or Editing Properties for a Data Map Table

You can view or edit properties for a table in a data map.

You might change a table name or define a hierarchy in a complex table. Also, you can define how records in a complex table appear in a database row test or are mapped in a target table.

To view or edit properties for a data map table:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click a table and click **Properties**.
The **Table Properties** dialog box appears.

3. Depending on the data source for the data map, view or edit table properties on some or all tabs:
 - **Definition.** Properties for a complex table, including settings for how records in a complex table appear in a database row test or are mapped in a target table, and how changes in records on which the table is based are reflected in the corresponding columns in the table.
 - **Keys.** For Adabas, the primary key for the table.
 - **IMS Options.** For IMS DL/1 batch and IMS ODBA, properties for segments in a complex table.

RELATED TOPICS:

- [“Adding a Data Map” on page 62](#)
- [“Adding an Element to a Datacom Record” on page 67](#)
- [“Adding a Key to a Datacom Record” on page 68](#)
- [“Adding a Key Segment to a Datacom Record” on page 69](#)
- [“Adding a CALC Element to an IDMS Record” on page 70](#)
- [“Adding an Owner Record and Set to an IDMS Record” on page 71](#)
- [“Adding a CCK Field to an IMS Record” on page 73](#)
- [“Adding Search Fields to an IMS Record” on page 74](#)
- [“Data Map Properties” on page 328](#)
- [“Table Properties - Definition” on page 364](#)
- [“Table Properties - Keys ” on page 368](#)
- [“Table Properties - IMS Options ” on page 367](#)

Viewing or Editing Properties for a Column in a Table

You can view or edit properties for a column in a table in a data map.

You might change a column name or the field on which the column is based.

1. Open the data map and the table.
2. In the **Table** window, right-click a column and click **Properties**.
The **Column Properties** dialog box appears.
3. On the **Name** tab, view or edit the column properties.

RELATED TOPICS:

- [“Adding a Column to a Table” on page 76](#)

Copying, Cutting, and Pasting Items in a Data Map

You can copy or cut and paste records, fields, tables, and columns in the same data map.

For example, if you copy a table to a data map, PowerExchange inserts the copied table at the selected position in the data map and appends an underscore and a numeric value to its name. If you copy a table named EMPLOYEE, PowerExchange inserts a copy of the EMPLOYEE table with the name EMPLOYEE_1 into the data map. If you copy another copy of the EMPLOYEE table, the second copy has the name EMPLOYEE_2.

To copy an item to a data map:

1. Open the data map.
2. Complete one of the following actions:
 - To copy or cut a record or a table, right-click the record or table and click **Copy** or **Cut**.
 - To copy or cut a field in a record, open the record. In the **Record** window, right-click the field and click **Copy** or **Cut**.
 - To copy or cut a column in a table, open the table. In the **Table** window, right-click the column and click **Copy** or **Cut**.
3. Right-click the item before or after which you want to paste the copied or cut item and click **Paste**.
4. In the **Paste Position** dialog box, select the option to paste the copied or cut item either before or after the selected item.

The copied or cut item appears before or after the selected item.

Editing Logon Information for Accessing Data Maps and Data

You can edit the user credentials for logging in to a remote system to access a data map or data.

1. Open the data map.
2. Click **Resource > Logons**.
3. In the **Data Map Logons** dialog box, enter credentials for logging in to a remote system to access a data map and data.

The following table describes the information to enter:

Option	Description
Location	<p>Select the location of the data source. Select local if the data source is on the PowerExchange Navigator system.</p> <p>This field lists the locations that are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.</p> <p>Default is local.</p>
Data Map Logon	<p>Enter the following credentials for accessing the data map:</p> <ul style="list-style-type: none"> - UserID. Enter the database user ID to access the data source, if required. For databases on supported Linux, UNIX, or Windows systems, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user name is the enterprise user name. For more information, see the <i>PowerExchange Reference Manual</i>. - Password. Enter a password for the user ID, if specified. For access to data maps on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters: <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p> - File Password. For Adabas, if the FDT file is password-protected, enter the password to access the file.

Option	Description
Data Access Logon	<p>Enter the following credentials for accessing the data source:</p> <ul style="list-style-type: none"> - UserID. For z/OS, enter the database user ID to access the data source, if required. - Password. For z/OS, required. Enter a password for the user ID. <p>For access to data on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:</p> <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>The allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: A valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p> <p>Note: You must enter credentials for both the Data Map Logon and Data Access Logon, even if the logon user IDs and passwords are the same. On z/OS, different users can own the data maps and actual data.</p>
Allow logons to be changed next time they are required	<p>Select this option to prompt for credentials again before proceeding with the next requested function.</p>

Sending a Data Map to a Remote System

If the data source for a data map is on a remote system, send the data map to the remote system so that it is available to PowerExchange for bulk data movement operations.

When you send the data map to the remote system, PowerExchange converts the data map into a platform-independent file to use for bulk data extraction. The original data map remains on the PowerExchange Navigator system. If you change the local PowerExchange Navigator data map, you must send the data map to the remote system again to update the data map on that system.

1. Open the data map.
2. Click **File > Send to Remote Node**.
3. In the **Data Map Remote Node** dialog box, enter connection information for the remote system to which to send the data map.

The following table describes the connection information to enter:

Option	Description
User ID	<p>If a user ID is required to access the remote system, enter a user ID.</p> <p>For supported Linux, UNIX, or Windows systems, if you have enabled PowerExchange LDAP user authentication, the user name is the enterprise user name. For more information, see the <i>PowerExchange Reference Manual</i>.</p>
Password	<p>If you specified a user ID, enter a password for that user ID.</p> <p>For access to an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:</p> <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p>
Location	<p>Select the node name of the location of the system to which to send the data map.</p> <p>This field lists the locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.</p> <p>Default is local.</p>
Save User ID and Password(s) for session	<p>Select this option to save the information that you entered for the duration of the session.</p> <p>Default is selected.</p>

4. Click **OK**.

A message box displays the status of the send operation.

5. Click **OK**.

PowerExchange saves the data map on the remote system.

RELATED TOPICS:

- [“Exported Data Maps” on page 60](#)

Importing Data Maps Sent to a Remote System

If you sent, or exported, data maps to a remote system, you can import one or more of those data maps to the PowerExchange Navigator system.

For each data map that you import, the data map remains on the remote system and a local copy is saved on the PowerExchange Navigator system.

1. Close any open PowerExchange resources.
2. Click **File > Exported Data Maps**.
3. In the **Data Map Import Node** dialog box, enter connection information for the remote system from which you are importing data maps.

The following table describes the connection information to enter:

Option	Description
User ID	<p>If a user ID is required to access the remote system, enter a user ID.</p> <p>For supported remote Linux, UNIX, or Windows systems, if you have enabled PowerExchange LDAP user authentication, the user name is the enterprise user name. For more information, see the <i>PowerExchange Reference Manual</i>.</p>
Password	<p>If you specify a user ID, enter a password for that user ID.</p> <p>For access to an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:</p> <ul style="list-style-type: none">- Uppercase and lowercase letters- The numbers 0 to 9- Spaces- The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p>

Option	Description
Location	Select the node name of the remote system from which you are importing the data map. This field lists the locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is local .
File Password	For Adabas, if the FDT file is password-protected, enter a password that allows access to the file.
Save User ID and Password(s) for session	Select this option to save the information you entered for the duration of the session. Default is selected.

4. In the **Exported Data Maps** dialog box, import one or more of the data maps. A message box displays the status of the import operation.
5. Click **OK**.
The data map or maps appear on the **Resources** tab in the **Resource Explorer**.

RELATED TOPICS:

- [“Exported Data Maps” on page 60](#)

Associating Multiple Data Files with a Data Map

You can associate multiple data files with a data map by using PowerExchange file-list processing.

Instead of associating a single data file with a data map, create a file-list file. In the file-list file, you list multiple data files. Then, associate the file-list file with the data map and select the **File List Processing** option in the data map properties. PowerExchange concatenates and processes the data in the data files in the order in which the data files are listed in the file-list file.

File-list processing is supported for flat files, sequential data sets, ESDSs, and tape data sets.

If you view the data file associated with a data map that uses file-list processing, the file-list file appears rather than the listed data files.

RELATED TOPICS:

- [“Data Files and File-List Processing” on page 52](#)

PowerExchange File-List Processing

When you run a database row test or a bulk data movement operation using a data map configured to use file-list processing, PowerExchange processes the file-list file as follows:

1. PowerExchange ignores any blank lines and any leading and trailing blanks on a line.
2. If PowerExchange encounters an error in the file-list file, file-list processing ends. Errors might include a file name that is not valid, nonexistent files, or read errors.
3. PowerExchange logs any file names that are not valid in the PowerExchange Listener log. File names are logged in hexadecimal format.

4. If PowerExchange does not detect any errors, PowerExchange concatenates and processes the data in the data files that are listed in a file-list file in the order in which they are listed.

File-List File Syntax Rules

You must follow syntax rules when you create a file-list file.

Use the following syntax rules when creating a file-list file:

- List one data file per line.
- Begin comment lines with forward slash and asterisk, as follows:
/*
- Any characters specified between leading and trailing blanks on a line are considered a data file name. On z/OS, if you set sequence numbers to ON, the sequence number is also considered part of the data file name.

Creating a File-List File and Associating It with a Data Map

To associate multiple data files with a data map, create a file-list file that lists the data files and associate the file-list file with a data map.

Note: To use file-list file processing with ODBC, set SQL escape sequences instead of defining data map properties.

1. Create a file-list file on the system where the data files are located. The file-list file lists the path and file name of each data file.
2. In the PowerExchange Navigator, edit a data map.
Note: File-list processing is supported for data maps defined with the ESDS, SEQ, and TAPE access methods only.
3. On the **Access Method** tab in the **Data Map Properties** dialog box, enter the path and file name of the file-list file in the **File Name** box, or click the Browse button to browse to the file.
4. Select the **File List Processing** option.
5. Click **OK**.

ODBC and File-List Processing

You can use file-list processing with ODBC.

To use file-list processing with ODBC, use the following SQL escape sequences:

DTLDSN

Specifies the file-list file.

DTLFILELIST

Indicates that file-list processing is to be used.

For more information about ODBC, file-list processing, and SQL escape sequences, see the *PowerExchange Reference Manual*.

File-List File Example for Linux, UNIX, and Windows

On Linux, UNIX, or Windows, you can define a file-list file that lists data files.

You might list the following data files in a file-list file named `filelist.dat`:

```
c:\pwx\data\filelist1.dat
c:\pwx\data\filelist2.dat
c:\pwx\data\filelist3.dat
```

When you add a data map, associate the `filelist.dat` file with the data map and select the **File List Processing** option in the data map properties.

File-List File Example for z/OS

On z/OS, you can define a file-list member that lists data file members.

You might list the following data file members in a file-list member named `DTLUSR.V901.V1.DEMO.ESDS`:

```
***** Top of Data *****
000001 /* FILE LIST PROCESSING DEFINITION FILE
000002 /*
000003 DTLUSR.V901.V1.DEMO.ESDS1
000004 /*
000005 DTLUSR.V901.V1.DEMO.ESDS2
000006 /*
000007 DTLUSR.V901.V1.DEMO.ESDS3
000008 /*
***** Bottom of Data *****
```

When you add a data map, associate the `DTLUSR.V900.V1.DEMO.ESDS` member with the data map and select the **File List Processing** option in the data map properties.

Printing a Data Map or Items in a Data Map

You can print a data map, or records, fields, tables, or columns in a data map, to a printer or to a file.

If you choose to print to a file, the default file name is `PowerExchange Navigator.xxx` in the My Documents folder, where `xxx` is a file extension appropriate for the printer type.

Note: You cannot print user-defined fields.

In a printed document or saved file, you can choose the following information:

- Include detailed information in addition to summary information for an item.
- Identify the hierarchical level of an item by using numbers or indentation.
- Include information for any dependent objects.

Printing a Data Map

You can print one data map at a time.

To print a data map:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the data map and click **Print**.

- In the **Print** dialog box, view or edit print options.

The following table describes the print options:

Option	Description
Name	Select the printer. Click Properties to set the properties for the printer.
Status	Read-only. The status of the selected printer.
Type	Read-only. The printer type of the selected printer.
Where	Read-only. The location of the selected printer.
Comment	Read-only. Any comments defined for the selected printer.
Print to file	Select this option to save the item in plain text format to a file. Select a printer type that is appropriate for the print to file option.
All	Select this option to print all pages in the document. Default is selected, which is the only valid option.
Pages	Not available.
Selection	Not available.
Number of copies	Enter the number of copies to print on a printer. This value does not apply to items printed to a file. Default is 1.
Print details	Select this option to print detailed information, in addition to summary information, for the selected item or items. Default is to print summary information.
Print dependents	Select this option to print summary information, and, if selected, detailed information, for any dependent objects of the selected item or items. Default is to not print information for dependent objects.
Use level number	Select this option to print level numbers that identify any hierarchy in the selected item or items. Default is to use indentation.

- Click **OK**.

PowerExchange prints the data map on the selected printer unless you selected the **Print to file** option.

- If you printed the data map to a file, enter the file properties in the **Save As** dialog box, and click **Save**.
PowerExchange saves the document to the specified file in the specified location.

Printing Data Map Records

You can print one or more records in a data map.

To print data map records:

- Open the data map.

2. On the **Data Map** tab in the **Resource Explorer**, print one or more records.
3. In the **Print** dialog box, define the print options. For information about print options, see [“Printing a Data Map” on page 95](#).
PowerExchange prints the records on the selected printer unless you selected the **Print to file** option.
4. If you printed the records to a file, enter the file properties in the **Save As** dialog box, and click **Save**.
PowerExchange saves the document to the specified file in the specified location.

Printing Fields in a Record

You can print one or more fields in a record in a data map.

Note: You cannot print user-defined fields.

To print fields in a record:

1. Open the data map.
2. Open the record.
3. On the **Fields** tab in the **Record** window, print one or more fields.
4. In the **Print** dialog box, define the print options. For information about print options, see [“Printing a Data Map” on page 95](#).
PowerExchange prints the fields on the selected printer unless you selected the **Print to file** option.
5. If you printed the fields to a file, enter the file properties in the **Save As** dialog box, and click **Save**.
PowerExchange saves the document to the specified file in the specified location.

Printing Data Map Tables

You can print one or more tables in a data map.

To print data map tables:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, print one or more tables.
3. In the **Print** dialog box, define the print options. For information about print options, see [“Printing a Data Map” on page 95](#).
PowerExchange prints the tables on the selected printer unless you selected the **Print to file** option.
4. If you printed the tables to a file, enter the file properties in the **Save As** dialog box, and click **Save**.
PowerExchange saves the document to the specified file in the specified location.

Printing Columns in a Table

You can print one or more columns in a table in a data map.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the table.
3. In the **Table** window, print one or more columns.
4. In the **Print** dialog box, define the print options. For information about print options, see [“Printing a Data Map” on page 95](#).
PowerExchange prints the columns on the selected printer unless you selected the **Print to file** option.

5. If you printed the columns to a file, enter the file properties in the **Save As** dialog box, and click **Save**. PowerExchange saves the document to the specified file in the specified location.

Deleting Data Maps or Items in a Data Map

You can delete data maps or items in a data map.

You can delete multiple data maps or data map items at a time.

You can delete the following data map items:

- Records
- Fields from a record
- Tables
- Columns from a table
- For Datacom data maps, elements, keys, and key segments
- For IDMS data maps, CALC elements, and owner records and sets
- For IMS data maps, CCK fields and search fields

When you delete data maps or data map items, PowerExchange deletes the data map or data map items from the current resource configuration directory on the PowerExchange Navigator system.

Deleting Data Maps

You can delete data maps.

To delete data maps:

1. In the **Resources** window, delete one or more data maps.
2. In the message box that prompts you to confirm the deletion, click **Yes**.

PowerExchange deletes the data maps from the current resource configuration directory on the PowerExchange Navigator system.

Deleting Data Maps Exported to a Remote System

You can delete data maps that you sent, or exported, to a remote system.

To delete data maps exported to a remote system:

1. Close any open PowerExchange resources.
2. Click **File > Exported Data Maps**.
3. In the **Data Map Import Node** dialog box, enter connection information for the system on which you want to delete exported data maps.
4. In the **Exported Data Maps** dialog box, delete one or more exported data maps.
5. In the message box that prompts you to confirm the deletion, click **Yes**.

PowerExchange deletes the data maps on the remote system.

RELATED TOPICS:

- [“Sending a Data Map to a Remote System” on page 90](#)

Deleting Data Map Records

You can delete records from a data map.

To delete data map records:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, delete one or more records.
A message box prompts you to confirm the deletion of the record or records.
3. If any cross-references exist to the record or records, the message box notifies you that the deletion of the record also deletes all cross-references to the record or records. To view the cross-references to the record or records, click **More Details**. In the **References affected by current delete** list, the icons for the items that PowerExchange will delete when it deletes the record or records appear in gray.
4. In the message box that prompts you to confirm the deletion, click **Yes**.
PowerExchange deletes the records and any cross-references to the records from the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting Fields from a Record

You can delete fields from a record.

To delete fields from a record:

1. Open the data map and the record.
2. On either the **Fields** tab or **Expr(n)** tab in the **Record** window, delete one or more records.
A message box prompts you to confirm the deletion of the field or fields. If any tables in the data map contain columns that are based on the field or fields, the message box notifies you that the deletion of the field or fields also deletes any columns that cross-reference the field or fields.
3. To view which columns in which tables cross-reference the field or fields, click **More Details**. In the **References affected by current delete** list, the icons for the items that PowerExchange will delete when it deletes the field or fields appear in gray.
4. In the message box that prompts you to confirm the deletion, click **Yes**.
PowerExchange deletes the field or fields from the record. If you delete all fields from a record, PowerExchange does not delete the record.

If any columns in tables were based on the field or fields, PowerExchange deletes those columns from the tables. If you delete all fields from a record on which all columns in a table are based, PowerExchange deletes all the columns in the table but does not delete the table.

Deleting Data Map Tables

You can delete tables from a data map.

To delete data map tables:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, delete one or more tables.

3. In the message box that prompts you to confirm the deletion, click **Yes**.
PowerExchange deletes the tables from the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting Columns from a Table

You can delete columns from a table.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the table.
3. In the **Table** window, delete one or more columns.

A message box prompts you to confirm the deletion of the column or columns.

4. In the message box that prompts you to confirm the deletion, click **Yes**.

PowerExchange deletes the columns from the table and any cross-references to the columns from the data map in the current resource configuration directory on the PowerExchange Navigator system.

If you delete all columns from a table, PowerExchange does not delete the table.

Deleting Elements from a Datacom Record

You can delete elements from a Datacom record.

To delete elements from a Datacom record:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **Elements** tab in the **Record Properties** dialog box, delete one or more elements.

PowerExchange deletes the elements from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting Keys from a Datacom Record

You can delete keys from a Datacom record.

To delete keys from a Datacom record:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **Keys** tab in the **Record Properties** dialog box, delete one or more keys.

PowerExchange deletes the keys from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting Key Segments from a Datacom Record

You can delete key segments from a Datacom record.

To delete key segments from a Datacom record:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **Keys** tab in the **Record Properties** dialog box, click the **Properties** button on the dialog box.

4. On the **Segment Details** tab in the **Key Properties** dialog box, delete one or more key segments.
PowerExchange deletes the key segments from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting CALC Elements from an IDMS Record

You can delete CALC elements from an IDMS record.

To delete CALC elements from an IDMS record:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **CALC Element Names** tab in the **Record Properties** dialog box, delete one or more CALC elements.

PowerExchange deletes the CALC elements from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting Owner Records and Sets from an IDMS Record

You can delete owner records and sets from an IDMS record.

To delete owner records and sets from an IDMS record:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **Owner Records and Set Names** tab in the **Record Properties** dialog box, delete one or more owner records and sets.

PowerExchange deletes the owner records and sets from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting CCK Fields from an IMS Record

You can delete CCK fields from a record in an IMS DL/1 batch or IMS ODBA data map.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **CCK Fields** tab in the **Record Properties** dialog box, delete one or more CCK fields.

PowerExchange deletes the CCK fields from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

Deleting Search Fields from an IMS Record

You can delete search fields from a record in an IMS DL/1 batch or IMS ODBA data map.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, click the right-click the record and click **Properties**.
3. On the **Search Fields** tab in the **Record Properties** dialog box, delete one or more search fields.

PowerExchange deletes the search fields from the record in the data map in the current resource configuration directory on the PowerExchange Navigator system.

CHAPTER 3

Data Maps for Specific Data Sources

This chapter includes the following topics:

- [Data Maps for Specific Data Sources Overview, 102](#)
- [Adabas Data Maps, 103](#)
- [Datacom Data Maps, 105](#)
- [DB2 Catalog Data Maps, 107](#)
- [DB2 Unload File Data Maps, 109](#)
- [IDMS Data Maps, 112](#)
- [IMS Data Maps, 115](#)
- [SEQ Data Maps, 126](#)
- [VSAM Data Maps, 127](#)

Data Maps for Specific Data Sources Overview

The options you configure when you add a data map depend on the data source type of the data map.

Also, the type of copybook that you can import into the data map depend on the data source type for the data map.

The following table lists the topics that provide details for adding data maps and importing copybooks into data maps for each data source:

Data Source	PowerExchange Access Method	Reference
Adabas file	ADABAS	"Adabas Data Maps" on page 103
CA Datacom/DB file	DATAKOM	"Datacom Data Maps" on page 105
CA IDMS/DB database	IDMS	"IDMS Data Maps" on page 112
DB2 catalog	DB2	"DB2 Catalog Data Maps" on page 107
DB2 unload file	DB2UNLD	"DB2 Unload File Data Maps" on page 109

Data Source	PowerExchange Access Method	Reference
IMS database	DL/1 BATCH	"IMS Data Maps" on page 115
IMS database accessed through a PowerExchange Listener	IMS ODBA	"IMS Data Maps" on page 115
Sequential data set or flat file	SEQ	"SEQ Data Maps" on page 126
User-defined user access method program	USER	Appendix B, "User Access Method Programs" on page 317
VSAM entry-sequenced data set (ESDS)	ESDS	"VSAM Data Maps" on page 127
VSAM key-sequenced data set (KSDS)	KSDS	"VSAM Data Maps" on page 127
VSAM relative record data set (RRDS)	RRDS	"VSAM Data Maps" on page 127

RELATED TOPICS:

- ["Data Maps" on page 49](#)
- ["Copybooks" on page 133](#)
- ["Database Row Test" on page 212](#)

Adabas Data Maps

You can import a copybook, an Adabas field definition table (FDT), and Adabas keys when you add an Adabas data map or after you add it.

The FDT defines the layout of the data in a data source. When you import an FDT, the data definitions in the data map are stored on the local system. If you import Adabas keys into the data map, you can add these keys to tables in the data map.

You can refresh the FDT after you add the data map.

You can import any of the following copybook types into an Adabas data map:

- COBOL copybook. If you import a COBOL copybook, you must edit the data map to associate the imported field definitions with Adabas field definitions.
- PL/I copybook. If you import a PL/I copybook, you must edit the data map to associate the imported field definitions with Adabas field definitions.
- File created by using the ADACMP utility.
- Natural Data Definition Module (DDM). To import a DDM, you must enter the FDIC database ID and file number values.

Note: If you import DDM metadata for Adabas databases on z/OS that contain wide character datatypes, PowerExchange does not select the **Wide Char** option on the **Code Page** tab for these fields. You must either import Predict or FDT metadata, or edit field properties in the data map to select the Wide Char option for all wide-character datatypes.

- Adabas field definition table (FDT) that contains the database definition, which provides short names and no scale information. An FDT enables validation of Adabas-related information such as field names, array sizes, and override lengths. After you import an FDT, it is cached for subsequent sessions, which enables certain validation checks to occur prior to run time. You can refresh an imported FDT.
- Predict data dictionary, which provides long name and scale information. To import a Predict data dictionary, you must enter the FDIC database ID and file number values.
- Text file, which uses a delimited Informatica internal format.

You can subdivide, or redefine, any PowerExchange field mapped to an Adabas field. To subdivide a field, define it as a group field with subfields that remap the group.

To test an Adabas data map, run a database row test.

Adding an Adabas Data Map

When you add an Adabas data map, you can import an Adabas FDT and a copybook.

To add an Adabas data map:

1. On the **Resources** tab in the **Resource Explorer**, right-click the **Data Maps** folder and select **Add Data Map**.

The wizard for adding a data map starts.

2. On the **Name** page, enter the following information:

- Enter a schema name.
- Select **ADABAS** as the access method.
- Enter a name for the data map.
- Select the **Import Record Definitions** and **Import Key Fields/FDT** check boxes.

Then click **Next**.

3. On the **Adabas Access Method** page, enter data source information and click **Finish**.
4. In the **Adabas FDT Import** dialog box, select the node name for the Adabas source system and click **OK**.
If the source system is z/OS and PowerExchange is configured for security, you must also enter a valid z/OS user ID and password. If you use Adabas file security, enter the file password.
5. On the **Import Copybook - Source Details** page, select one of the following copybook types and enter any additional information that is required for the selected type:

- **ADACMP**
- **COBOL**
- **DDM**. Also enter the FDIC database ID and file number values.
- **FDT**
- **PL/1**
- **PREDICT**. Also enter the FDIC database ID and file number values.
- **TEXT**. Available only if you selected **Local** as the source.

Default is **DDM**.

Click **Next**.

6. On the **Import Copybook - Remote Details** page, enter information for the remote copybook.
If you selected the **DDM** or **PREDICT** source type, enter a name in the **Name** box or click the **Browse** button to browse to the file.

Click **Next**.

7. On the **Import Copybook - Configuration Details** page, select the actions for imported records, fields, and tables, and click **Next**.

For more information, see [“Importing a Copybook into a Data Map” on page 135](#).

8. In the **Import Copybook Information** window, review the information for the import operation and click **OK**.

9. In the **Record Definition** dialog box, select an action for each imported record and click **OK**.

For more information, see [“Importing a Copybook into a Data Map” on page 135](#).

The **Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.

RELATED TOPICS:

- [“Importing an FDT into an Adabas Data Map” on page 139](#)
- [“Data Map Properties - ADABAS Access Method” on page 328](#)
- [“Import Copybook - Source Details Dialog Box” on page 140](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)

Adding a Key to a Table in an Adabas Data Map

If you imported Adabas keys into an Adabas data map, you can define a key for each table in the data map.

To add a key to a table in an Adabas data map:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the table and click **Properties**.
3. On the **Table Properties** dialog box, click the **Keys** tab.
4. In the **Primary Key** list, select a key for the table.

The selected key appears in the **Key Details** list.

The validity of a key is determined by checking that the table contains all the fields that are defined in the record as having the Adabas field names needed by the descriptor. If any of these fields are mapped into groups, all members of the group must be present. Fields in vertical arrays are not valid.

5. Click **OK**.

Datacom Data Maps

You can import a copybook when you add a CA Datacom data map or after you add it.

You can import a COBOL, DATACOM, or PL/I copybook into a Datacom data map.

After you add a Datacom data map, you can add elements, keys, and key segments to the data map.

To test a Datacom data map, run a database row test.

RELATED TOPICS:

- [“Datacom Data Map Items” on page 61](#)

Adding a Datacom Data Map

When you add a Datacom data map, you can import a copybook.

To add a Datacom data map:

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. Enter the following information in the **Name** dialog box:
 - Select the **DATACOM** access method.
 - Select the **Import Record Definitions** option.
3. Click **Next**.
4. In the **DATACOM Access Method** dialog box, enter information for the access method and click **Finish**.
5. In the **Import Copybook - Source Details** dialog box, select **Remote**.

Select one of the following copybooks types:

- **COBOL**
- **DATACOM**
- **PL/1**

Default is **COBOL**.

6. Click **Next**.
7. In the **Import Copybook - Remote Details** dialog box for the copybook type, enter connection information.

For the **DATACOM** copybook type, enter two user IDs and passwords, as required by the configuration of PowerExchange and CA Datacom/DB database. The first user ID and password are required if security is enabled by using the SECURITY statement in the DBMOVE configuration file. The **Dictionary UserID** and **Dictionary Password** values are required if CA Datacom/DB requires them.

To import a copybook with the metadata, select **Copybook** and select **COBOL** or **PL/1**.

The **Table Status**, **Key**, and **Elements** options are disabled by default.

8. Click **Next**.
9. In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records, fields, and tables and click **Next**. For more information, see [“Adding an Adabas Data Map” on page 104](#).
10. In the **Import Copybook Information** window, review the information for the import and click **OK**.
11. In the **Record Definition** dialog box, select an action for each imported record and click **OK**. For more information, see [“Importing a Copybook into a Data Map” on page 135](#).

The **Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.

RELATED TOPICS:

- [“Adding an Element to a Datacom Record” on page 67](#)
- [“Adding a Key to a Datacom Record” on page 68](#)

- [“Adding a Key Segment to a Datacom Record” on page 69](#)
- [“Data Map Properties - DATACOM Access Method” on page 330](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)
- [“Import Copybook - Remote Datacom Details Dialog Box” on page 146](#)

DB2 Catalog Data Maps

Because DB2 is a relational database, a DB2 data map is not required for PowerExchange to access DB2 tables.

However, in certain cases, defining a data map allows you to process the data in a way that is not otherwise possible. For more information, see [“Use Cases for DB2 Data Maps” on page 107](#).

You can add data maps for the following DB2 data sources:

- DB2 for i5/OS
- DB2 for Linux, UNIX, and Windows
- DB2 for z/OS

To test a DB2 data map, run a database row test.

Note: Do not attempt to create capture registrations with BI and CI columns for DB2 data maps that include merged expression columns. If you attempt to register an expression column with BI or CI columns, the BI column reflects the after-image data, rather than the before-image data. The CI column can also contain incorrect values.

Use Cases for DB2 Data Maps

The following use cases illustrate some of the ways that you can use DB2 data maps.

Packed Decimal Data Stored in Character Columns

Some DB2 tables use a single DB2 column to store an array of fields in a format that might not be consistent with the column type. For example, a VARCHAR column might contain multiple packed decimal fields.

To solve this problem, you can create a data map by importing record definitions, change the problem VARCHAR column to a GROUP field, and add an array of child packed decimal fields. The numeric data can then be columnized and loaded into numeric columns on the target.

Code Page Dependent on Values Within the Data

Your company amalgamates data from multiple international sites into a single DB2 subsystem. Although the DB2 catalog indicates that the data is stored in a single code page, in reality the data is stored in multiple code pages, with the code page being dependent on values within the data. DB2 conversion is turned off because the updating applications set the client and server CCSIDs to the same value.

To manage code pages correctly, you can create a data map by importing record definitions. Then add a record ID condition and set the code page as required. You can set the code page either once in the properties of the data map or multiple times, once for each character field. Repeat the process of adding a record ID condition and setting the code page for each type of code page.

Expansion Routine Called Through a User-Defined Function

Instead of using the DB2 compressed table space facility, your company stores data using an in-house compression and encryption routine.

To solve the problem of expanding and decrypting the data, you can create a data map by importing record definitions, add user-defined functions to expand the problem DB2 columns, and include the user-defined columns in the PowerCenter mapping.

Character Data that Contains Hexadecimal '0' Values

PowerCenter workflows truncate character data that they read from a DB2 relational source at the first hexadecimal '0' value. In certain cases you want the workflow to preserve hexadecimal '0' values and to write the entire column string to the target.

To accomplish this goal, you can define a DB2 data map by importing record definitions. Then perform the following actions:

- Specify `PreserveLowValues=Yes` in the **Custom Properties** field of the **Config Object** tab in the Powercenter Workflow Manager.
- Specify `LOWVALUES=Y` in the DBMOVER configuration file on the Integration Service machine.

Adding a DB2 Catalog Data Map

When you add a DB2 catalog data map, you can import a copybook.

To add a DB2 catalog data map:

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. Enter the following information in the **Name** dialog box:
 - Select the **DB2** access method.
 - Select the **Import Record Definitions** option.
3. Click **Next**.
4. In the **DB2 Access Method** dialog box, enter information for the access method and click **Finish**.
5. In the **Import Copybook - Source Details** dialog box, select **Remote**.
In the **Type** list, select the **DB2 Catalog** copybook type.
6. Click **Next**.
7. In the **Import Copybook - Remote DB2 Catalog Details** dialog box, enter the following information:
 - Optionally, change the import details. The input fields are populated from previous selections.
 - In the **Save File Locally As** box, enter a file name.
 - To select specific columns, click **Preview/Amend Columns**.
8. **Click Next**.
9. In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records, fields, and tables and click **Next**. For more information, see ["Importing a Copybook into a Data Map" on page 135](#).
10. In the **Import Copybook Information** window, review the information for the import and click **OK**.
11. In the **Record Definition** dialog box, select an action for each imported record and click **OK**. For more information, see ["Importing a Copybook into a Data Map" on page 135](#).

The **DB2 Catalog Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.

RELATED TOPICS:

- [“Data Map Properties - DB2 Access Method” on page 330](#)
- [“Import Copybook - Remote DB2 Catalog Details Dialog Box” on page 148](#)

Editing the Record Layout in a DB2 Data Map

You can edit the record layout of a DB2 table in a DB2 data map.

For example, you can edit a field in a record that contains three items to separate the data in the field into child fields.

In this example, the MISC_DATA field is defined as a CHAR(15) field. However, the field is used to store three Packed decimal fields, each with a precision of 9 and scale of 2.

To edit the record layout in a DB2 data map:

1. Open the data map.
2. Open the record.
3. In the **Record** window, right-click the MISC_DATA field and click **Properties**.
The **Field Properties** dialog box appears.
4. At **Field Type**, enter `GROUP`.
5. Click **OK**.
6. In the **Record** window, right-click the MISC_DATA field and click **Add Field as Child**.
7. Add the MISC_1, MISC_2, and MISC_3 fields as children of the MISC_DATA field. Define each field as a **PACKED** field with a precision of 9 and a scale of 2.

DB2 Unload File Data Maps

In addition to direct data mapping of DB2 tables, you can also map DB2 unload files.

For DB2 unload files, PowerExchange supports the following IBM and BMC Software formats:

- DB2 for z/OS online REORG TABLESPACE utility with UNLOAD EXTERNAL
- DB2 for z/OS online UNLOAD utility
- DB2 for z/OS sample unload program DSNTIAUL
- BMC Software Unload Plus
- DB2 for Linux, UNIX, and Windows High Performance Unload utility

When you run a database row test to test a DB2 unload file data map, select the NRDB data source type.

Note: When you add a DB2 unload data map and select **DSNUTILB UNLOAD** for the **Unload Type**, PowerExchange includes a field for the OBID in the record. If you create DB2 unload files specifying **HEADER NONE** in the DB2 UNLOAD control statements, select **UNDEFINED** for the **Unload Type**.

Note: PowerExchange cannot read large format sequential data sets. If you specify **DSNTYPE=LARGE** in the DD statement when you create the unload file, PowerExchange cannot read the file. Instead, you can specify **DSNTYPE=EXTREQ** in the DD statement.

Adding a DB2 Unload File Data Map

When you add a DB2 unload file data map, you can import a copybook.

To add a DB2 unload file data map:

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. Enter the following information in the **Name** dialog box:
 - Select the **DB2UNLD** access method.
 - Select the **Import Record Definitions** option.
3. In the **DB2UNLD Access Method** dialog box, enter the following information:
 - In the **File Name** box, enter the name of the target DB2 unload file.
 - In the **DB Instance** box, enter the DB instance, such as DSN9.
 - In the **Table Name** box, enter the table name or click the **Browse** button to browse to the file. In the **DB2 Table Filter** dialog box, enter filter criteria based on the creator or schema name, database name, or table name. Click **Next**. Select a table and click **Finish**.
 - In the **Unload Type** list, select the unload type. For bulk access to unload files, the following IBM and BMC formats are supported: **.REORG UNLOAD**, **.DSNTIAUL/BMC UNLOAD+**, **.DSNUTILB UNLOAD**, and **.UNDEFINED**. By default, the **.UNDEFINED** type is generated with null indicators, and count fields in the same position as in the **REORG UNLOAD EXTERNAL** type. Instead of using these unload types, you can modify these statements to meet specific requirements. You can move or delete these statements to correspond to user-defined unload formats, but PowerExchange interprets them if they precede the data field to which they apply.
4. Click **Finish**.
5. In the **Import Copybook - Source Details** dialog box, select **Remote** and select the **DB2 Catalog** copybook type.
6. Click **Next**.
7. In the **Import Copybook - Remote DB2 Catalog Details** dialog box, enter the following information:
 - In the **Save File Locally As** box, enter a file name.
 - In the **Null Indicator** box, enter the hexadecimal value of the null indicator that is used in the unload file that you are mapping.
 - If the variables are padded in the unload file that you are mapping, select the **Pad Variable** option. This option corresponds to the **BMC UNLOAD PLUS FIXEDVARCHAR** option.
 - To select specific columns, click **Preview/Amend Columns**.
8. Click **Next**.
9. In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records, fields, and tables and click **Next**. For more information, see [“Importing a Copybook into a Data Map” on page 135](#).
10. In the **Import Copybook Information** window, review the information for the import and click **OK**.
11. In the **Record Definition** dialog box, select an action for each imported record and click **OK**. For more information, see [“Importing a Copybook into a Data Map” on page 135](#).

The **DB2 Catalog Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.

RELATED TOPICS:

- [“Data Map Properties - DB2UNLD Access Method” on page 331](#)
- [“Import Copybook - Remote DB2 Catalog Details Dialog Box” on page 148](#)

Accessing DB2 Unload Data from Multiple Tables

Multiple table tablespaces result in a single unload file that contains data from several tables.

You can create a data map that can access multiple tables in a single unload file.

You can select individual tables from the unload by using the OBID of the table. The following types of unload files are supported for this type of multiple table access:

- Type 1 REORG UNLOAD

The OBID is held in the fourth and fifth bytes of each record.

- Type 3 DSNUTILB UNLOAD

The OBID is held in the first two bytes of each record.

Perform the following tasks to create a data map that can be used to access multiple tables in the same unload file:

1. Add a data map.
2. Add the underlying tables.
3. To specify the underlying table for each record type in the data map, enter the appropriate OBID in the **Record Properties** dialog box for the DTL__OBID field.

Adding a DB2 Unload File Data Map - Example

This example demonstrates how to use a Type 3 DSNUTILB UNLOAD file. You can use this same procedure for a Type 1 REORG UNLOAD file.

The unload file contains data from the PWXTAB3 and PWXTAB4 tables, each with its own structure. For Type 3, the underlying table is specified by the OBID in the first two bytes. The OBIDs for the tables are x'0004' and x'0005' respectively.

To add DB2 unload file data map:

1. Add a data map to map the PWXTAB3 table.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the DTL_OBID field and click **Properties**.
3. In **Record ID Values** box, enter the OBID hex value of x'0004' and click **OK**.
The icon for the DTL_OBID field appears in green, indicating that the **Record ID** filter is set for the field.
4. Import the next underlying table, PWXTAB4.
5. In **Record ID Values** box, enter the OBID hex value of x'0005' and click **OK**.
The icon for the DTL_OBID field appears in green, indicating that the **Record ID** filter is set for the field.
6. To test the PWXTAB4 table, run a database row test.
The **Database Row Test Output** window displays two rows with an OBID of x'0005'.

IDMS Data Maps

You can import a COBOL, IDMS, or PL/I copybook when you add an IDMS data map or after you add it.

After you add an IDMS data map, you can add CALC elements, and owner records and sets.

Run a database row test to test an IDMS data map.

RELATED TOPICS:

- [“IDMS Data Map Items” on page 61](#)

Adding an IDMS Data Map

This example demonstrates how to add an IDMS data map and import a copybook into the data map.

PowerExchange downloads metadata from z/OS to create the data maps for CA IDMS/DB. When running with a security level of 0 or 1, data sets are created on z/OS prefixed with the user ID under which the PowerExchange Listener is running. If this user ID is not acceptable, include the TEMPHLQ statement in the DBMOVER configuration file.

To add an IDMS data map:

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. Enter the following information in the **Name** dialog box:
 - Select the **IDMS** access method.
 - Select the **Import Record Definitions** option.
3. Click **Next**.
4. In the **IDMS Access Method** dialog box, enter information for the access method and click **Finish**.
5. In the **Import Copybook - Source Details** dialog box, select **Remote** and select one of the following copybooks types:
 - **COBOL**
 - **IDMS**
 - **PL/1**

Default is **COBOL**.

Note: COBOL and PL/I imports are allowed with IDMS. However, you must manually add the IDMS record navigation information.

6. Click **Next**.
7. Enter information in the **Import Copybook - Remote Details** dialog box for the copybook type.

Note: IDMS parameters are determined by the installation configuration and standards.
8. If you selected the **IDMS** copybook type, in the **Import Copybook - Remote IDMS Details** dialog box, click **Advanced**.

In the **IDMS Advanced Properties** dialog box, enter the additional information for acquiring the IDMS metadata. For more information, see [“IDMS Advanced Properties Dialog Box” on page 150](#).

Click **OK**.

9. Click **Finish**.
10. Click **OK**.

If you click **OK**, the PowerExchange Navigator communicates with the PowerExchange Listener on the z/OS system specified by the node name in the **Location** list. The PowerExchange Listener submits batch job IDMSMJCL to import the IDMS metadata. IDMSMJCL is a member in the RUNLIB library on that z/OS system.

11. Click **Next**.

The **Import Copybook - Configuration Details** dialog box appears.

12. Enter configuration details for the copybook.

IDMS log-based change data capture works at a single-record level. If the data map is to be used for CDC processing, select the **Create Table on Each Record Imported** option.

Also, for both bulk data movement and CDC processing, you can create a table for each imported record and create expressions that preserve set relationships among tables. To do so, select the following options:

- **Create Table on Each Record Imported**
- **Create expressions for IDMS set relationships**

Alternatively, in some cases for bulk data movement you might want to create a complex table for each hierarchical path. To do so, select **Create Tables for IDMS Hierarchical Paths**.

Caution: Selecting this option might result in significant data redundancy, as parent data is repeated for each row down to the lowest child in the hierarchy. Also, CDC workflows cannot correctly maintain these hierarchical paths. To maintain data relationships, in most cases it is best to create single-record tables and create expressions for IDMS set relationships.

For more information, see ["Import Copybook - Configuration Details Dialog Box" on page 150](#).

13. Click **Finish**.
14. Click **OK**.
15. In the **Record Definition** dialog box, click **Apply**.
16. For the COVERAGE record, click **OK**.

RELATED TOPICS:

- ["Adding a CALC Element to an IDMS Record" on page 70](#)
- ["Adding an Owner Record and Set to an IDMS Record" on page 71](#)
- ["Sending a Data Map to a Remote System" on page 90](#)
- ["Import Copybook - Remote Details Dialog Box" on page 145](#)
- ["IDMS Advanced Properties Dialog Box" on page 150](#)
- ["Data Map Properties - IDMS Access Method" on page 336](#)

Editing IDMS Record Properties

You can change IDMS record properties after you import the IDMS metadata.

To edit IDMS record properties:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click a record and click **Properties**.
The **Record Properties** dialog box appears.
3. On the **Name** tab, define the record properties.
4. On the **Owner Records and Set Names** tab, define the properties for owner records and sets.

RELATED TOPICS:

- [“Adding a CALC Element to an IDMS Record” on page 70](#)
- [“Adding an Owner Record and Set to an IDMS Record” on page 71](#)
- [“IDMS Data Map Items” on page 61](#)
- [“Record Properties - Name \(IDMS\)” on page 351](#)

Adding Expressions to an IDMS Record

To return additional information for an IDMS data map record, add user-defined fields that use one or more PowerExchange functions in an expression to process source data.

These functions provide operations such as field splitting and concatenation.

You can also call the following functions to get CA IDMS/DB source data:

- GetDatabaseKey
- GetDataFlowType
- GetDbKeyOfFirstMember
- GetDbKeyOfLastMember
- GetDbKeyOfNextMember
- GetDbKeyOfOwner
- GetDbKeyOfPriorMember
- GetFullDbKey
- GetFullDbKeyOfFirstMember
- GetFullDbKeyOfLastMember
- GetFullDbKeyOfNextMember
- GetFullDbKeyOfOwner
- GetFullDbKeyOfPriorMember
- GetPageGroup
- GetPageGroupOfOwner
- GetPgGrpAndRdx
- GetPgGrpAndRdxOfOwner
- GetPgGrpOfOwner
- GetRadix
- GetRadixOfOwner
- GetSeqWithinLevel
- GetSeqWithinParent

Use expressions to create relational structures, such as data warehouses, from CA IDMS/DB source data. The IDMS functions that return database keys return results for both bulk and CDC source data.

You can also use the CallProg function in an expression to invoke a user-defined program to process source data.

RELATED TOPICS:

- [“Adding a User-Defined Field to a Record” on page 66](#)
- [“GetDataFlowType” on page 292](#)
- [“GetDbKeyOfOwner” on page 294](#)
- [“Adding Expressions to an IMS Record” on page 120](#)
- [“PowerExchange Functions for User-Defined Fields” on page 282](#)

IMS Data Maps

To use an IMS database as a data source or target, define an IMS data map. The data map is required to create SQL statements for relational-type access to IMS data during session processing.

When creating the data map for an IMS source or target database, add one or more IMS segments, and the fields in the segments, to define the physical layout of the data. A data map also includes table views of the data. The segment and field definitions provide the nonrelational view of the data, and the tables and columns provide the relational view of the data.

Note: For IMS data sources, *record* refers to a root segment and its children.

A data map can include one segment or multiple segments:

- Define a single segment in a data map if the associated data file contains one segment type. A single-segment data map can define multiple tables.
- Define multiple segments in a data map if the source database contains multiple segment types. The fewer columns that PowerExchange must retrieve results in faster file processing and data extraction.

You can use data maps to align fields, convert dates, filter data, and enhance data to improve data accuracy and reduce data volume. You can also use data maps for Lookup transformations that look up related values and determine if rows exist in an IMS target.

To define an IMS data map, complete the following tasks:

1. Add an IMS data map.

When you define the data map, select the **DL/1 BATCH** or **IMS ODBA** access method and associate the data map with a data file that defines the record types and data. You must also enter the IMS source or target properties that the access method requires. DL/I batch access requires a PCB number. ODBA access requires an IMS SSID, PSB name, and PCB name.

Note: You can override the access method in the data map when you configure a PowerCenter session or perform a database row test. If you override the access method, you must also enter the properties that the new access method requires.

2. If the DBD source is available for the IMS data, import the DBD source into the data map.

The DBD source defines segments and the hierarchical sequence for the data map. During the import process, PowerExchange maps the IMS segments and fields to tables to create the relational view of source data that PowerExchange requires.

3. To overlay each segment with its COPYLIB, import a COBOL copybook into each IMS segment in the data map.
4. Send the data map to the PowerExchange Listener on the local or remote node to store the data map in PowerExchange.

This action converts the data map into an operating-system-independent file that PowerExchange can access for extraction processing.

5. Run a database row test to test the data map.

A database row test accesses actual source data, displays it in table format, and verifies that you can access data from the source database.

The PowerExchange installation includes the example IMSDEMOS file that includes a example IMS DBD source that you can use to create a example IMS data map.

Prerequisites for Adding an IMS Data Map

Before you add an IMS data map, you must gather information about the IMS database.

- Gather the following information about the IMS source or target:
 - The IMS SSID, which is required for ODBA access and IMSUNLD access to unload files.
 - The IMS database name and database data set name.
 - The DBD data set name and location.
 - The PSB name.
 - If you use IMS DL/I batch or BMP access to IMS data, determine the PCB number by counting the number of PCBs in the PSB. If the PSB has COMPAT=YES, add one to that number.
 - If you use IMS ODBA access, record the PCB name.
 - The names of the IMS segments, and which segments, if any, are of variable length. Additionally, note the hierarchical sequence as defined in the DBD source.
- To reduce the number of locking conflicts, set PROCOPT to read-only for the PCB for the IMS database.
- If the PSB is used with the IMS ODBA interface or a Netport BMP job, define the PSB in the IMS SYSGEN.
- Update the DBMOVER configuration member in the RUNLIB library.
- Update the dbmover.cfg configuration file on the PowerExchange Navigator system.
- If you plan to perform INSERTs or UPDATEs on an IMS segment, the data map must represent the complete segment length as defined in the IMS DBD. Otherwise, an INSERT or UPDATE to the segment might write nonblank data to the end of the segment not defined as FILLER. To avoid this issue, you can add a FILLER definition to the COPYLIB before you import it to PowerExchange.

Configuring DL/I Batch or BMP Access to IMS Data

To use the DL/I batch or BMP interfaces to access IMS source or target data, you must perform some PowerExchange configuration tasks on the z/OS system.

PowerExchange uses DL/I batch or BMP access when the IMS data map specifies the **DL/1 BATCH** access method or when you override the **IMS ODBA** access method to **DL/1 BATCH** for a database row test or PowerCenter bulk data movement session.

Before you run a database row test or PowerCenter session that uses DL/I batch or BMP access, complete the following tasks:

- Customize the IMSJCL member in the RUNLIB library so that it works as a DL/I batch or BMP netport job. You can include the %PSBNAME substitution variable if you want to override the PSB name when you run the bulk data movement session in PowerCenter. By using the %PSBNAME variable with the session override, you can use the same netport JCL member to access multiple PSBs.
- If you use BMP access, APF-authorize the netport job.

- Create an IMS data map that specifies the **DL/1 BATCH** access method, or override the the **IMS ODBA** access method in an IMS data map with **DL/1 BATCH**. You can override the access method in the PowerCenter session properties for the IMS source or target or when configuring a database row test.
- In the DBMOVER configuration member in the RUNLIB, add a NETPORT statement and corresponding LISTENER statement for each PSB. You can enter up to ten pairs of statements in the DBMOVER member.

Tip: To avoid exceeding this limit when you need to process more than ten PSBs, specify the %PSBNAME in the netport JCL and an **IMS PSBNAME Override** value in the PowerCenter session properties. You can then use the same pair of NETPORT and LISTENER statements for multiple PSBs. For more information, see *PowerExchange Interfaces for PowerCenter*.

For the DBMOVER changes to take effect, you must restart the PowerExchange Listener.

Note: You can also override the PSB name and PCB number for a PowerCenter session or database row test.

Configuring IMS ODBA Access to IMS Data

To use IMS ODBA to access IMS data, you must complete some PowerExchange and IMS configuration tasks on the z/OS system.

PowerExchange uses IMS ODBA access when the IMS data map specifies **IMS ODBA** as the access method or when you override the **DL/1 BATCH** access method to **IMS ODBA** for a database row test or PowerCenter bulk data movement session.

IMS ODBA queries run as PowerExchange Listener subtasks and use the Listener port.

To use ODBA access, perform the following configuration tasks:

- APF-authorize the PowerExchange Listener STEPLIB libraries.
- Include the RESLIB library in the STEPLIB library for the PowerExchange Listener for ODBA access, or verify that the RESLIB is accessible through the LNKST concatenation on the z/OS system.
- Verify that each PSB contains defined PCBNAMEs.
- Create an IMS data map that specifies **IMS ODBA** as the access method, or override the **DL/1 BATCH** access method in an IMS data map with **IMS ODBA**. You can override the access method in the PowerCenter session properties or when configuring a database row test.

Note: You can also override the IMS SSID, PSB name, and PCB name for a PowerCenter session or database row test.

- In the DBMOVER configuration member in the RUNLIB, add the IMSID=ssid and ODBASUPP=YES statements. Then restart the PowerExchange Listener for these DBMOVER changes to take effect.
- Verify that the IMS Resource Recovery Services (RRS) is active on the system. In the DFSPBxxx member of the IMS PROCLIB, specify RRS=Y.
- Assemble and linkedit the IMS load module for configuring ODBA access to data. Use JCL that is similar to the following JCL:

```
//ULIA      JOB      ( ) ,CLASS=A,MSGCLASS=H,MSGLEVEL=(1,1) ,REGION=4M
//*
//STEP1 EXEC HLASMCL,
//          PARM.C=('RENT')
//C.SYSLIB DD DISP=SHR,DSN=SYS1.MACLIB
//          DD DISP=SHR,DSN=IMS710.ADFSMAC
//C.SYSIN  DD *
DFSIMS70 CSECT
*
*      THE DFSPRP MACRO GENERATES THE DRA STARTUP TABLE
*      THERE ARE A FEW MORE POSSIBLE PARMS
*          DFSPRP DSECT=NO,          GENERATE          X
*          FUNCLV=1,                FUNCTION LEVEL    X
```

```

DDNAME=CCTLDD,          DDNAME FOR DRA RESLIB      X
DSNAME=IMS1210.SDFSRESL, DSNAME FOR DRA RESLIB      X
DBCTLID=IMSC,          DBCTL IDENTIFIER          X
USERID=,              USERID              X
MINTHRD=2,           MIN NUMBER OF THREADS      X
MAXTHRD=7,           MAX NUMBER OF THREADS      X
TIMER=60,            IDENTIFY TIMER VALUE DEFAULT X
FPBUF=5,             NUMBER OF FP BUFFERS PER THREAD X
FPBOF=5,             NUMBER OF FP OVERFLOW BUFFERS X
SOD=A,              SNAP DATASET OUTPUT CLASS  X
AGN=,              APPLICATION GROUP NAME      X
TIMEOUT=60,         DRATERM TIMEOUT VALUE      X
IDRETRY=0,          IDENTIFY RETRY COUNT        X
CNBA=10            TOTAL FP NBA BUFFERS FOR CCTL
END
//L.SYSLMOD DD DSN=ABC.A.LOAD(DFSIMSC0),
//              DISP=SHR

```

The L.SYSLMOD DD statement must specify a module name that has in the following format: "DFS" + *DBCTLID_value* + "0". This module name must match the SSID for the control region and must be specified in the IMS data map. In this example JCL, the module name is DFSIMSC0.

Configuring the PowerExchange dbmover.cfg File on Windows

Before you add an IMS data map, update the dbmover.cfg file on Windows to provide access to the IMS source data.

Configuration settings depend on whether you access IMS by using DL/1 batch or IMS ODBA.

To configure the PowerExchange dbmover.cfg file on Windows:

1. Add a NODE statement to the dbmover.cfg file to access the remote IMS system, as follows:
 - To use DL/1 batch or BMP to access IMS, add a NODE statement that points to the netport port for the PowerExchange Listener on z/OS. The format is:

```
NODE=(location_name,TCPIP,[hostname_or_IPaddress],port_number)
```

- To use IMS ODBA to access IMS, add a NODE statement that points to the PowerExchange Listener port on z/OS. The format is:

```
NODE=(location_name,TCPIP,[hostname_or_IPaddress],port_number)
```

2. Restart the PowerExchange Listener to implement the configuration changes.

Note: You do not need to start the PowerExchange Listener on Windows to access the IMS data.

Adding an IMS Data Map

To add an IMS data map, you must specify a schema name, data map name, and access method. To define the data layout, you can import an IMS DBD when creating the data map or afterwards.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.

2. In the **Name** dialog box, enter the following information:

Field	Description
Schema Name	The name of the schema.
Data Map Name	A user-defined name for the data map.
Access Method	<p>One of the following options for accessing the source data:</p> <ul style="list-style-type: none"> - DL/1 BATCH. Accesses an IMS database through submitted JCL. Although the access method is DL/1 batch, you can choose either DL/1 or BMP for the netport job. - IMS ODBA. Connects to an IMS database through an open source database connection. - SEQ. Accesses an IMS database that you plan to use as a flat file. <p>Note: You can override the DL/1 BATCH or IMS ODBA access method later when you perform a database row test or run a PowerCenter bulk data movement session.</p>
Import Record Definitions	<p>Select this box to import a DBD into the data map. The DBD import adds segments and corresponding table views to the data map, along with key field, search field, CCK field definitions. It also defines the hierarchical sequence of segments.</p> <p>After you import a DBD, you can import a COBOL or PL/I copybook into each segment in the data map to overlay the segment with its COPYLIB.</p>

3. Click **Next**.
4. In the **Access Method** dialog box, enter information for the access method that you selected. Then click **Finish**.
5. If you selected **Import Record Definitions**, import the IMS DBD and segment copybooks.

After you complete the data map, perform database row tests on the tables in the data map to verify that the data map can be used to access data.

RELATED TOPICS:

- [“Importing a Copybook into a Data Map” on page 135](#)

Viewing IMS Data

In an IMS data map, records display the nonrelational view of the data and tables display the relational view of the data.

When you import data for a single segment, the record and table view for the segment appears in the data map. If you import data for multiple segments, multiple segments appear in the data map.

You can also view the IMS hierarchy created by the DBD source import.

Viewing an IMS Hierarchy

You can view the IMS hierarchy created by the DBD source import.

To view an IMS hierarchy:

- ▶ On the **Data Map** tab in the **Resource Explorer**, right-click a data map and click **Display IMS Hierarchy**. The IMS hierarchy for the IMS data map appears.

Adding Expressions to an IMS Record

To return additional information for an IMS data map record, add user-defined fields that use one or more PowerExchange functions in an expression to process source data.

These functions provide operations such as field splitting and concatenation.

You can also use the following functions to get IMS source data:

- `GetDatabaseKey`
- `GetIMSRBByLevel`
- `GetSeqWithinLevel`
- `GetSeqWithinParent`

You can also use the `CallProg` function in an expression to invoke a user-defined program to process IMS source data.

RELATED TOPICS:

- [“Adding Expressions to an IDMS Record” on page 114](#)
- [“PowerExchange Functions for User-Defined Fields” on page 282](#)

Getting the RBA for an IMS Record

To get the IMS RBA for an IMS record, add a user-defined field to the record in the IMS data map. In the user-defined field, use either the `GetDatabaseKey` or `GetIMSRBByLevel` function in an expression.

To add the segment RBA to segment data in an IMS data map, Informatica recommends that you use the `GetIMSRBByLevel` function rather than the `GetDatabaseKey` function. The `GetIMSRBByLevel` function enables you to get the RBA of an unkeyed or non-unique keyed parent segment.

Use the `GetIMSRBByLevel` function in IMS data maps used for bulk data movement operations or IMS synchronous CDC.

You cannot use the `GetIMSRBByLevel` function in records in the following types of IMS data maps:

- IMS data maps used for IMS log-based CDC.
- IMS data maps used to access IMS unload files that contain the following types of records:
 - Unload records of any format for HISAM data sets
 - Unload records that have the format BMC SHORT or BMC XSHORT for HDAM and HIDAM data sets
 - Unload records that have the standard IBM format for IMS Fast Path data sets

To run a database row test on the following types of data maps that use the `GetIMSRBByLevel` function, you must APF-authorize the following libraries:

- If you use IMS DL/1 batch data maps with BMP for the netport job, you must APF-authorize all libraries in the BMP STEPLIB concatenation. Otherwise, PowerExchange issues message PWX-02204 with return code 12.
- If you use IMS ODBA data maps, you must APF-authorize all libraries in the STEPLIB concatenation of the PowerExchange Listener. Otherwise, PowerExchange abends.

RELATED TOPICS:

- [“Adding a User-Defined Field to a Record” on page 66](#)
- [“GetDatabaseKey” on page 291](#)

Editing Properties in an IMS Data Map

To update segments and tables in an IMS data map, re-import the DBD or COBOL copybook into the data map.

Alternatively, you can manually edit properties for segments and tables in an IMS data map.

Warning: Manually editing segments and tables in a data map can result in loss of data in a data map.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)
- [“Importing a Copybook into a Data Map” on page 135](#)
- [“Importing a Copybook into a Record or a Segment” on page 137](#)

Sending an IMS Data Map to a Remote System

To store an IMS data map, send the data map to the PowerExchange Listener on the same system as the IMS source data.

Store each data map that you create so that PowerExchange can extract the related source data. When you store a data map, PowerExchange converts it to a operating system-independent file that PowerExchange can accessed for extraction processing.

RELATED TOPICS:

- [“Sending a Data Map to a Remote System” on page 90](#)

Using Lookup Transformations to Look up Data in IMS Databases

To look up data in IMS databases, use PowerCenter Lookup transformations.

Use a separate netport job for each Lookup transformation for PCBs with a “GOx” PROCOPT.

Note: Specify the IMS key values of a segment so that SSAs perform efficiently within IMS.

Writing Data to an IMS Database

You can write source data from a data map to an IMS database.

Use the PowerExchange Client for PowerCenter (PWXPC) to write data to IMS.

When you write data to IMS, consider the following guidelines:

- Specify the key values of a segment so that all the processing required to write data for the segment occurs within IMS.
- You cannot use SDEP segments.
- You cannot update a segment below an unkeyed segment.
- You can use segments that contain an OCCURS clause where the COPYLIB has been changed to identify each of the fields in the OCCURS clause as a separate field, which eliminates the OCCURS clause in the COPYLIB definition. You cannot generate a separate row for each occurrence of the OCCURS clause.
- When you import a data map to PowerCenter, verify that any CCK fields that you want to use as key fields are marked as key fields.
- In PowerCenter, use a lower Commit frequency to reduce the risk of locking segments.

- Use a separate netport job to write IMS data. Using a separate netport job enables you to access a PSB with write intent and to modify the JCL to update IMS data, such as the IEFRDER log.
 - When you write data for one of the following types of fields to the database, indicate whether the field is optional when you add the data map:
 - Spaces to a field
 - No data to a field

If you have several fields of this type, indicate that the first field in the data map is optional, and that all fields following this field are optional. If you do not specify the field is optional, errors appear in the PowerCenter session log.
1. To verify that the z/OS system is accessible, ping the IP address of the z/OS system where the target database is located.
 2. In the dbmover.cfg configuration file, include a NODE statement that points to the remote z/OS system.
 3. To verify that the remote PowerExchange Listener is started, issue the following command:


```
dtlrexe loc=xxxx prog=ping uid=userid pwd={password|passphrase}
```

Where xxxx is the NODE name from the PowerExchange configuration file dbmover.cfg.
 4. In the PowerExchange Navigator, add, test, and store an IMS data map.
 5. Use PowerCenter to move the data to IMS.

Viewing or Editing IMS Options for a Table

The options defined on the **IMS Options** tab in the **Table Properties** dialog box indicate how data is written to an IMS database, and provide default values for optimal performance when working with IMS.

These options are set to default values that cause PowerExchange to:

- Treat IMS as a relational target
- Behave like an IMS COBOL application program

Editing the default values for these options impacts the amount of data sent to the database and the time required to write the data.

To view or change IMS options:

1. Open an IMS data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click a table and select **Properties**.
3. On the **IMS Options** tab in the **Table Properties** dialog box, define IMS options for the table.

Warning: Do not edit the default value for the **Delete Options** option unless you are familiar with the IMS hierarchy. Changing these settings can result in loss of IMS data.

RELATED TOPICS:

- [“Table Properties - IMS Options ” on page 367](#)

Testing an IMS Data Map

After you add a data map and store the data map in PowerExchange, run a database row test to test the data map and to verify that you can access the data source.

A database row test accesses actual source data and displays it in table format. A database row test can read data from an IMS unload file or an IMS database.

To test an IMS data map, select the data source type in the **Database Row Test** dialog box, as follows:

- **IMS unload file.** In the **DB Type** list, select **IMSUNLD**.
- **IMS database.** In the **DB Type** list, select **NRDB** or **NRDB2**.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)

IMS Data Map Examples

The following examples demonstrate how to create a single-segment data map and a multiple-segment data map by using the example DBD source that PowerExchange provides.

PowerExchange provides the following example DBD source:

```
DBD   NAME=IMSDMO, ACCESS= (HDAM, VSAM) , X
      RMNAME= (DFSHDC40, 50, 80) , X
      EXIT= (*, KEY, PATH, (CASCADE, KEY, PATH) , LOG)
DATASET DD1=IMSDMO, DEVICE=3380, SIZE=8192, SCAN=3
SEGM  NAME=IMSDMOS, BYTES=80FIELD  NAME= (DTLKEY, SEQ) , BYTES=9, START=3
FIELD  NAME=RECTYPE, BYTES=1, START=12FIELD  NAME=AMOUNT, BYTES=4, START=13
FIELD  NAME=BINNO, BYTES=4, START=17FIELD  NAME=DECNO, BYTES=3, START=21
FIELD  NAME=DTYY, BYTES=2, START=24FIELD  NAME=DTMM, BYTES=2, START=26
FIELD  NAME=DTDD, BYTES=2, START=28FIELD  NAME=ACCT1, BYTES=10, START=30
FIELD  NAME=ACCT2, BYTES=10, START=40
FIELD  NAME=ACCT3, BYTES=10, START=50
DBDGEN
FINISH
```

When you install PowerExchange on z/OS, PowerExchange creates libraries with the high-level qualifier that you specify in the MVS Installation Assistant.

On z/OS, run the following job to create the IMS environment for the example installation:

```
hlq.DTLDEMO (IMSDEF)
```

Adding an IMS Single-Segment Data Map - Example

This example demonstrates how to create a data map for a single IMS segment.

In this example, you use the IMSDEMOS example and import the example DBD source file to define the segment and the hierarchical sequence. You also import a COBOL copybook for the segment. This example uses the DL/1 batch access method.

To add a single-segment IMS data map:

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. Enter the following information in the **Name** dialog box:
 - In the **Schema Name** box, enter **IMSS1**.
 - In the **Data Map Name** box, enter **IMSM1**.
 - In the **Access Method** list, select **DL/1 BATCH**.
 - Select the **Import Record Definitions** option.
3. Click **Next**.
4. In the **DL/1 Batch Access Method** dialog box, specify information about the data source .

If you are reading an IMS unload data set, the **IMS SSID** and **DBD Name** values are required.

If you are reading the IMS database by using a netport job that runs either DL/1 batch or BMP, the **PCB Number** value is required.

5. Click **Finish**.
On the **Data Map** tab in the **Resource Explorer**, the IMSS1.IMSM1 data map appears.
6. In the **Import Copybook - Source Details** dialog box, enter the following information:
 - Select the **Remote** option.
 - In the **Type** list, select **DBD**.
7. Click **Next**.
8. In the **Import Copybook - Remote DBD Details** dialog box, enter the following information:
 - In the **File Name** box, enter the PDS name and DBD member with the high-level qualifier (HLQ) that you specified at installation. For example:
hlq.DTLDEMO(IMSDBD)
 - In the **Location** list, select the z/OS location.
 - The locations displayed in the **Location** list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.
 - If security has been implemented for the PowerExchange Listener on z/OS, enter the z/OS user ID and password.
 - In the **Save File Locally As** box, enter `IMSDBD`.
9. Click **Next**.
10. In the **Import Copybook - Configuration Details** dialog box, select the following options:
 - Select the **Create tables for DL1 hierarchical paths** option.
 - Accept the other defaults.
 For more information, see [“Importing a Copybook into a Data Map” on page 135](#).
11. Click **Finish**.
12. In the **Import Copybook Configuration Details** dialog box, review your selections and click **Finish**.
13. To import the DBD source, click **OK**.
14. In the **Record Definition** dialog box, click **Import**. If necessary, skip a record type or stop an import. For more information, see [“Importing a Copybook into a Record or a Segment” on page 137](#).
15. To complete the import, click **Next Redefinition** or **Previous Redefinition** to move the blue arrow to select a line.
16. If prompted, click **Resume Import** to resume the import.
17. View the log information for the import process.

Importing a Copybook into an IMS Single-Segment Data Map - Example

This example demonstrates how to import a copybook into an IMS single-segment data map.

To import a copybook into an IMS single-segment data map - example:

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, select the IMS segment.
3. Click **File > Import Copybook**.
4. On the **Import Copybook - Source Details** dialog box, enter the following information:
 - Select the **Remote** option.
 - In the **Type** list, select **COBOL**.
5. Click **Next**.

6. In the **Import Copybook Remote DBD Details** dialog box, enter the following information:
 - In the **File Name** box, enter the copybook name and location.
 - In the **Location** list, select the z/OS location.

The locations displayed in the **Location** list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator machine.

 - If security has been implemented for the PowerExchange Listener on z/OS, enter the z/OS user ID and password.
 - In the **Save File Locally As** box, enter a file name.
7. Click **Next**.
8. In the **Import Copybook - Configuration Details** dialog box, accept the defaults and click **Finish**. For more information, see [“Importing a Copybook into a Data Map” on page 135](#).
9. In the **Import Copybook Configuration Details** dialog box, review your selections and click **Finish**.
10. Respond to the import prompts.
11. When the import is complete, review the data.
12. Store the data map.
13. Run a database row test to test the data map.

Adding an IMS Multiple-Segment Data Map - Example

This example demonstrates how to add an IMS multiple-segment data map that defines a complex table.

In this example, you add multiple segments, one at a time, to create a hierarchical relationship between the segments in the complex table.

To create an IMS multiple-segment data map:

1. Open a data map that contains multiple segments.
2. On the **Data Map** tab in the **Resource Explorer**, right-click a table and click **Properties**.
3. In the **Table Properties** dialog box, enter the following information:
 - In the **Record Dependencies** list, select a segment to make it the parent.
 - In the **Available Records** list, right-click a segment to add it as a child and click **Add Record as Child**.
 - In the **Record Dependencies** list, select the parent segment.
 - In the **How do you want to handle multiple instances of selected record** list, select **New Row**.
 - In the **Record Dependencies** list, select each child segment and set the multiple instances option to **New Row**.
4. Click **OK**.

Note: For each segment in the hierarchy, you can set different options for the **How do you want to handle multiple instances of selected record** option. For example, set ROOT to **Ignore**, SEG1 to **New Row**, and SEG2 to **Array**.

If a parent segment is set to **Array**, set all child segments to **Ignore**.

You can also define segments at the same level. For example, add the SEG1 and SEG2 to ROOT segments as siblings.

RELATED TOPICS:

- [“Viewing an IMS Hierarchy” on page 119](#)
- [“Table Properties - Definition” on page 364](#)

SEQ Data Maps

To access files on Linux, UNIX, or Windows or sequential data sets on z/OS through PowerExchange, add a data map that uses the SEQ access method.

You can import a COBOL or PL/I copybook to describe the structure of the source file when you add an SEQ data map or later.

To test an SEQ data map, run a database row test.

Adding an SEQ Data Map

When you add a data map, you can optionally import a copybook to describe the structure of the sequential data set or flat file. The following steps assume that you use a copybook.

1. Click the **Resources** tab in the **Resource Explorer**.
2. On the menu bar, click **Add > Data Map**.
3. On the **Name** page, enter the following information:
 - Enter the **Schema Name** and **Data Map Name**.
 - In the **Access Method** list, select **SEQ**.
 - To import record definitions from a copybook, select the **Import Record Definitions** option.
4. Click **Next**.
5. On the **SEQ Access Method** page, enter the **File Name** and other required information, and click **Finish**.
For more information, see [“Data Map Properties - SEQ Access Method” on page 340](#).
6. On the **Import Copybook - Source Details** page, select the source type. Then select one of the following copybook types:
 - **COBOL**
 - **PL/1**Default is **COBOL**.
7. Click **Next**.
8. On the **Import Copybook - Remote Details** page for the copybook type, enter information for the copybook and click **Next**.
9. On the **Import Copybook - Configuration Details** page, select the actions for imported records, fields, and tables and click **Next**.
For more information, see [“Importing a Copybook into a Data Map” on page 135](#).
10. On the **Import Copybook Information** page, review the information for the import operation and click **OK**.
11. On the **Record Definition** page, select an action for each imported record and click **OK**.
For more information, see [“Importing a Copybook into a Data Map” on page 135](#).
The **Import** window displays the imported copybook.
The **Copybook Message Log** window displays the results of the import operation.
Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.

12. If you imported multiple records, you can perform the following actions for each record to define a record ID value:
 - a. On the **Data Map** tab in the **Resource Explorer**, click the record.
 - b. In the **Record** window, right-click the field that identifies the record, and click **Properties**.
The **Name** tab in the **Field Properties** dialog box appears.
 - c. In the **Record ID Values** box, enter a value and operator that defines the record.
13. To identify the relationships between record types, enter optional record properties. Perform the following actions for each record:
 - a. Right-click the record.
 - b. Select **Properties**.
 - c. Select the **Parent Record Name**, **Redefines Record Name**, or **Header/Trailer Record** option, and select the record name or type of record.

With this metadata, you can import a multiple-record data map into PowerCenter to perform multiple-record writes.

For more information about multiple-record writes, see the *PowerExchange Bulk Data Movement Guide* and *PowerExchange Interfaces for PowerCenter*.

Viewing an SEQ or VSAM Record Hierarchy

You can view the hierarchy for a record in an SEQ or VSAM data map. The hierarchy is based on the **Parent Record Name**, **Redefines Record Name**, and **Header/Trailer Record** optional record properties that you define.

- ▶ On the **Data Map** tab in the **Resource Explorer**, right-click a record and click **Display Hierarchy**.
The hierarchy that includes the selected record appears, with the name of the selected record displayed in bold.
A record that is a redefine of another record is labeled **REDEF** in the hierarchy diagram.

VSAM Data Maps

To access nonrelational VSAM data through PowerExchange, add a VSAM data map.

PowerExchange accesses the VSAM data by using SQL, as if the data source were relational.

You can add data maps for the following types of VSAM data sets:

- Key-sequenced data set (KSDS)
- Entry-sequenced data set (ESDS)
- Fixed-length relative record data set (RRDS)
- Variable-length relative record data set (VRRDS)

These data sets can be in basic format, in extended format, or compressed.

Run a database row test to test a VSAM data map.

Adding a VSAM Data Map

When you add a data map, you can optionally import a copybook to describe the structure of the VSAM data set. The following steps assume that you use a copybook.

1. Click the **Resources** tab in the **Resource Explorer**.
2. On the menu bar, click **Add > Data Map**.
3. On the **Name** page, enter the following information:
 - In the **Access Method** list, select **ESDS**, **KSDS**, or **RRDS**.
 - Select the **Import Record Definitions** option.

Note: Select the **RRDS** access method for both fixed-length and variable length RRDS (VRRDS) data sets.
4. Click **Next**.
5. On the **Access Method** page, enter information for the access method and click **Finish**.

Note: By default, PowerExchange uses the system-defined buffering for VSAM processing. To tune PowerExchange processing, specify additional VSAM buffers.
6. On the **Import Copybook - Source Details** page, select the source type. Then select one of the following copybook types:
 - **COBOL**
 - **PL/1**

Default is **COBOL**.
7. Click **Next**.
8. On the **Import Copybook - Remote Details** page for the copybook type, enter information for the copybook and click **Next**.
9. On the **Import Copybook - Configuration Details** page, select the actions for imported records, fields, and tables and click **Next**.

For more information, see [“Importing a Copybook into a Data Map” on page 135](#).
10. On the **Import Copybook Information** page, review the information for the import operation and click **OK**.
11. On the **Record Definition** page, select an action for each imported record and click **OK**.

For more information, see [“Importing a Copybook into a Data Map” on page 135](#).

The **Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.
12. If you imported multiple records, you can perform the following actions for each record to define a record ID value:
 - a. On the **Data Map** tab in the **Resource Explorer**, click the record.
 - b. In the **Record** window, right-click the field that identifies the record, and click **Properties**.

The **Name** tab in the **Field Properties** dialog box appears.
 - c. In the **Record ID Values** box, enter a value and operator that defines the record.
13. To identify the relationships between record types, enter optional record properties. Perform the following actions for each record:
 - a. Right-click the record.
 - b. Select **Properties**.

- c. Select the **Parent Record Name**, **Redefines Record Name**, or **Header/Trailer Record** option, and select the record name or type of record.

With this metadata, you can import a multiple-record data map into PowerCenter to perform multiple-record writes.

For more information about multiple-record writes, see the *PowerExchange Bulk Data Movement Guide* and *PowerExchange Interfaces for PowerCenter*.

RELATED TOPICS:

- [“Data Map Properties - ESDS Access Method” on page 335](#)
- [“Data Map Properties - KSDS Access Method” on page 337](#)
- [“Data Map Properties - RRDS Access Method” on page 339](#)

Viewing an SEQ or VSAM Record Hierarchy

You can view the hierarchy for a record in an SEQ or VSAM data map. The hierarchy is based on the **Parent Record Name**, **Redefines Record Name**, and **Header/Trailer Record** optional record properties that you define.

- ▶ On the **Data Map** tab in the **Resource Explorer**, right-click a record and click **Display Hierarchy**.
The hierarchy that includes the selected record appears, with the name of the selected record displayed in bold.
A record that is a redefine of another record is labeled **REDEF** in the hierarchy diagram.

Improving Bulk Read Performance for VSAM Data Sets

You can improve the performance of bulk read operations for VSAM data sets.

PowerExchange provides the following options to improve performance:

- Specify the number of data and index buffers to use when reading data.
- Specify control interval access.
- Use keys in WHERE clauses.

Number of VSAM Data Buffers

You can specify the number of I/O buffers that VSAM uses to transmit data between virtual and auxiliary storage.

A buffer is the size of a control interval in the data component. Valid values for the number of data buffers (BUFND) are from 0 through 65535. Minimum value is 1 plus the value specified for STRNO. If you omit STRNO, BUFND must be at least 2, because the default for STRNO is 1.

Note: The minimum buffer specification does not provide optimal sequential processing performance. Generally, the more data buffers that are specified, the better the performance. Additional data buffers can benefit direct inserts or updates during control area splits and benefits spanned record accessing.

Maximum number of buffers is 255, which includes 254 data buffers and one insert buffer.

Number of VSAM Index Buffers

Specifies the number of I/O buffers that VSAM uses to transmit the contents of index entries between virtual and auxiliary storage for keyed access.

A buffer is the size of a control interval in the index. Valid values for the number of index buffers (BUFNI) are from 0 through 65535. Minimum number is the number specified for STRNO. If you omit STRNO, BUFNI must be at least 1, because the default for STRNO is 1. The default is the minimum number required.

Additional index buffers improve performance by providing for the residency of some, or all, of the high-level index, which minimizes the number of high-level index records retrieved from DASD for key-direct processing.

Change the number of buffers used at a PowerExchange system level by including a VSAM statement in the DBMOVER configuration member. The value in this statement overrides any entry in the data map.

Maximum number of buffers is 255, which includes 254 data buffers and one insert buffer. Default is the minimum number required.

Using VSAM Control Interval Access

Control interval (CI) access is a read option that causes the VSAM access method to use direct control interval reads to retrieve data records.

If you extract all data from the VSAM data set, this option might improve sequential read performance. If you specify a WHERE clause when retrieving the data, CI access provides no performance benefit.

To select CI access at the data map level, select the CI ACCESS option on the **Access Method** tab in the **Data Map Properties** dialog box. The **CI ACCESS** option is valid for KSDS, ESDS, and RRDS access methods.

CI access is not supported for compressed VSAM data sets or VSAM data sets with spanned records.

Using VSAM Keys

PowerExchange reads key information for VSAM data sets from the catalog when the data set is allocated and opened.

Both the key length and the relative key position are retrieved.

For example, the following lines define a record layout for keypos=10 and keylen=4:

```
01 REC.
  04 HEADER      PIC X(10) .
  04 KEY.
    08 KEY1      PIC 99.
    08 KEY2      PIC 99.
  04 REST        PIC X(200) .
```

For optimization to work, include KEY1 in the WHERE clause.

Use key information to speed processing of WHERE clauses, which provides optimization.

Different WHERE clause formats produce different results.

For example:

```
Line 1   SELECT * FROM TEST.VSAM1.TAB WHERE (KEY1 > 50) AND (KEY1 < 90)
```

Line 1 translates to:

```
(50 < KEY1 < HIGH-VALUES) AND (LOW-VALUES < KEY1 < 90)
```

The following example generates the same results as the Line 1 example, but with a different elapsed time:

```
Line 2   SELECT * FROM TEST.VSAM1.TAB WHERE (KEY1 > 50 AND KEY1 < 90)
```

Line 2 translates to:

```
(KEY1 > LOW-VALUES AND KEY1 < HIGH-VALUES)
```

This format is still technically optimized, but when the file is processed, the read start is positioned at LOW-VALUES and read until HIGH-VALUES, which is not ideal.

Line 2 is the preferred syntax. This syntax positions the file at record 50 and reads data until KEY1 is greater than or equal to 90.

Optimization is used for key positioning, the data is selected against the WHERE clause to verify that the correct records are filtered.

Note: If you specify a value greater than 0 in the **Skip First Records from File** box in the **KSDS Access Method** dialog box, the SELECT statement is not optimized even if you specify KEY1.

Getting the RRN or RBA for VSAM Data Set Records

You can get the relative record number (RRN) or relative byte address (RBA) for VSAM entry-sequenced data sets (ESDSs) and relative record data sets (RRDSs).

Return the RRN or RBA in the following ways:

- Prefix records with the RRN or XRBA value when you select these options in the data map properties.
- Include the RRN or RBA value in a new field in the record when you use the GetDatabaseKey function in an expression.

Note: If you request offload processing, you cannot use the GetDatabaseKey function in an expression.

Setting Data Map Properties to Get the RRN or RBA in a VSAM Data Set Record

You can set data map properties to get the relative record number (RRN) or relative byte address (RBA) for VSAM data set records.

1. Open the data map.
2. Right-click the data map and click **Properties**.
3. On the **Access Method** tab in the **Data Map Properties** dialog box, select one of the following options:
 - To get the RBA for ESDS data sets, select **Prefix record with XRBA**.
 - To get the RRN for RRDS data sets, select **Prefix Record with RRN**.

Note: If you request that PowerExchange return the RRN or RBA value, you must add an 8-byte binary field at the beginning of the record definition to contain this value.

4. Open the record.
5. In the **Record** window, right-click the first field in the record and click **Add Field Before**.
6. Enter a value for **Field Name**. Select **BIN** for the **Field Type** and **8** for the **Length** field.
7. Click **OK**.
8. Open the table.
9. In the **Table** window, right-click the first column in the table and click **Add Column Before**.
10. In the **Name** field, enter a name for the new column.
11. In the **Base Field** list, select the new field.
12. To send the data map to the z/OS system where the VSAM data set resides, click **File > Send to Remote Node**.

Using the GetDatabaseKey Function to Get the RRN or RBA in VSAM Data Set Records

Use the GetDatabaseKey function in an expression to get the relative record number (RRN) or relative byte address (RBA) for VSAM data set records.

Note: If you use offload processing, you cannot use the GetDatabaseKey function in an expression.

Define expressions in the record definition in the data map. Add user-defined fields to the record to contain the results of the expression.

1. Open the data map and the record.
2. On the **Expr(n)** tab, right-click in the window and click **Add Field at End**.
3. In the **Name** column, enter a name.
4. In the **Type** column, enter **BIN**.
5. In the **Length** column, enter **8**.
6. In the **Expression** column, enter **GetDatabasekey()**.
If the expression field is valid, a green check mark icon appears. Otherwise, a red cross icon and a message appear.
7. Open the table.
8. In the **Table** window, right-click the table name and click **Add Column at End**.
9. In the **Base Field** list, select the new column.
10. In the **Name** field, enter a name for the column.
11. Click **OK**.
12. Click **File > Send to Remote Node** to send the data map to the z/OS system where the VSAM data set resides.

CHAPTER 4

Copybooks

This chapter includes the following topics:

- [Copybooks Overview, 133](#)
- [Importing a Copybook into a Data Map, 135](#)
- [Importing a Copybook into a Record or a Segment, 137](#)
- [Importing an FDT into an Adabas Data Map, 139](#)
- [Displaying and Refreshing the FDT for an Adabas Data Map, 139](#)
- [Import Copybook - Source Details Dialog Box, 140](#)
- [Import Copybook - Field Property Details Dialog Box, 143](#)
- [Import Copybook - Local Details Dialog Box, 143](#)
- [Import Copybook - Local DB2 Catalog Details Dialog Box, 144](#)
- [Import Copybook - Remote Details Dialog Box, 145](#)
- [Import Copybook - Remote Datacom Details Dialog Box, 146](#)
- [Import Copybook - Remote DB2 Catalog Details Dialog Box, 148](#)
- [DB2 Column Selection Dialog Box, 149](#)
- [IDMS Advanced Properties Dialog Box, 150](#)
- [Import Copybook - Configuration Details Dialog Box, 150](#)
- [Record Definition and Duplicate Record Definition Dialog Boxes, 152](#)

Copybooks Overview

In the PowerExchange Navigator, the term *copybook* refers generically to data definitions for records, segments, fields, keys, and other items from a data source that you import into a data map to define the layout of the data.

Import a copybook to eliminate the need to manually define the layout of these items in a data map.

You can import a copybook when you add a data map or after you add it. You can also import one or more copybooks into a record or segment in a data map to add data definitions to the record or segment.

After you import a copybook, it is stored on the local system. You can then re-import that copybook from the local system.

If a copybook contains a record or field that duplicates a record or field that is already in a data map, you can select options to indicate how to handle the duplicate item. For example, you can choose to import, overwrite, or skip the item, or stop the import.

Use example data maps and copybooks in the examples folder in the PowerExchange installation directory to learn how to import the following types of copybooks into data maps:

- COBOL copybook
- COBOL copybook with REDEFINES statements
- PL/I copybook with multiple definitions for a field

RELATED TOPICS:

- [“Importing a COBOL Copybook into a Data Map - Example” on page 248](#)
- [“Importing a COBOL Copybook with REDEFINES Statements - Example” on page 253](#)
- [“Importing a PL/I Copybook - Example” on page 256](#)

Copybook Types Supported in Data Maps

You select the type of copybook to import in the **Type** list in the **Import Copybook - Source Details** dialog box.

The type of copybook that you can import depends on the data source type of the data map into which you are importing the copybook, as follows:

- **Data maps for all data source types except DB2.** Import a COBOL or PL/I copybook from the local or a remote system to define the layout of records and fields.
- **Adabas data maps.** You can import both a field definition table (FDT) and a copybook into an Adabas data map, as follows:
 - **FDT.** Each database file has a corresponding FDT, which defines the record structure and the content of each field in the physical file. For each field in the record, the FDT defines the level, name, length, format, options, and special field and descriptor attributes. An FDT provides short names and no scale information. After you import an FDT, it is cached on the local system for subsequent sessions.

You can display an FDT for an Adabas data map. When you display the FDT, you can optionally refresh the locally saved FDT with data from the FDT on the remote system.
 - **ADACMP.** A file created by using the ADACMP utility.
 - **DDM.** A Natural data definition module (DDM) for Adabas for z/OS sources. Provides long name and scale information.
 - **PREDICT.** A Predict data dictionary. Provides long names and scale information.
 - **TEXT.** A text file that uses an Informatica internal delimited format.
 - **COBOL or PL/1.** If you import a COBOL or PL/I copybook into an Adabas data map, you must edit table properties in the data map to associate the imported field definitions with Adabas field definitions.
- **Datacom data maps.** Import a COBOL or PL/I copybook, or import a DATACOM copybook from a remote z/OS system or a local .dcm file. A DATACOM copybook provides combined CA Datacom/DB schema and COBOL copybook information.
- **IDMS data maps.** Import a COBOL or PL/I copybook, or import an IDMS copybook from a remote z/OS system or a local .idm file. An IDMS copybook provides combined CA IDMS/DB schema and COBOL copybook information.
- **DB2 for i5/OS, DB2 for Linux, UNIX, and Windows, DB2 for z/OS, and DB2 unload file data maps.** You can import a DB2 catalog.
- **DB2 for i5/OS data maps that are accessed sequentially.** Import a COBOL or PL/I copybook, or import a data description specifications (DDS), which describe data attributes in file descriptions that are external to the application program that processes the data.

- **IMS DL/1 batch and ODBA data maps.** Import a database description block (DBD) to add segment, key field, search field, and CCK field definitions from the IMS database, and to add tables to the data map. Importing a DBD also defines the hierarchical sequence of segments.

After you import a DBD, import a COBOL or PL/I copybook into each segment in the data map to overlay the segment with its COPYLIB. This action redefines the data map while maintaining the hierarchical metadata for the database.

Importing a Copybook into a Data Map

You can import a copybook when you add a data map or after you add it.

Tip: If you plan to import a COBOL copybook, you can run a trace at import time that provides an enhanced memory map, which is useful for troubleshooting mapping errors and other purposes. For more information, see [Appendix G, “Trace for Creating a Memory Map When Importing a COBOL Copybook” on page 372](#).

1. Complete one of the following steps depending on whether you are importing a copybook into an existing data map or adding a data map:
 - For an existing data map, open the data map and click **File > Import Copybook**.
 - If you are adding a data map, select the **Import Record Definitions** check box in the **Name** dialog box. Then, click **Next**. Enter information in the **Access Method** dialog box, and click **Finish**.

The **Import Copybook - Source Details** dialog box appears.

2. In the **Import Copybook - Source Details** dialog box, enter information about the location and type of copybook that you want to import.

For some copybook types, you must enter additional information.

For more information, see [“Import Copybook - Source Details Dialog Box” on page 140](#).

Note: For an IMS data map, import a DBD copybook. Then, import a COBOL or PL/I copybook into each segment in the data map.

3. To display information about the last import operation, click **Last Import**.

The **Import Copybook Information** window displays the following information about the last copybook import operation:

- Whether the copybook was imported from the local or a remote system
- The name and type of the copybook
- The default actions defined for imported records, fields, and tables

To close the dialog box, click **Cancel**.

4. Click **Next**.
5. Based on whether you specified a remote or local copybook location, enter information about the copybook in one of the following dialog boxes for the data source type:
 - **Import Copybook - Remote Details** dialog box. For more information, see [“Import Copybook - Remote Details Dialog Box” on page 145](#).
 - **Import Copybook - Local Details** dialog box. For more information, see [“Import Copybook - Local Details Dialog Box” on page 143](#).
6. Click **Next**.
7. In the **Import Copybook - Configuration Details** dialog box, select options to define the default actions to take for imported records, fields, and tables.

For more information, see [“Import Copybook - Configuration Details Dialog Box” on page 150.](#)

8. If you are importing a COBOL copybook, enter field property details in the **Import Copybook - Field Property Details** dialog box.

For more information, see [“Import Copybook - Field Property Details Dialog Box” on page 143.](#)

9. Click **Finish**.
10. In the **Import Copybook Information** window, review the information for the import operation.

The **Import Copybook Information** window displays the following information:

- Whether PowerExchange imports the copybook from the local or a remote system
- The name and type of the copybook
- The default actions defined for imported records, fields, and tables

11. Click **OK**.
12. In the **Record Definition** dialog box, select an action for each imported record or segment.
If the copybook contains a record or field that duplicates a record or field in the data map, the **Duplicate Record Definition** dialog box appears. In the **Duplicate Record Definition** dialog box, you can select options to indicate how to handle the duplicate item. For example, you can choose to import, overwrite, or skip the item, or stop the import.

For more information, see [“Record Definition and Duplicate Record Definition Dialog Boxes” on page 152.](#)

13. Click **OK**.
14. If the copybook contains REDEFINES statements, a **Copybook Redefines** window notifies you that the copybook contains REDEFINES statements.

Complete the following steps:

- a. To point to the line that contains the correct field, click **Redefinition > Next** or **Redefinition > Previous**.
- b. Click **Import > Resume**.

15. The **Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.

16. Close the **Import** window.
17. If you chose to overwrite a duplicate record or field in a data map, complete the following actions:
 - On the **Data Map** tab in the **Resource Explorer**, right-click the table that contains columns that are based on the updated record or field and click **Properties**.
 - In the **Record Dependencies** list in the **Table Properties** dialog box, select the table.
 - In the **Column Generation** list, select **Refresh with missing columns**.
 - Run a database row test on the table.

Based on the data source type for the data map and the type of copybook that you imported, you might need to complete one of the following tasks:

- If you imported a COBOL or PL/I copybook into an Adabas data map, edit the data map to associate the imported field definitions with Adabas field definitions. For information, see [“Viewing or Editing Properties for a Data Map Table” on page 86.](#)
- If you imported a DBD source into an IMS data map, import a COBOL or PL/I copybook into each segment in the data map. For more information, see [“Importing a Copybook into a Record or a Segment” on page 137.](#)

- If you imported a DDM copybook with fields defined with wide character datatypes into an Adabas for z/OS data map, PowerExchange does not select the **Wide Char** option on the **Code Page** tab in the **Field Properties** dialog box.

To set the **Wide Char** option for fields defined with the wide character datatype, complete one of the following tasks:

- Import a PREDICT or FDT copybook.
- Edit field properties in the data map.

RELATED TOPICS:

- [“Import Copybook - Local Details Dialog Box” on page 143](#)
- [“Import Copybook - Local DB2 Catalog Details Dialog Box” on page 144](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)
- [“Import Copybook - Remote Datacom Details Dialog Box” on page 146](#)
- [“Import Copybook - Remote DB2 Catalog Details Dialog Box” on page 148](#)

Importing a Copybook into a Record or a Segment

Import one or more copybooks into a record or segment to add data definitions to the record or segment.

Note: After you import a DBD into an IMS data map, import a COBOL copybook into each segment in the data map to overlay each segment with its COPYLIB. This action redefines the data map while maintaining the hierarchical metadata information for the IMS database.

1. Open the data map.
2. On the **Data Map** tab in the **Resource Explorer**, select a record or a segment.
3. Click **File > Import Copybook**.
4. In the **Import Copybook - Source Details** dialog box, enter information about the location and type of copybook to import.

For some copybook types, you must enter additional information.

For more information, see [“Import Copybook - Source Details Dialog Box” on page 140](#).

Note: For an IMS data map, import a DBD copybook. Then, import a COBOL or PL/I copybook into each segment in the data map.

5. Click **Next**.
6. Based on whether you chose a remote or local copybook location, enter information about the copybook in one of the following dialog boxes:
 - **Import Copybook - Remote Details** dialog box. For more information, see [“Import Copybook - Remote Details Dialog Box” on page 145](#).
 - **Import Copybook - Local Details** dialog box. For more information, see [“Import Copybook - Local Details Dialog Box” on page 143](#).

Note: If security is enabled on z/OS, enter a user ID and password.

7. Click **Next**.

8. In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records or segments, fields, and tables, and click **Finish**. For more information, see [“Import Copybook - Configuration Details Dialog Box” on page 150](#).
9. In the **Import Copybook Information** window, review the information for the import operation.
The **Import Copybook Information** window displays the following information:
 - Whether PowerExchange imports the copybook from the local or a remote system
 - The name and type of the copybook
 - The default actions defined for imported records, fields, and tables
10. Click **OK**.
11. In the **Record Definition** dialog box, select an action for each imported record or segment.
If the copybook contains a record or field that duplicates a record or field in the data map, the **Duplicate Record Definition** dialog box appears. In the **Duplicate Record Definition** dialog box, you can select options to indicate how to handle the duplicate item. For example, you can choose to import, overwrite, or skip the item, or stop the import.
For more information, see [“Record Definition and Duplicate Record Definition Dialog Boxes” on page 152](#).
12. Click **OK**.
13. If the copybook contains REDEFINES statements, a **Copybook Redefines** window notifies you that the copybook contains REDEFINES statements.
Complete the following steps:
 - a. To point to the line that contains the correct field, click **Redefinition > Next** or **Redefinition > Previous**.
 - b. Click **Import > Resume**.
14. The **Import** window displays the imported copybook.
The **Copybook Message Log** window displays the results of the import operation.
Note: If an error message appears in the **Copybook Message Log** window, double-click the message to find the associated line in the copybook.
15. Close the **Import** window.
16. If you chose to overwrite a duplicate record or field in a data map, complete the following actions:
 - On the **Data Map** tab in the **Resource Explorer**, right-click the table that contains columns that are based on the updated record or field and click **Properties**.
 - In the **Record Dependencies** list in the **Table Properties** dialog box, select the table.
 - In the **Column Generation** list, select **Refresh with missing columns**.
 - Run a database row test on the table.

RELATED TOPICS:

- [“Import Copybook - Source Details Dialog Box” on page 140](#)
- [“Import Copybook - Local Details Dialog Box” on page 143](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)

Importing an FDT into an Adabas Data Map

You can import an FDT into an Adabas data map if you have not previously imported an FDT into the data map.

When you import an FDT into a data map, you also import Adabas keys.

To import an FDT into an Adabas data map:

1. Open the data map.
2. If the PowerExchange Navigator does not prompt you to import an FDT, click **Resource > FDT**.
3. In the **Adabas FDT Import** dialog box, enter connection information for the system where the FDT is located. For more information, see [“Adding a Data Map” on page 62](#).
4. Click **OK**.

Edit table properties in the data map to add the imported keys to the table.

RELATED TOPICS:

- [“Adding a Key to a Table in an Adabas Data Map” on page 105](#)

Displaying and Refreshing the FDT for an Adabas Data Map

You can display the FDT that is saved locally for an Adabas data map in the PowerExchange Navigator.

When the FDT is displayed, you can optionally refresh it with data from the original FDT that is on the remote Adabas system.

To display and refresh the FDT for an Adabas data map:

1. Open the data map.
2. Click **Resource > FDT**.
The **Adabas FDT** dialog box displays the FDT.
3. In the **Adabas FDT** dialog box, click **Refresh FDT**.
4. In the **Adabas FDT Import** dialog box, enter connection information for the system from which you are reading the FDT. For more information, see [“Importing an FDT into an Adabas Data Map” on page 139](#).
5. Click **OK**.

A message box displays the status of the refresh operation.

If the refresh operation was successful, the copy of the FDT on the local system is refreshed with data from the remote FDT file.

Import Copybook - Source Details Dialog Box

Enter information about the copybook that you are importing into a data map or a record.

Source

Select the location of the copybook.

The following table describes the options:

Option	Description
Local	The copybook is located on the PowerExchange Navigator system.
Remote	The copybook is located on a remote system.

Type

Select the type of copybook. The type of copybook that you can import depends on the access method of the data map into which you are importing the copybook. For example, you can import a copybook type of ADACMP into a data map defined with the ADABAS access method only.

To import a COBOL or PL/1 copybook, select **COBOL** or **PL/1**, respectively.

The following table lists the types of data maps, by access method and corresponding data source, into which you can import a COBOL or PL/1 copybook:

Import into Data Maps Defined with Access Methods	Corresponding Data Source
ADABAS	Adabas file
DATACOM	CA Datacom/DB file
DB2UNLD	DB2 unload file
DL/1 BATCH	IMS database
ESDS	VSAM ESDS
IDMS	CA IDMS/DB database
IMS ODBA	IMS database accessed through a PowerExchange Listener
KSDS	VSAM KSDS
MQSERIES	IBM MQSeries message queue
RRDS	VSAM RRDS
SEQ	Flat file or sequential data set
TAPE	Tape data set
USER	User access method program

The following table describes the **Type** options and lists the types of data maps, by access method and corresponding data source, into which you can import each type of copybook other than a COBOL or PL/1 copybook:

Option	Description	Import into Data Maps Defined with Access Methods	Corresponding Data Source
ADACMP	File created by using the ADACMP utility.	ADABAS	Adabas file
DATAKOM	Datacom copybook. Provides combined CA Datacom/DB schema and COBOL copybook information.	DATAKOM	CA Datacom/DB file
DB2 Catalog	DB2 catalog.	DB2	DB2 for i5/OS, DB2 for Linux, UNIX, and Windows, or DB2 for z/OS database
DB2 Catalog	DB2 catalog.	DB2UNLD	DB2 unload file
DBD	Database description block (DBD). Adds segment, key field, search field, and CCK field definitions from the IMS database, and adds tables to the data map. Also defines the hierarchical sequence of segments. ¹	DL/1 BATCH	IMS database
DBD	Database description block (DBD). Adds segment, key field, search field, and CCK field definitions from the IMS database, and adds tables to the data map. Also defines the hierarchical sequence of segments. ¹	IMS ODBA	IMS database accessed through a PowerExchange Listener
DDM	Natural data definition module (DDM). Provides long name and scale information.	ADABAS	Adabas file on z/OS
DDS	Data description specifications (DDS). Describes data attributes in file descriptions that are external to the application program that processes the data.	SEQ	DB2 for i5/OS data that is accessed sequentially
FDT	Field definition table (FDT). Each database file has a corresponding FDT, which defines the record structure and the content of each field in the physical file. For each field in the record, the FDT defines the level, name, length, format, options, and special field and descriptor attributes. An FDT provides short names and no scale information.	ADABAS	Adabas file
IDMS	IDMS copybook. Provides combined CA IDMS/DB schema and COBOL copybook information.	IMDS	CA IDMS/DB database
PREDICT	Predict data dictionary. Provides long name and scale information.	ADABAS	Adabas file on z/OS

Option	Description	Import into Data Maps Defined with Access Methods	Corresponding Data Source
TEXT	Text file, which is a delimited format file that uses an Informatica internal format.	ADABAS	Adabas file
1. When you import a DBD, PowerExchange provides full support for DBD statements that preceded the introduction of the IMS catalog in IMS 12. PowerExchange ignores all DBD enhancements that are related to the IMS catalog.			

Column Range

The following table describes the start and end column range for the copybook:

Option	Description
Start	The start column number. Valid values are from 1 through 999. For the COBOL copybook type, default is 7. For all other copybook types, default is 1.
End	The end column number. Valid values are from 1 through 999. For the COBOL and PL/I copybook types, default is 72. For all other copybook types, default is 80.

Note: You cannot set a column range for DBDs imported into IMS DL/1 batch data maps.

FDIC File Details

The following table describes the FDIC file details for DDM or PREDICT copybook types:

Option	Description
Database ID	The FDIC database ID. Valid values are from 1 through 99999.
File Number	The FDIC file number. Valid values are from 1 through 9999. Default is 1.

RELATED TOPICS:

- [“Importing a Copybook into a Data Map” on page 135](#)

Import Copybook - Field Property Details Dialog Box

View or change the double-byte character set (DBCS) code page value that is selected for each field that is defined with picture clause G (PIC G) in a copybook.

Use Data Map Default Codepage

Select this option to have PowerExchange use the default operating system code page for each field that is defined with PIC G in a copybook.

By default, this option is selected.

Use Selected Codepage

Select a code page for copybook fields that are defined with PIC G.

Note: Multibyte code pages are available for Datacom, DB2, VSAM KSDSs, sequential data sets, and flat files.

Each generated character is of type CHAR in the selected code page.

Import Copybook - Local Details Dialog Box

Enter information for the local copybook.

File Name

Enter the path and file name for the copybook, or click the Browse button to browse to the file.

Default is the last copybook imported into the data map from the local system.

Preview

Click **Preview** to view the copybook.

Concatenate Field Name and Description

For a DDS, select this option to concatenate the DDS field name and description.

Default is cleared.

RELATED TOPICS:

- [“Importing a Copybook into a Data Map” on page 135](#)

Import Copybook - Local DB2 Catalog Details Dialog Box

Enter information for the local DB2 catalog.

DB Instance

Enter the instance name of the DB2 database.

Location

Select **Local** for the location of the DB2 catalog. The catalog is on the system where the PowerExchange Navigator runs.

Table Name

Enter the table name for which you are importing a DB2 catalog, or click the Browse button to enter filter criteria for a table.

UserID

Enter a user ID that has authority to access the database.

Password

Enter a password for the specified user ID.

Save File Locally As

Enter the path and file name on the local system where a copy of the DB2 catalog is saved, or click the **Browse** button to browse to a file.

Null Indicator

For a DB2 unload file data map, enter the hexadecimal value of the null indicator that is used in the unload file.

Pad Variable

For a DB2 unload file data map, select this option if variables in the unload file are padded.

Note: This option corresponds to the BMC UNLOAD PLUS FIXEDVARCHAR.

Preview/Amend Columns

Displays the **DB2 Column Selection** dialog box, which enables you to view or select the columns in the table that will be accessed.

RELATED TOPICS:

- [“Importing a Copybook into a Data Map” on page 135](#)
- [“DB2 Column Selection Dialog Box” on page 149](#)

Import Copybook - Remote Details Dialog Box

Enter information for the remote copybook.

File Name

Enter the fully qualified file name of the copybook on the remote system.

For a DBD, enter the partitioned data set (PDS) file name and member in the following format:

HLQ.DTLDEMO (dbdname)

Note: The z/OS data set that you import must not be created in packed format using the PACK option in the data set profile.

Location

Location from which to import the copybook. Options are:

- **Local.** The copybook is located on the PowerExchange Navigator system.
- **Remote.** The copybook is located on a remote system.

If you select **Remote**, the **Location** list includes the locations that are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Select **local** if the copybook is located on the PowerExchange Navigator system

UserID

For z/OS or i5/OS access, enter an operating system user ID if the SECURITY statement specifies (1,x) or (2,x) in the DBMOVER member.

Password

For access to a copybook on a z/OS system or to a data description specification (DDS) on i5/OS, enter a password for the specified user ID or a valid PowerExchange passphrase.

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

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

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

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

Save File Locally As

Enter the path and file name on the local system where a copy of the copybook is saved, or click the Browse button to browse to a file.

Name

For a DDM, enter the DDM name.

Name Browse

For a DDM, click **Name Browse** to browse to the DDM.

Preview

Displays the copybook.

Not available for an IDMS copybook.

DBName

For an IDMS copybook, enter the IDMS database name.

Advanced

For an IDMS copybook, click **Advanced** to enter information in the **IDMS Advanced Properties** dialog box.

RELATED TOPICS:

- [“Importing a Copybook into a Data Map” on page 135](#)
- [“IDMS Advanced Properties Dialog Box” on page 150](#)

Import Copybook - Remote Datacom Details Dialog Box

Enter information for the remote Datacom copybook.

Table Name (Long)

Enter the table name.

Location

Location from which to import the Datacom copybook. Options are:

- **Local.** The copybook is located on the PowerExchange Navigator system.
- **Remote.** The copybook is located on a remote system.

If you select **Remote**, the **Location** list includes the locations that are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.

UserID

For z/OS, enter an operating system user ID, which is required if security is enabled.

Password

For z/OS, enter a password for the specified user ID.

For access to a DB2 for z/OS subsystem, you can enter a valid PowerExchange passphrase instead of a password. A PowerExchange passphrase can be from 9 to 128 characters in length and can contain the following characters:

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

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

Note: The first character is an apostrophe.

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

The allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

Note: A valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.

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

Save File Locally As

Enter the path and file name on the local system where a copy of the copybook is saved, or click the Browse button to browse to a file.

Dictionary UserID

Enter the Datacom dictionary user ID.

Password

Enter a password for the user ID.

Table Status

Read-only. Specifies the value **PROD**.

Copybook

Select the this option to enable selection of one of the following copybooks types:

- **COBOL**
- **PL/1**

By default, the **Copybook** option is cleared, and the default copybook type is **COBOL**.

Key

Read-only. Key information is imported.

Default is selected.

Elements

Read-only. Element information is imported.

Default is selected.

RELATED TOPICS:

- [“Importing a Copybook into a Data Map” on page 135](#)

Import Copybook - Remote DB2 Catalog Details Dialog Box

Enter information for the remote DB2 copybook.

DB Instance

Enter the name of the DB2 database instance.

Location

Location from which to import the DB2 catalog. Options are:

- **Local.** The catalog is located on the PowerExchange Navigator system.
- **Remote.** The catalog is located on a remote system.

If you select **Remote**, the **Location** list includes the locations that are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Select **local** in the list if the copybook is located on the PowerExchange Navigator system

Table Name

Enter the table name, or click the Browse button to browse to enter filter criteria for a table.

UserID

Enter a user ID that has the authority to access the database.

For databases on supported Linux, UNIX, or Windows systems, if you have enabled PowerExchange LDAP user authentication and disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the *PowerExchange Reference Manual*.

Password

Enter a password for the specified user ID.

To access to a DB2 catalog on z/OS, you can enter a valid PowerExchange passphrase instead of a password. A PowerExchange passphrase can be from 9 to 128 characters in length and can contain the following characters:

- Uppercase and lowercase letters
- The numbers 0 to 9
- Spaces
- The following special characters:
' - ; # \ , . / ! % & * () _ + { } : @ | < > ?

Note: The first character is an apostrophe.

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

The allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

Note: A valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.

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

Save File Locally As

Enter the path and file name on the local system where a copy of the copybook is saved, or click the Browse button to browse to a file.

Null Indicator

For a DB2 unload file data map, enter the hexadecimal value of the null indicator that is used in the unload file.

Pad Variable

For a DB2 unload file data map, select this option if variables are padded in the unload file.

Note: This option corresponds to the BMC UNLOAD PLUS FIXEDVARCHAR.

Preview/Amend Columns

Displays the **DB2 Column Selection** dialog box, which enables you to view or select the columns in the table to be accessed.

RELATED TOPICS:

- ["Importing a Copybook into a Data Map" on page 135](#)
- ["DB2 Column Selection Dialog Box" on page 149](#)

DB2 Column Selection Dialog Box

Enter information to select DB2 columns when you import a DB2 catalog.

Available Columns

Select or clear individual columns in the selected table.

Select All

Click **Select All** to select all columns.

Deselect All

Click **Deselect All** to clear all columns.

RELATED TOPICS:

- [“Importing an FDT into an Adabas Data Map” on page 139](#)

IDMS Advanced Properties Dialog Box

Enter information for the IDMS copybook.

Schema Name

Enter the IDMS database schema name.

Schema Version

Select the version of the IDMS database schema. Options are:

- HIGHEST
- LOWEST

Dictionary UserID

Enter the user ID to access the IDMS schema or sub-schema, if required.

Dictionary Password

Enter the password for the user ID.

Node Name

Enter the DDS node name of the node that contains the following dictionary:

```
//SYSIDMS
```

Dictionary Node

Enter the name of the node that contains the following dictionary for the IDMSSUBSC or IDMSCHEM utilities:

```
//SYSIPT
```

Import Copybook - Configuration Details Dialog Box

Select options to define the default actions to take for imported records, fields, and tables.

The following table describes the options:

Option	Description
Prompt on record import	Select this option to prompt for a record name when you import a record. Default is selected.
Prompt on field import	Select this option to prompt for a field name when you import a field. Default is cleared.
Prompt on table creation	Select this option to prompt for a table name when you create a table. Default is cleared.

Option	Description
Create table on each record imported	Select this option to create a table based on each imported record. Default is selected.
Create tables for DL1 hierarchical paths	For a DBD, select this option to create all tables for all DL/1 hierarchical paths. Default is cleared.
Create tables for IDMS hierarchical paths	For an IDMS copybook, select this option to create all tables for all IDMS hierarchical paths. Default is cleared. If you select this option, the Create expressions for IDMS set relationships option is unavailable.
Create expressions for IDMS set relationships	Create expressions to represent IDMS set relationships. Applications can use this metadata to determine the logical foreign key relationships among tables. When you select this option, PowerExchange generates the following fields for each record in the data map: <ul style="list-style-type: none"> - A field that calls the GetDatabaseKey function. - For each set of which the record is the owner, fields that call the GetDbKeyOfFirstMember and GetDbKeyOfLastMember functions. - For each of which the record is a member, fields that call the GetDbKeyOfOwner, GetDbKeyOfNextMember, and GetDbKeyOfPriorMember functions. If you select this option, the Create tables for IDMS hierarchical paths option is unavailable.
Refresh table columns for imported records	For COBOL or PL/I copybooks imported into an IDMS, IMS DL/1 batch, or IMS ODBA data map, select this option to update columns in tables that are based on records that are overwritten during the copybook import operation. Default is cleared.
Select first data redefinition	Select this option to use the first data definition if PowerExchange finds a REDEFINES clause. Default is cleared.
Start import automatically	Select this option to start the import automatically. Default is selected.
Action on duplicate record Action on duplicate field Action on duplicate table	Specifies the action to take if PowerExchange finds a duplicate record, field, or table during the import operation. The options are: <ul style="list-style-type: none"> - PROMPT. Prompts for the action to take. - UNIQUE NAME. Imports the record or segment using a unique name. - OVERWRITE. Overwrites the record or segment. - SKIP. Skips the record or segment. - APPEND. Appends the record or segment from the copybook to the record or segment in the data map. Default is PROMPT .

Record Definition and Duplicate Record Definition Dialog Boxes

Select an action for each imported record or segment.

If the copybook contains a record or field that duplicates a record or field in the data map, you can select options to indicate how to handle the duplicate item. For example, you can choose to import, overwrite, or skip the item, or stop the import.

The following table lists the options that you can select in the **Record Definition** and **Duplicate Record Definition** dialog boxes:

Option	Description
Import	Select this option to import the item. Default is selected.
Name	Enter a name for the imported item. Default is the name in the copybook file.
Skip	Select this option to skip the import of the item. Default is cleared.
Stop Import	Select this option to stop the import. Default is cleared.
Generate Unique Name	Select this option to generate a unique name for a duplicate item. Default is cleared.
Overwrite	Select this option to overwrite the item for which PowerExchange found a duplicate item. If you select this option, select the type of item to overwrite, which is one of the following: - Fields Only - Records Default is cleared.
Append	Select this option to append the duplicate item from the copybook to the item in the data map. Default is cleared.
Record Browse	Select a record in the data map to browse.
Apply to rest of import session	Select this option to apply the selected options to the rest of the import session. Default is cleared.

CHAPTER 5

Registration Groups and Capture Registrations

This chapter includes the following topics:

- [Registration Group and Capture Registration Overview, 153](#)
- [Registration Tag Names, 154](#)
- [Registration Groups, 157](#)
- [Adding a Registration Group, 158](#)
- [Adding a Capture Registration, 163](#)
- [Viewing a Capture Registration, 170](#)
- [Viewing or Editing Properties for a Registration Group, 171](#)
- [Viewing or Editing Properties for a Capture Registration, 173](#)
- [Viewing Elements for a Datacom Capture Registration, 178](#)
- [Viewing Paths for an IDMS Capture Registration, 178](#)
- [Changing the DB2 Columns in a Capture Registration, 178](#)
- [Deleting a Capture Registration, 179](#)
- [Deleting a Registration Group, 180](#)
- [Deleting All Capture Registrations in a Registration Group, 180](#)

Registration Group and Capture Registration Overview

For PowerExchange to capture changes, you must register source data sets, databases, segments, or tables for change data capture (CDC).

Registration groups and capture registrations define the source data for which you want PowerExchange to capture changes and are required for CDC processing.

A registration group defines the data source location, the data source type, the collection identifier, and, optionally, the user ID and password for accessing the source data. When you add a registration group, PowerExchange adds an extraction group and application group with the same name.

A registration group contains one or more capture registrations. A capture registration specifies a data source for which to capture changes. A capture registration also specifies the condense option to use and

the registration status. The capture registration name is the name that you specify when you create the registration.

When you add a capture registration, PowerExchange adds an extraction map for that registration. PowerExchange assigns a registration tag name to both the capture registration and the generated extraction map. If you manually add an additional extraction map for a capture registration, the extraction map has the same registration tag name as in its associated capture registration. The registration tag name format varies depending on the source type.

To create a capture registration for nonrelational sources such as Adabas, IMS, and VSAM sources, you must first create a data map for the source. For nonrelational sources, data maps provide the metadata used to define capture registrations and extraction maps for CDC.

Note: The PowerExchange Navigator does not create capture registrations for certain types of source objects for which change data cannot be captured. For more information, see the PowerExchange CDC guide that covers your data source type.

The CCT file contains information for capture registrations defined in the PowerExchange Navigator. The registration group filters the capture registrations from the CCT file for display in the PowerExchange Navigator.

After you add a capture registration, you can change some of its properties.

RELATED TOPICS:

- [“Registration Groups” on page 157](#)

Registration Tag Names

A registration tag name is a unique identifier that PowerExchange generates and assigns to a data source registered for CDC.

PowerExchange uses these registration tags to identify the change records of interest for CDC processing. The PowerExchange Logger for z/OS, PowerExchange Condense, and PowerExchange Logger for Linux, UNIX, and Windows store a registration tag name with each change record in their log or condense files.

For z/OS data sources that are processed by the z/OS ECCRs and PowerExchange Logger for z/OS, the registration tag name is also referred to as the EDMNAME.

When you create a capture registration, PowerExchange assigns the same registration tag name to the capture registration and the corresponding extraction map that PowerExchange generates. If you manually add another extraction map for the capture registration, this additional extraction map has the same registration tag name.

The registration tag name format varies depending on the source type. The following list describes the format of the registration tag name for each source type:

Note: In this list, the different types of values that compose the registration tag name are separated by underscores for legibility. The underscores do not appear in the tag names.

Adabas

ADAinstanceDBIDnnnnnFILEIDnnnnn

This registration tag name format is composed of the following values:

- *ADAinstance*. The prefix "ADA" followed by the collection identifier that you specified when creating the registration group.

- *DBIDnnnnn*. The Adabas database ID (DBID) number.
- *FILEIDnnnnn*. The Adabas file number.

Datcom table-based

DCMMUFnameDBIDdatabaseIDSTNshortTableNameVregVersion

This registration tag name format is composed of the following values:

- *DCMMUFname*. The prefix "DCM" followed by the Datcom MUF name that you specified when creating the registration group.
- *DBIDdatabaseID*. The Datcom database ID.
- *STNshortTablename*. The three-character short form of the table name that PowerExchange displays in the **Short Table Name** field of the Record Properties dialog box for the Datcom table.
- *VregVersion*. A registration version number that PowerExchange generates.

Example: DCMR11G4DBID00800STNTB1V1

DB2 for Linux, UNIX, and Windows

UDBdatabase_registrationName_n

This registration tag name format is composed of the following values:

- *UDBdatabase*. The prefix "UDB" followed by the DB2 for Linux, UNIX, and Windows database name that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- *n*. A generated version number.

DB2 for z/OS

DB2instance_registrationName_n

This registration tag name format is composed of the following values:

- *DB2instance*. The prefix "DB2" followed by the DB2 for z/OS database instance that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- *n*. A generated version number.

DB2 for i (i5/OS)

AS4instance_registrationName_n

This registration tag name format is composed of the following values:

- *AS4instance*. The prefix "AS4" followed by the collection identifier that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- *n*. A generated version number.

IDMS log-based

IDLlogsid_registrationName_n

This registration tag name format is composed of the following values:

- *IDLlogsid*. The prefix "IDL" followed by the Logsid value that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.

- *n*. A generated version number.

IMS log-based

```
IMLreconid_registrationName100000
```

This registration tag name format is composed of the following values:

- *IML reconid*. The prefix "IML" followed by the IMS RECON identifier that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- 100000. A static version number.

IMS synchronous

```
IMS.databaseName.segmentName
```

This registration tag name format is composed of the following values:

- *IMS.databaseName*. The prefix "IMS" followed by the IMS database name that you specified when creating the data map.
- *segmentName*. The IMS segment name.

Microsoft SQL Server

```
MSSinstance_registrationName_n
```

This registration tag name format is composed of the following values:

- *MSSinstance*. The prefix "MSS" followed by the instance value that is generated from the Microsoft SQL Server database name that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- *n*. A generated version number.

MySQL

```
MYSinstance_registrationName_n
```

This registration tag name format is composed of the following values:

- *MYSinstance*. The prefix "MYS" followed by the MySQL instance name that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- *n*. A generated version number.

Oracle

```
ORAinstance_registrationName_n
```

This registration tag name format is composed of the following values:

- *ORAinstance*. The prefix "ORA" followed by the collection identifier that you specified when creating the registration group.
- *registrationName*. The registration name that you assigned to the capture registration.
- *n*. A generated version number.

PostgreSQL

```
PGSinstance_registrationName_n
```

This registration tag name format is composed of the following values:

- *PGSinstance*. The prefix "PGS" followed by the PostgreSQL instance name that you specified when creating the registration group.
- *registrationName*. The registration name that you specified for the capture registration.
- *n*. A generated version number.

VSAM

The registration tag name that is assigned to a VSAM data set depends on the length of the VSAM data set name in the data map.

- If the VSAM data set name is less than or equal to 28 characters, the registration tag name has the following format:

VSAMdatasetName

For example, the data set name of USERID1.VSAM.EDMSRCV.MAKEITA is 28 characters in length. For this data set, PowerExchange generates the following registration tag name:

VSAMUSERID1.VSAM.EDMSRCV.MAKEITA

- If the VSAM data set name is equal to or greater than 29 characters, PowerExchange truncates the data set name to 23 characters and appends an ascending numeric value, *nnnnn*, to create the registration tag name. The registration tag name has the following format :

VSAMdataset_namennnnn

For example, the data set name of USERID1.VSAM.EDMSRCV.MAKEITAB is 29 characters in length. For this data set, PowerExchange generates the following registration tag name:

VSAMUSERID1.VSAM.EDMSRCV.MA00001

- if you have subsequent data set names that are longer than 28 characters and have the same first 23 characters as a previous data set name that is longer than 28 characters, the PowerExchange Navigator increments the appended numeric value *nnnnn* by 1:

VSAMUSERID1.VSAM.EDMSRCV.MAnnnnn+1

For example, if the data set name is USERID1.VSAM.EDMSRCV.MAKEITAV, PowerExchange generates the following registration tag name:

VSAMUSERID1.VSAM.EDMSRCV.MA00002

Registration Groups

Before you can add capture registrations, you must add a registration group.

The group defines the data source location, the data source type, the collection identifier, and, optionally, the user ID and password for accessing the source data. When you add a registration group, PowerExchange also creates an extraction group and application group with the same name. You can also manually add an extraction group or application group.

PowerExchange stores registration group, extraction group, and application group information on the Windows system where the groups are added. The groups are unique for that system.

When you delete a registration group, extraction group, or application group, PowerExchange deletes local group information on the PowerExchange Navigator system. PowerExchange does not delete the capture registrations, extraction maps, or applications in the deleted group. Also, when you delete an application group or extraction group that PowerExchange generated, PowerExchange does not delete the registration group that generated the creation of this application group or extraction group.

If you add another group with the same properties as a group that you deleted, the new group contains any existing capture registrations, extraction maps, or applications for the deleted group.

RELATED TOPICS:

- [“Adding a Registration Group” on page 158](#)
- [“Adding an Application Group” on page 323](#)
- [“Adding an Extraction Group” on page 184](#)

Adding a Registration Group

A registration group is a named group of capture registrations that have the same source type and instance.

You must add a registration group before you can add capture registrations. You can add capture registrations immediately after defining the registration group or later.

1. On the **Resources** tab in the **Resource Explorer**, right-click **Registration Groups** and click **Add Registration Group**.
2. In the **Add Registration Group** dialog box, enter the following information:

Name

Enter a user-defined name for the registration group. This value can be an alphanumeric string of up to 16 characters in length. You cannot enter spaces or special characters.

Location

Select the location of the source from which you want to capture changes. This field applies to all source types except IDMS_L.

The location is a node name that is defined in a NODE statement in the dbmover.cfg configuration file on the PowerExchange Navigator system, or the value "local" for the local Windows system where the Navigator runs.

Default is **local**. Select **local** if the source is on the PowerExchange Navigator system.

Note: For a remote Microsoft SQL Server source, specify the actual server name. Do not use a local alias to identify the server. PowerExchange does not support the use of local aliases when connecting to a remote server and creating SQL Server publications at registration creation.

Type

Select the data source type. Options are:

- **ADABAS** for Adabas files
- **AS4** for DB2 for i (i5/OS) tables
- **DATACOM** for CA Datacom/DB tables
- **DB2** for DB2 for z/OS tables
- **DB2UDB** for DB2 for Linux, UNIX, and Windows tables
- **IDMS_L** for IDMS tables to be registered for IDMS log-based CDC
- **IMS** for IMS databases
- **MSSql** for Microsoft SQL Server tables

Note: For the PowerExchange Navigator to create capture registrations for Microsoft SQL Server sources, the license.key file on the PowerExchange Navigator system must contain a key that authorizes use of SQL Server CDC. Otherwise, when you try to add a capture registration for a SQL Server source, the PowerExchange Navigator fails with a license key error. This failure occurs even if the PowerExchange Listener is running with a license key that includes SQL Server CDC.

- **MySQL** for MySQL tables
- **ORACLE** for Oracle tables
- **PostgreSQL** for PostgreSQL tables
- **VSAM** for VSAM ESDS, KSDS, RRDS, and VRRDS data sets

UserID

Enter a user ID that allows access to the source node that is specified in the **Location** field. This field applies to all source types except IDMS_L.

The type of user ID depends on the source type and PowerExchange security settings:

- If the source is on an IBM i or z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x).
- For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the *PowerExchange Reference Manual*.

Note: If you use Microsoft NTLM and Active Directory authentication to control access to a Microsoft SQL Server source, enter the user ID in the format `domain\user_name`. Then in the **Password** field, enter a valid password for this user ID.

- For a source on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the *PowerExchange Reference Manual*.

Password

Enter a password for the specified user ID that allows access to the source node. This field applies to all source types except IDMS_L.

If the source is on an IBM i or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An IBM i passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

- Uppercase and lowercase letters
- The numbers 0 to 9
- Spaces
- The following special characters:
`' - ; # \ , . / ! % & * () _ + { } : @ | < > ?`

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

If you specify a user ID but omit the password or passphrase, when you open the registration group, the PowerExchange Navigator prompts you for a password in the **Missing Password** dialog box.

Source-specific fields

Depending on the source type that you entered in the **Type** field, some source-specific fields are displayed. The following table describes these fields by source type:

Source Type	Source-Specific Fields
ADABAS	In the Collection Identifier field, enter the instance name.
AS4 (DB2 for i)	In the Collection Identifier field, enter the instance name specified in the INST parameter of the AS4J CAPI_CONNECTION statement in the DBMOVER member of the <i>dtllib/CFG</i> file on the source system.
DATACOM	In the MUF Name field, enter the name of Multi-User Facility (MUF).
DB2 (DB2 for z/OS)	In the Database Instance box, enter the DB2 subsystem ID or the DB2 data-sharing group name.
DB2UDB (DB2 for Linux, UNIX, and Windows)	In the Database field, enter the name of the DB2 database.
IDMS_L	<ul style="list-style-type: none"> - In the Registration Location list, select the location where the capture registrations will be stored. The location is a node name that is defined in a NODE statement in the <i>dbmover.cfg</i> configuration file on the PowerExchange Navigator system. - In the UserID and Password fields under the registration location, enter a user name and password that allow access to that location. If the registrations are on the z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x). - In the Source Map Location list, select the location where the data maps for IDMS source tables are located. - In the UserID and Password fields under the source map location, enter a user name and password that allow access to that location. If the maps are on the z/OS system and PowerExchange security is enabled, enter an operating system user ID. - In the Logsid field, enter the value specified in the LOGSID statement in the DBMOVER configuration file on the source system.
IMS	In the Recon Identifier field, enter one of the following values: <ul style="list-style-type: none"> - For IMS synchronous CDC, enter the IMS subsystem ID. - For IMS log-based CDC, enter the value that is specified in the IMSID statement in the DBMOVER configuration member on the z/OS system where the RECON data sets are located.

Source Type	Source-Specific Fields
MSSql	<ul style="list-style-type: none"> - In the Database Server list, select the server where the SQL Server source is located. You can optionally append the SQL Server instance name and also a port number if you use a port other than the default port number of 1433. Use the following format: <code>server[\instance_name][,port_number]</code> For example: MSSQL001\MSSQLINST5,6983 If you include the instance name, the PowerExchange Navigator can use the SQL Server Browser, provided that it is active, to find a port for connecting to SQL Server if you did not specify the port or if you specified the port but the first connection attempt on that port failed. Note: If you do not use the default port and do not specify an instance or port number in this field, attempts to create capture registrations for source tables for this group will fail. - In the Database Name list, select or enter the name of a SQL Server publication database. - In the Instance field, optionally enter a unique user-defined identifier for the specified database server and database name combination. Maximum length is seven characters. This identifier is used to identify a set of registrations for the publication database. This identifier is also incorporated into the names of the extraction maps that are generated for capture registrations in the registration group. If you use the PowerExchange Logger for Linux, UNIX, and Windows, ensure that the instance identifier matches the DBID parameter value in the Logger configuration file. If you do not enter an Instance value, PowerExchange generates a unique instance identifier that is composed of all or part of the publication database name followed by a 3-digit number if a number is required to make the identifier unique. This user-defined instance identifier is useful in migration scenarios. If you need to migrate change capture from one environment to another, such as from test to production, and you do <i>not</i> define an instance identifier, PowerExchange uses a generated instance identifier in the new environment. The generated instance identifier might be different from the one in the original environment. To avoid having to update the extraction map names in PowerCenter workflows and edit the DBID parameter value for the PowerExchange Logger, enter an instance identifier that matches the instance identifier in the original environment when creating registrations for the new environment. Tip: In this migration scenario, ensure that the paths in CAPT_PATH and CAPT_XTRA statements in the dbmover.cfg configuration file are unique in the original environment and new environment.
MySQL	<ul style="list-style-type: none"> - In the Database Server field, enter the server name or IP address of the MySQL server where the source database is located. If the MySQL server is on the local host where the Navigator runs, you can enter localhost. Optionally, you can append the port number if you use a port other than the default port of 3306. Use the following format: <code>server[,port_number]</code> For example: "localhost,3400" - In the Instance field, optionally enter a unique user-defined identifier for the MySQL server that you specified in the Database Server field. Maximum length is seven characters. This identifier is used to identify a set of registrations for tables in the database on the MySQL server. This identifier is also incorporated into the names of the extraction maps that are generated for capture registrations in the registration group. If you use the PowerExchange Logger for Linux, UNIX, and Windows, ensure that this instance identifier matches the DBID parameter value in the Logger configuration file. If you do not enter an Instance value, PowerExchange generates a unique instance identifier that consists of first four characters of the database server name followed by a three-digit number if a number is required to make the identifier unique.

Source Type	Source-Specific Fields
ORACLE	In the Collection Identifier field, enter the Oracle instance identifier specified in the ORACLEID statement in the DBMOVER configuration file on the source system.
PostgreSQL	<ul style="list-style-type: none"> - In the Database Server list, select the server where the PostgreSQL source is located. You can optionally append a port number if you use a port other than the default port number of 5432. Use the following format: <i>server[,port_number]</i> For example: PGSL001,6983 Note: If the server does not use the default port and you do not specify a port number in this field, attempts to create capture registrations for this registration group will fail. - In the Database Name list, select the name of the PostgreSQL source database on the specified server. - In the Instance field, optionally enter a unique user-defined identifier for the specified database server and database name combination. Maximum length is seven characters. This identifier is used to identify a set of registrations for the database. This identifier is also incorporated into the names of the extraction maps that are generated for capture registrations in the registration group. If you use the PowerExchange Logger for Linux, UNIX, and Windows, ensure that the instance identifier matches the DBID parameter value in the Logger configuration file. If you do not enter this value, PowerExchange generates a unique instance identifier that is composed of the first four characters of the database server name followed by a 3-digit number if a number is required to make the identifier unique.
VSAM	In the Collection Identifier field, enter the instance name.

Add Registration

Select this option to add capture registrations to the group. Clear this option to add only the registration group.

By default, this option is selected.

3. Add capture registrations to the group or complete the registration group definition, as follows:
 - If you selected the **Add Registration** option, click **Next** and proceed to [“Adding a Capture Registration” on page 163](#).
 - If you cleared the **Add Registration** option, click **Finish**. The **Registration Group** tab in the **Resource Inspector** displays the properties for the registration group. Additionally, if the PowerExchange Listener on the source system is running, the **Registration Group** tab in the **Resource Explorer** displays any existing capture registrations for the source instance associated with the registration group.

RELATED TOPICS:

- [“Registration Groups” on page 157](#)
- [“Adding a Capture Registration” on page 163](#)

Adding a Capture Registration

A capture registration defines a source object such as an IMS segment, DB2 table, or VSAM data set for PowerExchange change capture processing.

PowerExchange stores capture registrations in the CCT file on the source system.

Before adding a capture registration, you must define the registration group. Within a registration group, you can add a capture registration for a specific source only once. The capture registration provides source object metadata and identifies the columns for which you want to capture change data.

Note: You cannot register a source object that has column names greater than 128 characters in length.

1. Open a registration group.
2. On the **Registration Group** tab in the **Resource Explorer**, right-click the registration group and click **Add Capture Registration**.
3. In the **Add Capture Registration - Name and Table Filter** dialog box, enter a capture registration name and optional information as needed.

The following table describes the fields in this dialog box:

Field	Description
Name	Required. A user-defined name for the capture registration. The name must be from one to eight lowercase alphanumeric characters in length and begin with a letter.
Table Filter	<p>Optional. To filter the tables, enter values in any of the following filter fields, depending on the source type:</p> <ul style="list-style-type: none"> - For Adabas, enter filter information in the Schema and Table boxes. The maximum length for both fields is 128 characters. - For Datacom, enter filter information in the Schema and Table boxes. The maximum length for both fields is 128 characters. - For all DB2 types, enter filter information in the Creator/Schema, DBName/Definer, and Table boxes. For DB2 for i5/OS, maximum lengths are: 10 characters for the Creator/Schema field and 32 characters for the Table field For DB2 for z/OS, the maximum length of both the Creator/Schema and Table fields is 128 characters. - For IDMS, enter filter information in the Schema and Table boxes. The maximum length of both fields is 128 characters. - For IMS, enter filter information in the Schema, MapName, and Table boxes. In the MapName field, enter a PowerExchange IMS data map name. Maximum lengths are: 10 characters for the Schema and MapName values and 128 characters for the Table value. - For MSSql, enter filter information in the Owner and Table boxes. The maximum length for both fields is 128 characters. - For MySQL, enter filter information in the Schema and Table fields. The maximum length for both fields is 64 characters. In MySQL, the schema name is equivalent to the database name. - For Oracle, enter filter information in the Schema and Table fields. The maximum length for both fields is 128 characters. - For PostgreSQL, enter filter information in the Schema and Table fields. The maximum length for both fields is 63 characters. - For VSAM, enter filter information in the Schema, MapName, and Table boxes. In the MapName box, enter the name of a PowerExchange VSAM data map. Maximum lengths are: 10 characters for the Schema and MapName fields and 128 characters for the Table field. <p>In any filter field, you can include the following wildcards to create a mask:</p> <ul style="list-style-type: none"> - An asterisk (*) to represent one or more characters. - A question mark (?) to represent a single character.
Respect Case	<p>By default, PowerExchange converts text that you enter in the filter boxes to the standard case for the data source. For example, on Oracle, uppercase is the standard. A value of <code>scott</code> or <code>SCOTT</code> would both produce <code>SCOTT</code>.</p> <p>Select the Respect Case option for the filter fields for which you do <i>not</i> want PowerExchange to use the standard case for the data source.</p>
Escape Character	<p>Enter the escape character to use in the filter boxes to delimit an asterisk (*) or question mark (?) that is used a literal value instead of a wildcard.</p> <p>For example, if you specify an escape character of <code>~</code>, a filter value of <code>tab*</code> returns all tables that have table names beginning with <code>tab</code>. A filter value of <code>tab~*</code> returns the table that has the name <code>tab*</code>.</p> <p>Note: Do not use an escape character with multibyte-enabled data sources.</p>

4. Click **Next**.

The **Add Capture Registration - Tables and Columns** dialog box appears.

If you entered filter criteria, the **Tables** list displays the tables that matched your filter criteria. Otherwise, the **Tables** list displays all tables.

The **Tables** list can include the one or more of the following fields, depending on the source type:

Column	Description
Creator Name	The creator name for the table
Schema	The schema name
Name or Table	The table name
Type	The object type, such as TABLE
DataSet Name	The data set name

5. In the **Tables** list, double-click a table to select it.

Note: If the table contains column names that are longer than 128 characters, an error message is issued and you cannot register the table.

The **Columns** list displays the columns in the selected table. The list can include the following fields, depending on the source type:

Field	Description
Select Column	A check box next to each column for selecting columns for registration. Note: Index columns are selected by default.
Name	The column name.
CCSID	For a DB2 source, the coded character set identifier (CCSID) related to the column.
PWXCP	For a DB2 source, the PowerExchange internal representation of the code page.
Codepage	For a DB2 source, the code page related to the column.

6. In the **Columns** list, select the columns for which you want PowerExchange to capture changes.

To select columns individually, select the **Select Column** check box next to each column. To select all columns, click **Select All Columns** or **Select all and notify changes**. The following table describes the options for selecting all columns:

Check Box	Description
Select All Columns	<p>Available for source types that support selective column capture. Selects all columns in the Columns list for change capture.</p> <p>After you add the registration, any columns that are added to the source table are ignored until you edit the capture registration to select them.</p>
Select all and notify changes	<p>Available for DB2 and Oracle sources. If you select this option, PowerExchange captures changes for all columns. Any change to the schema for the table causes PowerExchange CDC to fail and log an error message.</p> <p>For a DB2 for z/OS source, the DB2 ECCR abnormally ends after it reads the first change record for the table after the schema change.</p> <p>For an Oracle source, Oracle CDC fails and logs an error message in the following situations:</p> <ul style="list-style-type: none"> - If a change record for a table that you registered for capture contains a column that you did not register for capture - If a change record does not contain a column that you registered for capture <p>For Oracle CDC, if a definition for a table changes in a way that is compatible with the PowerExchange capture registration, Oracle CDC continues to capture changes for that table.</p> <p>For example, if the length of a character column decreases but the capture registration does not reflect this change, Oracle CDC continues to capture changes for the table.</p> <p>Conversely, if the datatype of a column changes from numeric to character without a change in the capture registration, Oracle CDC continues to capture changes for the table until it encounters the first change record that contains nonnumeric data for the column. When Oracle CDC encounters a change record containing nonnumeric data for the column, it fails and logs an error message.</p> <p>Note: If you clear the Select all and notify changes option, the check boxes appear next to the columns are cleared. To select the columns again, you must select their check boxes.</p>

Tip: When selecting columns, consider the following recommended practices:

- Informatica recommends that you select all columns for which you want to capture change data because a registration can be defined only once in a registration group for specific source. You can edit the corresponding extraction map to deselect columns.
- A key symbol indicates that a column is an index column that Condense processing uses. If you clear the check box for an index column, you might compromise the uniqueness of the row.

7. Click **Next**.
8. In the **Add Capture Registration - Type** dialog box, enter additional information about the capture registration.

The following table describes the fields in this dialog box:

Field	Source Type	Description
Condense	All	<p>Whether PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows is used and the type of PowerExchange Condense processing on i5/OS or z/OS. Options are:</p> <ul style="list-style-type: none"> - None. PowerExchange does not use PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows to process changes for the registered source. - Part. PowerExchange uses PowerExchange Condense or the PowerExchange Logger to process changes for the registered source. PowerExchange Condense uses partial condense processing. Informatica recommends this option because it maintains transactional consistency. - Full. PowerExchange uses PowerExchange Condense full condense processing for i5/OS or z/OS sources but some limitations apply. This option does not maintain transactional consistency. <p>You can change the Condense option later without incrementing the registration version number. For more information, see "Condense Options" on page 169.</p>
Status	All	<p>The status of the capture registration with regard to availability for change capture use. Options are:</p> <ul style="list-style-type: none"> - Inactive. PowerExchange does not yet use the registration to capture changes for the source. This option is the initial status setting. You must manually set the option to Active to make the registration eligible for change capture use. - Active. PowerExchange can use the registration to capture changes for the source. <p>Default is Inactive.</p> <p>Note: You can change the status from Inactive to Active later to make it eligible for change capture processing.</p> <p>For more information about status settings, see "Status Options" on page 168.</p>
Comment	All	User-defined comments about the capture registration.
Database ID	Adabas	The DBID of the Adabas database that contains the file from which changes are captured.
File Number	Adabas	The file number of the Adabas file in the database identified by the Database ID field.
Primary Dataset Name	IMS synchronous	<p>The name of the primary data set for the IMS database. This name can be up to 44 characters in length. Depending on the database type, the primary data set is one of the following:</p> <ul style="list-style-type: none"> - Full-function database: The primary data set is the data set that contains the primary (or only) data set group for the database. - Fast Path database: The primary data set is the data set that contains the first (or only) area for the database. - HALDB database: The primary data set is the data set prefix for the first partition, which is used for the data set name.

Field	Source Type	Description
Supplement Log Group Name	Oracle	The name of the supplemental log group that you will create for the source. PowerExchange generates DDL for creating supplemental log groups when you complete the capture registration. You must define a supplemental log group for each table for which you want PowerExchange to capture Oracle change data. This supplemental log group must contain all table columns for which change data is to be captured.
Execute DDL now	Oracle	For an Oracle source, select this option to run DDL to create a supplemental log group when you complete the capture registration. Clear this option to run the DDL later. You must also enter a value in the Supplement Log Group Name box. Note: The GRANT statements in the ORACAPT.SQL file for the Oracle capture user do not include the authority that is required to run the DDL.

9. Click **Finish**.
10. For an Oracle source, in the **Save Supplement Log Group SQL to File** dialog box, enter a file name in the **File Name** field for the file that contains the generated DDL and click **Save**.

PowerExchange saves the DDL to the specified file. By default, the file is located in the PowerExchange root installation directory.

The PowerExchange Listener writes the capture registration to the CCT file on the source system. In the PowerExchange Navigator, the capture registration appears in the list of capture registrations on the **Registration Group** tab in the **Resource Explorer**. Also, the registration appears at the following locations in the interface:

- The **Capture Registration** window displays the columns in the capture registration.
- The **Capture Registration** tab in the **Resource Inspector** displays the properties for the capture registration. Use the arrows to toggle between the **Capture Registration** and **Registration Group** tabs in the **Resource Inspector**.
- The status bar displays information about the open registration group and capture registration.

RELATED TOPICS:

- [“Viewing a Capture Registration” on page 170](#)
- [“Status Options” on page 168](#)
- [“Condense Options” on page 169](#)

Status Options

The capture registration **Status** option indicates if the registration is available for change capture use, temporarily suspended, or retained only for history.

When you create a capture registration, its initial **Status** setting is **Inactive**.

From the PowerExchange Navigator, you can change the status with limitations. The following one-way transitions are allowable:

- **Inactive > Active > History**
- **Suspended > History**

Use the following descriptions to determine which status option to use:

Inactive

The initial status that prevents data from being captured for the source before the target is materialized.

Active

The status that allows changes to be captured for the registered source based on the capture registration.

History

The status that permanently disables the capture registration for change capture use but that retains the capture registration for audit and historical purposes. After you set the registration status to History, you cannot perform the following actions:

- Use the same registration name again unless you delete the registration and create another registration with that name.
- Set the registration to Active again. To resume change capture for the source, you must create another capture registration.

Suspended

The status that is set after you run the PWXUCREG utility SUSPEND_REGISTRATION command to temporarily suspend change capture for the registered source. You cannot select this status from the PowerExchange Navigator. Only the following types of CDC data sources can have this status: Adabas, Datacom table-based, IDMS log-based, and IMS log-based. While the status is Suspended, PowerExchange does not capture changes for the registered source. You can reactivate the registration for change capture use by issuing a different PWXUCREG utility command.

Note: You can change the status of Suspended back to Active by issuing the ACTIVATE_REGISTRATION command with the PWXUCREG utility. For more information, see the *PowerExchange Utilities Guide*.

Condense Options

The **Condense** option in a capture registration controls whether you can use PowerExchange Condense or PowerExchange Logger for Linux, UNIX, and Windows processing for the registered source. For PowerExchange Condense on i5/OS or z/OS, the option also indicates the type of condense processing.

From the PowerExchange Navigator, you can edit the **Condense** option for an existing capture registration without versioning the registration. For example, you can switch from None to Part or Full to Part.

The Condense options are:

None

The capture registration is not eligible for use by PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows.

Part

The capture registration is eligible for use by PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows. PowerExchange Condense uses the registration for partial condense processing.

With this option, only the changes in UOWs that complete successfully on the source are written to PowerExchange Condense condense files or PowerExchange Logger log files. PowerExchange writes all changes for columns of CDC interest, not just the latest changes, in chronological order based on UOW end time. This option maintains transactional consistency.

Informatica recommends that you select the **Part** option even if you do not plan to use PowerExchange Condense or the PowerExchange Logger. You can implement PowerExchange Condense or the PowerExchange Logger later without having to update the registration.

Full

The capture registration is eligible for use by PowerExchange Condense on i5/OS or z/OS for full condense processing.

PowerExchange Condense accumulates changes in keyed condense files by having later changes replace earlier changes. This option does not maintain transactional consistency.

The following restrictions apply to full condense processing:

- For Adabas and IDMS log-based CDC sources, full condense processing is not available.
- For PowerExchange Condense on i5/OS, full condense processing is available only for tables that have primary keys and DDS files that are defined with a unique key.
- For PowerExchange Condense on z/OS, full condense processing is available for tables or data maps that specify key columns.

Use this option only if the source has a unique key. PowerExchange Condense uses the unique key as part of the key in the condense file to maintain a single before and after image.

Tip: To be allowed to update a unique key while using full condense processing, you must specify `KEY_CHANGE_ALW=Y` in the PowerExchange Condense configuration file. With this setting, PowerExchange Condense ignores the key update and continues full condense processing.

Viewing a Capture Registration

You can view the columns in a capture registration.

1. Open the registration group that contains the capture registration.
2. Open the capture registration that you want to view.

The **Capture Registration** window appears. The following table describes the fields that are displayed for each registered column, depending on the source type:

Field	Description
Name	The column name
Table Name	The name of the table that contains the column
Column Num	The column number
Type	The column datatype
Length	The column length
Scale	The data scale

Field	Description
Key	YES or NO - to indicate whether the column is an index key column
CCSID	For a DB2 source, the coded character set identifier (CCSID) for the code page used for the column
PWXCP	For a DB2 source, the PowerExchange code page used for the column
Codepage	For a DB2 source, the code page used for the column
Nullable	YES or NO - to indicate whether the column is nullable.

- To sort the columns in alphabetical order by column name, click the **Name** column heading in a **Capture Registration** window.

Viewing or Editing Properties for a Registration Group

You can view or edit properties for a registration group.

You can edit the registration group name, source location, or the user ID and password for accessing the source. You cannot change the database type.

- Open the registration group.

The following table describes the properties that appear on the **Registration Group** tab in the **Resource Inspector**, depending on the data source type:

Property	Description
Name	User-defined name for the registration group.
Access Method	Read-only. Access method associated with the data source type.
Location	Location of the source for which you want to capture changes. The location value is a node name in a NODE statement in the dbmover.cfg configuration file on the PowerExchange Navigator system.
UserID	A user ID that allows access to the source. The type of user ID depends on the source type and PowerExchange security settings: <ul style="list-style-type: none"> - If the source is on an i5/OS or z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x). - For a DB2 LUW, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the <i>PowerExchange Reference Manual</i>.

Property	Description
Password	<p>A password for the specified user ID.</p> <p>If the data source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:</p> <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p>
Instance	<p>For an Adabas, DB2 for i5/OS, Oracle, or VSAM source, the instance name is the collection identifier that you entered when defining the registration group.</p> <p>For a DB2 for z/OS source, the instance name is the database instance name.</p> <p>For a Microsoft SQL Server source, the instance name is the Instance value that you entered when creating the registration group. If you did not enter an instance name, this value is a generated value that is composed of the first four characters of the database name followed by an ascending number, which starts at 000.</p> <p>For a MySQL or PostgreSQL source, the instance name is the Instance value that you entered when creating the registration group. If you did not enter an instance name, this value is a generated value that consists of the first four characters of the database server name followed by an ascending number, which starts at 000.</p>
MUF Name	For a CA Datacom/DB source, the name of the Multi-User Facility (MUF).
Server	For a Microsoft SQL Server source, the name of the publication server. For a PostgreSQL source, the name of the database server.
MySQL Server	For a MySQL source, the name of the database server.
Database	For a DB2 LUW, Microsoft SQL Server, or PostgreSQL source, the name of the database.
DB Name	For an IDMS log-based source, the name of the database.
Recon Identifier	For an IMS source, the RECON identifier for the database.
Oracle SID	Read-only. For an Oracle source, the system identifier (SID) for the database.

2. To edit a property, click a property in the **Resource Inspector** and enter a value.

After you edit a property, a red square appears next to the changed property.

Tip: To undo a change before you apply it, click **Reset**.

3. To apply a change, click **Apply**.

If you edit the registration group name, you are prompted to confirm the change. If the new name matches the name of an existing registration group, you are prompted to overwrite the existing group.

If you change the location to a location that is not valid for the instance, PowerExchange displays an error message.

If you change the instance to an instance that does not exist, PowerExchange displays an error message.

4. Click **Yes** to confirm any changes.

The **Resource Inspector** updates any fields that you edited.

RELATED TOPICS:

- [“Adding a Registration Group” on page 158](#)

Viewing or Editing Properties for a Capture Registration

After you register a source for change capture, you can edit some capture registration properties.

The following properties can be edited:

- **Status.** For all source types, you can change the status setting for a capture registration. The PowerExchange Navigator allows the following status transitions: Inactive to Active, Active to History, and Suspended to History. You must use the PWXUCREG utility to change a registration status from Active to Suspended and from Suspended back to Active.
- **Condense.** For all source types, you can change the Condense option for a capture registration from None to Part or from Part to None. For i5/OS and z/OS data sources that have keys, you can also change the Condense option from None or Part to Full or from Full or None to Part.
- **Comments.** For all source types, you can add or edit comments for the capture registration.

For a DB2 data source, you can also add or remove columns.

If you need to make any other changes to a capture registration, you must re-create the registration.

Warning: Editing an active capture registration can have various implications to CDC.

1. Open the registration group.
2. Open the capture registration.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays, depending on the data source type, for the capture registration.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for all data source types:

Property	Description
Name	Read only. The user-defined name for the capture registration.
Type	Read only. The PowerExchange capture type. Options are: <ul style="list-style-type: none"> - Log-Based. PowerExchange captures changes from database logs. - Synchronous. PowerExchange captures changes in a synchronous manner.
Status	The status of the capture registration with regard to availability for change capture use. Options are: <ul style="list-style-type: none"> - Inactive. PowerExchange cannot yet use the registration to capture changes for the source. This option is the initial status setting when you create the registration. You must manually set the option to Active to make the registration eligible for change capture use. - Active. PowerExchange can use the registration to capture changes for the source. - History. PowerExchange has permanently stopped using the registration for change capture. Select this option if the registration is obsolete and you want to retain it for audit purposes or delete it. After you set the status to History, you cannot make the registration active again. - Suspended. PowerExchange change capture processing is temporarily suspended for the registered source. This status is set by the PWXUCREG utility SUSPEND_REGISTRATION command and is available only for an Adabas, Datacom table-based, IDMS log-based, or IMS log-based CDC source. You can change this status to History from the PowerExchange Navigator. However, to change this status back to Active, you must use the PWXUCREG utility ACTIVATE_REGISTRATION command. <p>For more information, see "Status Options" on page 168.</p>
Suspend Time	Read only. The current system timestamp, without adjustment for local time, that indicates when the PWXUCREG utility SUSPEND_REGISTRATION command was processed to temporarily suspend the registration for change capture use. The timestamp format is YYYY/MM/DD HH:MM:SS.ssssss, where ssssss are fractional seconds. This property is blank if the registration has never been suspended. <p>This property is displayed for Adabas, Datacom table-based, IDMS log-based, and IMS log-based CDC sources because the PWXUCREG utility supports only these sources. This property is not displayed for any other source types.</p> <p>For more information about the PWXUCREG suspension time, see the <i>PowerExchange Utilities Guide</i>.</p>
Active Time	Read only. The current system timestamp, without adjustment for local time, that indicates when the PWXUCREG utility ACTIVATE_REGISTRATION command was processed to reactivate the registration for change capture use. The timestamp format is YYYY/MM/DD HH:MM:SS.ssssss, where ssssss are fractional seconds. <p>This field is blank if the registration status was set to Active for the first time from the PowerExchange Navigator or if the registration has never been activated or reactivated with the PWXUCREG utility. This field remains blank until you issue a PWXUCREG utility SUSPEND_REGISTRATION command followed by an ACTIVATE_REGISTRATION command.</p> <p>This property is displayed for Adabas, Datacom table-based, IDMS log-based, and IMS log-based CDC sources because the PWXUCREG utility supports only these sources. This property is not displayed for any other source types.</p> <p>For more information about the PWXUCREG activation time, see the <i>PowerExchange Utilities Guide</i>.</p>

Property	Description
Condense	<p>Whether PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows is used and the type of PowerExchange Condense processing on i5/OS or z/OS. Options are:</p> <ul style="list-style-type: none"> - None. PowerExchange does not use PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows to process changes for the registered source. - Part. PowerExchange uses PowerExchange Condense or the PowerExchange Logger to process changes for the registered source. PowerExchange Condense uses partial condense processing. This option maintains transactional consistency and is recommended by Informatica. - Full. PowerExchange uses PowerExchange Condense full condense processing for i5/OS or z/OS sources but some limitations apply. This option does not maintain transactional consistency. <p>For more information, see "Condense Options" on page 169.</p>
Version	Read only. The registration version number.
Capture Type	Read only. The data source type.
Comments	User-defined comments about the capture registration.
Tag	Read only. A PowerExchange-generated tag name that is assigned to the registered source and that is used to identify the change records of CDC interest.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for Adabas data sources:

Property	Description
Database ID	The DBID of the database.
File Number	The file number of the database.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for CA Datacom/DB data sources:

Property	Description
Database ID	The database ID.
Short Table Name	The short name of the table.
Table ID	The internal table identifier.
Record Size	The record size.
Table Recovery	An option that indicates whether recovery facilities of CA DATACOM/DB are used for this table. Valid values are YES and NO.
Element Count	The number of elements.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for DB2 for i5/OS data sources:

Property	Description
File Name	The file name.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for IDMS log-based data sources:

Property	Description
Area Name	The name of the IDMS area that contains a set of pages that store the database records.
SubSchema Type	The name of the subschema. A subschema provides a program view of the database and is used at runtime. It includes data elements, record types, set types, areas, and logical-record components. Valid characters are ASCII characters and the pound and yen sign.
DB Name	The name of the IDMS database. Valid characters are ASCII characters and the pound and yen sign.
Base Record	The IDMS base record name.
Location Mode	One of the following modes that determine how a record is assigned to a particular page in the area: - C . Indicates CALC mode. - V . Indicates VIA mode. - D . Indicates DIRECT mode.
Base Record ID	The IDMS base record identifier.
Compressed	Indicates whether IDMS native compression is used to compress records. Valid values are YES and NO.
Presspack Comp	Indicates whether the IDMS Presspack compression tool is used to compress records. Valid values are YES and NO.
Presspack Dctable	If Presspack compression is used, the name of the data characteristic table (DCT) that contains information about the data and that is required for compression and decompression.
Variable	Indicates whether the record occurrences are of variable length. Valid values are YES and NO.
Page Group	The page group number.
Radix	The radix number. A radix is the database key format that describes the structure of the database keys that identify records in the database.
Min Root Length	The length of the minimum root portion of a variable-length record on the home page.
Data Length	The length of the Data component of a record occurrence. The Data component is a set of bytes that represents the record value.

Property	Description
Prefix Length	The length of the Prefix component of a record occurrence. The prefix points to the related NEXT, PRIOR and OWNER records.
Ctrl Portion Len	The length of the control portion of a record up to and including the symbolic key.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for IMS synchronous data sources:

Property	Description
Database Name	Read-only. The database name.
Primary Dataset Name	Read-only. The data set file name.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for Microsoft SQL Server data sources:

Property	Description
Server	The server where the source is located.
Database Name	The database name.
Dist DBName	The SQL Server distribution database name.
Article Id	The SQL Server article ID.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for Oracle data sources:

Property	Description
Supplemental Log	Read-only. The name of the supplemental log group.

The following table describes the properties that the **Capture Registration** tab in the **Resource Inspector** displays for VSAM data sources:

Property	Description
Data Set Name	A VSAM ESDS, KSDS, or RRDS data set name.

- Click the property in the **Resource Inspector** that you want to edit and then change the property value. After you edit a property, a red square appears next to it.
Tip: To undo a change, click **Reset**.
- To apply the change, click **Apply**.

RELATED TOPICS:

- [“Adding a Capture Registration” on page 163](#)
- [“Changing the DB2 Columns in a Capture Registration” on page 178](#)
- [“Registration Tag Names” on page 154](#)
- [“Status Options” on page 168](#)
- [“Condense Options” on page 169](#)

Viewing Elements for a Datacom Capture Registration

You can view the elements for the Datacom table registered for change capture.

1. Open the registration group, and open the capture registration.
2. On the **Registration Group** tab in the **Resource Explorer**, right-click the capture registration and click **View Elements**.

The **Capture Registration - Elements Details** window displays the following information for the elements in the registered table:

- **Name.** The name of the element.
- **Position.** The position offset of the element from the start of the record.
- **Length.** The length of the element.

Viewing Paths for an IDMS Capture Registration

You can view paths, or the parent hierarchy for an IDMS base record, for an IDMS table registered for change capture.

To view paths for an IDMS capture registration:

1. Open the registration group, and open the capture registration.
2. On the **Registration Group** tab in the **Resource Explorer**, right-click the capture registration and click **View Paths**.

The **View Paths** window displays the parent hierarchy for the base record.

Changing the DB2 Columns in a Capture Registration

For a DB2 data source, you can add columns to and remove columns from a capture registration.

Note: Changing an active capture registration has implications to the capture process. For more information, contact Informatica Global Customer Support.

1. Open the registration group.

2. On the **Registration Group** tab in the **Resource Explorer**, right-click the capture registration and click **Amend Columns**.
3. The **Capture Registration - Columns** dialog box appears.
4. Set the **Select all and notify changes** option to indicate whether you want to select all columns and be notified of any column changes in the schema:
 - Clear the option to not be notified of column changes.
 - Select the option to be notified of column changes.
5. To add a column, select a column in the **Available Columns** list and click **Add**.
6. To remove a column, select a column in the **Current Columns** list and click **Remove**.
7. Click **OK**.

Deleting a Capture Registration

You can delete capture registrations that you no longer need to use. You must delete one registration at a time.

When you delete a registration, PowerExchange deletes the registration from the CCT file, which is on the system where the data source resides. PowerExchange does not delete the extraction map associated with a deleted registration. However, you can manually delete the extraction map.

After a registration is deleted, if you want to capture changes for the same source object again, you must re-create the registration. When you re-create the registration, the extraction map is regenerated.

1. If the capture registration is open, close it.
2. On the **Registration Group** tab in the **Resource Explorer**, select the capture registration and click the **Delete** button on the toolbar.

A message box prompts you to confirm the deletion.
3. To confirm the deletion, click **Yes**.
4. If the registration is active, a second message box prompts you to confirm that you want to delete the registration rather set its status to **History**. Click **Yes** to confirm the deletion, or click **No** to cancel the deletion.

Note: If you click **Yes** to delete a Microsoft SQL Server registration, PowerExchange attempts to delete the associated SQL Server article from the publication database. If PowerExchange cannot delete the SQL Server article for any reason, an error is displayed and an additional message box prompts you to confirm the registration deletion again. If you click **Yes** to confirm the deletion, you must manually delete the associated SQL Server article.

After a registration is deleted, PowerExchange no longer captures change data for the source object that was defined in a deleted registration.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Capture Registration” on page 173](#)

Deleting a Registration Group

When you delete a registration group, PowerExchange deletes local group information on the PowerExchange Navigator system.

PowerExchange does not delete the capture registrations in the group. If you create another registration group with the same properties as a group that you deleted, the new group contains any existing capture registrations for the deleted group.

Also, PowerExchange does not delete the extraction group and application group that PowerExchange generated when the registration group was created.

To delete a registration group:

1. If the registration group is open, close it.
2. In the **Resources** window, delete one or more registration groups.
3. In the message box that prompts you to confirm the deletion of the group or groups, click **Yes**.
PowerExchange deletes the registration group or groups.

RELATED TOPICS:

- [“Deleting All Capture Registrations in a Registration Group” on page 180](#)

Deleting All Capture Registrations in a Registration Group

To completely delete a registration group and its capture registrations, delete the capture registrations, the associated extraction maps and extraction group, and the registration group.

You cannot open or use an extraction map that is associated with a deleted capture registration. Alternatively, you can edit the extraction maps to change the capture registrations associated with them.

1. Delete each capture registration in the registration group.
2. Delete the extraction maps associated with the deleted registrations, and then delete the extraction group.
3. Delete the registration group.

RELATED TOPICS:

- [“Deleting a Registration Group” on page 180](#)
- [“Deleting a Capture Registration” on page 179](#)
- [“Changing the Capture Registration Associated with an Extraction Map” on page 191](#)
- [“Deleting an Extraction Map” on page 196](#)
- [“Deleting an Extraction Group” on page 197](#)

CHAPTER 6

Extraction Groups and Extraction Maps

This chapter includes the following topics:

- [Extraction Groups and Extraction Maps Overview, 181](#)
- [Extraction Map Names, 182](#)
- [Adding an Extraction Group, 184](#)
- [Adding an Extraction Map, 186](#)
- [Viewing an Extraction Map, 188](#)
- [Adding or Removing Change Indicator and Before Image Columns for a Data Column, 189](#)
- [Adding Columns to or Removing Columns from Extraction Processing, 190](#)
- [Changing the Capture Registration Associated with an Extraction Map, 191](#)
- [Merging a Data Map with an Extraction Map, 192](#)
- [Viewing or Editing Properties for an Extraction Group, 192](#)
- [Viewing Properties for an Extraction Map, 195](#)
- [Creating a Data Map from an Extraction Map, 195](#)
- [Deleting an Extraction Map, 196](#)
- [Deleting an Extraction Group, 197](#)
- [Deleting All Extraction Maps in an Extraction Group, 197](#)
- [Extract Definition Window - Map Tab, 198](#)
- [Extract Definition Window - Extensions Tab, 198](#)
- [PowerExchange-Generated Columns in Extraction Maps, 199](#)

Extraction Groups and Extraction Maps Overview

Use extraction maps to extract change data captured by PowerExchange.

Extraction groups and extraction maps define the extraction information for registered data sources and are required for CDC processing.

An extraction group defines the database instance, node location, and data source type. When you create a registration group, PowerExchange creates an extraction group with the same name. You can also manually add an extraction group.

An extraction group contains one or more extraction maps. An extraction map maps a data source. When you create a capture registration, PowerExchange creates an extraction map that includes the same columns and registration tag name as in the capture registration. PowerExchange assigns a unique name to the extraction map.

You can also manually add additional extraction maps for a capture registration. A user-defined extraction map includes the same columns and registration tag name as in its associated capture registration. PowerExchange assigns a unique name to a user-defined extraction map that includes the map name that you specified when you added the extraction map.

The DTLCAMAP file, on a z/OS system, or the CAMAPS file, on a non-z/OS system, contain extraction map information defined in the PowerExchange Navigator. The extraction group filters the data in the DTLCAMAP or CAMAPS file for display in the PowerExchange Navigator.

You can edit extraction maps to add change indicator (CI) and before image (BI) columns for data columns, and to add columns to or remove columns from extraction processing.

For a DB2 source, you can merge a data map with an extraction map. The extraction map uses the associated data map to map the change data.

To test an extraction map, run a database row test.

RELATED TOPICS:

- [“Testing an Extraction Map” on page 215](#)
- [“Registration Tag Names” on page 154](#)
- [“Registration Groups” on page 157](#)

Extraction Map Names

An extraction map name has a specific format that varies by source type.

The following general format is initially used:

xnninstance.mapname

After you extract changes using an extraction map, the table name is appended to the extraction map name:

xnninstance.mapname_tablename

The variables in the extraction map name are:

x

A character that indicates whether the extraction map was generated or user-defined. Valid values are:

- **d.** Default extraction map that PowerExchange generated.
- **u.** User-defined extraction map.

nn

A one- or two-character value that defines the source type. Valid values are:

- **1** for DB2 for z/OS.
- **2** for IMS.
- **3** for DB2 for i5/OS.
- **4** for Adabas.

- **6** for VSAM.
- **7** for CA Datacom/DB.
- **8** for Oracle.
- **10** for Microsoft SQL Server.
- **B** for DB2 for Linux, UNIX, and Windows.
- **D** for IDMS log-based.
- **e** for MySQL.
- **f** for PostgreSQL.

instance

The instance name that was specified or generated when the registration group was created. The instance value varies by source type.

mapname

One of the following values:

- For a default extraction map, the user-defined name that was specified for the capture registration.
- For a user-defined extraction map, the **Map Name** value that was specified when the extraction map was manually created.

tablename

The table name of the relational source table, or the source data map name.

In extraction maps, the *xnninstance* variable string is called the *schema*. When you open an extraction group, PowerExchange Navigator displays the extraction maps in the group as *schema.mapname*. When you extract changes using an extraction map, the full extraction map name of *schema.mapname.tablename* is used.

For example, when you run a database row test, PowerExchange Navigator generates the following SQL statement to extract the changes:

```
SELECT * FROM schema.mapname_tablename
```

In the database row test, all columns are selected by default. However, you can edit the SQL statement to select specific columns in either of the following ways:

- In the **Database Row Test** dialog box, manually edit the SQL statement in the **SQL Statement** box to select specific columns.
- In the **Extract Definition** window, select specific columns and run a database row test.

PowerExchange generates an SQL statement in the **Database Row Test** dialog box that selects the specified columns.

Note: You can override the schema name and map name by using the **Schema Name Override** and **Map Name Override** fields in the PowerExchange Client for PowerCenter (PWXPC).

RELATED TOPICS:

- [“Adding a Registration Group” on page 158](#)
- [“Adding a Capture Registration” on page 163](#)
- [“Adding an Extraction Map” on page 186](#)

Adding an Extraction Group

An extraction group is a named group of extraction maps.

An extraction group defines the data source instance, node location, and data source type. When you add a registration group, PowerExchange adds an extraction group with the same name. You can also manually add an extraction group as described in this procedure. Add extraction maps to the extraction group when you define the group or later.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Extraction Group**.
2. In the **Add Extraction Group** dialog box, enter the following information:

Name

A user-defined name for the extraction group. The maximum length is 16 characters.

Location

The location of the source for which you want to extract change data. Select **local** if the source is on the PowerExchange Navigator system.

This field lists the locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.

Default is **local**.

Type

Data source type.

Select a data source type and enter the associated information, which must match the information in the corresponding registration group:

- **ADABAS**. Adabas files.
In the **Collection Identifier** box, enter the instance name.
- **AS4**. DB2 for i5/OS tables.
In the **Collection Identifier** box, enter the instance name specified in the INST parameter of the AS4J CAPI_CONNECTION statement in the DBMOVER member of the *dtllib*/CFG file on the source system.
- **DATACOM**. CA Datacom/DB tables registered for CA Datacom/DB table-based CDC.
In the **MUF Name** box, enter the name of the source Multi-User Facility (MUF).
- **DB2**. DB2 for z/OS tables.
In the **Database Instance** box, enter the DB2 subsystem ID or the DB2 data-sharing group name.
- **DB2UDB**. DB2 for Linux, UNIX, and Windows tables.
In the **Database** box, enter the name of the database.
- **IDMS_L**. IDMS tables registered for IDMS log-based CDC.
In the **Registration Location** list, select the location where the capture registrations are to be stored.
In the **Source Map Location** list, select the location where the data maps for the IDMS tables are located.
In the **Logsid** box, enter the value specified in the LOGSID parameter in the DBMOVER configuration file on the source system.
- **IMS**. IMS databases.

For IMS synchronous CDC, in the **Recon Identifier** box, enter the IMS subsystem ID.

For IMS log-based CDC, in the **Recon Identifier** box, enter the value specified in the IMSID parameter in the DBMOVER configuration member on the z/OS system where the RECON data sets are located.

- **MSSql.** Microsoft SQL Server tables.

In the **Database Server** list, select the server where the source is located.

In the **Database Name** list, select or enter the name of the database.

- **ORACLE.** Oracle tables.

In the **Collection Identifier** box, enter the Oracle instance identifier specified in the ORACLEID statement in the DBMOVER configuration file on the source system.

- **VSAM.** VSAM ESDS, KSDS, RRDS, and VRRDS data sets.

In the **Collection Identifier** box, enter the instance name.

UserID

Enter a user ID that allows access to the source. The type of user ID depends on the source type and PowerExchange security settings:

- If the source is on an i5/OS or z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x).
- For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the *PowerExchange Reference Manual*.
- For a source on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the *PowerExchange Reference Manual*.

Password

A password for the specified user ID.

If the source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

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

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

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

If you specify a user ID but omit the password, when you open the extraction group, the PowerExchange Navigator prompts you for the password in the **Missing Password** dialog box.

Add Extraction Definition

Adds extraction maps to the group.

Clear this option to add only the extraction group.

Default is selected.

3. Perform one of the following actions:
 - If you selected the **Add Extract Definition** option, click **Next** and proceed to ["Adding an Extraction Map" on page 186](#) in the add an extraction map procedure.
 - If you cleared the **Add Extract Definition** option, click **Finish**. The **Extraction Group** tab in the **Resource Inspector** displays the properties for the extraction group. Additionally, if the PowerExchange Listener on the source system is running, the **Extraction Group** tab in the **Resource Explorer** displays any existing extraction maps for the source instance associated with the extraction group.

RELATED TOPICS:

- ["Registration Groups" on page 157](#)
- ["Adding an Extraction Map" on page 186](#)

Adding an Extraction Map

You can manually add an extraction map to an extraction group, if necessary.

PowerExchange generates an extraction map when you define a capture registration. However, you might want to add an additional extraction map for a registration.

1. Open an extraction group.
2. On the **Extraction Group** tab in the **Resource Explorer**, right-click the extraction group and click **Add Extract Definition**.
3. In the **Add Extract Definition - Name** dialog box, enter the following information:
 - a. In the **Map Name** field, enter a unique user-defined name for the extraction map. Maximum length is eight characters.
 - b. In the **Table Name** field, enter a user-defined name for the table portion of the extraction map name. Maximum length is 20 characters.

The **Schema Name** field displays the generated name for the schema portion of the extraction map name in read only format. The name has the following format:

unninstance

Where:

- u indicates that the extraction map is user-defined.

- *nn* is a one- to two-character value that indicates the source type. For example, 10 represents Microsoft SQL Server.
- *instance* is the instance value specified or generated when the registration group was created. It varies by source type.

4. Click **Next**.

The **Add Extract Definition - Capture Registrations** dialog box appears. The **Available Capture Registrations** list displays the capture registrations that are available to associate with the extraction map.

The following table describes the columns in this list:

Column	Description
Name	The user-defined name of the capture registration.
Version	The registration version number.
Condense	Whether PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows is used and the type of PowerExchange Condense processing on i5/OS or z/OS. Options are: <ul style="list-style-type: none"> - None. PowerExchange does not use PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows to process changes for the registered source. - Part. PowerExchange uses PowerExchange Condense or the PowerExchange Logger to process changes for the registered source. PowerExchange Condense uses partial condense processing. This option maintains transactional consistency. - Full. PowerExchange uses PowerExchange Condense full condense processing for an i5/OS or z/OS source but some limitations apply. This option does not maintain transactional consistency. For more information, see "Condense Options" on page 169 .
Status	The status of the capture registration with regard to availability for change capture use. Options are: <ul style="list-style-type: none"> - I for Inactive. PowerExchange cannot yet use the registration to capture changes for the source. This option is the initial status setting for a new registration. - A for Active. PowerExchange can use the registration to capture changes for the source. - H for History. PowerExchange has permanently stopped using the registration for change capture. The registration is obsolete but retained for audit purposes. - S for Suspended. PowerExchange change capture processing is temporarily suspended for the registered source. This status is set by the PWXUCREG utility SUSPEND_REGISTRATION command and is available only for an Adabas, Datacom table-based, IDMS log-based, or IMS log-based CDC source. For more information, see "Status Options" on page 168 .

5. In the **Available Capture Registrations** list, select a capture registration and click **Add**.

The selected capture registration moves to the **Current Capture Registration** list.

Tips:

- To change your selection, double-click a registration in the **Available Capture Registrations** list. The registration moves to the **Current Capture Registration** list, and the previous registration moves back to the **Available Capture Registrations** list.
- To remove a capture registration from the **Current Capture Registration** list, select the registration and click **Remove** or double-click the registration. The registration moves back to the **Available Capture Registrations** list.

6. Click **Finish**.

PowerExchange creates the extraction map in the DTLCAMAP file on a z/OS source system or in the CAMAPS file on a non-z/OS source system. The extraction map name has the following format:

unnschema.mapname

Where:

- "u" indicates that the extraction map is user-defined.
- *nn* is a one- to two-character value that indicates the source type.
- *schema* is the PowerExchange-generated schema name.
- *mapname* is the map name that you specified in Step 3.

For more information, see ["Extraction Map Names" on page 182](#).

The new extraction map appears in the list of extraction maps on the **Extraction Group** tab in the **Resource Explorer**. Also, the extraction map appears in the following locations in the interface:

- The **Extract Definition** window displays the columns in the extraction map.
- The **Extract Definition** tab in the **Resource Inspector** displays the properties for the extraction map. Use the arrows to toggle between the **Extract Definition** and **Extraction Group** tabs in the **Resource Inspector**.
- The status bar displays information about the open extraction group and extraction map.

RELATED TOPICS:

- ["Extraction Map Names" on page 182](#)
- ["Adding a Registration Group" on page 158](#)
- ["Status Options" on page 168](#)
- ["Condense Options" on page 169](#)

Viewing an Extraction Map

You can view the columns in an extraction map, including the columns that are generated by PowerExchange.

1. Open the extraction group.
2. Open the extraction map.

The **Extract Definition** window appears and displays column information on the following tabs:

- The **Map** tab displays metadata for the columns for which PowerExchange captures changes and for the PowerExchange-generated DTL__ columns if you choose to show the generated columns.
- The **Extensions** tab displays the PowerExchange-generated DTL__CI and DTL__BI columns if you added them to one or more extraction map columns.

To show the PowerExchange-generated columns, right-click anywhere on the **Map** tab and click **Show Auto Generated Columns**.

To hide the PowerExchange-generated columns, right-click anywhere on the **Map** tab and click **Hide Auto Generated Columns**.

To add DTL__CI and DTL__BI columns for one or more data columns, right-click anywhere on the **Map** tab and click **Amend Change Indicator/Before Image Extensions**. These columns will appear on both tabs.

To sort the list of columns on either tab by column name or metadata field, click the metadata column heading.

RELATED TOPICS:

- [“Extract Definition Window - Map Tab” on page 198](#)
- [“Extract Definition Window - Extensions Tab” on page 198](#)

Adding or Removing Change Indicator and Before Image Columns for a Data Column

You can edit an extraction map for a source table to add or remove change-indicator (CI) and before-image (BI) columns for one or more data columns.

In an extraction map, the CI column name is `DTL__CI_data_column_name` and the BI column name is `DTL__BI_data_column_name`.

A `DTL__CI` column indicates whether an UPDATE operation changed data in the associated column. You can use a CI column in an SQL WHERE clause filter for a CDC session to filter the change records that PWXPC passes to PowerCenter for processing. The `DTL__CI` column contains one of the following values:

- **Y**. An UPDATE changed the column data.
- **N**. An UPDATE did not change the column data.
- **Null**. An INSERT or DELETE operation on the column occurred.

A `DTL__BI` column contains the BI for an UPDATE operation on the associated column. If the source database allows changes to primary keys, you can use the BI fields with a PowerCenter Flexible Target Key Custom transformation to trigger updates to primary keys on the target when source keys change.

Tip: If you use the PowerExchange Logger for Linux, UNIX, and Windows and specify AI for the `CAPT_IMAGE` parameter in the `pwxcl.cfg` file, do not add change-indicator or before-image columns. The AI setting causes the PowerExchange Logger to store only after images. Consequently, PowerExchange cannot provide BIs to PWXPC for downstream processing in PowerCenter. Also, if you enter BA for the `CAPT_IMAGE` parameter, the PowerExchange Logger stores both before images and after images for UPDATES. Any INSERT or DELETE operations on the source result in null values in the change-indicator columns.

1. Open the extraction group.
2. Right-click the extraction map and click **Amend Change Indicator/Before Image Extensions**.
The **Extract Definition - Amend Columns** dialog box appears.
3. On the **Change Indicators** tab, select the columns for which to add `DTL__CI` columns. In the **Available Columns** list, select one or more columns and click **Add**.
The selected columns appear in the **Selected Columns** list.
4. On the **Before Images** tab, select the columns for which to add `DTL__BI` columns. In the **Available Columns** list, select one or more columns and click **Add**.
The selected columns appear in the **Selected Columns** list.

Note: If the selected columns are user-defined fields with expressions that invoke PowerExchange functions, ensure that the functions support BI buffering. For more information about functions, see Appendix A, [Appendix A, “PowerExchange Functions for User-Defined Fields” on page 282](#).

5. To remove `DTL__CI` or `DTL__BI` columns, in the **Selected Columns** list, select the columns and click **Remove**.
The selected columns move back to the **Available Columns** list.

6. Click **OK**.
If you added DTL__CI or DTL__BI columns, the columns appear on both the **Map** and **Extensions** tabs in the **Extract Definition** window.
7. To verify the changes, run a database row test on the extraction map.

RELATED TOPICS:

- [“Testing an Extraction Map” on page 215](#)

Adding Columns to or Removing Columns from Extraction Processing

You can edit an extraction map to add columns to or remove columns from extraction processing.

Adding or removing columns from extraction processing does not affect the columns in the associated capture registration for which PowerExchange captures change data.

You can also associate DTL__CI and DTL__BI columns with columns in the extraction map. A change indicator (DTL__CI) column indicates whether the contents of the column changed in a change record. A before image (DTL__BI) column contains the contents of the column from before the last update operation.

1. Open the extraction group.
2. Open the extraction map.

The **Extract Definition** window displays column metadata on the following tabs:

- The **Map** tab displays metadata for the columns for which PowerExchange captures change data and any DTL__CI and DTL__BI columns that you added to those data columns. The tab also displays the metadata for the PowerExchange-generated columns if you choose to show these columns.

To show the PowerExchange-generated columns, right-click anywhere in the **Map** tab and click **Show Auto Generated Columns**.

- The **Extensions** tab displays CI and BI columns if they are added for one or more extraction map columns.

3. To remove a column from extraction processing, clear the check box next to the column name.

PowerExchange stops extracting change data for the removed columns but continues extracting data for the selected columns.

4. To add a column to extraction processing, select the check box next to the column name.
5. After adding or removing columns, verify that the extraction map is still valid by running a database row test on it.

RELATED TOPICS:

- [“Adding or Removing Change Indicator and Before Image Columns for a Data Column” on page 189](#)
- [“Testing an Extraction Map” on page 215](#)
- [“Extract Definition Window - Map Tab” on page 198](#)
- [“Extract Definition Window - Extensions Tab” on page 198](#)

Changing the Capture Registration Associated with an Extraction Map

You can edit an extraction map to change its associated capture registration.

You might change the capture registration associated with an extraction map if a new version of the capture registration has been added.

1. Open the extraction group.
2. Open the extraction map.
3. Right-click the extraction map and click **Amend Capture Registrations**.

The **Extract Definition - Amend Capture Registrations** dialog box appears.

On the **Amend Capture Registration** tab, the **Current Capture Registration** list shows the capture registration that is currently associated with the extraction map, and the **Available Capture Registration** list shows the registrations that are available for selection.

The following table describes the columns in each list:

Column	Description
Name	The user-defined name for the capture registration.
Version	The registration version number.
Condense	Whether PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows is used and the type of PowerExchange Condense processing on i5/OS or z/OS. Options are: <ul style="list-style-type: none">- None. PowerExchange does not use PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows to process changes for the registered source.- Part. PowerExchange uses PowerExchange Condense or the PowerExchange Logger to process changes for the registered source. PowerExchange Condense uses partial condense processing. This option maintains transactional consistency.- Full. PowerExchange uses PowerExchange Condense full condense processing for an i5/OS or z/OS source but some limitations apply. This option does not maintain transactional consistency. For more information, see "Condense Options" on page 169 .
Status	The status of the capture registration with regard to availability for change capture use. Options are: <ul style="list-style-type: none">- Inactive. PowerExchange cannot yet use the registration to capture changes for the source. This option is the initial status setting for a new registration.- Active. PowerExchange can use the registration to capture changes for the source.- History. PowerExchange has permanently stopped using the registration for change capture. The registration is obsolete but retained for audit purposes.- Suspended. PowerExchange change capture processing is temporarily suspended for the registered source. This status is set by the PWXUCREG utility SUSPEND_REGISTRATION command and is available only for an Adabas, Datacom table-based, IDMS log-based, or IMS log-based CDC source. For more information, see "Status Options" on page 168 .

4. In the **Current Capture Registration** list, select the current registration and click **Remove**.
5. In the **Available Capture Registrations** list, select the registration to associate with the extraction map and click **Add**.

That capture registration appears in the **Current Capture Registration** list.

6. Click **OK**.

7. To verify the change, run a database row test on an extraction map.

RELATED TOPICS:

- [“Testing an Extraction Map” on page 215](#)
- [“Status Options” on page 168](#)
- [“Condense Options” on page 169](#)

Merging a Data Map with an Extraction Map

For a DB2 source, you can merge a data map with an extraction map.

For example, if a source table contains columns in which you store data in a format that is inconsistent with the column datatype, you can create a data map to manipulate that data with PowerExchange expressions. If you store packed data in a CHAR column, you might create a data map to manipulate and prepare that data for loading to a target. You can then merge the data map with an extraction map. The extraction map uses the associated data map to map the change data.

1. Open the extraction group.
2. Open the extraction map.
3. On the **Extraction Group** tab in the **Resource Explorer**, right-click the extraction map and select **Amend Capture Registrations**.
4. In the **Extract Definition - Amend Capture Registrations** dialog box appears, select **Merge**.
5. In the **Bulk Datamap Name** box, enter a data map name or select a data map:
 - Click the Browse button to browse to the file.
 - In the **DB2 Datamaps Selection** dialog box, specify filter criteria under **DB2 Map Filter Data** to restrict the list of data maps. You can include wildcards.
 - Click **Next**.
 - In the **Available Datamaps** list, select a data map and click **Finish**.
6. Click **OK**.

The extraction map uses the associated data map to map the change data. The merged data map appears in the **Merged With** field on the **Extract Definition** tab of the **Resource Inspector**.

RELATED TOPICS:

- [“DB2 Catalog Data Maps” on page 107](#)

Viewing or Editing Properties for an Extraction Group

You can view or edit properties for an extraction group.

You might edit the properties for an extraction group to change its name, location, or the user ID and password used to access the source. You cannot change the database type for an extraction group.

1. Open the extraction group.

The **Extraction Group** tab in the **Resource Inspector** displays the following properties for the extraction group, depending the source type:

Property	Description
Name	A user-defined name for the extraction group.
Access Method	Read-only. Access method associated with the data source type.
Location	Location of the source for which you want to capture changes.
UserID	<p>A user ID that allows access to the source. The type of user ID depends on the source type and PowerExchange security settings:</p> <ul style="list-style-type: none"> - If the source is on an i5/OS or z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x). - For a DB2 LUW, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the <i>PowerExchange Reference Manual</i>. - For a source on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the <i>PowerExchange Reference Manual</i>.
Password	<p>A password for the specified user ID.</p> <p>If the data source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:</p> <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p>

Property	Description
Instance	The instance name from the registration group. For an Adabas, DB2 for i5/OS, Oracle, or VSAM source, the instance name is the collection identifier. For a DB2 for z/OS source, the instance name is the database instance name. For a Microsoft SQL Server source, the instance name is the Instance value that you entered when creating the registration group. If you did not enter an instance name, this value is a generated value that is composed of the first four characters of the database name followed by an ascending number, which starts at 000. For a MySQL or PostgreSQL source, the instance name is the Instance value that you entered when creating the registration group. If you did not enter an instance name, this value is a generated value that consists of the first four characters of the database server name followed by an ascending number, which starts at 000.
MUF Name	For a CA Datacom/DB source, the name of the Multi-User Facility (MUF).
Server	For a Microsoft SQL Server source, the name of the publication server. For a PostgreSQL source, the name of the database server.
MySQL Server	For a MySQL source, the name of the database server.
Database	For a DB2 LUW, Microsoft SQL Server, or PostgreSQL source, the name of the source database.
DB Name	For an IDMS log-based source, the name of the database.
Recon ID	For an IMS source, the RECON identifier for the database.
Oracle SID	Read-only. For an Oracle source, the system identifier (SID) for the database.

- To edit a property, click the property box in the **Resource Inspector** and enter a value.
After you edit a property, a red square appears next to the changed property.
Tip: To undo a change before you apply it, click **Reset**.
- To apply a change, click **Apply**.
If you edit the extraction group name, you are prompted to confirm the change. If the new name matches the name of an existing extraction group, you are prompted to overwrite the existing group.
If you edit the location, PowerExchange displays an error message if the location is not valid for the instance.
If you edit the instance, PowerExchange displays an error message if the specified instance does not exist.
- To confirm the changes, click **Yes**.
The **Resource Inspector** updates the fields that you edited.

RELATED TOPICS:

- [“Adding an Extraction Group” on page 184](#)

Viewing Properties for an Extraction Map

You can view but not change extraction map properties.

1. Open the extraction group.
2. Open the extraction map.

In the **Resource Inspector**, the **Extract Definition** tab displays extraction map properties. The following table describes these properties:

Property	Description
Schema	The PowerExchange-generated schema name.
Map	The PowerExchange-generated map name.
Registration	The name of the associated capture registration.
Version	The version number.
Table	The table to which the extraction map refers.
Merged With	For a DB2 source, if the extraction map was merged with a data map, displays the data map name.
Tag	The unique identifier that PowerExchange generates and assigns to a data source registered for capture.

RELATED TOPICS:

- [“Adding an Extraction Group” on page 184](#)
- [“Registration Tag Names” on page 154](#)

Creating a Data Map from an Extraction Map

You can create a PowerExchange data map from an extraction map.

For example, you might want to create a sequential data map with the same layout as a DB2 table or an Adabas table.

PowerExchange stores the data map in the current resource configuration directory on the local system.

To create a data map from an extraction map:

1. Open the extraction group, and open the extraction map.
2. Right-click the extraction map and click **Create Data Map**.

3. In the **Create Map** dialog box, enter the following information:

Schema Name

Enter the schema name, which PowerExchange uses as the first part of the data map name, as follows:

schema_name.map_name

Map Name

Enter the map name, which PowerExchange uses as the second part of the data map name, as follows:

schema_name.map_name

Include Capture Control Fields

Includes the following PowerExchange-generated columns in the data map:

- DTL__CAPXtype
- DTL__columnname_BI
- DTL__columnname_CI

Clear this option to exclude these columns from the data map.

Default is selected.

Include Binary Count and Null Indicator Fields

Includes the PowerExchange-generated columns, DTL__columnname_CNT and DTL__columnname_IND, in the data map.

Clear this option to exclude these columns from the data map.

Default is cleared.

4. Click **OK**.
PowerExchange saves the data map in the current resource configuration directory on the local system.

Deleting an Extraction Map

You can delete extraction maps that you no longer intend to use.

After you delete an extraction map, you can no longer use it to extract change data. You must update any PowerCenter mappings and sessions that reference the deleted extraction map.

When you delete an extraction map, PowerExchange does not delete the capture registration associated with the extraction map.

To delete an extraction map:

1. If the extraction map is open, close it.
2. On the **Extraction Group** tab in the **Resource Explorer**, select the extraction map and click the **Delete** button on the toolbar.
3. In the message box that prompts you to confirm the deletion of the map, click **Yes**.
PowerExchange deletes the extraction map but does not delete the capture registration associated with the extraction map.
4. In PowerCenter, update any PowerCenter mappings and sessions that reference the deleted extraction map.

RELATED TOPICS:

- [“Deleting a Capture Registration” on page 179](#)

Deleting an Extraction Group

You can delete extraction groups that you no longer intend to use.

When you delete an extraction group, PowerExchange deletes local group information on the PowerExchange Navigator system but does not delete the extraction maps in the group. If you create another extraction group with the same properties as a group that you deleted, the new group contains any existing extraction maps for the deleted group.

PowerExchange does not delete the registration group for which this extraction group was generated.

To delete an extraction group:

1. If the extraction group is open, close it.
2. In the **Resources** window, delete one or more extraction groups.
3. In the message box that prompts you to confirm the deletion of the group or groups, click **Yes**. PowerExchange deletes the extraction group or groups.

Deleting All Extraction Maps in an Extraction Group

To completely delete an extraction group and its extraction maps, you must delete the extraction maps, the associated capture registrations and their registration group, and the extraction group.

To delete all extraction maps in an extraction group:

1. Delete each extraction map in the extraction group.
2. Delete the associated capture registrations and their registration group.
3. Delete the extraction group.

RELATED TOPICS:

- [“Deleting an Extraction Map” on page 196](#)
- [“Deleting an Extraction Group” on page 197](#)
- [“Deleting a Capture Registration” on page 179](#)
- [“Deleting a Registration Group” on page 180](#)

Extract Definition Window - Map Tab

The **Map** tab of an **Extract Definition** window displays metadata for each column in an extraction map.

The following table describes the metadata columns:

Column	Description
Name	The name of the column.
ColType	The type of the column, which is one of the following values: <ul style="list-style-type: none">- Real. An original column of the extraction map.- BI. A before image column that was added through the Amend Change Indicator/ Before Image Extensions command.- CI. A change indicator image column that was added through the Amend Change Indicator/ Before Image Extensions command.- Auto. A PowerExchange-generated column.
DataType	The datatype.
Precision	The precision of the field.
Scale	The data scale.
Length	The column length. For columns of variable-length datatypes, the data might not be as long as the column allows.

Extract Definition Window - Extensions Tab

The **Extensions** tab in the **Extract Definition** window displays the before-image columns (DTL__BI) and change indicator columns (DTL__CI) columns that you added for one or more data columns in the extraction map.

The following table describes the columns of information on the **Extensions** tab:

Column	Description
Name	A data column name, or the name of a change-indicator or before-image column that was added for a data column.
CI Column	A change-indicator column name in the format <code>DTL__CI_data_column_name</code> .
BI Column	A before-image column name in the format <code>DTL__BI_data_column_name</code> .
Real Column	A data column for which a change-indicator or before-image column was added.

Note: The check boxes for the CI and BI columns are disabled and appear in grey.

RELATED TOPICS:

- [“Adding or Removing Change Indicator and Before Image Columns for a Data Column” on page 189](#)

PowerExchange-Generated Columns in Extraction Maps

Besides the table columns that are defined in capture registrations, extraction maps include columns that PowerExchange generates.

These PowerExchange-generated columns contain CDC-related information, such as the type of SQL change and time stamp.

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

When you run a database row test on an extraction map, the PowerExchange Navigator displays the PowerExchange-generated columns in the results. By default, the PowerExchange Navigator hides these columns from view when you open the extraction map. To display these columns, open the extraction map, right-click anywhere within the **Extract Definition** window, and select **Show Auto Generated Columns**.

Note: By default, all columns are selected in extraction maps except the `DTL__columnname_CNT`, `DTL__columnname_IND`, and `DTL__CI_columnname` columns. To add these columns, you must edit the extraction map.

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

Column	Description	Datatype	Length
DTL__CAPXRESTART1	<p>Provides a binary value that represents the position of the end of the UOW for that change record followed by the position of the change record itself.</p> <p>The length of a sequence token varies by data source type, except on z/OS where sequence tokens for all data source types have the same length.</p> <p>The value of DTL__CAPXRESTART1 is also known as the <i>sequence token</i>, which when combined with the <i>restart token</i> comprises the restart token pair.</p> <p>A sequence token for a change record is a strictly ascending and repeatable value.</p>	VARBIN	255
DTL__CAPXRESTART2	<p>Provides a binary value that represents a position in the change stream that can be used to reconstruct the UOW state for the change record, with the following exceptions:</p> <ul style="list-style-type: none"> - Microsoft SQL Server CDC. A binary value that contains the DBID of the distribution database and the name of the distribution server. - Change data extracted from full condense files on z/OS or i5/OS. A binary value that contains the instance name from the registration group of the capture registration. <p>The length of a restart token varies by data source type. On z/OS, restart tokens for all data source types have the same length, except for change data extracted from full condense files.</p> <p>The value of DTL__CAPXRESTART2 is also known as the <i>restart token</i>, which when combined with the <i>sequence token</i> comprises the restart token pair.</p>	VARBIN	255
DTL__CAPXROWID	<p>For PowerExchange Express CDC for Oracle, provides the physical rowid value. PowerExchange can include rowid values in change records for Oracle tables only if the tables do not have row movement enabled.</p> <p>To enable the capture of rowid values, you must configure the OPTIONS ROWID=Y statement in the Express CDC configuration file.</p> <p>The rowid is useful for processing rows in unkeyed tables during CDC extraction sessions.</p>	CHAR	18
DTL__CAPXRRN	For Db2 for i sources only, provides the relative record number.	DECIMAL	10
DTL__CAPXUOW	Provides a binary value that represents the position in the change stream of the start of the UOW for the change record.	VARBIN	255

Column	Description	Datatype	Length
DTL__CAPXUSER	<p>Provides the user ID of the user who made the change to the data source, with the following exceptions:</p> <ul style="list-style-type: none"> - For Adabas 8.3 CDC sources, this value is the Security User-id (SECUID) of the user if the Adabas File Definition includes the system field SY=SECUID. - For Datacom table-based CDC sources, this value is the MUF name. - For Db2 for i (i5/OS) CDC sources, this value depends on the LIBASUSER parameter in the AS4J CAPI_CONNECTION statement. If LIBASUSER=Y, this value is the library name and file name of the file where the change was made. If LIBASUSER=M, this value is the library name, file name, and data member name of the file where the change was made. If LIBASUSER=N, this value is the user ID of the user who made the change. If LIBASUSER=P, this value is the name of the program that made the change. - For Db2 for z/OS CDC sources, this value depends on the UIDFMT parameter in the LRAP CAPI_CONNECTION. Depending on the parameter setting, this value can be a Db2 connection identifier, correlation identifier, connection type, plan name, user ID, or all of these values in the format <code>UID:PLAN:CORR:CONN:CTYPE</code>. If you do not specify the UIDFMT parameter, this value is the user ID of the user who made the change. - For IDMS CDC sources, this value is the value that the user program puts in the program name field of the application subschema control block. Usually, this value is the user program name. - For IMS synchronous CDC sources, this value depends on the UIDFMTIMS parameter in the LRAP CAPI_CONNECTION statement. Depending on the parameter setting, this value can be a user ID, a PSB name, or both values in the format <code>userid:psbname</code>. If you do not specify the UIDFMTIMS parameter, the user ID is used by default. - For Microsoft SQL Server CDC sources, this value depends on the UIDFMT parameter in the MSQL CAPI_CONNECTION statement. If UIDFMT=DBNAME, this value is the SQL Server publication database name. If UIDFMT=NONE, this value is a null. - For Oracle CDC sources, this value is a user ID that PowerExchange gets from Oracle, if available. Otherwise, this value is null. 	VARCHAR	255

Column	Description	Datatype	Length
DTL__CAPXTIMESTAMP	<p>Provides the time stamp that the source DBMS records for the database change record.</p> <p>This value can be either the time stamp that the source DBMS writes to the change record in the database logs or the time stamp of the transaction commit on the source database.</p> <p>The type of time stamp depends on the source type and certain parameters:</p> <ul style="list-style-type: none"> - For Db2 sources on Linux, UNIX, or Windows, the transaction commit time stamp. - For Microsoft SQL Server sources, the time at which the change was written to the distribution database. - For MySQL sources, the time at which MySQL recorded the change event in the binary log. - For Oracle sources, the timestamp type is controlled by the TIME_STAMP_MODE parameter in the OPTIONS statement of the Express CDC configuration file. - For all sources that require a UOWC CAPI_CONNECTION statement, the time stamp type is controlled by the TIMESTAMP parameter in the UOWC CAPI_CONNECTION statement in the DBMOVE file. <p>For more detailed information about time stamps for each source type, see Appendix F, "DTL__CAPXTIMESTAMP Time Stamps" on page 370.</p> <p>The time stamp format is:</p> <pre>YYYYMMDDhhmssnnnnnn</pre> <p>Where:</p> <ul style="list-style-type: none"> - YYYY is the four-digit year. - MM is the month. - DD is the day. - hhmssnnnnnn is hours, minutes, seconds, and microseconds. <p>Note: Db2 on Linux, UNIX, or Windows and Oracle do not support microseconds in the time stamp.</p>	CHAR	20

Column	Description	Datatype	Length
DTL__CAPXACTION	<p>Indicates the type of change record that PowerExchange passed to the target during extraction processing. This indicator corresponds to the type of SQL change operation on the source database.</p> <p>Valid values:</p> <ul style="list-style-type: none"> - I. Insert. - D. Delete. - U. After image of an UPDATE. - T. Before image of an UPDATE. (ODBC connections only) <p>If you specify an Image Type of BA on the connection for a CDC session, PowerExchange generates a delete record followed by an insert record for a source update. In the delete record, the DTL__CAPXACTION column contains the value D. In the insert record, the DTL__CAPXACTION column contains the value I.</p> <p>If you specify an Image Type of AI on the connection for a CDC session, PowerExchange generates one record for an update. In this record, the DTL__CAPXACTION column contains the U value.</p> <p>If you use an ODBC connection to write change data to a staging table and either set the ODBC driver CAPXIMAGETYPE parameter to TU or enter the SQL escape sequence DTLIMTYPE=TU in PowerCenter, this column can contain a value of T or U. For each source update, PowerExchange delivers two records to the staging table: one for the before image and another for the after image. In the before image record, the DTL__CAPXACTION column contains the T value. In the after image record, The DTL__CAPXACTION column contains the U value.</p>	CHAR	1
DTL__CAPXCASDELIND	<p>For Db2 for z/OS sources only, indicates whether or not Db2 deleted the row because the table specifies the ON DELETE CASCADE clause. Valid values:</p> <ul style="list-style-type: none"> - Y. Indicates that Db2 deleted the row because of a cascade delete rule. - N. Indicates that Db2 did not delete the row because of a cascade delete rule. 	CHAR	1
DTL__BI_< i>columnname	Provides the before image of a column that an UPDATE operation changed.	Datatype of the source column	Length of the source column
DTL__CI_< i>columnname	<p>Indicates whether or not an UPDATE operation changed the column value. Valid values:</p> <ul style="list-style-type: none"> - Y. The column value was changed by an UPDATE operation. - N. The column was changed by an UPDATE operation. - <i>null</i> . The column was changed by an INSERT or DELETE operation. It was not changed by an UPDATE. <p>Note: By default, the change indicator column is not included in extraction maps. To add it, you must edit an extraction map and select this auto-generated column.</p>	CHAR	1

Column	Description	Datatype	Length
DTL__ST_lob_columnname	<p>For a Db2 for z/OS LOB column, indicates whether or not the column contains all of the LOB data. The ECCR provides incomplete LOB data if the data is not stored fully inline in the base table space or exceeds 32 KB in size. Valid values:</p> <ul style="list-style-type: none"> - C. The column contains all of the LOB data. The ECCR was able to capture all of the LOB data because the data is stored fully inline in the base table space and does not exceed 32 KB. - I. The column contains incomplete LOB data. The ECCR was unable to capture all of the LOB data because the data is stored in an auxiliary table space, or the data is stored fully inline but exceeds 32 KB in size. - <i>null.</i> The column contains null data only. <p>If you have Db2 for z/OS source tables that include LOB data that is not stored fully inline in the base table, include this column. You can then use this column with PowerCenter transformations to retrieve all of the current LOB data for columns with incomplete data (DTL__ST_columnname=I) and write the data to the target.</p> <p>Note: This field is included in extraction maps by default. To remove it, open the extraction map in the PowerExchange Navigator and deselect this auto-generated column.</p>	CHAR	1
DTL__columnname_CNT	<p>A binary count that PowerExchange generates for a variable length column of the type VARCHAR and VARBIN. The count is used to determine the length of the column during change data extraction processing.</p> <p>Note: By default, the binary count column is not included in extraction maps. To add it, you must edit an extraction map and select this auto-generated column.</p>	NUM32U	0
DTL__columnname_IND	<p>Indicates whether or not a nullable column contains a null. PowerExchange generates this column only for nullable columns.</p> <p>Note: By default, the null indicator column is not included in extraction maps. To add it, you must edit an extraction map and select this auto-generated column.</p>	BIN	1

CHAPTER 7

Personal Metadata

This chapter includes the following topics:

- [Personal Metadata Overview, 205](#)
- [Adding a Personal Metadata Profile, 206](#)
- [Viewing Metadata for Tables in a Personal Metadata Profile, 209](#)
- [Viewing Properties for a Table in a Personal Metadata Profile, 210](#)
- [Viewing or Editing Properties for a Personal Metadata Profile, 210](#)
- [Refreshing Personal Metadata, 211](#)
- [Deleting a Personal Metadata Profile, 211](#)

Personal Metadata Overview

You can define personal metadata profiles for relational and nonrelational sources and targets to test connectivity and access metadata and data.

Personal metadata profiles are not required to perform bulk data movement operations or change data capture (CDC).

To view metadata for a source or target, create a personal metadata profile. View tables in the profile to see metadata for each column in the table such as datatype, date format, and CCSID. You can view metadata for multiple tables in a profile at the same time. To preview data for the source or target, run a database row test on tables in the profile.

For information about opening and closing personal metadata profiles, see [“Opening a PowerExchange Resource” on page 30](#) and [“Closing a PowerExchange Resource” on page 32](#).

For information about searching for tables and columns in tables in personal metadata profiles, see [“Searching for a Resource” on page 30](#).

PowerExchange provides an example personal metadata profile, demo_oracle. To use this example profile, edit its properties to point to the Oracle instance.

Adding a Personal Metadata Profile

Add a personal metadata profile that you can use to connect to a source or target.

You can include filters to limit the tables that the profile contains.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Personal Metadata**.
2. In the **Personal Metadata - Name** dialog box, enter the following information:

Name

A user-defined name for the personal metadata profile.

Location

The location of the source or target for which you want to view metadata. Select **local** if the source or target is on the PowerExchange Navigator system.

This field lists locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.

Default is **local**.

Type

The data source or target type. Select one of the following types and then complete the fields for that type:

- **DB2**. DB2 for i5/OS or DB2 for z/OS sources or targets.
For DB2 for i5/OS, optionally enter the database name in the **SSID or DBName** field. Default is the default database.
For DB2 for z/OS, enter the SSID in the **SSID or DBName** field.
- **DB2400C**. DB2 Call Level Interface (CLI) for i5/OS sources.
Enter the database name in the **SSID or DBName** field.
- **DB2UDB**. DB2 for Linux, UNIX, and Windows sources or targets.
Optionally enter a database name in the **DBName** field.
- **MSSQL**. Microsoft SQL Server sources or targets.
Enter the SQL Server instance name in the **DSN** field. Optionally, enter a database name to override the database for the connection in the **Database** field.
- **MYSQL**. MySQL sources.
Enter the MySQL database server name in the **Server** field.
- **NRDB** or **NRDB2**. PowerExchange data maps for nonrelational sources such as Adabas, CA Datacom/DB, CA IDMS/DB, IMS, VSAM, and flat files.
Optionally enter an override file name in the **Override File Name** field.
The **NRDB2** data source type uses a two-tier naming convention for generated SQL in a database row test while **NRDB** uses a three-tier naming convention. You can use **NRDB** or **NRDB2** for PowerExchange data maps.
- **ORACLE**. Oracle sources or targets.
Optionally, enter a TNS name for the Oracle instance in the **SQL*Net Name** field.
- **PGSQL**. PostgreSQL sources.

Note: Do not select any of the following deprecated source or target types:

- AS4_META
- CAPX
- CAPXRT
- ODBC
- TASK_CNTL

UserID

Enter a user ID that allows access to the source. The type of user ID depends on the source type and PowerExchange security settings:

- If the source is on an i5/OS or z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x).
- For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, MySQL, Oracle, or PostgreSQL source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the *PowerExchange Reference Manual*.
- For a source on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the *PowerExchange Reference Manual*.

Password

A password for the specified user ID.

If the source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

- Uppercase and lowercase letters
- The numbers 0 to 9
- Spaces
- The following special characters:
' - ; # \ , . / ! % & * () _ + { } : @ | < > ?

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

If you specify a user ID and omit the password or passphrase, when you open the personal metadata profile, the PowerExchange Navigator prompts you for the password in the **Missing Password** dialog box.

Type-specific fields

Type	Fields
DB2, DB2400C, or DB2UDB	SSID or DBNAME. For a DB2 for z/OS, enter a DB2 subsystem ID. For DB2400C or DB2UDB, enter a DB2 database name.
MSSQL	Server. The SQL Server publication server name. Database. The SQL Server database name.
MYSQL	Server. The MySQL database server name.
ORACLE	SQL*Net Name. The TNS name for the Oracle instance.
PGSQL	Server. The PostgreSQL database server name. Database. The PostgreSQL database name.

Description

A user-defined description for the personal metadata profile.

- Click **Next**.
- In the **Personal Metadata - Filters** dialog box, enter the following information to limit the tables that the profile contains:

Filter Data

Enter optional filter criteria in the following filter fields, depending on the source or target type:

- For the DB2, DB2400C, or DB2UDB type, enter filters in the **Creator/Schema**, **DBName/Definer**, and **Table** boxes.
- For the MYSQL, ORACLE, or PGSQL type, enter filters in the **Schema** and **Table** boxes.
- For the MSSQL type, enter filters in the **Owner** and **Table** boxes.
- For the NRDB or NRDB2 type, enter filters in the **Schema**, **Mapname**, and **Table** boxes.

In any filter field, you can use the following wildcard characters:

- An asterisk (*) to represent one or more matching characters.
- A question mark (?) to represent a single matching character.

Respect Case

By default, text entered in any of the filter boxes is converted to the standard case for the data source. For example, on Oracle, uppercase is the standard and so a value of `scott` or `SCOTT` produces the same result: `SCOTT`.

Select the **Respect Case** check box for those filters for which you do not want PowerExchange to switch to the standard case for the data source.

Escape Character

Enter the escape character to use for delimiting an asterisk (*) or question mark (?) that is used as a literal value rather than as a wildcard in the filter fields.

For example, if you specify an escape character of `~`, a filter value of `tab*` returns all tables beginning with `tab`. A filter value of `tab~*` returns the table named `tab*`.

Note: Do not use an escape character with data sources that are multibyte-enabled.

5. Click **Finish**.

The **Personal Metadata** tab in the **Resource Explorer** displays the personal metadata profile and all tables that match the specified filters.

6. To verify that the personal metadata profile can preview data, run a database row test on a table in the profile.

RELATED TOPICS:

- [“Testing a Personal Metadata Profile” on page 216](#)

Viewing Metadata for Tables in a Personal Metadata Profile

You can view metadata for the columns in one or more tables in a personal metadata profile.

You can view metadata for a single table at a time or display metadata for multiple tables, each in a separate Table window.

To view metadata for tables in a personal metadata profile:

1. On the **Personal Metadata** tab in the **Resource Explorer**, complete one of the following actions:

- To view a single table, double-click the table.
- To view multiple tables, each in a separate **Table** window, right-click each table and click **Explore**.

A **Table** window displays the following metadata for each table column:

- **Name**. The name of the column.
- **Type**. The datatype.
- **Precision**. The data precision.
- **Scale**. The data scale.
- **Nullable**. The value YES or NO to indicate whether the column is nullable.
- **PowerExchange Type**. The PowerExchange datatype. PowerExchange has different designations for some datatypes.
- **Comments**. Comments for the table.

2. To sort the metadata in alphabetical order by name or datatype, click the Name or Type heading in a **Table** window. To sort the metadata in reverse alphabetical order, click the column heading again.

A small arrow appears in the **Name** or **Type** heading that indicates the sort order, as follows:

- A down arrow indicates alphabetical order.
- An up arrow indicates reverse alphabetical order.

Viewing Properties for a Table in a Personal Metadata Profile

You can view the table name, object type, and comments for a table in a personal metadata profile.

- ▶ On the **Personal Metadata** tab in the **Resource Explorer**, right-click a table in a personal metadata profile and click **Properties**.

The **Personal Metadata Table Properties** dialog box displays the following information:

Name

The name of the table in one of the following formats, depending on the personal metadata type:

- For NRDB:

schema.map.table

- For NRDB2:

schema.map_table

- For relational sources:

schema.table

Type

The object type.

Comments

Comments for the table.

Viewing or Editing Properties for a Personal Metadata Profile

You can view or edit properties for a personal metadata profile.

You might edit a personal metadata profile to change filter criteria for tables.

1. Open the personal metadata profile.
2. On the **Personal Metadata** tab in the **Resource Explorer**, right-click a personal metadata profile and click **Properties**.

The **Personal Metadata Properties** dialog box appears.

3. On the **Name** tab, **Filter** tab, or both, view or edit the profile properties.
4. Click **OK**.

RELATED TOPICS:

- [“Adding a Personal Metadata Profile” on page 206](#)

Refreshing Personal Metadata

You can refresh the objects displayed by a personal metadata profile.

You might want to refresh personal metadata if you defined new tables or views in the source. Or, if the PowerExchange Listener was not running on the source system when you opened the profile, you can start the PowerExchange Listener and refresh the personal metadata.

To refresh personal metadata:

- ▶ On the **Personal Metadata** tab in the **Resource Explorer**, right-click a personal metadata profile and click **Refresh**.

The metadata in the personal metadata profile is refreshed with current metadata from the data source.

Deleting a Personal Metadata Profile

You can delete a personal metadata profile.

To delete a personal metadata profile:

1. If the personal metadata profile is open, close it.
2. On the **Resources** tab of the **Resource Explorer**, right-click the profile and click **Delete**.
3. In the message box that prompts you to confirm the deletion of the profile, click **Yes**.

PowerExchange deletes the profile.

CHAPTER 8

Database Row Test

This chapter includes the following topics:

- [Database Row Test Overview, 212](#)
- [Testing a Data Map, 213](#)
- [Testing an Extraction Map, 215](#)
- [Testing a Personal Metadata Profile, 216](#)
- [Issuing PowerExchange Listener Commands in a Database Row Test, 217](#)
- [Generating Restart Tokens for Change Data Capture, 218](#)
- [Database Row Test Dialog Box, 219](#)
- [CAPX Advanced Parameters Dialog Box - General Tab, 228](#)
- [CAPX Advanced Parameters Dialog Box - AS400 Tab, 230](#)
- [CAPXRT Advanced Parameters Dialog Box - General Tab, 230](#)
- [CAPXRT Advanced Parameters Dialog Box - Adabas Tab, 234](#)
- [CAPXRT Advanced Parameters Dialog Box - AS400 Tab, 234](#)
- [CAPXRT Advanced Parameters Dialog Box - DB2UDB Tab, 235](#)
- [CAPXRT Advanced Parameters Dialog Box - MSSQL Tab, 235](#)
- [CAPXRT Advanced Parameters Dialog Box - Oracle Tab, 235](#)
- [DB2390IMG Advanced Parameters Dialog Box, 236](#)
- [Advanced Parameters Dialog Box \(IMS\), 237](#)

Database Row Test Overview

You can run a database row test on a PowerExchange resource to preview data for all data sources and to generate restart tokens for PowerExchange change data capture sources.

You can also run a database row test to issue PowerExchange Listener commands through the PowerExchange Navigator.

To preview data, you can run a database row test on the following PowerExchange resources:

- Data maps. A database row test validates data maps and enables you to preview source data for tables in data maps.
- Extractions maps. A database row test enables you to preview captured change data.

- Personal metadata profiles. A database row test enables you to preview source data for tables in personal metadata profiles.

A database row test accesses data in a data source and displays it in table format in the **Database Row Test Output** window. The results of a data map or extraction map row test indicate that PowerExchange can access data when a bulk data movement operation or a change data extraction runs.

In addition to previewing data for data sources in a database row test, you can use SQL DTLDESCRIBE statements to preview metadata in a data map or an extraction map. Depending on the data source type, you can preview metadata for columns, foreign and primary keys, procedures and procedure columns, records, schemas, and tables. If you preview metadata in the **Database Row Test** dialog box, PowerExchange generates an SQL DTLDESCRIBE statement in the **SQL Statement** box. You can edit this statement.

Note: The SQL DTLDESCRIBE syntax is valid for both relational and nonrelational sources.

For information about the DTLDESCRIBE syntax, see the *PowerExchange Reference Manual*.

If you run a database row test to generate restart tokens for PowerExchange change data capture sources, the PowerExchange Navigator displays the generated restart tokens in the **Database Row Test Output** window. You can copy the output from this window to populate the PWXPC restart token file.

Showing or Hiding the Database Row Test Output Window

You can show or hide the **Database Row Test Output** window.

If you show it, the window displays the results of the last database row test at the bottom of the PowerExchange Navigator GUI.

To show or hide the **Database Row Test Output** window:

- ▶ Click **View > Output**.

Note: This option is enabled after you run a database row test.

Database Row Test and Nonstandard Code Page Data

Nonstandard code page data, including multibyte data, is available for selection from DB2 on both z/OS and i5/OS, and the PowerExchange Navigator can display double-byte characters.

The PowerExchange Navigator uses a UTF-8 code page, which enables the display of a wide-character set.

Before you preview data with a database row test, verify that data will appear in the correct format by editing the following statements in the PowerExchange configuration file on the system from which you access data:

- DB2CODEPAGE for DB2 on z/OS
- DB2_BIN_CODEPAGE for DB2 on i5/OS

For more information about code pages, see the *PowerExchange Reference Manual*.

Testing a Data Map

You can run a database row test to validate a data map and preview data for the tables in a data map.

A database row test accesses data in a data source and displays it in table format in the **Database Row Test Output** window.

1. Open a data map.

2. On the **Data Map** tab in the **Resource Explorer**, click a table.
3. To get data for specific columns, select these columns in the **Table** window.
PowerExchange adds the selected columns to the SQL SELECT statement that is generated in the **SQL Statement** box in the **Database Row Test** dialog box.
Note: This setting does not affect SQL DTLDESCRIBE requests.
4. Click **File > Database Row Test**.
The **Data Map Remote Node** dialog box appears.
5. Enter connection information for the remote system and click **OK**.
The **Database Row Test** dialog box appears.
6. In the **DB Type** list, click the data source type. Then, enter any information that is associated with the data source type.
Note: Select the **AS4_META** data source type only at the direction of Informatica Global Customer Support.
7. In the **Fetch** list, select **Data** to preview data, or select another option to preview metadata for an item such as a table or a schema.
The row test generates an SQL statement in the **SQL Statement** box based on the selected **Fetch** option:
 - For the **Data** option, the row test generates an SQL SELECT statement.
 - For a metadata option, the row test generates an SQL DTLDESCRIBE statement.
8. Optionally, edit the SQL SELECT or DTLDESCRIBE statement in the **SQL Statement** box.
9. If you selected the **CAPX** or **CAPXRT** access method, enter an application name in the **Application** box.
10. If you selected the **CAPX**, **CAPXRT**, **DB2390IMG**, or **NRDB** access method, click **Advanced** to enter override values for the row test. Then, in the **Advanced Parameters** dialog box, enter any of the available override values for the source type. These values override certain values in PowerExchange definitions or configuration files, such as data maps or the DBMOVER configuration file.
11. In the **Get n Rows** field, enter the number of rows that you want the row test to fetch. You can display up to 99999 rows of data at a time. By default, PowerExchange displays ten rows of data.
12. If you selected the **CAPX**, **CAPXRT**, **NRDB**, or **NRDB2** data source type and selected **Data** in the **Fetch** list, you can have the row test skip a specific number of initial rows before displaying data. To skip rows, specify a number from 1 to 99999 in the **Skip First n Rows** field.
Restriction: If you are testing a SEQ data map that has the **Variable** property set to **VS** to indicate a variable-length stream data file, the database row test returns warning message PWX-03042 and the **Skip First n Rows** option is ignored. The PowerExchange Navigator cannot skip rows for a variable-length stream data file.
13. Click **Go**.
The **Database Row Test Output** window displays the results of the database row test.

RELATED TOPICS:

- [“Database Row Test Dialog Box” on page 219](#)
- [“CAPX Advanced Parameters Dialog Box - General Tab” on page 228](#)
- [“CAPX Advanced Parameters Dialog Box - AS400 Tab” on page 230](#)
- [“CAPXRT Advanced Parameters Dialog Box - General Tab” on page 230](#)
- [“CAPXRT Advanced Parameters Dialog Box - Adabas Tab” on page 234](#)
- [“CAPXRT Advanced Parameters Dialog Box - AS400 Tab” on page 234](#)
- [“CAPXRT Advanced Parameters Dialog Box - DB2UDB Tab” on page 235](#)

- [“CAPXRT Advanced Parameters Dialog Box - MSSQL Tab” on page 235](#)
- [“CAPXRT Advanced Parameters Dialog Box - Oracle Tab” on page 235](#)
- [“DB2390IMG Advanced Parameters Dialog Box” on page 236](#)
- [“Advanced Parameters Dialog Box \(IMS\)” on page 237](#)

Testing an Extraction Map

You can run a database row test for an extraction map to preview captured change data and test changes to a capture registration.

1. Open an extraction group and an extraction map.
2. To get data for specific columns, select the columns in the **Extract Definition** window.
PowerExchange adds the selected columns to the SQL SELECT statement that is generated in the **SQL Statement** box in the **Database Row Test** dialog box.
Note: Selecting specific columns in the table does not affect SQL DTLDESCRIBE requests.
3. Click **File > Database Row Test**.
The **Database Row Test** dialog box appears.
4. In the **DB Type** list, click the data source type. Then, enter any information that is associated with the data source type.
Note: Select the **AS4_META** data source type only at the direction of Informatica Global Customer Support.
5. In the **Fetch** list, select **Data** to preview data, or select another option to preview metadata for an item such as a table or a schema.
The row test generates an SQL statement in the **SQL Statement** box based on the selected **Fetch** option:
 - For the **Data** option, the row test generates an SQL SELECT statement.
 - For a metadata option, the row test generates an SQL DTLDESCRIBE statement.
6. Optionally, edit the SQL SELECT or DTLDESCRIBE statement in the **SQL Statement** box.
7. If you selected the **CAPX** or **CAPXRT** access method, enter an application name in the **Application** box.
8. If you selected the **CAPX**, **CAPXRT**, **DB2390IMG**, or **NRDB** access method, click **Advanced** to enter override values for the row test. Then, in the **Advanced Parameters** dialog box, enter any of the available override values for the source type. These values override certain values in PowerExchange definitions or configuration files, such as data maps or the DBMOVER configuration file.
9. In the **Get n Rows** field, enter the number of rows that you want the row test to fetch. You can display up to 99999 rows of data at a time. By default, PowerExchange displays ten rows of data.
10. If you selected the **CAPX**, **CAPXRT**, **NRDB**, or **NRDB2** data source type and selected **Data** in the **Fetch** list, you can have the row test skip a specific number of initial rows before displaying data. To skip rows, specify a number from 1 to 99999 in the **Skip First n Rows** field.
11. Click **Go**.
The **Database Row Test Output** window displays the results of the database row test, which show the changes that occurred on the table that you registered for data capture.
If you added before or after image columns to the extraction map, these columns also appear in the database row test output.

RELATED TOPICS:

- [“Adding or Removing Change Indicator and Before Image Columns for a Data Column” on page 189](#)

Testing a Personal Metadata Profile

You can run a database row test for a table in a personal metadata profile to preview data.

Warning: The PowerExchange Navigator allows you to select columns with datatypes that PowerExchange does not support. A database row test fails when you include columns with unsupported datatypes.

1. Open a personal metadata profile.
2. On the **Personal Metadata** tab in the **Resource Explorer**, double-click a table to open it.
3. To get data for specific columns, select these columns in the **Table** window.
PowerExchange adds the selected columns to the SQL SELECT statement that is generated in the **SQL Statement** box in the **Database Row Test** dialog box.
4. Click **File > Database Row Test**.
The **Database Row Test** dialog box appears.
5. In the **DB Type** list, click the data source type. Then, enter any information that is associated with the data source type.
Note: Select the **AS4_META** data source type only at the direction of Informatica Global Customer Support.
6. The **Fetch** list is disabled. For a personal metadata profile, the default option is **Data**.
7. Optionally, edit the SQL SELECT or DTLDESCRIBE statement in the **SQL Statement** box.
8. If you selected the **CAPX** or **CAPXRT** access method, enter an application name in the **Application** box.
9. If you selected the **CAPX**, **CAPXRT**, **DB2390IMG**, or **NRDB** access method, click **Advanced** to enter override values for the row test. Then, in the **Advanced Parameters** dialog box, enter any of the available override values for the source type. These values override certain values in PowerExchange definitions or configuration files, such as data maps or the DBMOVER configuration file.
10. In the **Get n Rows** field, enter the number of rows that you want the row test to fetch. You can display up to 99999 rows of data at a time. By default, PowerExchange displays ten rows of data.
11. If you selected the **CAPX**, **CAPXRT**, **NRDB**, or **NRDB2** data source type and selected **Data** in the **Fetch** list, you can have the row test skip a specific number of initial rows before displaying data. To skip rows, specify a number from 1 to 99999 in the **Skip First n Rows** field.
12. Click **Go**.
The **Database Row Test Output** window displays the results of the database row test.

Issuing PowerExchange Listener Commands in a Database Row Test

You can use the Database Row Test dialog box to issue a LISTTASK or STOPTASK command to a PowerExchange Listener that is running on any system or to issue a LISTLOCATIONS command.

The PowerExchange Listener LISTTASK command lists the client tasks that are running under the Listener, and the STOPTASK command stops a task. The LISTLOCATIONS command provides information about the locations that are defined in the NODE and SVCNODE statements of the DBMOVER configuration file.

1. In the **Resource Explorer**, double-click a data map that is defined for the location where the PowerExchange Listener is running to open the data map.
Note: Alternatively, you can open an extraction map or personal metadata profile that is defined for the PowerExchange Listener location.
2. On the **Data Map** tab, select a table view, and then click **File > Database Row Test** on the menu bar. A message might prompt you to send the data map to a remote location.
3. In the **Data Map Remote Node** dialog box, enter connection information for the location where the PowerExchange Listener is running, and click **OK**.
The **Database Row Test** dialog box appears.
4. In the **DB Type** list, select **TASK_CNTL**.
5. In the **Fetch** list, select one of the following commands:
 - **List Locations.** Displays information about the locations that are defined in NODE or SVCNODE statements in the DBMOVER configuration file on the system where the PowerExchange Listener is running. The output includes the node name, IP address, port number, send and receive buffer sizes and lengths, receive timeout, and SSL use.
 - **List Task.** Displays information about each active task that is running under the PowerExchange Listener. The output includes the task ID, TCP/IP address, port number, application name, access type, and status.
 - **Stop Task.** Stops a specific PowerExchange Listener task. To identify the task, you must enter a task ID or application name.
6. If you are issuing a STOPTASK command, enter a task ID or application name in the **SQL Statement** box. Use the following syntax:

```
stoptask {taskid=task_id|appname=application_name}
```

Do not include the curly brackets. These brackets indicate a choice of either taskid or appname is required.
Note: Do not enter the application name in the **Application** field. The **Application** field is ignored for TASK_CNTL commands.
7. If you are issuing a LISTLOCATIONS command, optionally enter the node type in the **SQL Statement** box. Use the following syntax:

```
listlocations nodetype={N|A|S}
```

Specify one of the following values for nodetype:
 - N. Default. List locations that are defined in NODE statements in the DBMOVER file.
 - A. List locations that are defined in NODE or SVCNODE statements in the DBMOVER file.
 - S. List locations that are defined in SVCNODE statements in the DBMOVER file.
8. Click **Go**.
The **Database Row Test Output** window displays the output for the command.

RELATED TOPICS:

- [“Sending a Data Map to a Remote System” on page 90](#)

Generating Restart Tokens for Change Data Capture

You can run a database row test to generate restart tokens for data sources.

Use these generated restart tokens to populate the PWXPC restart token file for a PowerCenter session. PowerExchange generates restart tokens using the location and source type that you specify. These restart tokens represent the current end of the change stream at the time when you ran the database row test. The PowerExchange Navigator displays the generated restart tokens in the **Database Row Test Output** window. Copy the output from this window to populate the PWXPC restart token file.

Alternatively, use one of the following methods to generate restart tokens:

- Include the special override statement with CURRENT_RESTART in the PWXPC restart token file. PWXPC and PowerExchange generate restart tokens that represent the current end of the change stream at the time the PowerCenter session runs.
- Use the DTLUAPPL utility with the GENERATE RSTTKN parameters and a valid capture registration. DTLUAPPL generates restart tokens that represent the current end of the change stream at the time that the utility runs.

1. Open the extraction map for the capture registration for which you want to generate restart tokens.
2. On the **Extraction Map** tab in the **Resource Explorer**, click the extraction map and click **File > Database Row Test**.

The **Database Row Test** dialog box appears.

3. In the **DB Type** list, select **CAPXRT**.
4. In the **Location** list, select the node name that represents the location of the data source.
The locations displayed in the **Location** list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.
5. In the **Application** box, enter an application name from 1 to 20 characters in length.
6. To generate current restart tokens, enter the following PowerExchange SQL statement in the **SQL Statement** box:

```
SELECT CURRENT_RESTART [WHERE [{CONNAME=capi_connection_name|
CONTYPE=connection_type}] [{AND] INSTANCE=db2_database_name]]
```

This SELECT CURRENT_RESTART statement has the following parameters:

CONNAME=capi_connection_name

Specifies the name of a CAPI_CONNECTION statement in the DBMOVER file at the source location that PowerExchange uses to determine the data source type. PowerExchange requires this information because restart token formats vary by source type. Enter this parameter if multiple CAPI_CONNECTION statements are defined at the source location. Otherwise, this parameter is optional, and you can specify CONTYPE instead. Do not specify both CONNAME and CONTYPE.

CONTYPE=connection_type

Specifies the connection type that PowerExchange uses to determine the data source type. PowerExchange requires this information because restart token formats vary by source type.

Specify one of the following values for the *conn_type* variable:

- **ADA** for Adabas sources (z/OS).
- **AS4** for DB2 for i5/OS sources.
- **DCM** for CA Datacom/DB sources (z/OS).
- **DB2**. DB2 for z/OS sources (z/OS).
- **IDL** for CA IDMS/DB log-based sources (z/OS).
- **IML** for IMS log-based sources (z/OS).
- **IMS** for IMS synchronous sources (z/OS).
- **ORA** for Oracle sources.
- **MSS** for Microsoft SQL Server sources.
- **MYSQL** for MySQL sources.
- **PGS** for PostgreSQL sources.
- **UDB** for DB2 for Linux, UNIX, and Windows sources.
- **VSAM** or **VSM**. VSAM sources (z/OS).

For z/OS sources, all z/OS connection types result in restart tokens of the same format because PowerExchange records all change data from z/OS sources in the PowerExchange Logger for z/OS log files. Restart tokens for z/OS sources represent locations within the PowerExchange Logger logs.

INSTANCE=db2_database_name

For a DB2 for Linux, UNIX, and Windows source, specifies the DB2 database name. This parameter value is required unless you enter the DB2 database name in the **Override File Name** field. In the WHERE clause of the SELECT CURRENT_RESTART statement, you can specify this parameter only, or use the optional AND keyword to specify this parameter with the CONNAME or CONTYPE parameter.

7. Click **Go**.

The **Database Row Test Output** window displays the generated restart tokens.

8. To copy the generated restart tokens, right-click in the **Database Row Test Output** window and click **Copy Output**.

You can use the generated restart tokens in the PWXPC restart token file.

Database Row Test Dialog Box

In the Database Row Test dialog box, you can set options for a database row test.

The dialog box includes the following options:

DB Type

The type of data source or access method. For extraction maps, default is **CAPXRT**.

Select an option and enter information in the related fields that depend on the option type:

Option	Description	Related Fields
AS4_META - Use this option only at the direction of Informatica Global Customer Support.	-	-
CAPX	An access method that reads changes from PowerExchange Condense condense files or PowerExchange Logger for Linux, UNIX, and Windows log files in batch and then stops.	In the Override File Name field, optionally enter the name of the file that overrides the default file name.
CAPXRT	An access method that reads changes directly from the change stream or from PowerExchange Condense condense files or PowerExchange Logger for Linux, UNIX, and Windows log files in real time.	In the Override File Name field, optionally enter the name of the file that overrides the default file name.
DB2	DB2 for z/OS, DB2 for i5/OS, and DB2 for Linux, UNIX, and Windows databases	In the SSID or DBName field, enter the DB2 database name, or for DB2 for z/OS, enter the subsystem ID or data-sharing group name.
DB2390IMG	DB2 for z/OS image copy of uncompressed table spaces from a remote system	In the Override File Name field, optionally enter the name of the file that overrides the default file name. In the SSID field, enter the DB2 for z/OS subsystem ID.
DB2400C	An access method that uses ODBC standard calls to access DB2 for i5/OS data	In the SSID or DBName field, enter the DB2 database name.
DB2UDB	DB2 for Linux, UNIX, and Windows database	In the SSID or DBName field, enter the DB2 database name.
IMSUNLD	IMS unload file	In the File Name field, enter the name of the IMS unload file.
MSSQL	Microsoft SQL Server database	In the DSN field, enter the SQL Server instance name. In the Database field, optionally enter a database name that overrides the database name that is specified for the connection.
MYSQL	MySQL database server	In the Server field, enter the server name or IP address of the MySQL server where the source database is located.

Option	Description	Related Fields
NRDB	Nonrelational data sources such as a VSAM data sets, IMS databases, or flat files. Note: This data source type uses a three-tier naming convention in the generated SQL.	In the Override File Name field, optionally enter the name of a file that overrides the default file name from the data map. In the File Password field, enter the password for the file, if required.
NRDB2	Nonrelational data sources Note: This data source type uses a two-tier naming convention in the generated SQL.	In the Override File Name field, optionally enter the name of a file that overrides the default file name from the data map. In the File Password field, enter the password for the file, if required.
ODBC	ODBC data source	In the DSN field, enter the ODBC data source name. In the Database field, enter the database name.
ORACLE	Oracle database	In the SQL*Net Name field, optionally enter a TNS name for the Oracle instance.
PGSQL	PostgreSQL database server	Server. The database server name.
Simulator	This option is primarily for Informatica internal use. Do not select it.	Ignore the DB Qual1 field.
TASK_CTRL	Option that enables you to issue a PowerExchange Listener command through the PowerExchange Navigator. For more information, see "Issuing PowerExchange Listener Commands in a Database Row Test" on page 217.	-

Location

Select the node name of the location of the data source.

This field lists the locations that are defined in NODE statements in the DBMOVER configuration file on the PowerExchange Navigator system. A value of **local** indicates that the source is on the PowerExchange Navigator system.

If you are testing an IMS data map with the DL/1 access method, enter the node name of the location where the netport job runs. If you testing an IMS data map with the ODBA access method, enter the node name for the location where the PowerExchange Listener runs.

User ID

A user ID that allows access to the source. The type of user ID depends on the source type and PowerExchange security settings in the SECURITY statement of the DBMOVER configuration file.

For a source on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the *PowerExchange Reference Manual*.

For a Microsoft SQL Server source for which you use Microsoft NTLM and Active Directory authentication, enter the user ID in the format *domain\user_name*. Then in the **Password** field, enter the password for this user ID.

Password

A password for the user ID, if specified.

If the source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

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

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

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

Fetch

The type of information that the database row test accesses and displays. The options that are available depend on the **DB Type** value. For the **TASK_CTRL** type, this field lists the type of command to issue to a PowerExchange Listener.

Note: For personal metadata profiles, the **Fetch** list is not available, and the default value is **Data**.

The following table describes the options in the **Fetch** list:

Option	Available for Data Source Types	Description
Columns	All, except TASK_CTRL	Displays metadata for columns.
Data	All, except TASK_CTRL	Displays rows of data.
Foreign keys	All, except TASK_CTRL	Displays metadata for foreign keys.
Primary keys	All, except TASK_CTRL	Displays metadata for primary keys.
Procedure Cols	- AS4_META - DB2 - DB2400C - ODBC - Simulator	Displays metadata for specific stored procedures.

Option	Available for Data Source Types	Description
Procedures	<ul style="list-style-type: none"> - AS4_META - DB2 - DB2400C - ODBC - Simulator 	Displays metadata for the available stored procedures.
Records	<ul style="list-style-type: none"> - NRDB - NRDB2 	Displays metadata for NRDB and NRDB2 records.
Schemas	<ul style="list-style-type: none"> - CAPX - CAPXRT - DB2 - DB2390IMG - DB2400C - DB2UDB - IMSUNLD - MSSQL - MYS - NRDB - NRDB2 - ODBC - ORACLE - PGSQL 	Displays metadata for schema names for the specified data source type.
Tables	All, except TASK_CTRL	Displays metadata for tables.
List Locations	TASK_CTRL	Displays information about the NODE statements that are defined in the DBMOVER configuration file on the system where the PowerExchange Listener is running.
List Task	TASK_CTRL	Displays information about each active PowerExchange Listener task, including the TCP/IP address, port number, application name, access type, and status.
Stop Task	TASK_CTRL	Stops an individual PowerExchange Listener task based on the application name that you specify.

Default is **Data**. Use **Data** to preview data.

Access Methods

The **Access Methods** list is available if you selected the following combinations of options:

- The **DB Type** value is **NRDB** or **NRDB2**, and the **Fetch** value is **Tables** or **Columns**
- The **DB Type** value is **CAPX** or **CAPXRT**, and the **Fetch** value is **Tables**.

Select one or more access methods if you want to retrieve information for specific access methods. PowerExchange adds filtering keywords to the DTLDESCRIBE statement in the **SQL Statement** box.

If you do not select an access method, the DTLDESCRIBE statement returns information for all access methods.

The following table describes the options in the **Access Methods** list:

Access Method	Description	DTLDESCRIBE Keyword
DM_ADABAS	Adabas	A
DM_CAPX	- CAPX - CAPXRT	C
DM_DATACOM	Datacom	X
DM_DB2	- DB2 for i5/OS - DB2 for z/OS	Z
DM_DB2390IMG	DB2390IMG	G
DM_DB2UNLD	DB2 unload file	W
DM_DL1	IMS DL/1 batch	D
DM_ESDS	VSAM ESDS	E
DM_IDMS	IDMS	I
DM_KSDS	VSAM KSDS	K
DM_MQ	MQSeries	Q
DM_ODBA	IMS ODBA	O
DM_RRDS	VSAM RRDS	N
DM_SEQ	Sequential data set or flat file	S
DM_TAPE	Tape	T
DM_USER	User access method	U
DM_XDB2	DB2 for i5/OS or DB2 for z/OS extraction map. Displayed only if the data source type is CAPX or CAPXRT,	B
DM_XDB2UDB	DB2 for Linux, UNIX, and Windows extraction map . Displayed only if the DB Type is CAPX or CAPXRT,	V
DM_XMSSQL	Microsoft SQL Server extraction map. Displayed only if the DB Type is CAPX or CAPXRT,	L
DM_XORACLE	Oracle extraction map. Displayed only if the DB Type is CAPX or CAPXRT,	P

Tip: To clear all selections in the **Access Methods** list, right-click in the **Access Methods** list and click **Deselect All**.

SQL File

Uses a file that contains an SQL statement instead of using the generated SQL in the **SQL Statement** box. If you select this option, enter the path and file name of the SQL file in the **SQL File** box, or click the Browse button to browse to the file.

Default is cleared.

Edit

Enabled if the **SQL File** option is selected.

Click **Edit** to edit the SQL file specified in the **SQL File** box.

SQL Record Length

Enabled if the **SQL File** option is selected.

The record length of the SQL in the SQL file.

SQL Statement

Displays the generated SQL, DTLDESCRIBE SQL, or PowerExchange Listener command statement for the specified data source type and **Fetch** options.

You can edit the statement in the **SQL Statement** box.

Note: If you select the **SQL File** option, the **SQL Statement** box is disabled.

PowerExchange generates the following statements in the **SQL Statement** box for the following data source type and **Fetch** option combinations:

- Any data source type except **TASK_CTRL**, and **Data** in the **Fetch** list

An SQL statement is generated and displayed in the **SQL Statement** box. PowerExchange generates the following SELECT statement for data maps that use a two-tier naming convention:

```
select * from schema.map_table
```

PowerExchange generates the following SELECT statement for data maps that use a three-tier naming convention:

```
select * from schema.map.table
```

- Any data source type except **TASK_CTRL** and a metadata option in the **Fetch** list

A DTLDESCRIBE statement is generated and displayed in the **SQL Statement** box. The metadata options include **Columns**, **Foreign Keys**, **Primary Keys**, **Procedure Cols**, **Procedures**, **Schemas**, and **Tables**.

- The **TASK_CTRL** data source type and a PowerExchange Listener command in the **Fetch** list

A PowerExchange Listener command is generated and displayed in the **SQL Statement** box.

Note: Use the **AS4_META** data source type at the direction of Informatica Global Customer Support.

Meta Data Filters

Depending on the selected **DB Type** option and the **Fetch** option, optionally enter filter criteria in the **Meta Data** fields.

Note: None of the filter fields are available if you select **Data** in the **Fetch** list.

In any filter field, you can use the following wildcard characters:

- Asterisk (*) represents one or more matching characters.
- Question mark (?) represents a single matching character.

The following table identifies the filter fields that are available for different combinations of **DB Type** and **Fetch** options:

DB Type	Fetch Options	Filter Fields
AS4_META, Simulator	Columns, Foreign Keys, Primary Keys, Tables	- Table
CAPX, CAPXRT, DB2390IMG, IMSUNLD, NRDB2, MYSQL, Oracle, PGSQL	Columns, Foreign Keys, Primary Keys, Tables	- Schema - Table
CAPX, CAPXRT, DB2390IMG, IMSUNLD, NRDB2, MYSQL, Oracle, PGSQL	Schemas	- Schema
DB2, DB2400C, DB2UDB	Columns, Foreign Keys, Primary Keys, Tables	- Creator/Schema For DB2 for i5/OS, maximum length is 10 characters. For DB2 for z/OS, maximum length is 128 characters - DBName/Definer - Table For DB2 for i5/OS, maximum length is 32 characters. For DB2 for z/OS, maximum length is 128 characters.
DB2, DB2400C, DB2UDB	Schemas	- Creator/Schema
DB2, DB2400C, ODBC	Procedure calls, Procedures	- Catalog - Schema - Procedure Name
MSSQL, ODBC	Columns, Foreign Keys, Primary Keys, Tables	- Owner - Table For both fields, maximum length is 128 characters.
MSSQL, ODBC	Schemas	- Owner
NRDB	Columns, Foreign Keys, Primary Keys, Records, Tables	- Schema Maximum length is 10 characters. - Mapname The name of a PowerExchange data map. Maximum length is 10 characters. - Table Maximum length is 128 characters.
NRDB	Schemas	- Schema

Also, in the **Meta Data** group box, select any of the following check boxes if available:

- **Respect case.** By default, text entered in the filter fields is converted to the standard case of the data source. Select this check box for the filter fields for which you do *not* want PowerExchange to convert text to the standard case of the data source. For example, for an Oracle source, uppercase is the standard. If this check box is cleared, the entries `scott` and `SCOTT` would both be converted to `SCOTT`. However, if you selected this check box, the `scott` entry would be retained in lowercase.
- **Comments.** Select this check box to include any comments in the metadata that the row test fetches for database objects.
- **Extensions.** Select this check box to return extended metadata, if appropriate for the select data source type.

Advanced

Available for the **CAPX**, **CAPXRT**, **DB2390IMG**, **IMSUNLD**, **IMS NRDB**, and **NRDB2** data source types.

Click **Advanced** to display the **Advanced Parameters** dialog box. In this dialog box, you can enter parameter values that override certain settings in PowerExchange configuration files, data maps, and extraction maps.

Application

Required for the **CAPX** and **CAPXRT** data source types.

The application name. Enter a character string of 1 to 20 characters.

Escape Character

Enter the escape character that to use to delimit an asterisk (*) or question mark (?) that is used as a literal value rather than as a wildcard in the **Meta Data** filter boxes.

For example, if you specify an escape character of `~`, a filter value of `tab*` returns all tables beginning with `tab`. A filter value of `tab~*` returns the table named `tab*`.

Note: Do not use an escape character with data sources that are multibyte-enabled.

Get *n* Rows

Indicates the number of rows to fetch.

Valid values are 1 to 99999. Default is 10.

Note: A large value can increase PowerExchange Navigator response time and CPU usage.

Skip First *n* Rows

Indicates the number of initial rows of data to skip when the row test fetches data for display. If you are performing a row test on a large file, you can use this option to have the row test skip to a specific point in the file before displaying data. This option can help you diagnose problems in large files faster because only the data of potential interest is returned for analysis.

This option is available only if you select **CAPX**, **CAPXRT**, **NRDB**, or **NRDB2** in the **DB Type** list and select **Data** in the **Fetch** list.

Valid values are 0 to 99999. Default is 0, which causes no rows to be skipped.

A large value can increase PowerExchange Navigator response time and CPU usage.

Restriction: If you are testing a SEQ data map that has the **Variable** property set to **VS** to indicate a variable-length stream data file, the database row test returns warning message PWX-03042 and the **Skip First *n* Rows** option is ignored. The PowerExchange Navigator cannot skip rows for a variable-length data file.

Valid values are:

- 0. Returns the EOF when the end of the current log is reached. The end of the log is determined at the start of the extraction process, because the actual end point constantly changes.
- 2 to 86399. Waits the specified number of seconds for data before returning the EOF.
- 86400. Waits continually for an unlimited amount of time, without returning the EOF.

Default is 15.

No Progress Update

Indicates whether to add database row test output to the PowerExchange CDC audit trail for the application.

- Select this check box to omit database row test output from the PowerExchange CDC audit trail.
- Clear this check box to include database row test output in the PowerExchange CDC audit trail.

By default, the check box is selected.

Restart Token 1

This field is available only if you selected **CC** in the **Extract Type** field. If you want the database row test to start fetching data from a specific point in the change stream, enter a valid sequence token value in this field. The sequence token in combination with the restart token value in the **Restart Token 2** field identify a specific change record in the change stream from which the database row test will begin retrieving data. You can use these fields to bypass old records that have been archived off disk or to bypass records that are not consistent with an updated extraction map.

The sequence token is a binary value with a length that varies by data source type. You can generate a sequence token and restart token for the current end of the change stream in the following ways:

- In the PowerExchange Navigator, perform a CAPXRT database row test that specifies the SELECT CURRENT_RESTART statement in the **SQL Statement** box. The **Database Row Test Output** window displays the generated token values.
- Use the DTLUAPPL utility to generate the sequence token and restart token values. In the ADD or MOD APPL statement, include the RSTTKEN GENERATE parameter. Then use the PRINT APPL statement to print the generated token values.

Note: The generated sequence token is in the format that is used by ODBC extractions. To use the token value for a database row test, add eight zeroes to the end of the value.

Alternatively, if you have run PWXPC CDC sessions recently, you can copy the token values from the following messages:

- In the PowerExchange message log, the PWX-04565 and PWX-09959 messages show the sequence token in the **Sequence** field and the restart token in the **PowerExchange Logger** field
- In the PowerCenter workflow session log, the PWXPC_12060 and PWXPC_12068 messages show the sequence token in the **Restart Token 1** field and the restart token in the **Restart Token 2** field. Also, the PWXPC_10081, PWXPC_10082, and PWXPC_12128 messages show the sequence token as the first token value and the restart token as the second token value.

If you need assistance with creating valid token values, contact Informatica Global Customer Support.

Restart Token 2

This field is available only if you selected **CC** in the **Extract Type** field. If you want the database row test to start fetching data from a specific point in the change stream, enter a valid restart token value in this field. The restart token value in combination with the sequence token value in the **Restart Token 1** field identify a specific change record in the change stream from which the database row test will begin

retrieving data. You can use these fields to bypass old records that have been archived off disk or bypass records that are not consistent with an updated extraction map.

The restart token is a binary value with a length that usually varies by data source type. On z/OS, restart tokens for all data source types have the same length, except for change data extracted from full condense files. You can generate restart token and sequence token values that indicate the current end of the change stream from the PowerExchange Navigator or with the DTLUAPPL utility. Alternatively, if you have run PWXPC CDC sessions recently, you can copy the token values from messages in the session log or the PowerExchange message log. If you need assistance with creating valid token values, contact Informatica Global Customer Support.

Extraction Schema

Overrides the schema that is specified in the extraction map.

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPX Advanced Parameters Dialog Box - AS400 Tab

On the **AS400** tab of the **CAPX Advanced Parameters** dialog box, define an optional override parameter for the database row test of batch access to DB2 for i5/OS source data.

AS400 Instance

Overrides the instance name, or collection ID, of the extraction map that you are testing. Use this override field in the following situation:

- You have identical tables that use different AS400 objects, such as different journals or schemas.
- PowerExchange Condense used an alternative DBID setting (an instance name) for each of these tables when writing change data to condense files.
- For the database row test, you want to override the instance name from the extraction map to test another table in the set of identical tables for which PowerExchange Condense processed changes.

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPXRT Advanced Parameters Dialog Box - General Tab

On the **General** tab of the **CAPXRT Advanced Parameters** dialog box, define any optional override parameters that you want to use for the database row test of real-time access to change data.

Extract Type

The extraction type. Select one of the following options:

- **RS.** Restarts the extraction from the start point of the last extraction. This option enables you to run the same extraction multiple times.

- **SL.** Starts the extraction from the point at which the last extraction ended. The extraction uses restart tokens for z/OS or checkpoint timestamps for i5/OS.
- **CC.** Starts the extraction from the point that is indicated by the token values the **Restart Token 1** and **Restart Token 2** fields on this tab. If you select this option, you must specify valid token values. For z/OS sources, if you use the default value of Y for the FUZZYRSTART parameter in the LRAP CAPI_CONNECTION statement, you can enter a **Restart Token 2** position other than a begin-uow position from which to start extraction processing in the PowerExchange Logger for z/OS logs. In the following example, the **Restart Token 2** value points to a RBA position that does not coincide with a begin-uow record and the **Restart Token 1** value points to a position before the **Restart Token 2** position:
 - In the **Restart 1** field, specify 00000000000100. This value consists of 48 zeroes with a "1" in the twelfth position. With this value, the Restart Token 2 value will determine the records to be returned by the row test.
 - In the **Restart 2** field, specify a value that consists of a 6-byte Logger ID in EBCDIC format (padded with spaces) + a 6-byte RBA position + 8 zeroes. The RBA position can be any RBA in the logs near where you want to restart extraction processing. For example: E2C2F2D340400000000AEF4000000000

With these restart token settings, PowerExchange begins extraction processing at the first record that is at or past the Restart 1 position and in the first UOW that is returned based on the Restart 2 position.

Default is **RS**.

Image Type

The capture image type. Select one of the following options:

- **AI.** Captures the after-image data.
- **BA.** Captures the before-image data as it existed before the latest change. Also, captures the after-image data as it existed after the latest change.

The before-image data is shipped with an action character forcing the deletion of the record. The after-image data, which includes the change data, is delivered as an insert operation.
- **TU.** Captures the before- and after-image data, as it occurs. The application uses the before-image data to verify that no changes occurred in the record since the request. The after-image data appears as an update operation. The update is a full image, but the application that captures this update can change the target record instead of deleting the record and then inserting a new record, as occurs with BA. The TU option with ODBC connections.

Default is **AI**.

Timeout

The maximum amount of time, in seconds, to wait for data before returning an end-of-file (EOF).

Valid values are:

- 0. Returns the EOF when the end of the current log is reached. The end of the log is determined at the start of the extraction process, because the actual end point constantly changes.
- 2 to 86399. Waits the specified number of seconds for data before returning the EOF.
- 86400. Waits continually for an unlimited amount of time, without returning the EOF.

Default is 15.

No Progress Update

Indicates whether to add database row test output to the PowerExchange CDC audit trail for the application.

Select this check box to omit database row test output from the PowerExchange CDC audit trail.

Clear this check box to include database row test output in the PowerExchange CDC audit trail.

By default, this check box is selected.

Restart Token 1

This field is available only if you selected **CC** in the **Extract Type** field. If you want the database row test to start fetching data from a specific point in the change stream, enter a valid sequence token value in this field. The sequence token in combination with the restart token value in the **Restart Token 2** field identify the specific change record in the change stream from which the database row test will begin retrieving data. You can use these fields to bypass old records that have been archived off disk or to bypass records that are not consistent with an updated extraction map.

The sequence token is a binary value with a length that varies by data source type. You can generate a sequence token and restart token for the current end of the change stream in the following ways:

- In the PowerExchange Navigator, perform a CAPXRT database row test that specifies the SELECT CURRENT_RESTART statement in the **SQL Statement** box. The **Database Row Test Output** window displays the generated token values.
- Use the DTLUAPPL utility to generate the sequence token and restart token values. In the ADD or MOD APPL statement, include the RSTTKEN GENERATE parameter. Then use the PRINT APPL statement to print the generated token values.

Note: The generated sequence token is in the format that is used by ODBC extractions. To use the token value for a database row test, add eight zeroes to the end of the value.

Alternatively, if you have run PWXPC CDC sessions recently, you can copy the token values from the following messages:

- In the PowerExchange message log, the PWX-04565 and PWX-09959 messages show the sequence token in the **Sequence** field and the restart token in the **PowerExchange Logger** field
- In the PowerCenter workflow session log, the PWXPC_12060 and PWXPC_12068 messages show the sequence token in the **Restart Token 1** field and the restart token in the **Restart Token 2** field. Also, the PWXPC_10081, PWXPC_10082, and PWXPC_12128 messages show the sequence token as the first token value and the restart token as the second token value.

If you need assistance with creating valid token values, contact Informatica Global Customer Support.

Restart Token 2

This field is available only if you selected **CC** in the **Extract Type** field. If you want the database row test to start fetching data from a specific point in the change stream, enter a valid restart token value in this field. The restart token value in combination with the sequence token value in the **Restart Token 1** field identify the specific change record in the change stream from which the database row test will begin retrieving data. You can use these fields to bypass old records that have been archived off disk or bypass records that are not consistent with an updated extraction map.

The restart token is a binary value with a length that usually varies by data source type. On z/OS, restart tokens for all data source types have the same length, except for change data extracted from full condense files. You can generate restart token and sequence token values that indicate the current end of the change stream from the PowerExchange Navigator or with the DTLUAPPL utility. Alternatively, if you have run PWXPC CDC sessions recently, you can copy the token values from messages in the session log or the PowerExchange message log. If you need assistance with creating valid token values, contact Informatica Global Customer Support.

Extraction Schema

Overrides the schema that is specified in the extraction map.

CAPI Connection Name

Overrides the default database connection that is defined in a CAPI_CONNECTION statement in the DBMOVER configuration file.

To point to the relevant data source, use the value that you defined in the NAME parameter of the CAPI_CONNECTION statement in the DBMOVER configuration file.

Required when multiple CAPI connection statements are included in the DBMOVER configuration file.

Extraction Map - Location

If you are extracting change data in continuous extraction mode from PowerExchange Logger for Linux, UNIX, and Windows log files that are remote from the extraction maps on the data source, enter the node name of the system where the extraction maps are located.

You must define this node in a NODE statement in the dbmover.cfg file where the PowerExchange Logger for Linux, UNIX, and Windows runs. Usually, this node name is the node name that you defined in the CAPTURE_NODE parameter in the pwxcl.cfg configuration file.

For example, if a PowerExchange Logger for Linux, UNIX, and Windows process captures change data from a remote IMS source on an z/OS system, specify the node name for the z/OS system.

Extraction Map - UserID

The user ID that is used to authorize PowerExchange Logger for Linux, UNIX, and Windows access to extraction maps on the remote node that is selected in the **Location** list.

Whether this user ID is required depends on the SECURITY setting in the DBMOVER configuration file at the remote location. This user ID can be either an operating system user ID or database ID, depending on the data source type.

Usually, this user ID is the one that you defined in the CAPTURE_NODE_UID parameter in the pwxcl.cfg configuration file.

For data sources on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name. For more information, see the *PowerExchange Reference Manual*.

Extraction Map - Password

A clear text password for the user ID that is specified in the **UserID** box. Usually, this value matches the CAPTURE_NODE_PWD parameter value in the pwxcl.cfg configuration file.

If you are using the PowerExchange Logger to access extractions maps that are on a remote i5/OS or z/OS node, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

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

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

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPXRT Advanced Parameters Dialog Box - Adabas Tab

On the **Adabas** tab of the **CAPXRT Advanced Properties** dialog box, you must define a parameter for the database row test of real-time access to Adabas source data if the Adabas FDT is password protected.

ADABAS File Password

The Adabas FDT password.

Required if the Adabas FDT is password protected in Adabas.

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPXRT Advanced Parameters Dialog Box - AS400 Tab

On the **AS400** tab of the **CAPXRT Advanced Properties** dialog box, define an optional override parameter for the database row test of real-time access to DB2 for i5/OS source data.

AS400 Library/Journal

Overrides the library and journal name that are specified in the JOURNAL parameter in the AS4J CAPI_CONNECTION statement in the DBMOVER configuration file on the i5/OS system. For example:

```
STQA/NEWJOURNAL
```

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPXRT Advanced Parameters Dialog Box - DB2UDB Tab

On the **DB2UDB** tab of the **CAPXRT Advanced Parameters** dialog box, define an optional override parameter for the database row test of real-time access to DB2 for Linux, UNIX, and Windows source data.

DB2 UDB Database

Overrides the DB2 for Linux, UNIX, and Windows database name from the PowerExchange extraction map.

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPXRT Advanced Parameters Dialog Box - MSSQL Tab

On the **MSSQL** tab of the **CAPXRT Advanced Parameters** dialog box, you must define a parameter for the database row test of real-time access to Microsoft SQL Server source data if you use the PowerExchange Logger for Linux, UNIX, and Windows in continuous mode and capture data for SQL Server articles in multiple publication databases.

MSS LUW DBId

If you use the PowerExchange Logger for Linux, UNIX, and Windows in continuous extraction mode and capture change data from a Microsoft SQL Server distribution database for articles in multiple publication databases, enter the DBID parameter value that is specified in the PowerExchange Logger configuration file, `pwxcl.cfg`. This DBID value identifies the collection of registrations to use for row test processing. If you do not specify this value, the row test fails.

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

CAPXRT Advanced Parameters Dialog Box - Oracle Tab

On the **Oracle** tab of the **CAPXRT Advanced Parameters** dialog box, define optional overrides for the database row test of real-time access to Oracle source data.

Oracle Instance

Overrides the Oracle instance in the second positional parameter, `oracle_sid`, in the ORACLEID statement in the `dbmover.cfg` configuration file.

```
ORACLEID=(coll_id,oracle_sid,connect_string,cap_connect_string)
```

This parameter is used in conjunction with the **Oracle Connection String** parameter to enable the use of a single set of capture registrations for capturing data from multiple Oracle instances.

If you specify an **Oracle Instance** value but no **Oracle Connection String** value, PowerExchange uses the Oracle connection string value that you define in the ORACLEID statement in the dbmover.cfg file.

Oracle Connection String

Overrides the Oracle connection string value in the third positional parameter, *connect_string*, in the ORACLEID statement in the dbmover.cfg configuration file.

This parameter is used in conjunction with the **Oracle Instance** value to enable the use of a single set of capture registrations for capturing data from multiple Oracle instances.

If you specify an **Oracle Connection String** value but no **Oracle Instance** value, PowerExchange uses the Oracle connection string value that you define in the ORACLEID statement in the dbmover.cfg file.

Oracle Schema

Overrides the Oracle schema that is specified in the registration group to enable the use of a single set of capture registrations for capturing data from multiple schemas, which might exist in the Oracle instance.

Click **OK** to save changes. The source database connection closes. In the **Database Row Test** dialog box, click **Go** to establish the database connection again and perform the database row test with the updated advanced properties.

DB2390IMG Advanced Parameters Dialog Box

The DB2390IMG access method creates a data map dynamically as part of its processing.

Enter a schema name and data map name for the data map. PowerExchange constructs the data map name from the schema name and map name.

For example:

schema_name.data_map_name.dmp

Enter information to define DB2390IMG advanced parameters in a database row test.

Schema Name

Schema name for the data map.

Must begin with an alphabetic character.

Maximum length of ten bytes.

Data Map Name

Map name for the data map.

Must begin with an alphabetic character.

Maximum length of ten bytes.

Advanced Parameters Dialog Box (IMS)

Enter overrides for a database row test of an IMS data map that uses the DL/1 BATCH or IMS ODBA access method or that accesses an IMS unload file source. You can override the access method and related properties.

Access Method

To override the access method that is specified in the data map, select one of the following options:

- **DL/1 BATCH.** If the data map specifies the IMS ODBA access method, select this option to use DL/1 batch or BMP access to IMS data for the database row test. Also, set the **Location** value in the **Database Row Test** dialog box to point to the node where the netport job runs.
- **IMS ODBA.** If the data map specifies the DL/1 BATCH access method, select this option to use IMS ODBA access to IMS data for the database row test. Also, set the **Location** value in the **Database Row Test** dialog box to point to the node where the PowerExchange Listener runs.

By default, the access method from the data map is initially displayed. To override it, select the other available access method in the list.

If you select the blank row in the list, the database row test uses the access method from the data map. You can still enter any of the other overrides in this dialog box that are valid for that access method.

If you change the access method without also changing the **Location** value, PowerExchange issues one of the following error messages:

- For DL/1 access:

```
PWX-02200 Incorrect environment for DL1 execution.  
Check NETPORT for server job or executable for client.
```
- For ODBA access:

```
PWX-02220 ODBA "ALLOCATE" Call Failed RC=00000104 Reason=00001110.
```

Important: If you switch the access method, ensure that you configured PowerExchange to use the new access method. For more information, see [“Configuring DL/1 Batch or BMP Access to IMS Data” on page 116](#) and [“Configuring IMS ODBA Access to IMS Data” on page 117](#).

IMS SSID

If you are testing an IMS ODBA data map and did not override the access method, use this value to override the IMS subsystem ID (SSID) from the data map. If you specified IMS ODBA access as an override in the **Access Method** field, you must enter this value. A SSID is required for ODBA access.

If you are testing a data map with an IMS unload file source, use this override to point to another IMSID statement in the DBMOVER member for the purpose of changing from one DBD library to another DBD library. For example, if you have a DL/I data map but want to run the row test against an IMS unload file to reduce queries against the database, enter the database type of **IMSUNLD**, the location of the unload file, and the unload file name in the **Database Row Test** dialog box. Then, in this field, enter the IMS SSID that is associated with the IMSID statement that includes the DBD library you want to use for the IMS unload file.

If you are testing a data map for a NRDB database type and using a BMP netport job, you can use this override with the %IMSID substitution variable in the netport JCL. This override replaces the substitution variable to specify the IMS SSID to use for the row test. By using the override with the substitution variable, you can use the same netport JCL to access multiple IMS environments, such as development, testing, and production environments.

Note: An IMS SSID is not required for DL/I batch access to IMS data or for access to an IMS unload file.

PSB Name

Enter an override IMS PSB name.

If you are testing an ODBA data map and did not override the access method, this value overrides the PSB name from the data map. If you specified **IMS ODBA** as an override in the **Access Method** field, you must enter this value. A PSB name is required for ODBA access. By using this override, you can use the same data map to test access to different PSBs.

If you are testing a DL/I batch or BMP data map and specified the PSB=%PSBNAME substitution variable in the netport JCL, this value overrides the PSB name in the JCL. If you specified the PSB=%1 substitution variable instead of %PSBNAME in the netport JCL, the row test uses the PSB name in the NETPORT statement, if specified. Informatica recommends using this override with %PSBNAME substitution variable to avoid editing or adding NETPORT statements and restarting the PowerExchange Listener.

Note: A PSB name is not used for access to an IMS unload file (IMSUNLD database type).

PCB Number

Enter an override PCB number. This field is available only for DL/I or BMP access to the IMS data.

If you are testing a DL/I batch or BMP data map and did not override the access method, this value overrides the PCB number from the data map. If you specified **DL/1 BATCH** as an override in the **Access Method** field, you must enter this value. A PCB number is required for DL/I or BMP access.

Note: A PCB number is not used for ODBA access or for access to an IMS unload file (IMSUNLD database type).

PCB Name

Enter an override IMS PCB name. The PCB name is available only for ODBA access to IMS data.

If you are testing an IMS ODBA data map for the source and did not override the access method, this value overrides the PCB name from the data map. If you specified **IMS ODBA** as an override in the **Access Method** field, you must enter this value. A PCB name is required for ODBA access.

Note: A PCB name is not used for DL/I batch or BMP access or for access to an IMS unload file (IMSUNLD database type).

CHAPTER 9

PowerExchange Navigator Examples

This chapter includes the following topics:

- [PowerExchange Navigator Examples Overview, 239](#)
- [Adding a Single-Record Data Map - Example, 240](#)
- [Adding a Single-Record Data Map with an Array - Example, 244](#)
- [Adding a Multiple-Record Data Map with a Complex Table - Example, 246](#)
- [Importing a COBOL Copybook into a Data Map - Example, 248](#)
- [Importing a COBOL Copybook with REDEFINES Statements - Example, 253](#)
- [Importing a PL/I Copybook - Example, 256](#)
- [Defining Table Properties in a Data Map to Normalize an Array - Example , 259](#)
- [Calling a User Exit Program by Using the PowerExchange CallProg Function - Example , 263](#)

PowerExchange Navigator Examples Overview

PowerExchange provides example data maps, copybooks, and data files that you can use to learn how to create data maps.

Additionally, PowerExchange provides the following types of example programs written in Assembler, C, and PL/I:

- User access method programs, named `uam*. *`
- Record-level exit programs, named `urle*. *`
- Programs called by using the CallProg function, named `ucpe*. *`

Example header files and JCL are provided for these programs.

The example files are located in the examples subdirectory in the PowerExchange installation directory on Windows. For example:

```
C:\Informatica\PowerExchange.n.n\examples\
```

In this section, you use the example files to create the following types of data maps:

- Single-record data map
- Single-record data map that contains arrays

- Multiple-record data map with complex tables
- Data map with an imported COBOL copybook
- Data map with an imported COBOL copybook with REDEFINES statements
- Data map with an imported PL/I copybook with multiple definitions for a field
- Data map that normalizes a field defined as an array with multiple occurrences
- Data map with user-defined fields that use the PowerExchange CallProg function to call a user exit program that returns the class type of a field in the data map record

Adding a Single-Record Data Map - Example

In this example, you create a single-record data map.

This example shows how to complete the following tasks:

1. Add a data map.
2. Add a record to the data map.
3. Add fields to the record in the data map.
4. Add a table to the data map.
5. Test the single-record data map.

This example uses the demo1.dat data file.

Step 1. Add a Data Map

Create the demo.map1 data map by using the example demo1.dat file.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, enter the required information.

The following table describes the required information:

Option	Description
Schema Name	Enter <code>demo</code> .
Data Map Name	Enter <code>map1</code> .
Access Method	Select SEQ .
Import Record Definitions	Clear this option. You specify record definitions later.

3. Click **Next**.
4. In the **SEQ Access Method** dialog box, enter the required information.

The following table describes the required information:

Option	Description
File Name	Click the Browse button to browse to the demo1.dat file in the examples directory.
Field Separator	Select this option and enter the comma (,) separator character. If you cannot enter the separator character through the keyboard, enter its hexadecimal representation as: x'nn'. Because the demo1.dat data file is a character-separated file, you are not prompted for the separator character.

5. Click **Finish**.

The demo.map1 data map appears on the **Data Map** tab in the **Resource Explorer**.

RELATED TOPICS:

- [“Adding a Data Map” on page 62](#)

Step 2. Add a Record to the Data Map

Add the record1 record to the demo.map1 data map.

1. On the **Data Map** tab in the **Resource Explorer**, right-click demo.map1 and click **Add Record**.
The **Add Record** dialog box appears.
2. In the **Record Name** box, enter `record1`.
3. Click **OK**.

The record1 record appears in the **Record** window.

RELATED TOPICS:

- [“Adding a Data Map Record” on page 65](#)

Step 3. Add Fields to the Record in the Data Map

Add the seqno, name, and gender fields to the record1 record in the demo.map1 data map.

1. On the **Data Map** tab in the **Resource Explorer**, click **record1**.
2. Click **Add > Field**.
3. In the **Add Field** dialog box, enter information about the field.

The following table describes the information to enter:

Option	Description
Field Name	Enter <code>seqno</code> .
Field Type	Select NUMCHAR .
Precision	Enter <code>2</code> .

Option	Description
Scale	Enter 0.
Length	Enter 2. The Length field indicates the maximum size of the input, allowing for leading blanks or other characters. The length value must be equal to or greater than the precision value.

4. Click **OK**.
The **Record** window displays the seqno field.
5. In the **Record** window, click **record1**.
6. Click **Add > Field**.
7. In the **Add Field** dialog box, enter information about the field.

The following table describes the information to enter:

Option	Description
Field Name	Enter name.
Field Type	Select CHAR .
Length	Enter 20.

8. Click **OK**.
The **Record** window displays the name field.
9. On the **Data Map** tab, click **record1**.
10. Click **Add > Field**.
11. In the **Add Field** dialog box, enter information about the field.

The following table describes the information to enter:

Option	Description
Field Name	Enter gender.
Field Type	Select CHAR .
Length	Enter 1.

12. Click **OK**.
The **Record** window displays the gender field.

RELATED TOPICS:

- [“Adding a Field to a Record” on page 66](#)

Step 4. Add a Table to the Data Map

Add the table1 table to the demo.map1 data map. The columns in the table1 table are based on fields in the record1 record.

1. On the **Data Map** tab in the **Resource Explorer**, right-click demo.map1 and click **Add Table**.
2. In the **Add Table** dialog box, enter information about the table.

The following table describes the information to enter:

Option	Description
Table Name	Enter <code>table1</code> .
Record Dependencies	Verify that the record1 record appears.
Groups and Arrays only	Select this option. This option causes the fields defined as a group field, or any array, to be displayed. If you clear this option, all fields in the record appear.
How do you want to handle multiple instances of selected records	Select New Row . This option causes a new row to be created for every instance of the record.

3. Click **OK**.

The **Table** window displays the table1 table, which contains columns that are based on the fields that you added to the record1 record.

RELATED TOPICS:

- [“Adding a Data Map Table” on page 75](#)

Step 5. Test the Single-Record Data Map

Run a database row test on the demo.map1 data map. You access the data map locally.

1. In the **Table** window, select `table1`.
2. Click **File > Database Row Test**.
3. In the message box that prompts you to send the data map to a remote location, click **Yes**.
4. In the **Data Map Remote Node** dialog box, click **local** in the **Location** list.
5. Click **OK**.
6. In the **Database Row Test** dialog box, click **Go**.

The **Database Row Test Output** window displays the output.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)

Adding a Single-Record Data Map with an Array - Example

In this example, you add a single-record data map by using a data file that contains multiple record types. You set options for a table to control how PowerExchange displays output from records that contain arrays.

This example shows how to complete the following tasks:

1. Associate a data file with a data map.
2. Filter the data in the data map.
3. Create an output row for each element in an array.
4. Create one output row for all elements in an array.

This example uses the demo.map2.dmp and demo2.dat files.

Step 1. Associate a Data File with a Data Map

Associate a data file with a data map.

1. On the **Resources** tab in the **Resource Explorer**, double-click **demo.map2**.
2. On the **Data Map** tab, right-click demo.map2 and click **Properties**.
The **Data Map Properties** dialog box appears.
3. On the **SEQ Access Method** tab, click the Browse button to browse to the demo2.dat file in the `examples` directory.
4. Click **OK**.
5. Click **OK**.

RELATED TOPICS:

- [“Data Files and File-List Processing” on page 52](#)

Step 2. Filter the Data in the Data Map

Record IDs provide a filtering mechanism at run-time, eliminating the need for SQL statements to select the desired records.

1. On the **Data Map** tab in the **Resource Explorer**, click **record1**.
2. In the **Record** window, right-click the **rectype** field and click **Properties**.
The **Field Properties** dialog box appears.

3. In the **Record ID Values** list, click the = or <> button to select the operators for filtering the data. The record IDs 10 and 20 combined with the operator setting determine which data appears. Complete one of the following actions:
 - Click the = button to display records with a record ID value equal to either 10 or 20. In the **Database Row Test Output** window, any record with a record ID that is not equal to 10 or 20 does not appear.
 - Click the <> button to display records with a record ID value that is greater than or less than but not equal to 10 or 20.

Note: The data map stores the record IDs but not the data. Record ID fields might not be displayed in the column view of the data. You cannot use record ID filtering on WRITE.
4. Click **OK**.

Step 3. Create an Output Row for Each Element in an Array

Create an output row for each element in an array.

1. In the **Data Map** tab in the **Resource Explorer**, right-click the **row_out** table and click **Properties**. The **Table Properties** dialog box appears.
2. In the **Table Properties** dialog box, select options to create an output row for each element in the array. The following table describes these options:

Option	Description
Fields	Select Groups and Arrays only . This option displays fields defined as a group or an array field in the list.
Fields	Select the items field. In this list, you can select fields that are defined as groups or arrays.
How do you want to handle multiple instances of selected records	Select New Row . This option causes each occurrence of the data in the items field to appear in its own row.

3. Click **OK**.
4. On the **Data Map** tab, double click the **row_out** table to open the **Table Properties** dialog box. The **items_L** column appears in turquoise, which indicates that it is part of an array.
5. To test the data map, run a database row test. Access the data map locally and click **NRDB** in the **DB Type** list in the **Database Row Test** dialog box.

The **Database Row Test Output** window displays multiple output rows from a single input record, depending on the number of elements in the array.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)
- [“Defining Array Properties in the Table Properties Dialog Box” on page 57](#)

Step 4. Create One Output Row for All Elements in an Array

Create a single row of output for a record that contains an array. All elements in the array appear in a single row.

1. In the **Table Properties** dialog box for the row_out table, select options to create one output row for all elements in the array.

The following table describes these options:

Option	Description
Fields	Clear the items field.
How do you want to handle multiple instances of selected records	Select Array .
Array	Select 1 .
New Row on Overflow	Select this option. PowerExchange populates the output row until it is full and then displays a new row with the overflow records.

2. Click **OK**.
3. To test the data map, run a database row test. Access the data map locally and select **NRDB** in the **DB Type** list in the **Database Row Test** dialog box.

In the **Database Row Test Output** window, one row appears for each record. All elements of the array in a record appear in a single row.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)
- [“Defining Array Properties in the Table Properties Dialog Box” on page 57](#)

Adding a Multiple-Record Data Map with a Complex Table - Example

In this example, you define a complex table with a hierarchical relationship between records of different types.

This example illustrates the following concepts:

- A data map table can have multiple data map records associated with it.
- You can define a hierarchical relationship between the records.

This example shows how to complete the following tasks:

1. Associate a data file with a multiple-record data map.
2. View records in the multiple-record data map.
3. Define the hierarchy of records in the multiple-record data map.
4. Test the multiple-record data map.

This example uses the demo.map3.dmp and demo3.dat files.

Step 1. Associate a Data File with a Multiple-Record Data Map

Associate a data file with a multiple-record data map.

1. On the **Resources** tab in the **Resource Explorer**, double-click the **demo.map3** data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click **demo.map3** and click **Properties**.
The **Data Map Properties** dialog box appears.
3. On the **SEQ Access Method** tab, click the Browse button to browse to the demo3.dat file in the `examples` directory.
4. Click **OK**.
5. Click **OK**.

RELATED TOPICS:

- [“Data Files and File-List Processing” on page 52](#)

Step 2. View Records in a Multiple-Record Data Map

Record IDs identify multiple records in a flat file.

The first character of each line in the demo3.dat file, which the demo.map3 data map defines as the `rectype` field, distinguishes the `rec1` record from the `rec2` record.

1. On the **Resources** tab in the **Resource Explorer**, double-click **demo.map3**.
2. On the **Data Map** tab in the **Resource Explorer**, click the **rec1** record.
3. In the **Record - rec1** window, double-click the **rectype** field.
The **Record ID Values** box indicates that the record ID for the `rectype` field is 1.
4. On the **Data Map** tab in the **Resource Explorer**, click the **rec2** record.
5. In the **Record - rec2** window, double-click the **rectype** field.
The **Record ID Values** box indicates that the record ID for the `rectype` field is 2.

Step 3. Define a Hierarchy of Records in a Multiple-Record Data Map

Define a hierarchical relationship between the `rec1` and `rec2` records to reflect the actual hierarchy in the data source.

1. On the **Data Map** tab in the **Resource Explorer**, right-click the **row_out** table and click **Properties**.
The **Record Dependencies** list displays the hierarchy between the `rec1` and `rec2` records.
2. To delete this hierarchy, in the **Record Dependencies** list, right-click **rec2** and click **Delete**.
This action moves the `rec2` record to the **Available Records** list.
3. In the **Record Dependencies** list, right-click **rec1** and click **Delete**.
This action moves the `rec1` record to the **Available Records** list.
4. To redefine the hierarchy, in the **Available Records** list, right-click **rec1** and click **Add Record**.
This action moves the `rec1` record to the **Record Dependencies** list.

For this record, the **How do you want to handle multiple instances of selected record** option is set to **New Row**.

5. In the **Available Records** list, right-click **rec2** and click **Add Record as Child**.
This action moves the rec2 record to the **Record Dependencies** list as a child record of the rec1 parent record.
For this record, the **How do you want to handle multiple instances of selected record** option is set to **New Row**.
6. Click **OK**.
In the **Table** window, the name column is based on the name field in the rec1 record, and the address column is based on the address field in the rec2 record.
The name for each column in the table is prefixed with the record name on which the column is based.

Step 4. Test a Multiple-Record Data Map

Run a database row test on a multiple-record data map. You access the data map locally.

1. On the **Data Map** tab in the **Resource Explorer**, click **row_out**.
2. Click **File > Database Row Test**.
3. In the message box that prompts you to send the data map to a remote location, click **Yes**.
4. In the **Data Map Remote Node** dialog box, select **local** in the **Location** list.
5. Click **OK**.
6. In the **Database Row Test** dialog box, click **Go**.

In the **Database Row Test Output** window, the name column is populated from the rec1 record, and the address column is populated from the rec2 record.

If you change the hierarchy in the **Table Properties** dialog box for the row_out table to make the rec2 record a sibling of the rec1 record, the output in the **Database Row Test Output** window changes.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)

Importing a COBOL Copybook into a Data Map - Example

In this example, you add a multiple-record data map and import a COBOL copybook into it.

This example shows how to complete the following tasks:

1. Add a data map and import a COBOL copybook that contains an OCCURS DEPENDING ON clause into the data map.
When you add the data map, you use a fixed-length data file. After you add the data map, you create a hierarchical view of the flat file.
2. Apply a date mask to a field in a data map record.
3. Use a WHERE clause to filter an empty record from the output.
4. Display multiple output rows for items in an OCCURS DEPENDING ON clause.

This example uses the train3.dat and train3.cob files.

Step 1. Add a Data Map and Import a COBOL Copybook into the Data Map

Add a data map for a sequential flat file and import a COBOL copybook into the data map.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, enter information about the data map.

The following table describes the information to enter:

Option	Description
Schema Name	Enter <code>COBOL</code> .
Data Map Name	Enter <code>map2</code> .
Access Method	Select SEQ .
Import Record Definitions	Select this option.

3. Click **Next**.
4. In the **SEQ Access Method** dialog box, enter information about the data file.

The following table describes the information to enter:

Option	Description
File Name	Click the Browse button to browse to the <code>train3.dat</code> file in the <code>examples</code> directory.
Fixed	Select this option.
Size	Enter <code>60</code> .

5. Click **Finish**.
6. In the **Import Copybook - Source Details** dialog box, enter copybook information.

The following table describes the information to enter:

Option	Description
Source	Select Local .
Type	Select COBOL .
Start	Enter <code>7</code> .
End	Enter <code>72</code> .

7. Click **Next**.
8. In the **Import Copybook - Local Cobol Details** dialog box, click the Browse button to browse to the `train3.cob` file in the `examples` directory.

Note: To view the copybook, click **Preview**.

9. Click **Next**.

10. In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records, fields, and tables and click **Finish**. For more information, see [“Import Copybook - Configuration Details Dialog Box” on page 150](#).
11. In the **Import Copybook Information** window, review the information for the import and click **OK**.
12. In the **Record Definition** dialog box for the NAME_REC record and the ACCOUNT_REC record, click **OK**. For more information, see [“Record Definition and Duplicate Record Definition Dialog Boxes” on page 152](#).
The **Cobol Import** window displays the imported copybook.
The **Copybook Message Log** window displays the results of the import operation.
Note: To find the associated line in the copybook for an error that appears in the **Copybook Message Log** window, double-click the message.
13. Close the **Cobol Import** window.
14. On the **Data Map** tab in the **Resource Explorer**, click the **NAME_REC** record.
15. In the **Record** window, right-click the **RECTYPE** field and click **Properties**.
The **Name** tab in the **Field Properties** dialog box appears.
16. In the **Record ID Values** box, enter 01.
Note: To view the original properties for the field from the imported copybook, click the **Extra Properties** tab.
17. Click **OK**.
In the **Record** window, the **RECTYPE** field icon appears in green.
18. On the **Data Map** tab in the **Resource Explorer**, click the **ACCOUNT_REC** record.
19. In the **Record** window, right-click the **RECTYPE** field and click **Properties**.
The **Name** tab in the **Field Properties** dialog box appears.
20. In the **Record ID Values** box, enter 02.
21. Click **OK**.
In the **Record** window, the **RECTYPE** field icon appears in green.
22. On the **Data Map** tab in the **Resource Explorer**, right-click **COBOL.map2** and click **Add Table**.
23. In the **Add Table** dialog box, enter information about the table.

The following table describes the information to enter:

Option	Description
Table Name	Enter <i>both</i> .
Available Records	Right-click the NAME_REC record and click Add Record . Then, right-click the ACCOUNT_REC record and click Add Record as Child .

24. Click **OK**.
25. To view the table output before applying a date mask to the POLICY_DATE record, run a database row test. On the **Data Map** tab in the **Resource Explorer**, click the **both** table and click **File > Database Row Test**.
26. In the **Database Row Test** dialog box, enter information about the database row test.

The following table describes the information to enter:

Option	Description
DB Type	Select NRDB .
Location	Select local .

27. Click **Go**.
The **Database Row Test Output** window displays the output.
28. Close the **Database Row Test Output** and the **Database Row Test** windows.
29. Close the **Database Row Test Output** and **Database Row Test** windows.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)
- [“Import Copybook - Local Details Dialog Box” on page 143](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)

Step 2. Apply a Date Mask to a Field in the Data Map Record

Apply a date mask to a field in a data map and delete extraneous fields from a table before running a database row test.

1. In the **Table** window for the **both** table, right-click the following columns one by one, and click **Delete**:
 - RECTYPE
 - ACCOUNT_REC_ACCOUNT
 - ACCOUNT_REC_RECTYPE

For each record, click **OK** in the window that prompts you to confirm the deletion.

2. On the **Data Map** tab in the **Resource Explorer**, click the **ACCOUNT_REC** record.
3. In the **Record** window, right-click the **POLICY_DATE** field and click **Properties**.
4. In the **Field Format** list, select **Y2-MM-D2**.
5. Click **OK**.
6. On the **Data Map** tab in the **Resource Explorer**, click the **both** table.
7. Click **File > Database Row Test**.
8. On the **Database Row Test** dialog box, enter information about the database row test.

The following table describes the information to enter:

Option	Description
DB Type	Select NRDB .
Location	Select local .

9. Click **Go**.
In the **Database Row Test Output** window, the **POLICY_DATE** column includes the century in the date format.

RELATED TOPICS:

- [“Step 3. Use a WHERE Clause to Filter an Empty Record from the Output” on page 252](#)
- [“Testing a Data Map” on page 213](#)

Step 3. Use a WHERE Clause to Filter an Empty Record from the Output

Use a WHERE clause to filter an empty record from the output in the **Database Row Test Output** window.

1. In the **Database Row Test** dialog box, enter the following statement in the **SQL Statement** box:

```
select * from COBOL.map2.both where itemct > 0
```

2. In the **Database Row Test** dialog box, click **Go**.

In the **Database Row Test Output** window, the fourth record, which contains no items, disappears from the output.

3. Close the **Database Row Test Output** and **Database Row Test** windows.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)

Step 4. Display Multiple Output Rows for Items in an OCCURS DEPENDING ON Clause

Display multiple output rows in the **Database Row Test Output** window for items in an OCCURS DEPENDING ON clause.

1. On the **Data Map** tab in the **Resource Explorer**, right-click the **both** table and click **Properties**.
2. In the **Table Properties** dialog box, select the **ITEMS** check box in the **Fields** list.
3. Click **OK**.

In the **Table** window, the icon for the ITEMS_L column appears in turquoise.

4. On the **Data Map** tab in the **Resource Explorer**, click the **both** table.
5. Click **File > Database Row Test**.
6. In the **Database Row Test** dialog box, click **Go**.

In the **Database Row Test Output** window, one row appears for each item in the record.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)

Importing a COBOL Copybook with REDEFINES Statements - Example

You can add a multiple-record data map and import a complex COBOL copybook with REDEFINES statements into it.

This example shows how to complete the following tasks:

1. Add a data map and import a COBOL copybook into it.
When you add the data map, you use a fixed-length binary data file of EBCDIC data with 58320 records. Each record is 57 bytes long. This EBCDIC data file simulates reading data from an IBM mainframe system.

In the **Add Data Map** dialog box, you set the skip records option to skip a header record in the data file.

You import a COBOL copybook with line numbers in columns 1 through 6 and comments in columns 73 through 80. The COBOL copybook also contains REDEFINES statements.

Note: PowerExchange ignores the comment lines in the copybook.

2. Apply a date mask to a GROUP field in the data map.

This example uses the train5.dat and train5.cob files.

Step 1. Add a Data Map and Import a COBOL Copybook with REDEFINES Statements into the Data Map

Add a multiple-record data map and import a COBOL copybook with REDEFINES statements into the data map.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, enter information about the data map.

The following table describes the information to enter:

Option	Description
Schema Name	Enter COBOL.
Data Map Name	Enter map3.
Access Method	Select SEQ .
Import Record Definitions	Select this option.

3. Click **Next**.
4. In the **SEQ Access Method** dialog box, enter information about the data file.

The following table describes the information to enter:

Option	Description
File Name	Click the Browse button to browse to the train5.dat file in the <code>examples</code> directory.
Fixed	Select this option.
Size	Enter 57.
Default	Select this option.
Encoding	Select EBCDIC .
Codepage	Select Default .
Skip First	Enter 1.

After you set the encoding and code page, select the **Fixed** option again.

- Click **Finish**.
- In the **Import Copybook - Source Details** dialog box, enter information about the copybook.

The following table describes the information to enter:

Option	Description
Source	Select Local .
Type	Select COBOL .
Start	Enter 7.
End	Enter 72.

- Click **Next**.
- In the **Import Copybook - Local Cobol Details** dialog box, click the Browse button to browse to the train5.cob file in the examples directory.
Note: To view the copybook, click **Preview**.
- Click **Next**.
- In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records, fields, and tables and click **Finish**. For more information, see ["Import Copybook - Configuration Details Dialog Box" on page 150](#).
- In the **Import Copybook Information** window, review the information for the import and click **OK**.
- In the **Record Definition** dialog box for the MASTER_REC record, click **OK**.
For more information, see ["Record Definition and Duplicate Record Definition Dialog Boxes" on page 152](#).
The **Copybook Redefines** window notifies you that the copybook contains REDEFINES statements.
The **Cobol Import** window appears.
- To point to the line that contains the PIC S9(8) field, click **Redefinition > Next** or **Redefinition > Previous**.
- Click **Import > Resume**.

The **Cobol Import** window displays the imported copybook.

The **Copybook Message Log** window displays the results of the import operation.

Note: The first line in the message log refers to a record in the **Cobol Import** window. Click the first line in the message log to highlight the specified record in the **Cobol Import** window.

15. Close the **Cobol Import** window.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)
- [“Import Copybook - Source Details Dialog Box” on page 140](#)
- [“Import Copybook - Local Details Dialog Box” on page 143](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)

Step 2. Apply a Date Mask to a GROUP Field in the Data Map

Apply a date mask to a GROUP field in a data map.

1. On the **Data Map** tab in the **Resource Explorer**, click the **MASTER_REC** record.
2. In the **Record** window, and click the **MASTER_DATE** field.

The **MASTER_DATE** field is defined as a GROUP field that contains the following subfields that are defined as UZONED fields:

- DATE_YY
- DATE_MM
- DATE_DD

In the next step, delete these subfields and set properties for the **MASTER_DATE** field to create a single date column in the database row test output from the **MASTER_DATE** GROUP field.

3. In the **Record** window, select the **DATE_YY** field. Press **SHIFT** and click **DATE_DD**. Click the **Delete** button on the toolbar.
4. On the confirmation dialog box, to view details about which cross-references will be deleted with the fields, click **More Details**.
5. Click **Yes**.
6. In the **Record** window, right-click the **MASTER_DATE** field and click **Properties**.
7. In the **Field Properties** dialog box, enter information about the field.

The following table describes the information to enter:

Option	Description
Field Type	Select CHAR .
Length	Enter 6.
Field Format	Select Y2-MM-D2 . Delete the hyphens.

8. Click **OK**.
9. On the **Data Map** tab in the **Resource Explorer**, click the **MASTER_REC** table.
10. Click **Add > Column**.

11. In the **Add Column** dialog box, enter information about the column.

The following table describes the information to enter:

Option	Description
Name	Enter <code>new_date</code> .
Base Field	Select MASTER_REC:MASTER_DATE .

12. Click **OK**.
13. On the **Data Map** tab in the **Resource Explorer**, click the **MASTER_REC** table.
14. Click **File > Database Row Test**.
15. In the **Database Row Test** dialog box, click **Go**.
In the **Database Row Test Output** window, the `new_date` column displays the date.

RELATED TOPICS:

- [“Adding a Column to a Table” on page 76](#)
- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)
- [“Testing a Data Map” on page 213](#)

Importing a PL/I Copybook - Example

You can substitute a PL/I copybook for a COBOL copybook. This example shows how to add a multiple-record data map and import a PL/I copybook into it.

This example shows how to complete the following tasks:

1. Add a data map and import a PL/I copybook with multiple definitions of a field into it.
When you add the data map, you use a fixed-length binary data file of 58320 EBCDIC data records, which follow a single ASCII header record. Each record is 57 bytes long. This EBCDIC data file simulates reading data from an IBM mainframe system.
2. Test the data map.

This example uses the `train5.dat` and `train5.pl1` files.

Step 1. Add a Data Map and Import a PL/I Copybook into the Data Map

Add a data map and import a PL/I copybook into the data map.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, enter information about the data map.

The following table describes the information to enter:

Option	Description
Schema Name	Enter <code>PLI</code> .
Data Map Name	Enter <code>map5</code> .
Access Method	Select SEQ .
Import Record Definitions	Select this option.

- Click **Next**.
- In the **SEQ Access Method** dialog box, enter information about the data file.

The following table describes the information to enter:

Option	Description
File Name	Click the Browse button to browse to the <code>train5.dat</code> file in the <code>examples</code> directory.
Fixed	Select this option.
Size	Enter <code>57</code> .
Default	Select this option.
Encoding	Select EBCDIC .
Skip First	Enter <code>1</code> .

After you set the encoding, select the **Fixed** option again.

- Click **Finish**.
- In the **Import Copybook - Source Details** dialog box, enter information about the copybook.

The following table describes the information to enter:

Option	Description
Source	Select Local .
Type	Select PL/1 .
Start	Enter <code>1</code> .
End	Enter <code>72</code> .

- Click **Next**.
- In the **Import Copybook - Local PL/1 Details** dialog box, click the Browse button to browse to the `train5.pl1` file in the `examples` directory.

Note: To view the copybook, click **Preview**.

- Click **Next**.

10. In the **Import Copybook - Configuration Details** dialog box, select the actions for imported records, fields, and tables and click **Finish**. For more information, see [“Import Copybook - Configuration Details Dialog Box” on page 150](#).
11. In the **Import Copybook Information** window, review the information for the import and click **OK**.
12. In the **Record Definition** dialog box for the MASTER_REC record, click **OK**.
For more information, see [“Record Definition and Duplicate Record Definition Dialog Boxes” on page 152](#).
The **Copybook Redefines** message box notifies you that the copybook contains two definitions of the BIN_NO field.
The **PL/1 Import** window appears.
13. To point to line 6, which contains the FIXED BINARY(16) field, click **Redefinition > Next** or **Redefinition > Previous**.
14. Click **Import > Resume**.
The **PL/1 Import** window displays the imported copybook.
The **Copybook Message Log** window displays the results of the import operation.
15. Close the **PL/1 Import** window.

RELATED TOPICS:

- [“Testing a Data Map” on page 213](#)
- [“Import Copybook - Source Details Dialog Box” on page 140](#)
- [“Import Copybook - Local Details Dialog Box” on page 143](#)
- [“Import Copybook - Remote Details Dialog Box” on page 145](#)

Step 2. Test the Data Map

Run a database row test on the MASTER_REC table in the data map into which you imported the PL/I copybook.

1. On the **Data Map** tab in the **Resource Explorer**, click the **MASTER_REC** table.
2. Click **File > Database Row Test**.
3. In the message box, click **Yes** to send the PL1.map5 data map to a remote location.
The **Data Map Remote Node** dialog box appears.
4. In the **Data Map Remote Node** dialog box, accept the default location of local and click **OK**.
The **Database Row Test** dialog box appears.
5. In the **Database Row Test** dialog box, click **Go**.
The **Database Row Test Output** window displays the output.

Defining Table Properties in a Data Map to Normalize an Array - Example

To define the layout of multiple records one time rather than multiple times, define an array with multiple occurrences. Then, define table properties in the data map to normalize the array and produce the correct output from the source file.

In this example, you add user-defined fields to the data map. These fields invoke the PowerExchange GenVRowKey and GetSeqWithinLevel functions, which demonstrate the normalization of the array in the database row test output.

This example shows how to complete the following tasks:

1. Add a data map for a sequential flat file.

When you add the data map, you import a COBOL copybook that defines an array with multiple occurrences into the data map.

2. Add user-defined fields to the data map.

These user-defined fields invoke the PowerExchange GenVRowKey and GetSeqWithinLevel functions, which demonstrate the normalization of the array in the database row test output.

For a source row that contains an array, the GenVRowKey function generates multiple table rows identified by unique row numbers.

The GetSeqWithinLevel function gets the sequence number of a record in the current hierarchical level.

3. Define table properties that normalize the array in the database row test output.

4. Run a database row test to display the results of normalization.

In the database row test output, one row appears for each occurrence in the array.

In this example, you use the demo2.dat data file and the demo2.cob copybook to create the demo.map2array data map.

The demo2.cob copybook uses the following OCCURS clause to define an array with multiple occurrences:

```
04 ITEM OCCURS 3 DEPENDING ON ITEMCT PIC X(15).
```

Step 1. Add a Data Map that Contains an Array Defined with Multiple Occurrences

In this step, you add a data map for a sequential flat file and import a COBOL copybook that defines an array with multiple occurrences into the data map.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, enter the following information:
 - In the **Schema Name** box, enter **demo**.
 - In the **Data Map Name** box, enter **map2array**.
 - Select the **Import Record Definitions** option.
 - In the **Access Method** list, select **SEQ**.
3. Click **Next**.

4. In the **SEQ Access Method** dialog box, enter the following information:
 - In the **File Name** box, click the Browse button to browse to the demo2.dat file in the examples directory.
 - In the **Record Format** area, select the **Field Separator** option and enter the comma (,) character in the **Field Separator** box.
The demo2.dat data file is a comma (,) separated file.
5. Click **Finish**.
6. In the **Import Copybook - Source Details** dialog box, enter the following information:
 - In the **Source** area, select **Local**.
 - In the **Type** list, select **COBOL**.
 - In the **Column Range** area, enter **7** in the **Start** box and **72** in the **End** box.
7. Click **Next**.
8. In the **Import Copybook - Local Cobol Details** dialog box, enter the following information:
 - Click the Browse button to browse to the demo2.cob file in the examples directory.
 - To view the copybook, click **Preview**. The copybook includes the following OCCURS clause to define an array with multiple occurrences:

```
04 ITEM OCCURS 3 DEPENDING ON ITEMCT PIC X(15).
```
9. Close the preview window. In the **Import Copybook - Local Cobol Details** dialog box, click **Next**.
10. In the **Import Copybook - Configuration Details** dialog box, review the selected actions for imported records, fields, and tables and click **Finish**.
11. In the **Import Copybook Information** window, review the information for the import and click **OK**.
12. To import the RECORD_IN record, click **OK** in the **Record Definition** dialog box.
13. To import the fields in the record, click **OK** in the **Field Definition** dialog box for each field.
14. To import the RECORD_IN table, click **OK** in the **Table Definition** dialog box.
The **Cobol Import** window displays the imported copybook. The **Copybook Message Log** window displays the results of the import operation.
15. Close the **Cobol Import** and the **Copybook Message Log** windows.
16. On the **Resources** tab in the **Resource Explorer**, select the RECORD_IN table and click **File > Database Row Test**.
The PowerExchange Navigator displays an error message that indicates that the RECTYPE field in the RECORD_IN record is an invalid type for a delimited data file. Additionally, the ITEMCT field is an invalid type.
17. Change the datatype of the RECTYPE and ITEMCT fields by completing the following actions:
 - In the **Record** window for the RECORD_IN record, double-click the RECTYPE field.
In the **Field Properties** dialog box, select a **Field Type** of NUMCHAR. Enter 2 in the **Length** box. Click **OK**.
 - In the **Record** window, double-click the ITEMCT field.
In the **Field Properties** dialog box, select a **Field Type** of NUMCHAR. Enter 1 in the **Length** box. Click **OK**.
18. On the **Resources** tab in the **Resource Explorer**, select the RECORD_IN table and click **File > Database Row Test**.
19. On the **Database Row Test** dialog box, click **Go**.
The **Database Row Test Output** window displays the results of the database row test.

Step 2. Add User-Defined Fields that Demonstrate the Normalization of an Array

In this step, you add user-defined fields that invoke PowerExchange functions that demonstrate the normalization an array with multiple occurrences.

The user-defined fields invoke the PowerExchange GenVRowKey and GetSeqWithinLevel functions. These functions add data to the database row test output that demonstrate how the table properties that you defined normalize the array. When you run a database row test, PowerExchange displays one row for each occurrence in the array. For more information about these functions, see [“GenVRowKey” on page 289](#) and [“GetSeqWithinLevel” on page 301](#).

1. Open the demo.map2array data map and the RECORD_IN record.
2. In the **Record** window, click the **Expr(0)** tab.
3. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
4. In the cell in the **Name** column, enter `rlgenvrow_idx`.
5. In the cell in the **Type** column, select **NUM32U**.
6. In the cell in the **Phase** column, select **R**.
This selection indicates a read operation.
7. To enter an expression for the field, click in the cell in the **Expression** column and click the Browse button.
The **Expression Editor** dialog box appears.
8. In the **Function List** list in the **Expression Editor** dialog box, double-click the GenVRowKey function.
9. In the **Expression List** list, enter `()` at the end of the GenVRowKey function name. Click **Validate**.
In the **Validate** box, the `No Errors` message appears.
10. Click **OK**.
11. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
12. In the cell in the **Name** column, enter `rlgenseq`.
13. In the cell in the **Type** column, select **NUM32**.
14. In the cell in the **Phase** column, select **R**.
This selection indicates a read operation.
15. To enter an expression for the field, click in the cell in the **Expression** column and click the Browse button.
The **Expression Editor** dialog box appears.
16. In the **Function List** list in the **Expression Editor** dialog box, double-click the GetSeqWithinLevel function.
17. In the **Expression List** list, enter `()` at the end of the GetSeqWithinLevel function name. Click **Validate**.
In the **Validate** box, the `No Errors` message appears.
18. Click **OK**.

Step 3. Define Table Properties to Normalize an Array

In this step, you define table properties that normalize an array in a data map.

1. Open the demo.map2array data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the RECORD_IN table and click **Properties**.

The **Table Properties - Definition** dialog box appears.

3. In the **Column Generation** list, select **Refresh with missing columns**.

Because the record on which the table is based contains new fields, corresponding columns are added to the table.

4. Click **OK**.

5. On the **Data Map** tab in the **Resource Explorer**, right-click the RECORD_IN table and click **Properties**.

The **Table Properties - Definition** dialog box appears.

6. In the **Table Properties - Definition** dialog box, complete the following actions:

- In the **Column Generation** list, select **Apply array format changes**.

Because the record on which the table is based contains changed fields that are defined as arrays, those changes are reflected in the corresponding columns in the table.

- In the **How do you want to handle multiple instances of selected record** list, select **New Row**.

This action causes PowerExchange to display a new row for each occurrence in the array in a database row test.

7. Click **OK**.

For more information about the **Table Properties - Definition** dialog box, see ["Table Properties - Definition" on page 364](#).

Step 4. Test the Normalization of the Array in the Data Map

In this step, you run a database row test to test the normalization of the array in the data map.

1. Open the demo.map2array data map.
2. On the **Data Map** tab in the **Resource Explorer**, select the RECORD_IN table and click **File > Database Row Test**.

The **Database Row Test** dialog box appears.

3. In the **Database Row Test** dialog box, click **Go**.

The **Database Row Test Output** window appears.

The following image displays the **Database Row Test Output** window:

Row Number	RE...	NAME	SEX	ITE...	ITEM_L	r1genvrow_idx	r1genseq
1	10	Mark Jones	M	3	apple	1	1
2	10	Mark Jones	M	3	orange	2	1
3	10	Mark Jones	M	3	pear	3	1
4	20	Shirley Wong	F	1	raspberry	1	2
5	20	John Jackson	M	2	pansy	1	3
6	20	John Jackson	M	2	daisy	2	3
7	10	Donald Leary	M	0	<null>	1	4
8	30	David Wu	M	1	fox	1	5
9	10	Jean Connor	F	3	dog	1	6
10	10	Jean Connor	F	3	cat	2	6
11	10	Jean Connor	F	3	rabbit	3	6
12	20	Ronald Rose	M	2	horse	1	7
13	20	Ronald Rose	M	2	pony	2	7
14	20	Betsy Martin	F	1	wolf	1	8

1. Results of the GenVRowKey function
2. Results of the GetSeqWithinLevel function

The r1genvrow_idx column displays the results of the GenVRowKey function, which is the number of the occurrence in the array. The r1genseq column displays the results of the GetSeqWithinLevel function, which is the number of the record.

For example, the following results appear in the output:

- The first row displays data for occurrence 1 in the array in record 1.
- The third row displays data for occurrence 3 in the array in record 1.
- The fourth row displays data for occurrence 1 in the array in record 2.

Calling a User Exit Program by Using the PowerExchange CallProg Function - Example

In this example, you call a user exit program by adding user-defined fields to a data map record to invoke the PowerExchange CallProg function.

The user exit program returns the class type for the data in a specified field. The user exit program tests any field with a maximum length of 15 bytes.

The following table describes the class types that the user exit program returns:

Class Type	Description
A	Alphabetic
H	High values
L	Low values
N	Numeric zoned decimal
S	Spaces

This example shows how to complete the following tasks:

1. Add a data map by using a sample data file and copybook that ship with PowerExchange on z/OS.
2. Create and compile a user exit program. Link-edit the program into the PowerExchange Listener LOADLIB library on z/OS.
3. Add the following user-defined fields to the MASTER_REC data map record:
 - The classtype_bin_no, classtype_dec_no, and classtype_rec_type fields.
Defined as a one-byte CHAR fields. The user exit program that is called by the CallProg function returns the class type of a specified field in these fields. You must define a separate classtype field for each field for which you want to check the class type.
 - The rc_bin_no, rc_decimal_no, and rc_rec_type fields.
Defined as NUM32 fields. Use these fields to call the CallProg function and to contain the return code from the user exit program call.

Before PowerExchange completes data checking for a data map record, it runs any expressions and program calls defined in user-defined fields in the data map record.

4. Refresh the columns in the MASTER_REC table to pick up the user-defined fields that you added to the MASTER_REC record.
5. Run a database row test on the data map record to test the results of the user exit program and to verify that the user exit program runs correctly.

Step 1. Add a Data Map

In this step, you add a data map for a sequential flat file and import a COBOL copybook.

To add the data map, use data set members that ship with PowerExchange on the z/OS system.

The following table describes the data set members:

Data Set Member	Description
KSDSDAT	Data file that contains source data.
KSDSCOB	COBOL copybook that you import to define the layout of the data.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, define properties for the data map.

The following table describes the properties:

Property	Value
Schema Name	demo
Data Map Name	userexit
Access Method	SEQ

Also, select the **Import Record Definitions** option.

3. Click **Next**.
4. In the **SEQ Access Method** dialog box, enter information about the data file.

The following table describes the information to enter:

Property	Value	Notes
File Name	<code>PWX_installation_dataset.DTLDEMO(KSDSDAT)</code>	Where <code>PWX_installation_dataset</code> is the PowerExchange installation data set. For example, you might enter the following value for the file name: <code>PWX.V901.DTLDEMO(KSDSDAT)</code>
Record Format	Default	-
Skip First	0	-

Also, verify that the **File List Processing** option is disabled. By default, this option is disabled.

5. Click **Finish**.

- In the **Import Copybook - Source Details** dialog box, define properties for the copybook.

The following table lists the values to enter:

Property	Value
Source	Remote
Type	COBOL
Column Range Start	7
Column Range End	72

- Click **Next**.
- In the **Import Copybook - Remote Cobol Details** dialog box, define properties for the copybook on the remote system.

The following table describes the information to enter:

Property	Value	Notes
File Name	<i>PWX_installation_dataset</i> .DTLDEMO (KSDSCOB)	Where <i>PWX_installation_dataset</i> is the PowerExchange installation data set. For example, you might enter the following value for the file name: <i>PWX.V901.DTLDEMO (KSDSCOB)</i>
Location	Node for the z/OS system	-
UserID	The user ID for the z/OS system	-

Property	Value	Notes
Password	<p>The password or valid PowerExchange passphrase for the user ID on the z/OS system.</p> <p>A passphrase for z/OS access can be from 9 to 128 characters in length and can contain the following characters:</p> <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p>	-
Save File Locally As	A name for the copybook file on the local Windows system	-

Note: If you want to preview the copybook, click **Preview**. After you preview the copybook, close the preview window.

9. Click **Next**.
10. In the **Import Copybook - Configuration Details** dialog box, review the selected actions for imported records, fields, and tables, and click **Finish**.
11. In the **Import Copybook Information** window, review the information for the import and click **OK**.
12. In the **Record Definition** dialog box for the MASTER_REC record, click **OK**.
The **Copybook Redefines** message box appears indicating that two definitions for the BIN_NO field exist in the copybook. For this example, the first definition is the correct one.
13. To accept the first definition for the BIN_NO field, click **Import > Goto Current Line**. Then, click **Import > Resume** to resume the import.
The **Cobol Import** window displays the imported copybook and the **Copybook Message Log** window displays the results of the import operation.
14. Close the **Cobol Import** window.
15. To verify that the data map was added correctly, on the **Resources** tab in the **Resource Explorer**, select the MASTER_REC table and click **File > Database Row Test**.
16. When you are prompted to send the data map to a remote location, click **Yes**.
The **Data Map Remote Node** dialog box appears.
17. In the **Data Map Remote Node** dialog box, enter the user ID, password, and the node for the z/OS system.
The **Database Row Test** dialog box appears.

18. In the **Database Row Test** dialog box, accept the default values and click **Go**.
 The **Database Row Test Output** window displays the results of the database row test.
 The following image displays the **Database Row Test Output** window:

RowNumber	ACCOUNT	REC_TY...	AMOUNT	BIN_NO	DECIMAL...	DAT...	DAT...	DAT...	ACCT_CODES_1	ACCT_CODES_2	ACCT_CODES_3
1	Abby	A	3312.34	-1	-111	99	12	11	Automobile	Automobile	Life
2	Charles	A	-112.34	65535	-119	99	12	3	Automobile	Automobile	Life
3	Cindy	B	12.34	-1	-111	99	12	2	Automobile	Life	Boat
4	Esther	B	112.34	-1	-111	99	12	5	Automobile	Automobile	Life
5	James	A	-1012.34	65535	-119	99	12	6	Automobile	Life	Boat
6	Jason	A	1312.34	15	111	99	12	10	House	Life	Motorcycle
7	John	A	112.34	15	111	99	12	4	House	Boat	Automobile
8	Larry	A	12.34	15	111	99	12	1	Automobile	Life	House
9	Luke	A	-112.34	65535	-119	99	12	3	House	Life	Life
10	Maggie	B	1012.34	-1	-111	99	12	8	Automobile	House	Life

Step 2. Create and Compile the User Exit Program

In this step, you copy the UCPEC user exit program and modify it. Then, compile the user exit program and link-edit the program into the PowerExchange Listener LOADLIB library.

PowerExchange ships sample user exit programs, including the UCPEC program, in the SRCLIB library in the installation data set on z/OS.

1. In the SRCLIB library in the PowerExchange installation data set on z/OS, copy the UCPEC user exit program and name it UCPGCLSC.
2. Edit the UCPGCLSC program and make the changes marked in bold, as shown in the following code:

```

IDENTIFICATION DIVISION.
PROGRAM-ID. UCPGCLSC.
*****
* GLOBAL CUSTOMER SUPPORT SAMPLE CLASS TEST
* EXAMPLE COBOL PROGRAM CALLED VIA CALLPROG.
*
*****          *****
* USER EXITS ARE NOT SUPPORTED BY INFORMATICA
* USER EXITS ARE USED AT THE CUSTOMERS OWN RISK
*
*
* USING SYNTAX :-
* CALLPROG('UCPGCLSC', 'UCPGCLSC', 'COBOL', 'VOID',
*          TEXT_FIELD, NUMBER1_FIELD)
*
* RECEIVES THE FOLLOWING ARGUMENTS :-
* 1. NUMBER-ARGUMENTS - REQUIRED
*    THE NUMBER OF ARGUMENTS WHICH FOLLOW.
*    THE PROGRAM WILL EXIT SETTING A BAD RETURN CODE
*    IF THE NUMBER IS NOT WHAT IT EXPECTS.
*
* 2. FAILURE-CODE. - REQUIRED
*    AN INTEGER PASSED BACK TO THE CALLER TO INDICATE IF
*    PROCESSING WAS WAS NOT SUCCESSFUL.
*    THE FAILURE-CODE IS MONITORED
*    SO THAT ACTION CAN BE TAKEN TO HANDLE ERRORS.
*
*    BECAUSE THE MVS COBOL LINKAGE TYPE ONLY SUPPORTS A
*    RETURN TYPE OF 'VOID', IT IS NECESSARY TO PASS IT
*    BACK AS A NORMAL FIELD WITH ITS ACCOMPANYING LENGTH.
*    (SEE CLLPRGL2 FOR HOW THE RETURN CODE CAN BE PASSED
*    USING A LINKAGE TYPE OF 'OS' RETURNING 'INT')
*
* 3. MESSAGE-BUFFER. - REQUIRED
*    AN ERROR INTO WHICH THE PROGRAM CAN PUT A MESSAGE
*    TO ACCOMPANY A NON-ZERO FAILURE CODE, INDICATING
*    THE REASON.

```

```

*
* 4. MESSAGE-BUFFER-LENGTH. - REQUIRED
*   THE LENGTH OF MESSAGE-BUFFER
*
* 5. TEXT-AREA.
*   THIS IS THE 5TH ARGUMENT TO CALLPROG DEFINED IN THE
*   NAVIGATOR EXPRESSIONS SCREEN.
*   IN THIS EXAMPLE, IT IS A FIELD CONTAINING A MAX OF 15 BYTES
*
* 6. TEXT-AREA-LENGTH.
*   THE LENGTH OF TEXT-AREA WHICH VARIES ACCORDING
*   TO THE ACTUAL FIELD LENGTH ON THE FILE.
*
* 7. CLASS-TYPE.
*   THIS IS THE 6TH ARGUMENT TO CALLPROG DEFINED IN THE
*   NAVIGATOR EXPRESSIONS SCREEN.
*   IN THIS EXAMPLE, IT IS A 1 BYTE CHARACTER FIELD WITH VALUES
*   S=SPACES, L=LOW-VALUES, H=HIGH-VALUES, A=ALPHABETIC, N=NUMERIC
*
* 8. CLASS-TYPE-LENGTH.
*   THE LENGTH OF FIELD CLASS-TYPE WHICH WILL ALWAYS
*   BE 1.
*****
*
ENVIRONMENT DIVISION.
*
DATA DIVISION.
WORKING-STORAGE SECTION.
*
01 WS-DATE      PIC X(6).
01 WS-TIME      PIC X(8).
01 WS-DATA.
   05 WS-DATA15  PIC X(15).
01 WS-DATAL14 REDEFINES WS-DATA.
   05 WS-DATA14  PIC X(14).
   05 FILLER     PIC X(01).
01 WS-DATAL13 REDEFINES WS-DATA.
   05 WS-DATA13  PIC X(13).
   05 FILLER     PIC X(02).
01 WS-DATAL12 REDEFINES WS-DATA.
   05 WS-DATA12  PIC X(12).
   05 FILLER     PIC X(03).
01 WS-DATAL11 REDEFINES WS-DATA.
   05 WS-DATA11  PIC X(11).
   05 FILLER     PIC X(04).
01 WS-DATAL10 REDEFINES WS-DATA.
   05 WS-DATA10  PIC X(10).
   05 FILLER     PIC X(05).
01 WS-DATAL09 REDEFINES WS-DATA.
   05 WS-DATA09  PIC X(09).
   05 FILLER     PIC X(06).
01 WS-DATAL08 REDEFINES WS-DATA.
   05 WS-DATA08  PIC X(08).
   05 FILLER     PIC X(07).
01 WS-DATAL07 REDEFINES WS-DATA.
   05 WS-DATA07  PIC X(07).
   05 FILLER     PIC X(08).
01 WS-DATAL06 REDEFINES WS-DATA.
   05 WS-DATA06  PIC X(06).
   05 FILLER     PIC X(09).
01 WS-DATAL05 REDEFINES WS-DATA.
   05 WS-DATA05  PIC X(05).
   05 FILLER     PIC X(10).
01 WS-DATAL04 REDEFINES WS-DATA.
   05 WS-DATA04  PIC X(04).
   05 FILLER     PIC X(11).
01 WS-DATAL03 REDEFINES WS-DATA.
   05 WS-DATA03  PIC X(03).
   05 FILLER     PIC X(12).

```

```

01 WS-DATAL02 REDEFINES WS-DATA.
   05 WS-DATA02 PIC X(02).
   05 FILLER PIC X(13).
01 WS-DATAL01 REDEFINES WS-DATA.
   05 WS-DATA01 PIC X(01).
   05 FILLER PIC X(14).
*
LINKAGE SECTION.

01 LK-NUMBER-ARGUMENTS PIC S9(9) COMP.

01 LK-FAILURE-CODE PIC S9(9) COMP.

01 LK-MESSAGE-BUFFER.
   05 LK-MESSAGE-BUFFER-BYTE PIC X(1)
      OCCURS 1 TO 255
      DEPENDING ON LK-MESSAGE-BUFFER-LENGTH.
01 LK-MESSAGE-BUFFER-LENGTH PIC S9(9) COMP.

01 LK-TEXT-AREA.
   05 LK-TEXT-AREA-BYTE PIC X(1) OCCURS 15.
01 LK-TEXT-AREA-LENGTH PIC S9(9) COMP.

01 LK-CLASS PIC X.
01 LK-CLASS-LENGTH PIC S9(9) COMP.

```

```

PROCEDURE DIVISION USING
  LK-NUMBER-ARGUMENTS
  LK-FAILURE-CODE
  LK-MESSAGE-BUFFER
  LK-MESSAGE-BUFFER-LENGTH
  LK-TEXT-AREA
  LK-TEXT-AREA-LENGTH
  LK-CLASS
  LK-CLASS-LENGTH
.

```

```

MAIN SECTION.
0100-MAIN.

```

```

MOVE ZERO TO LK-FAILURE-CODE.
MOVE ZERO TO LK-MESSAGE-BUFFER-LENGTH.

```

```

*****
* EXIT FLAGGING AN ERROR IF THE WRONG NUMBER OF ARGUMENT PAIRS
*****
IF LK-NUMBER-ARGUMENTS NOT = 2
  DISPLAY 'UCPGCLSC:NUMBER-ARGUMENTS=' LK-NUMBER-ARGUMENTS
    ' (REQUIRED 2)'
    ' EXITTING WITH RC=401'
  MOVE 401 TO LK-FAILURE-CODE
  MOVE 'UCPGCLSC:NOT ENOUGH ARGUMENTS '
    TO LK-MESSAGE-BUFFER
  GO TO 0900-MAIN-EXIT
END-IF.

```

```

*****
* IF DATA PRESENT FIND LENGTH AND TEST CLASS
*****
*
IF LK-TEXT-AREA-LENGTH = ZERO
  MOVE 'FIELD EMPTY' TO LK-MESSAGE-BUFFER
  MOVE 11 TO LK-MESSAGE-BUFFER-LENGTH
  MOVE 402 TO LK-FAILURE-CODE
  GO TO 0900-MAIN-EXIT
ELSE
  IF LK-TEXT-AREA-LENGTH > +15
    MOVE 'LENGTH > 15' TO LK-MESSAGE-BUFFER
    MOVE 11 TO LK-MESSAGE-BUFFER-LENGTH
  
```

```

        MOVE 403 TO LK-FAILURE-CODE
        GO TO 0900-MAIN-EXIT
    ELSE
        MOVE LK-TEXT-AREA TO WS-DATA
    END-IF.
*
    MOVE SPACES TO LK-CLASS.
    MOVE +1     TO LK-CLASS-LENGTH.
*
0100-CLASS15.
*
    IF LK-TEXT-AREA-LENGTH < +15
        GO TO 0100-CLASS14
    END-IF.
*
    IF WS-DATA ALPHABETIC
        MOVE 'A' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA NUMERIC
        MOVE 'N' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA = LOW-VALUES
        MOVE 'L' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA = HIGH-VALUES
        MOVE 'H' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA = SPACES
        MOVE 'S' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
    GO TO 0900-MAIN-EXIT.
*
0100-CLASS14.
*
    IF LK-TEXT-AREA-LENGTH < +14
        GO TO 0100-CLASS13
    END-IF.
*
    IF WS-DATA14 ALPHABETIC
        MOVE 'A' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA14 NUMERIC
        MOVE 'N' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA14 = LOW-VALUES
        MOVE 'L' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA14 = HIGH-VALUES
        MOVE 'H' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA14 = SPACES
        MOVE 'S' TO LK-CLASS

```

```

        GO TO 0900-MAIN-EXIT
    END-IF.
    GO TO 0900-MAIN-EXIT.
*
0100-CLASS13.
*
    IF LK-TEXT-AREA-LENGTH < +13
        GO TO 0100-CLASS12
    END-IF.
*
    IF WS-DATA13 ALPHABETIC
        MOVE 'A' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA13 NUMERIC
        MOVE 'N' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA13 = LOW-VALUES
        MOVE 'L' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA13 = HIGH-VALUES
        MOVE 'H' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA13 = SPACES
        MOVE 'S' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
    GO TO 0900-MAIN-EXIT.
*
0100-CLASS12.
*
    IF LK-TEXT-AREA-LENGTH < +12
        GO TO 0100-CLASS11
    END-IF.
*
    IF WS-DATA12 ALPHABETIC
        MOVE 'A' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA12 NUMERIC
        MOVE 'N' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA12 = LOW-VALUES
        MOVE 'L' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA12 = HIGH-VALUES
        MOVE 'H' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
*
    IF WS-DATA12 = SPACES
        MOVE 'S' TO LK-CLASS
        GO TO 0900-MAIN-EXIT
    END-IF.
    GO TO 0900-MAIN-EXIT.
*
0100-CLASS11.

```

```

*
IF LK-TEXT-AREA-LENGTH < +11
GO TO 0100-CLASS10
END-IF.
*
IF WS-DATA11 ALPHABETIC
MOVE 'A' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA11 NUMERIC
MOVE 'N' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA11 = LOW-VALUES
MOVE 'L' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA11 = HIGH-VALUES
MOVE 'H' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA11 = SPACES
MOVE 'S' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
GO TO 0900-MAIN-EXIT.
*
0100-CLASS10.
*
IF LK-TEXT-AREA-LENGTH < +10
GO TO 0100-CLASS09
END-IF.
*
IF WS-DATA10 ALPHABETIC
MOVE 'A' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA10 NUMERIC
MOVE 'N' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA10 = LOW-VALUES
MOVE 'L' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA10 = HIGH-VALUES
MOVE 'H' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
*
IF WS-DATA10 = SPACES
MOVE 'S' TO LK-CLASS
GO TO 0900-MAIN-EXIT
END-IF.
GO TO 0900-MAIN-EXIT.
*
0100-CLASS09.
*
IF LK-TEXT-AREA-LENGTH < +9
GO TO 0100-CLASS08
END-IF.
*

```



```

        IF WS-DATA09 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA09 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA09 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA09 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA09 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
*
0100-CLASS08.
*
        IF LK-TEXT-AREA-LENGTH < +8
            GO TO 0100-CLASS07
        END-IF.
*
        IF WS-DATA08 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA08 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA08 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA08 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA08 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
*
0100-CLASS07.
*
        IF LK-TEXT-AREA-LENGTH < +7
            GO TO 0100-CLASS06
        END-IF.
*
        IF WS-DATA07 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*

```

```

        IF WS-DATA07 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA07 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA07 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA07 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
    *
0100-CLASS06.
    *
        IF LK-TEXT-AREA-LENGTH < +6
            GO TO 0100-CLASS05
        END-IF.
    *
        IF WS-DATA06 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA06 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA06 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA06 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA06 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
    *
0100-CLASS05.
    *
        IF LK-TEXT-AREA-LENGTH < +5
            GO TO 0100-CLASS04
        END-IF.
    *
        IF WS-DATA05 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA05 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *

```

```

        IF WS-DATA05 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA05 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA05 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
*
0100-CLASS04.
*
        IF LK-TEXT-AREA-LENGTH < +4
            GO TO 0100-CLASS03
        END-IF.
*
        IF WS-DATA04 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA04 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA04 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA04 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA04 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
*
0100-CLASS03.
*
        IF LK-TEXT-AREA-LENGTH < +3
            GO TO 0100-CLASS02
        END-IF.
*
        IF WS-DATA03 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA03 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*
        IF WS-DATA03 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
*

```

```

        IF WS-DATA03 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA03 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
    *
0100-CLASS02.
    *
        IF LK-TEXT-AREA-LENGTH < +2
            GO TO 0100-CLASS01
        END-IF.
    *
        IF WS-DATA02 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA02 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA02 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA02 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA02 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
    *
0100-CLASS01.
    *
        IF LK-TEXT-AREA-LENGTH < +1
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA01 ALPHABETIC
            MOVE 'A' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA01 NUMERIC
            MOVE 'N' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA01 = LOW-VALUES
            MOVE 'L' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *
        IF WS-DATA01 = HIGH-VALUES
            MOVE 'H' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
    *

```

```

        IF WS-DATA01 = SPACES
            MOVE 'S' TO LK-CLASS
            GO TO 0900-MAIN-EXIT
        END-IF.
        GO TO 0900-MAIN-EXIT.
    *
    *
    0900-MAIN-EXIT.
    GOBACK.

```

3. Compile the UCPGCLSC program and link-edit the program into the PowerExchange Listener LOADLIB library.

Step 3. Add User-Defined Fields

In this step, you add user-defined fields.

The user-defined fields invoke the PowerExchange CallProg function, which calls the user exit program. For more information about the CallProg function, see [“CallProg” on page 283](#).

The user exit program processes data in and returns the class types of the REC_TYPE, BIN_NO, and DECIMAL_NO fields.

1. Open the demo.userexit data map and the MASTER_REC record.
2. In the **Record** window, click the **Expr(0)** tab.
3. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
4. Add the classtype_rec_type field, which is an output field that contains the result from the user exit program when it is invoked for the REC_TYPE field.

The following table describes the properties to define for the field:

Property	Value
Name	classtype_rec_type
Type	CHAR
Precision	0
Scale	0
Length	1

5. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
6. Add the rc_rec_type field, which calls the user exit program to process the REC_TYPE field.

The following table describes the properties to define for the field:

Property	Value	Notes
Name	rc_rec_type	-
Type	NUM32	-
Precision	0	-

Property	Value	Notes
Scale	0	-
Length	0	-
Phase	RW	Indicates that the operation is read or write.
Expression	CallProg('UCPGCLSC', 'UCPGCLSC','COBOL', REC_TYPE,classtype_rec_type)	To enter the expression for the field, complete the following steps: 1. Click in the cell in the Expression column and click the Browse button. The Expression Editor dialog box appears. 2. In the Function List list in the Expression Editor dialog box, double-click the CallProg function. 3. In the Expression List list, enter ('UCPGCLSC','UCPGCLSC','COBOL',REC_TYPE,classtype_rec_type) at the end of the CallProg function name. 4. Click Validate . In the Validate box, the No Errors message appears. 5. Click OK .

7. In the **Record** window, click the **Expr(0)** tab.
8. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
9. Add the classtype_bin_no field, which is an output field that contains the result from the user exit program when it is invoked for the BIN_NO field.

The following table describes the properties to define for the field:

Property	Value
Name	classtype_bin_no
Type	CHAR
Precision	0
Scale	0
Length	1

10. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
11. Add the rc_bin_no field, which calls the user exit program to process the BIN_NO field.

The following table describes the properties to define for the field:

Property	Value	Notes
Name	rc_bin_no	-
Type	NUM32	-
Precision	0	-
Scale	0	-

Property	Value	Notes
Length	0	-
Phase	RW	Indicates that the operation is read or write.
Expression	CallProg('UCPGCLSC', 'UCPGCLSC','COBOL', BIN_NO,classtype_bin_no)	To enter the expression for the field, complete the following steps: 1. Click in the cell in the Expression column and click the Browse button. The Expression Editor dialog box appears. 2. In the Function List list in the Expression Editor dialog box, double-click the CallProg function. 3. In the Expression List list, enter ('UCPGCLSC','UCPGCLSC','COBOL',BIN_NO,classtype_bin_no) at the end of the CallProg function name. 4. Click Validate . In the Validate box, the No Errors message appears. 5. Click OK .

12. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
13. Add the classtype_dec_no field, which is an output field that contains the result from the user exit program when it is invoked for the DECIMAL_NO field.

The following table describes the properties to define for the field:

Property	Value
Name	classtype_dec_no
Type	CHAR
Precision	0
Scale	0
Length	1

14. Right-click anywhere on the **Expr(0)** tab and click **Add Field at End**.
15. Add the rc_decimal_no field, which calls the user exit program to process a copy of the DECIMAL_NO field.

The following table describes the properties to define for the field:

Property	Value	Notes
Name	rc_decimal_no	-
Type	NUM32	-
Precision	0	-
Scale	0	-
Length	0	-

Property	Value	Notes
Phase	RW	Indicates that the operation is read or write.
Expression	CallProg('UCPGCLSC', 'UCPGCLSC','COBOL', DECIMAL_NO,classtype_dec_no)	To enter the expression for the field, complete the following steps: 1. Click in the cell in the Expression column and click the Browse button. The Expression Editor dialog box appears. 2. In the Function List list in the Expression Editor dialog box, double-click the CallProg function. 3. In the Expression List list, enter ('UCPGCLSC','UCPGCLSC','COBOL',DECIMAL_NO,classtype_dec_no) at the end of the CallProg function name. 4. Click Validate . In the Validate box, the No Errors message appears. 5. Click OK .

Step 4. Refresh Columns in the MASTER_REC Table

In this step, you refresh the columns in the MASTER_REC table to pick up the user-defined fields that you added to the MASTER_REC record.

1. Open the demo.userexit data map.
2. On the **Data Map** tab in the **Resource Explorer**, right-click the MASTER_REC table and click **Properties**.

The **Table Properties - Definition** dialog box appears.

3. In the **Column Generation** list, select **Refresh with missing columns**.

Because the record on which the table is based contains new fields, this action adds the corresponding columns to the table.

4. Click **OK**.

For more information about the **Table Properties - Definition** dialog box, see [“Table Properties - Definition” on page 364](#).

Step 5. Test the Results of the User Exit Program

In this step, you run a database row test to test the results of the user exit program in the data map record.

1. Open the demo.userexit data map.
2. On the **Data Map** tab in the **Resource Explorer**, select the MASTER_REC table and click **File > Database Row Test**.
3. When prompted to send the data map to a remote location, click **Yes**.

The **Data Map Remote Node** dialog box appears.

4. In the **Data Map Remote Node** dialog box, enter the user ID, a password or valid PowerExchange passphrase for the user, and the node name of the z/OS system.

The **Database Row Test** dialog box appears.

5. In the **Database Row Test** dialog box, accept the default values and click **Go**.

The **Database Row Test Output** window displays the results for the database row test, including user-defined fields.

The following image shows the **Database Row Test Output** window:

ACCT_CODES_2	ACCT_CODES_3	classtype_bin_no	classtype_dec_no	classtype_rec_type	rc_bin_no	rc_decimal_no	rc_rec_type
Automobile	Life	H		A	0	0	0
Automobile	Life			A	0	0	0
Life	Boat	H		A	0	0	0
Automobile	Life	H		A	0	0	0
Life	Boat			A	0	0	0
Life	Motorcycle		N	A	0	0	0
Boat	Automobile		N	A	0	0	0
Life	House		N	A	0	0	0
Life	Life			A	0	0	0
House	Life	H		A	0	0	0

The following table describes the user-defined fields:

User-defined Field	Value	Description
classtype_bin_no	H	Indicates that the BIN_NO field for that row contains a high value.
classtype_dec_no	N	Indicates that the DECIMAL_NO field for that row contains a numeric zoned decimal value.
classtype_rec_type	A	Indicates that the REC_TYPE field for all rows contains an alphabetic value.
rc_bin_no rc_decimal_no rc_rec_type	0	Indicates that the UCPGCLSC user exit program ran successfully for all rows.

APPENDIX A

PowerExchange Functions for User-Defined Fields

This appendix includes the following topics:

- [PowerExchange Functions for User-Defined Fields Overview, 282](#)
- [PowerExchange Functions, 283](#)
- [Calling User-Defined Programs with the CallProg Function, 307](#)
- [Testing User-Defined Fields in a Data Map, 316](#)

PowerExchange Functions for User-Defined Fields Overview

In data maps, you can define fields in records that use one or more PowerExchange functions in an expression to process source data.

These functions provide operations such as field splitting and concatenation. You can also use the CallProg function in an expression to invoke a user-defined program to process source data.

PowerExchange provides example user-defined programs with headers files and JCL. These files, named `ucpe*.*`, are located in the examples directory in the PowerExchange Navigator installation directory. Edit and compile these example programs to customize them for your environment.

After you define a user-defined field, run a database row test to test any PowerExchange and user-defined functions defined for the field.

If you plan to generate before-image and change-indicator columns for user-defined fields, ensure that the functions associated with the fields support before-image buffering. If the functions do not support before-image buffering, the before-image and change-indicator are not generated.

RELATED TOPICS:

- [“Testing User-Defined Fields in a Data Map” on page 316](#)

PowerExchange Functions

You can include the following PowerExchange functions in expressions in user-defined fields to process the source data in a record in a data map.

CallProg

Calls a user-defined program or subroutine to process the source data in a record.

Syntax:

```
[result=]CallProg('program','subroutine','linkage'[,arg1][,arg2][,...])
```

The parameters are:

- *result*. Optional. NUM32. This argument contains the return value from program called by the CallProg function, which is one of the following values:
 - **0**. Success.
 - **Non-zero**. Failure.
If you do not specify a result argument and a non-zero return code is returned from the external program, CallProg executes the following default map-level error responses:
 - Ends the extract.
 - Skips this subroutine.
- *program*. The name of the program that contains the subroutine. Depending on the operating system, the program is one of the following:
 - **i5/OS**. A service program.
 - **Linux or UNIX**. A shared object.
 - **Windows**. A DLL.
 - **z/OS**. A load module.
Enclose the program name in single quotes.
- *subroutine*. The name of the entry point in the program. Depending on the operating system, the subroutine is one of the following:
 - **i5/OS**. The subroutine name.
 - **Linux, UNIX, or Windows**. The subroutine name.
 - **z/OS for Assembler, C, or COBOL programs**. You must provide a value, but the value is ignored and the default entry point for the load module is used. Specify the same name as the program.
 - **z/OS for PL/I programs**. If multiple fetchable subroutines reside in the same load module, specify the subroutine name.
Enclose the subroutine name in single quotes.
- *linkage*. The type of linkage, which determines the way that arguments are passed to and return codes are returned from the program or subroutine.

The following table describes each linkage type:

Linkage Type	Supported Operating Systems	Arguments	Returns
C	- i5/OS - Linux, UNIX, and Windows - z/OS	Passed through the stack	Program return code
COBOL	z/OS	Passed as a list of addresses	Address of failure code integer
OS	z/OS	Passed as a list of addresses	Program return code
OS400	i5/OS	Passed as a list of addresses	Address of failure code integer
PLI	z/OS	Passed as a list of addresses	Address of failure code integer

Enclose the linkage type in single quotes.

- `[arg1][,arg2][,...]`. One or more optional arguments passed to the program or subroutine.

RELATED TOPICS:

- ["Calling User-Defined Programs with the CallProg Function" on page 307](#)

Check

Enables you to specify which return codes indicate success or failure from programs called by the CallProg function.

Compares the return code from a program called by the CallProg function against one or more specified return codes. If a match is found, writes a user-defined message to the PowerExchange message log.

Syntax:

```
Check(CallProgRC, 'message', "comparison", return_code1, return_code2, return_code3)
```

The following table describes the parameters:

Parameter	Description	Datatype
CallProgRC	The return code from a program called by the CallProg function. This name must match the name specified for the result argument in the CallProg function. The Check function compares this return code against the specified return code values.	NUM32
<i>message</i>	A user-defined message that is written to the PowerExchange message log when the return code either matches or does not match, depending on the type of comparison being performed, one of the specified return codes in the list. Enclose the message text in single quotes.	-

Parameter	Description	Datatype
<i>comparison</i>	The type of comparison to be performed. The options are: <ul style="list-style-type: none"> - EQ. A failure occurs if the return code matches one of the specified return codes in the list. - NE. A failure occurs if the return code does not match one of the specified return codes in the list. Enclose the comparison value in quotes.	-
return_code1 return_code2 return_code3	A list of one or more return code values. The Check function compares the return code from the program called by the CallProg function against this list of return code values. You must specify at least one return code value.	NUM32

Check Example

If a return code of 0, 3, or 4 is an acceptable return code from the Program2 subroutine of the Program1 program, then you might specify the following functions:

```
CallProgRC=CallProg('Program1','Program2','COBOL','field1','field2')
Check(CallProgRC,'Error in Program2','NE',0,3,4)
```

If the return code from the Program2 subroutine is 1, processing fails and the following message is written to the log:

```
Error in Program2
```

If the return code from the Program2 subroutine is 0, processing continues and the specified message is not written to the log.

CheckNumData

Enables a calling program to check whether a field contains valid numeric characters for field types CHAR, NUMCHAR, ZONED, and PACKED. If the program finds invalid data, it takes additional action.

CheckNumData performs a similar function to the data checking options in the data map properties. CheckNumData applies data checking to a single field, rather than to the entire data map. You can use CheckNumData to validate numeric data fields before conversion processing by a function such as the CopyData function.

Syntax:

```
CheckNumData(field_name,field_type,action)
```

The following table describes the parameters:

Parameter	Description
<i>field_name</i>	The name of the field to check for valid numeric data.
<i>field_type</i>	The datatype of the field to be checked by CheckNumData. The following types are valid: <ul style="list-style-type: none"> - CHAR - NUMCHAR - ZONED - PACKED
<i>action</i>	The action to take if invalid data is found in the field for the specified datatype. Options are: <ul style="list-style-type: none"> - REPLACE. Replaces the field value with a specific value that is valid. - SETNULL. Replaces the field value with a null character that is valid for the target. - SKIP. Skips the field in the target mapping if the field contains invalid data. - TERMINATE. Ends the mapping process if invalid data is found.

Concat

Concatenates two or more fields. You might use this function to create a group field from multiple fields.

This function does not complete any conversions on the data in the fields. The number of bytes to copy is determined from the length of the fields.

Note: You can only concatenate fields of similar data types. For example, you cannot concatenate a field that has a character data type with a field that has a numeric data type.

Syntax:

```
result=Concat(field1,field2[,...])
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Required. User-defined expression field populated with the concatenated input fields. Must be large enough to contain the concatenated fields.	Any
<i>field1</i>	Field to be concatenated to one or more fields.	A real or user-defined field of any data type, or a value enclosed in single quotes
<i>field2[,...]</i>	One or more fields to be concatenated to the first field.	A real or user-defined field of any data type, or a value enclosed in single quotes

Concat Example

If the field, Field1, contains the following characters:

```
The quick brown fox
```

And the field, Field2, contains the following characters:

```
jumped over the lazy dog
```

The function call, Concat(Field1,Field2), returns the following characters:

```
The quick brown fox jumped over the lazy dog
```

CopyData

Copies data from one field to another field. You can copy a source field to a target field or to the user-defined field that is specified for the expression that contains this function.

Syntax:

To copy data to a specific target field:

```
CopyData(source_field, target_field)
```

To copy data to the user-defined field that is specified for the expression:

```
CopyData(source_field)
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>source_field</i>	The field to be copied. This field can be a real field or user-defined field of any datatype, or the data value that you want to copy enclosed in single quotation marks, such as '123456'.	Any datatype
<i>target_field</i>	The target field to which the source field data is copied. The target field length must be equal to or greater than the source field length. Otherwise, a conversion truncation error occurs. Note: If you specify a target field in the CopyData expression, do not also define a field name and attributes on the Expr(n) tab.	Any datatype

Usage Notes:

- If you want to copy data to a specific target field, on the **Expr(n)** tab of the **Record** window, leave the **Name** value and field attributes in the expression row that includes this function blank. Otherwise, CopyData fails with RC=2660.
- If you want to copy data to the user-defined field that is associated with the expression, on the **Expr(n)** tab of the **Record** window, enter the **Name** value and field attributes in the expression row that includes this function. Ensure that the field length is long enough to contain the copied data. Use any appropriate datatype. The function uses this datatype instead of the source field datatype for the user-defined field. If you have a nonrelational source field that is defined with multiple datatypes, you can use expressions to write the data to separate user-defined fields that each have a different datatype.

Fragment

Splits and distributes the subfields of a source field among two or more specified target fields.

The amount of data that is distributed to each target field is determined by the length of the target fields. For example, if the first target field is a CHAR field with a length of two characters, the first two characters of the source field is copied to the target field. If the second target field is a CHAR field with a length of eight characters, the next eight characters of the source field is copied to that target field.

Syntax:

```
[result=]Fragment(source_field, target_field1, target_field2[, ...])
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Optional. Returns one of the following values: <ul style="list-style-type: none"> - 0. The source field was fragmented successfully to the target field or fields. - 1. All data in the source field was not distributed to target fields. - 2. The last target field is of a variable datatype and was not populated with data to its maximum length. - 3. Some target fields were not populated with data and were set to null. 	-
<i>source_field</i>	Field to be fragmented.	A real or user-defined field of any datatype, or a value enclosed in single quotes
<i>target_field1</i>	Target field into which to copy fragment from the source field.	A user-defined field of any datatype except a variable datatype
<i>target_field2</i> [...]	One or more additional target fields into which to copy fragments from the source field.	A user-defined field of any datatype except a variable datatype. The last target field can be of a variable datatype. If the last target field is not variable and its length exceeds the length of the remaining data in the source field, the target field is populated with null and the return code is 0. However, if you make the last target field variable, it is populated with data, and the return code is still 0, even if full length of the variable field is not populated.

Fragment Example

If you define the following fields:

```
SourceField CHAR (44)
TargetField1 CHAR (16)
TargetField2 CHAR (28)
```

The field, SourceField, contains the following characters:

```
The quick brown fox jumped over the lazy dog
```

The function call, `Fragment(SourceField,TargetField1,TargetField2)`, distributes fragments of the source field to the TargetField1 and Targetfield2 fields.

The TargetField1 field contains:

```
The quick brown
```

The TargetField2 field contains:

```
fox jumped over the lazy dog
```


GenVRowKey

Generates multiple table rows identified by unique row numbers from a row created from a source row that contains an array.

This function enables you to differentiate among multiple target rows created by a single source row. For example, if a source file contains an array and a child table is created from the array, use this function to populate a target field with unique multiple table rows for rows that would otherwise be identical.

Note: For data rows with an OCCURS clause, a single row is returned for each instance of the OCCURS.

Syntax:

```
result=GenVRowKey()
```

Returns:

NUM32U. Multiple unique table rows for a row created from a source row that contains an array.

GenVRowKey Example

Define a data map that contains a record with an array defined by a COBOL OCCURS clause, as follows:

```
01 RECORD.  
  03 FIELD1 PIC X(4).  
  03 FIELD2 PIC X(4) OCCURS 3 TIMES.
```

The data map includes two source rows that contain the following data:

```
AAAA111122221111  
BBBB555566667777
```

In this example, two rows contain the following identical data:

```
AAAA 1111
```

The GetVRowKey() function generates the following unique table rows:

```
AAAA 1111 1  
AAAA 2222 2  
AAAA 1111 3  
BBBB 5555 1  
BBBB 6666 2  
BBBB 7777 3
```

GetCurrentFileName

For a data map record defined for a nonrelational data source, gets the name of the source data file.

Use the GetCurrentFileName function to determine the data file from which data for a record is read. You might use this information to complete the following tasks:

- For a data map that is associated with a single data file, verify that all records from the data file were processed.
- For a data map that is associated with a file-list file, verify that all records from all data files listed in the file-list file were processed.
- Trace a row back to its original data file.

This function runs on all operating systems and runs in read mode.

Syntax:

```
result=GetCurrentFileName()
```

The following table describes the parameters:

Result	Description	Datatype
<i>result</i>	Returns the name of the the data file from which data for a record is read. If the data map is associated with a file-list file, records might be read from multiple data files.	VARCHAR. Maximum length is 512 bytes plus a null termination character.

Usage Notes:

- On i5/OS, the GetCurrentFileName function does not attempt to return the actual member name when a file is comprised of multiple members and *FIRST and *LAST are specified in the file name. The GetCurrentFileName function returns the file name that was passed to the open function.
- On Linux, UNIX, and Windows, the GetCurrentFileName function does not attempt to get the full file name if it is qualified by relative directory paths, such as: ../../myfile.dat. Instead, the GetCurrentFileName function returns the following value: ../../myfile.dat.
- In the following situations on z/OS, the GetCurrentFileName function might return file names that are more precise than the names listed in a file-list file:
 - Generation data group (GDG) file name in a data map or file-list file
 - DD name in a data map or file-list file

RELATED TOPICS:

- [“Data Files and File-List Processing” on page 52](#)

GetCurrentFileName Examples

The following examples demonstrate how to use the GetCurrentFileName function to return the data file or files associated with a data map.

Example 1

If a data map is associated with the demo3.dat data file that is located in the c:/Informatica/PowerExchange/examples directory, the GetCurrentFileName function returns the following value for all records in the data map:

```
c:/Informatica/PowerExchange/examples/demo3.dat
```

Example 2

If a data map is associated with a file-list file that lists multiple data files, the GetCurrentFileName function returns the name of the data file from which each record was read. For example, if a data map is associated with a file-list file that lists the following data files that each contain 10 records:

```
c:/Informatica/PowerExchange/examples/datafile1.dat
c:/Informatica/PowerExchange/examples/datafile2.dat
```

The GetCurrentFileName function returns the following value for records 1 through 10:

```
c:/Informatica/PowerExchange/examples/datafile1.dat
```

The GetCurrentFileName function returns the following value for records 11 through 20:

```
c:/Informatica/PowerExchange/examples/datafile2.dat
```

Example 3

If the file-list file lists the following data files that use a relative path and each data file contains 10 records:

```
../datafile1.dat  
../datafile2.dat
```

The `GetCurrentFileName` function returns the following value for records 1 through 10:

```
../datafile1.dat
```

The `GetCurrentFileName` function returns the following value for records 11 through 20:

```
../datafile2.dat
```

GetDatabaseKey

Gets the database key for a record or a segment.

Alternatively, you can return the extended relative byte address (XRBA) for a VSAM ESDS data set or the relative record number (RRN) for a VSAM RRDS data set by using the following methods:

- To get the XRBA for ESDS data sets, select the **Prefix record with XRBA** option on the **ESDS Access Method** tab in the **Data Map Properties** dialog box for the ESDS data map.
- To get the RRN for RRDS data sets, select the **Prefix Record with RRN** option on the **RRDS Access Method** tab in the **Data Map Properties** dialog box for the RSDS data map.

Syntax:

```
GetDatabaseKey()
```

Or, use the following alias:

```
GetDbKey()
```

Returns:

The following table describes the objects that are returned, based on the source type of the data map:

Access Method	Data Source	Database Key Object Returned	Datatype
ADABAS	Adabas	Internal sequence number (ISN)	NUM32
ESDS	VSAM entry-sequenced data set (ESDS) on z/OS	Extended relative byte address (XRBA)	BIN 8
IDMS	IDMS	Database key	NUM32
DL/1 BATCH IMS ODBA	IMS	Relative byte address (RBA)	BIN 8
SEQ	i5/OS files that are accessed sequentially	Relative record number (RRN)	NUM32
RRDS	VSAM relative record data set (RRDS) on z/OS	Relative record number (RRN)	NUM32

Note: Although an IMS RBA is 4 bytes long, the `GetDatabaseKey` function returns an 8-byte binary representation of the RBA value for the requested segment. Eight bytes are required to allow for future expansion and to guarantee uniqueness for FastPath and HALDB databases.

Usage Notes:

- To add the segment RBA to segment data in an IMS data map, Informatica recommends that you use the GetIMSRBByLevel function rather than the GetDatabaseKey function. The GetIMSRBByLevel function enables you to get the RBA of an unkeyed or non-unique keyed parent segment.

Use the GetIMSRBByLevel function in IMS data maps used for bulk data movement operations or IMS synchronous CDC.

You cannot use the GetIMSRBByLevel function in records in the following types of IMS data maps:

- IMS data maps used for IMS log-based CDC.
- IMS data maps used to access IMS unload files that contain the following types of records:
 - Unload records of any format for HISAM data sets
 - Unload records that have the format BMC SHORT or BMC XSHORT for HDAM and HIDAM data sets
 - Unload records that have the standard IBM format for IMS Fast Path data sets
- This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

RELATED TOPICS:

- [“Getting the RRN or RBA for VSAM Data Set Records” on page 131](#)
- [“GetIMSRBByLevel” on page 298](#)

GetDataFlowType

IDMS. Returns a two-character indicator that indicates whether the data comes from a bulk data movement operation, a log capture data extraction, or a real-time capture data extraction, and whether data is or is not attached.

Used in conjunction with the GetDbKeyOfOwner function, generates a relational view of IDMS data. You might use these functions to materialize a relational database by using a PowerExchange bulk data movement operation. Then, you can maintain the data by using PowerExchange CDC processing.

The GetDataFlowType function works with either:

- A single-record capture registration
- The base record, or lowest level, of a multiple-record capture registration

The GetDbKeyOfOwner function returns the database key and sets the page group and radix elements to zero.

Syntax:

```
result=GetDataFlowType()
```

The *result* parameter returns a two-character CHAR value that specifies the following information:

- The first character specifies one of the following values:
 - **B.** Bulk data extraction.
 - **L.** Log capture data extraction.
 - **R.** Real-time capture data extraction.
- The second character specifies one of the following values:
 - **D.** Normal data record.
 - **C.** Connect without data. Relevant for CDC processing. The data portion of the record contains null.

- U. Disconnect without data. Relevant for CDC processing. The data portion of the record contains null. If no owner exists, the GetDbKeyOfOwner function returns high values.

GetDbKey

Alias for GetDatabaseKey.

RELATED TOPICS:

- [“GetDatabaseKey” on page 291](#)

GetDbKeyOfFirstMember

IDMS. Returns the database key of the first member of the specified set. You can use this function with the record that is the owner of the specified set.

Because this function does not return the IDMS page group or radix, use this function when duplicate databases that are accessed using different page groups do not exist. If duplicate databases might exist, use the GetFullDbKeyOfFirstMember function.

Syntax:

```
GetDbKeyOfFirstMember('IDMS_set_name')
```

The function returns a value of datatype NUM32U or BIN4.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetDbKeyOfLastMember

IDMS. Returns the database key of the last member of the specified set. You can use this function with the record that is the owner of the specified set.

Because this function does not return the IDMS page group or radix, use this function when duplicate databases that are accessed using different page groups do not exist. If duplicate databases might exist, use the GetFullDbKeyOfLastMember function.

Syntax:

```
GetDbKeyOfLastMember('IDMS_set_name')
```

The function returns a value of datatype NUM32U or BIN4.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetDbKeyOfNextMember

IDMS. Returns the database key of the next member record of the specified set. You can use this function with a record that is a member of the specified set.

Because this function does not return the IDMS page group or radix, use this function when duplicate databases that are accessed using different page groups do not exist. If duplicate databases might exist, use the `GetFullDbKeyOfNextMember` function.

Syntax:

```
GetDbKeyOfNextMember('IDMS_set_name')
```

The function returns a value of datatype NUM32U or BIN4.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetDbKeyOfOwner

IDMS. Gets the database key of the owner of an IDMS set name. Enables you to identify the owner record without physically reading the record.

Because this function does not return the IDMS page group or radix, use this function when duplicate databases that are accessed using different page groups do not exist. If duplicate databases might exist, use the `GetFullDbKeyOfOwner` function.

Syntax:

```
GetDbKeyOfOwner('IDMS_set_name')
```

The function returns a value of datatype NUM32U or BIN4.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

RELATED TOPICS:

- [“GetFullDbKeyOfOwner” on page 297](#)

GetDbKeyOfPriorMember

IDMS. Returns the database key of the previous member record of the specified set. You can use this function with a record that is a member of the specified set.

Because this function does not return the IDMS page group or radix, use this function when duplicate databases that are accessed using different page groups do not exist. If duplicate databases might exist, use the `GetFullDbKeyOfPriorMember` function.

Syntax:

```
GetDbKeyOfPriorMember('IDMS_set_name')
```

The function returns a value of datatype NUM32U or BIN4.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetFullDbKey

IDMS. Returns the fully qualified database key of an IDMS record, which includes the four-byte concatenated page group and radix identifier prefix.

Use this function if duplicate database keys exist. If duplicate database keys do not exist, use the `GetDbKeyOfOwner` function.

Syntax:

```
GetFullDbKey()
```

The function returns a value of datatype BIN8.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetFullDbKeyOfFirstMember

IDMS. Returns the fully qualified database key of the first member of the specified set. The fully qualified database key includes the four-byte concatenated page group and radix identifier prefix. You can use this function with the record that is the owner of the specified set.

Use this function if duplicate databases exist that are accessed using different page groups. If duplicate databases do not exist, use the GetDbKeyOfFirstMember function.

Syntax:

```
GetFullDbKeyOfFirstMember('IDMS_set_name')
```

The function returns a value of datatype BIN8.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Note: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetFullDbKeyOfLastMember

IDMS. Returns the fully qualified database key of the last member of the specified set. The fully qualified database key includes the four-byte concatenated page group and radix identifier prefix. You can use this function with the record that is the owner of the specified set.

Use this function if duplicate databases exist that are accessed using different page groups. If duplicate databases do not exist, use the GetDbKeyOfLastMember function.

Syntax:

```
GetFullDbKeyOfLastMember('IDMS_set_name')
```

The function returns a value of datatype BIN8.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetFullDbKeyOfNextMember

IDMS. Returns the fully qualified database key of the next member of the specified set. The fully qualified database key includes the four-byte concatenated page group and radix identifier prefix. You can use this function with a record that is a member of the specified set.

Use this function if duplicate databases exist that are accessed using different page groups. If duplicate databases do not exist, use the GetDbKeyOfNextMember function.

Syntax:

```
GetFullDbKeyOfNextMember('IDMS_set_name')
```

The function returns a value of datatype BIN8.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetFullDbKeyOfOwner

IDMS. Gets the fully qualified database key of the owner of an IDMS owner record, which includes the four-byte concatenated page group and radix identifier prefix.

Use this function if duplicate databases exist that are accessed using different page groups. If duplicate databases do not exist, use the GetDbKeyOfOwner function.

Syntax:

```
GetFullDbKeyOfOwner('IDMS_set_name')
```

The function returns a value of datatype BIN8.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes. Note: Do not use an IDMS set name for a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

RELATED TOPICS:

- [“GetDbKeyOfOwner” on page 294](#)

GetFullDbKeyOfPriorMember

IDMS. Returns the fully qualified database key of the previous member of the specified set. The fully qualified database key includes the four-byte concatenated page group and radix identifier prefix. You can use this function with a record that is a member of the specified set.

Use this function if duplicate databases exist that are accessed using different page groups. If duplicate databases do not exist, use the GetDbKeyOfPriorMember function.

Syntax:

```
GetFullDbKeyOfPriorMember('IDMS_set_name')
```

The function returns a value of datatype BIN8.

The following table describes the parameter:

Parameter	Description
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotation marks. Note: Do not use an IDMS set name that represents a system index.

Usage Notes: This function supports before-image buffering, which is required for adding before-image and change-indicator columns for the user-defined expression fields.

GetIMSRBByLevel

For data map records defined for IMS data sources, gets the RBA value of an IMS segment at the specified level of the path of the current segment.

To add the segment RBA to segment data in an IMS data map, Informatica recommends that you use the GetIMSRBByLevel function rather than the GetDatabaseKey function. The GetIMSRBByLevel function enables you to get the RBA of an unkeyed or non-unique keyed parent segment.

Use the GetIMSRBByLevel function in IMS data maps used for bulk data movement operations or IMS synchronous CDC.

You cannot use the GetIMSRBByLevel function in records in the following types of IMS data maps:

- IMS data maps used for IMS log-based CDC.
- IMS data maps used to access IMS unload files that contain the following types of records:
 - Unload records of any format for HISAM data sets
 - Unload records that have the format BMC SHORT or BMC XSHORT for HDAM and HIDAM data sets
 - Unload records that have the standard IBM format for IMS Fast Path data sets

To run a database row test on the following types of data maps that use the GetIMSRBByLevel function, you must APF-authorize the following libraries:

- If you use IMS DL/1 batch data maps with BMP for the netport job, you must APF-authorize all libraries in the BMP STEPLIB concatenation. Otherwise, PowerExchange issues message PWX-02204 with return code 12.

- If you use IMS ODBA data maps, you must APF-authorize all libraries in the STEPLIB concatenation of the PowerExchange Listener. Otherwise, PowerExchange abends.

Syntax:

```
result=GetIMSRBByLevel([integer])
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the 8-byte binary value of the RBA of the requested segment.	NUM64, NUM64U, or BIN with a length of 8
<i>integer</i>	Optional. The type of request, which is one of the following: <ul style="list-style-type: none"> - No value. The RBA of the current segment is returned. - Positive integer. The 4-byte integer level number of the segment in the IMS hierarchy for which to get the RBA. The level number is limited to the level number of the current segment in the hierarchy or any ancestors of the segment. For example, for a segment at level 3 of a 5-level hierarchy, you can request the RBA of the segment at level 1, 2, or 3. If you request the RBA for segments in the hierarchy below the level of the current segment, a run-time error occurs. - Negative integer. Indicates a relative request. For example, you can request the RBA of the segment two levels above the level of the current segment by specifying -2. If you specify a negative integer with an absolute value that is greater than or equal to the level number of the current segment, a run time error occurs. 	NUM64

Note: Although an IMS RBA is four bytes long, the GetIMSRBByLevel function returns an 8-byte binary representation of the RBA value for the requested segment. Eight bytes are required to allow for future expansion and to guarantee uniqueness for FastPath and HALDB databases.

RELATED TOPICS:

- [“GetDatabaseKey” on page 291](#)

GetPageGroup

IDMS. Gets the page group in which an IDMS record exists.

Syntax:

```
result=GetPageGroup()
```

Returns:

NUM16 or NUM16U, or BIN with a length of 2. The page group.

GetPageGroupOfOwner

IDMS. Gets the page group of an IDMS owner record.

Syntax:

```
result=GetPageGroupOfOwner('IDMS_set_name')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the page group of the IDMS owner record.	NUM16 or NUM16U, or BIN with a length of 2
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes.	-

GetPgGrpAndRdx

IDMS. Gets the page group and radix of an IDMS record.

Syntax:

```
result=GetPgGrpAndRdx()
```

Returns:

NUM32 or NUM32U, or BIN with a length of 4. The concatenated page group and radix of the IDMS record.

GetPgGrpAndRdxOfOwner

IDMS. Gets the page group and radix of an IDMS owner record.

Syntax:

```
result=GetPgGrpAndRdxOfOwner('IDMS_set_name')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the concatenated page group and radix of the IDMS owner record.	NUM32 or NUM32U, or BIN with a length of 4
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes.	-

GetPgGrpOfOwner

IDMS. Gets the page group of an IDMS owner record.

Syntax:

```
result=GetPgGrpOfOwner('IDMS_set_name')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	The page group of the IDMS owner record.	NUM16 or NUM16U, or BIN with a length of 2
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes.	-

GetRadix

IDMS. Gets the radix of an IDMS record.

Syntax:

```
result=GetRadix()
```

Returns:

NUM16 or NUM16U, or BIN with a length of 2. The radix of the IDMS record.

GetRadixOfOwner

IDMS. Gets the radix of an IDMS owner record.

Syntax:

```
result=GetRadixOfOwner('IDMS_set_name')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	The radix of the IDMS owner record.	NUM16 or NUM16U, or BIN with a length of 2
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes.	-

GetSeqWithinLevel

Sequential flat files, IMS, and IDMS. Gets the sequence number of a record in the current hierarchical level.

Syntax:

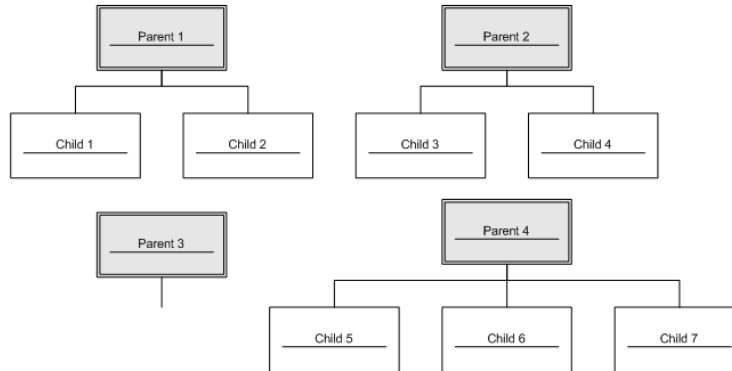
```
result=GetSeqWithinLevel()
```

Returns:

NUM32. The record sequence number.

GetSeqWithinLevel Example

The following figure shows an example hierarchy:



For this hierarchy, the GetSeqWithinLevel function returns the following record sequence numbers:

1, 2, 3, 4, 0, 5, 6, 7

The returned record sequence numbers reflect the following records:

- Children 1 and 2 from parent 1
- Children 3 and 4 from parent 2
- No children, 0, from parent 3
- Children 5, 6, and 7 from parent 4

GetSeqWithinParent

IMS and IDMS. Gets the record sequence number relative to the current parent record instance.

Syntax:

```
result=GetSeqWithinParent()
```

Returns:

NUM32. The record sequence number.

GetSeqWithinParent Example

For the hierarchy shown in GetSeqWithinLevel, the GetSeqWithinParent function returns the following record sequence numbers:

1, 2, 1, 2, 0, 1, 2, 3

The returned record sequence numbers reflect the following records:

- Children 1 and 2 from parent 1
- Children 1 and 2 from parent 2
- No children (0) from parent 3
- Children 1, 2, and 3 from parent 4

RELATED TOPICS:

- [“GetSeqWithinLevel” on page 301](#)

LengthOf

Gets the length of a field.

Syntax:

```
result=LengthOf(field)
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the length of the field.	NUM32
<i>field</i>	The field for which to return the length.	Any

LTrim

Trims a specified character from the left side of a field.

Note: The field to be trimmed must be a similar data type as the target field. For example, you cannot use the LTrim function to copy a field with a character data type to a target field with a numeric data type.

Syntax:

```
result=LTrim(field,'character')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the contents of the field trimmed of the specified character from the left side.	CHAR
<i>field</i>	The field to be trimmed.	CHAR
<i>character</i>	Optional. The character to trim from the left side of the field. If a character is not specified, space is the default. Enclose the character value in single quotes.	-

LTrim Example

If the field, Text, contains the following characters:

```
*****The quick brown fox*****
```

The function LTrim(Text,*) returns the following characters:

```
The quick brown fox*****
```

RTrim

Trims a specified character from the right side of a field.

Note: The field to be trimmed must be a similar data type as the target field. For example, you cannot use the RTrim function to copy a field with a character data type to a target field with a numeric data type.

Syntax:

```
RTrim(field, 'character')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the contents of the field trimmed of the specified character from the right side.	CHAR
<i>field</i>	The field to be trimmed.	CHAR
<i>character</i>	Optional. The character to trim from the right side of the field. If a character is not specified, space is the default. Enclose the character value in single quotes.	-

RTrim Example

If the field, Text, contains the following characters:

```
*****The quick brown fox*****
```

The function RTrim(Text,*) returns the following characters:

```
*****The quick brown fox
```

SetBitA

Returns a character value to indicate the on or off setting of a bit at a specified offset in a field.

Syntax:

```
result=SetBitA(field, offset)
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns one of the following values to indicate the on or off setting of the specified bit: - Y. On. - N. Off.	CHAR
<i>field</i>	The field to be checked.	Any
<i>offset</i>	The position of the bit within the field. For a 1-byte field, the offset begins at bit 0. For a 2-byte field, the offset begins at 8 in the second byte.	-

SetBitN

Returns a numeric value to indicate the on or off setting of a bit at a specified offset in a field.

Syntax:

```
result=SetBitN(field,offset[,on_value][,off_value])
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns one of the following values to indicate the on or off setting of the specified bit: - 1. On. - 0. Off.	NUM8
<i>field</i>	The field to be checked.	Any
<i>offset</i>	The position of the bit within the field. Offsets begin at 0 and end at the last bit of a field. For example, if you have a 4-byte CHAR field, the last offset is 30.	-
<i>on_value</i>	Optional. Overrides the value returned if the bit value is set to on. For example, if you set the on value to 3 and the bit value is set to on, SetBitN returns 3. Default is 1.	NUM8
<i>off_value</i>	Optional. Overrides the value returned if the bit value is set to off. For example, if you set the off value to 2 and the bit value is set to off, SetBitN returns 2. Default is 0.	NUM8

Split

Gets the contents of a field from a specified start point for a specified length. You might use this function to get a single field from a group field.

Note: The field to be split must be a similar data type as the target field. For example, you cannot use the Split function to copy a field with a character data type to a target field with a numeric data type.

To create a subfield with a datatype of other than CHAR from a CHAR source field, use the Split function in conjunction with the CopyData function, as follows:

1. Use the Split function on the source field to create a new CHAR field.
2. Use the CopyData function on the new CHAR field to create a target field with a different datatype, such as a numeric datatype.

Syntax:

```
result=Split(field,start_byte,length)
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the contents of the field from the specified start byte for the specified length.	Any type
<i>field</i>	Field from which to get a subfield.	Any type

Parameter	Description	Datatype
<i>start_byte</i>	The number of the byte where the subfield begins. Offsets begin at 1.	NUM32
<i>length</i>	The number of bytes to get.	NUM32

RELATED TOPICS:

- [“CopyData” on page 287](#)

Split Example

If the field, Text, contains the following string:

```
The quick brown fox jumped over the lazy dog
```

The function Split(Text,17,10) returns the following characters:

```
fox jumped
```

The function Split(Text,1,23) returns the following characters:

```
The quick brown fox
```

Strip

Strips a specified character from the left side, the right side, or both sides of a field.

Syntax:

```
Strip(field,['side'],['character'])
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>field</i>	The field to be stripped.	CHAR
<i>side</i>	Optional. The side of the field to be stripped of the specified character, which is one of the following values: <ul style="list-style-type: none"> - L. Left side. - R. Right side. - B. Both sides. If a side is not specified, B is the default. Enclose the side value in single quotes.	-
<i>character</i>	Optional. The character to remove from the field from the specified side or sides. If a character is not specified, space is the default. Enclose the character value in single quotes.	-

Returns:

CHAR. Contents of the field stripped of the specified character on the specified side or sides.

Strip Example

If the field, Text, contains the following characters:

```
*****The quick brown fox*****
```

The function Strip(Text,'R','*') returns the following characters:

```
*****The quick brown fox
```

The function Strip(Text,'L','*') returns the following characters:

```
The quick brown fox*****
```

The function Strip(Text,'B','*') returns the following characters:

```
The quick brown fox
```

ToLower

Converts all characters in a field to lowercase.

Syntax:

```
ToLower(field)
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>field</i>	The field to convert to lowercase.	CHAR or VARCHAR

Returns:

CHAR. Contents of the field in lowercase.

ToUpper

Converts all characters in a field to uppercase.

Syntax:

```
ToUpper(field)
```

Parameter	Description	Datatype
<i>field</i>	The field to convert to uppercase.	CHAR or VARCHAR

Returns:

CHAR. Contents of the field in uppercase.

Calling User-Defined Programs with the CallProg Function

You can use the CallProg function to call a user-defined program to process source data in a record.

Location of User-Defined Programs

Depending on the operating system, the PowerExchange Listener loads the user-defined program from one of the following locations:

- On Linux, UNIX, or Windows, the PowerExchange Listener loads the program from the first directory in the PATH that contains the named dll.
 - On z/OS, the PowerExchange Listener loads the program from one of the following locations:
 - If the DBMOVER configuration file includes the PC_AUTH=Y statement, the PowerExchange Listener loads the user-defined program from the DTLLOAD DD statement.
 - If the DBMOVER configuration file does not include the PC_AUTH=Y statement, the PowerExchange Listener loads the user-defined program from the JOBLIB/STEPLIB in the JCL for the PowerExchange Listener or from LINKLIST.
- Note:** Use PC_AUTH=Y and DTLLOAD DD if the user-defined program resides in a library that is not APF-authorized.
- On i5/OS, the PowerExchange Listener loads the program from the active library list for the PowerExchange Listener job.

Load Errors for User-Defined Programs Called by the CallProg Function

When you call a user-defined program with the CallProg function, the function address for the entry point of the program is loaded on the first call. The load might fail for one of the following reasons:

- The CallProg function did not find the program in the standard order of search for programs on the operating system.
- The following table lists the program types for which the CallProg function searches:

Program Type	Operating System
DLL	Windows
Load module	z/OS
Service program	i5/OS
Shared object	Linux and UNIX

- On some operating systems, the names of programs are case-sensitive and the name was specified incorrectly.
- The CallProg function found the module handle but did not find the routine in the module.

The following table describes possible causes of the error:

Problem	Linkage Type	Operating System
The symbol for the routine was not exported.	C OS400	i5/OS
The routine was not defined with a #pragma fetchable.	C PLI	z/OS
The routine was not exported.	C	Windows

Parameter List Passed to User-Defined Programs

When you call a user-defined program with the CallProg function, the CallProg function passes information to the user-defined program in a parameter list.

The user-defined program can set parameters in the parameter list to return information to the CallProg function.

The following table describes the input and output parameters in the parameter list:

Parameter	Datatype	Use	Description
NbrFlds	Integer	Input	<p>Specifies the number of arguments passed to the CallProg function. For example, if you pass the following two arguments to the CallProg function:</p> <pre>CallProg('program', 'subroutine', 'linkage', arg1, arg2)</pre> <p>PowerExchange sets the NbrFlds parameter to 2.</p> <p>Note: Internally, arguments are passed to the CallProg function through the ppData and ppDataLen arrays. The use of the NbrFlds integer and the ppData and ppDataLen arrays by PowerExchange is similar to the use of the argc integer and the argv array in a C program:</p> <pre>int NbrFlds, char *ppData[], int * pDataLen[] main(int argc, char *argv[])</pre>
pDataLen	Array of pointers	Input and output	<p>Specifies the lengths of the arguments passed to the CallProg function. The length of an argument is determined by its datatype, as follows:</p> <ul style="list-style-type: none"> - Fixed-length datatypes. For datatypes of DOUBLE, FLOAT, NUM8, NUM16, NUM32, NUM64 DATE, and TIME TIMESTAMP, the argument always has the same size if it is present, or a size of 0 if the argument is NULL. - Variable-length datatypes that are not NULL-terminated. For datatypes such as VARCHAR, the length of the data can be determined by using the pDataLen parameter. <p>When fields are returned to the CallProg function, you must reset the data length if it has changed.</p> <p>For example, to change the data and length of second argument with a datatype of VARCHAR, you might code the following statements:</p> <pre>char Name[50]; ... strcpy(ppData[1], Name); *pDataLen[1] = strlen(Name);</pre>

Parameter	Datatype	Use	Description
pMsgBuffer	Pointer to a character string	Output	Points to a message buffer into which the user-defined program can return a NULL-terminated message string that describes an error. PowerExchange issues the returned message if the user-defined program returns a non-zero return code. If no error occurs, Informatica recommends that the user-defined program leave the message buffer empty. Note: When returning error messages, do not exceed the memory buffer size allocated by the CallProg routine, which is 128 bytes.
pMsgBufferSize	Integer pointer	Input and output	Specifies the message buffer size. If an error occurs, the user-defined program can return an error message that includes the message buffer size by using the following syntax: <pre>strncpy(pMsgBuffer, "Error in program FRED", (*pMsgBufferSize) -1);</pre> Return the size of the message buffer if the returned message string is not NULL-terminated.
ppData	Array of pointers	Input and output	An array of pointers to the data for the arguments passed to the CallProg function. This data is accessed in the same way that data is accessed in the argv array on the C main function. For example, to get the data from the second field of field of type STRING, you might code the following statements: <pre>char Name[50]; ... strcpy(Name, ppData[1]);</pre>

User-Defined Programs Return Codes

To indicate whether processing succeeded or failed, a user-defined program returns a return code to the CallProg function.

Code user-defined programs to return the following return codes:

- **0.** Success.
- **Non-zero return code that is positive or negative.** Error.

If the CallProg function receives a non-zero return code from the user-defined program, it performs the following default error handling:

1. A map-level response is triggered.
2. The map-level response either ends the extraction or skips the user-defined program or subroutine.

In the Expression Editor in the PowerExchange Navigator, you can optionally specify a return argument in the call to the CallProg function. If you omit a return argument, the CallProg function detects the return code returned from the user-defined program and performs default error handling.

The Check function enables you to specify which return codes in a user-defined program called from the CallProg function indicate either success or failure.

RELATED TOPICS:

- [“Check” on page 284](#)

User-Defined Programs - Examples

PowerExchange provides example programs that you can use as templates for creating user-defined programs that you can call through the CallProg function.

Example programs are provided in the following languages:

- Assembler
- C
- COBOL
- PL/I
- RPG

User-Defined C Programs - Examples

PowerExchange provides example user-defined C programs on all operating systems.

User-defined C programs use the following prototype:

```
int CRoutine(int NbrFlds
             ,char *pMsgBuffer, int *pMsgBufferSize
             ,char *ppData[], int * pDataLen[]);
```

The following table lists the example C programs:

Operating System	Member or Program	Location	Linkage Type	Program Requirements
i5/OS	-	-	C ¹	Exported routines in a service program
Linux UNIX	-	-	C ¹	Exported routines in an executable shared object
Windows	ucpe.c	The examples directory in the PowerExchange installation directory	C ¹	Exported routines in the CDECL convention in a DLL
z/OS	UCPE	hlq.SRCLIB library	C ¹	<ul style="list-style-type: none">- Must be fetchable from a load module.- Link called modules with AMODE(31). If access is required to AMODE(24) load modules, call the module through an AMODE(31) program that relocates the parameters below the 16 MB line and then calls the AMODE(24) load module.

1. Specify as the third argument in the CallProg function call.

RELATED TOPICS:

- [“Parameter List Passed to User-Defined Programs” on page 309](#)

COBOL Program - Example

PowerExchange provides example COBOL code for a CallProg function in the UCPEC and UCPEC01 members of the SRCLIB library on z/OS.

COBOL Program Operating System Support

Use linkage type COBOL in the third argument to CallProg expression to link to a COBOL language program. COBOL programs can be used on z/OS.

Compile COBOL programs by using the Language Environment (LE) COBOL run-time routines. Results are unpredictable for COBOL programs compiled with non-LE COBOL run-time routines.

COBOL Program Linkage

In COBOL, the first four arguments are always the same and are mandatory.

The remaining fields vary according to the needs of the external program. The attributes of the user-defined expression fields must match the parameters that the external program expects.

COBOL Program Linkage - Example

The following code shows example COBOL program linkage:

```
003700 LINKAGE SECTION.
003800
003900 01  NUMBER-FIELDS          PIC S9(9) COMP.
003901
003902 01  FAILURE-CODE          PIC S9(9) COMP.
003903
003904 01  MESSAGE-BUFFER.
003905     05  MESSAGE-BUFFER-BYTE PIC X(1)
003906         OCCURS 1 TO 128
003907         DEPENDING ON MESSAGE-BUFFER-LENGTH.
003908 01  MESSAGE-BUFFER-LENGTH  PIC S9(9) COMP.
003910
004000 01  TEXT-AREA.
004010     05  TEXT-AREA-BYTE      PIC X(1) OCCURS 15.
004100 01  TEXT-AREA-LENGTH      PIC S9(9) COMP.
004111
004120 01  NUMBER1               PIC S9(9) COMP.
004130 01  NUMBER1-LENGTH        PIC S9(9) COMP.
005730
005800 PROCEDURE DIVISION USING
005801     NUMBER-FIELDS
005802     FAILURE-CODE
005803     MESSAGE-BUFFER
005807     MESSAGE-BUFFER-LENGTH
005809     TEXT-AREA
005810     TEXT-AREA-LENGTH
005811     NUMBER1
005820     NUMBER1-LENGTH
```

In the example program UPEC, two fields are passed to the COBOL program.

- The first field is TEXT, and is a maximum size of 15 bytes. Data can be moved from and to it using COBOL field TEXT-AREA. If it is not a fixed-length CHAR field, then the TEXT-AREA-LENGTH must be used to determine the actual length. If the COBOL program wants the length to change, it must store the required value in field TEXT-AREA-LENGTH.

- The second field is numeric, which in the PowerExchange Navigator is defined as NUM32. Data can be moved from and to it using COBOL field NUMBER1. Generally, this field is not used.

If the field is not nullable, the field contains a value of 4 on entry to the COBOL program. If NUMBER1-LENGTH is NULL, it is set to zero. This situation might occur if the field was NULL before the program was called. If NUMBER1-LENGTH is set to NULL, the program sets the NUMBER1-LENGTH field to 0.

Assembler Program - Example

PowerExchange provides example Assembler code for a CallProg program in the UCPEA member of the SRCLIB library.

Assembler Program Operating System Support

Use linkage type OS in the third argument to CallProg expression to link to an Assembler language program. Assembler language programs can be used on z/OS.

Assembler programs receive a list of parameter addresses in a parameter list pointed to by Register 1. An Assembler program places the return code in register 15.

Assembler Program Linkage

In Assembler, the first three arguments are always the same and are mandatory. Usage of the arguments and return code is similar to that used for C and COBOL routines.

The definitions for the fields vary according to the number of fields and their types.

Assembler Program Linkage - Example

The following code shows an example of Assembler program linkage:

```

L      R3,0(R1)                get address of argument 1
      L      R4,0(R3)          get NumberFields value
      C      R4,=F'2'         Required value of 2 ?
      BNE   BADARGS

*-----
* Get arguments
*-----
NBRARGOK DS      0H
      L      R3,4(R1)          get address of argument 2
      ST     R3,AMGBFF         = address of message buffer
      L      R3,8(R1)          get address of argument 3
      ST     R3,AMGBFSZ        = size of message buffer
      L      R3,12(R1)         get address of argument 4
      ST     R3,ATEXT          = address of text argument
      L      R3,16(R1)         get address of argument 5
      ST     R3,ATEXTLEN       = length of text argument (15)
      L      R3,20(R1)         get address of argument 6
      ST     R3,ANUMBER        = address of number
      L      R3,24(R1)         get address of argument 7
      ST     R3,ANUMBLLEN      = length of number (always 4)

      ...
AMGBFF DS      F
AMGBFSZ DS     F
ATEXT DS      F
ATEXTLEN DS    F
ANUMBER DS     F
ANUMBLLEN DS   F

```

In the example program, two fields are passed to the Assembler program. The first field is called TEXT. Data can be moved to it and from it using the fourth address in the list - the address in 12(R1). If it is not a fixed-length CHAR field, then use the data length to determine the actual length the address in 16(R1).

RPG Program - Example

PowerExchange provides example RPG code for a CallProg program in the UCPERPGL member of the *dtllib*/RPGLE, where *dtllib* is the PowerExchange software library that you specified during PowerExchange installation.

RPG Program Operating System Support

PowerExchange supports C and CL programs on i5/OS.

Use linkage type OS400 in the third argument to CallProg expression to link to a non-C language program. The OS400 linkage type is required for any language other than C, i5/OS, such as CL, COBOL, PL/I, and RPG. Linkage type OS400 can be used on i5/OS.

CL Program Linkage

The first four arguments are always the same and are mandatory. Usage of the arguments and return code is similar to that used in the other languages.

The definitions for the fields vary according to the number of fields and their types.

CL Program Linkage - Example

The following code shows an example of CL program linkage:

```
PGM PARM(&NBRFLDS &RC &MSGBUFF &MSGBUFFSZ &TEXT &TEXTLEN &NUM &NUMLEN)
/*-----*/
/* ARGUMENTS                                     */
/*-----*/
DCL      VAR(&NBRFLDS) TYPE(*CHAR) LEN(4)
DCL      VAR(&RC) TYPE(*CHAR) LEN(4)
DCL      VAR(&MSGBUFF) TYPE(*CHAR) LEN(128)
DCL      VAR(&MSGBUFFSZ) TYPE(*CHAR) LEN(4)
DCL      VAR(&TEXT) TYPE(*CHAR) LEN(15)
DCL      VAR(&TEXTLEN) TYPE(*CHAR) LEN(4)
DCL      VAR(&NUM) TYPE(*CHAR) LEN(4)
DCL      VAR(&NUMLEN) TYPE(*CHAR) LEN(4)
```

In the example program, two fields are passed to the CL program.

The first field is called TEXT that has a fixed length of 15. Data can be moved to it and from it using the 5th argument &TEXT. If it is not a fixed-length CHAR field, then the data length must be used to determine the actual length, such as field &TEXTLEN.

Unfortunately, the CL language does not support integers directly. To get the numeric value, the contents of &TEXTLEN must be moved to a packed decimal using a statement like:

```
CHGVAR  VAR(&DECTEXTLEN) VALUE(%BIN(&TEXTLEN 1 4))
```

If the length is changed by the program then the new length has to be moved into the TEXTLEN field using a statement like:

```
CHGVAR  VAR(%BINARY(&TEXTLEN)) VALUE(&DECTEXTLEN)
```

PL/I Program - Example

PowerExchange provides example PL/I code for a CallProg program in the UCPEP member of the SRCLIB library.

PL/I Program Operating System Support

Use linkage type PLI in the third argument to CallProg expression to link to a PL/I language program.

PL/I language programs can be used on z/OS. Use a C interface layer to call PL/I programs on Windows and UNIX. Use linkage type OS400 to call PL/I programs on i5/OS.

Compile PL/I programs by using the Language Environment (LE) PL/I run-time routines. Results are unpredictable for PL/I programs compiled with non-LE PL/I run-time routines.

PL/I Program Linkage

The first four arguments are always the same and are mandatory. The definitions for the fields vary according to the number of fields and their types.

PL/I Program Linkage - Example

The following code shows an example of PL/I program linkage:

```
PROC (NUMBER_ARGUMENTS,
      FAILURE_CODE,
      MESSAGE_BUFFER_PTR,
      MESSAGE_BUFFER_LENGTH,
      TEXT_AREA_PTR,
      TEXT_AREA_LENGTH,
      NUMBER1,
      NUMBER1_LENGTH)
  OPTIONS (FETCHABLE)
  REORDER;

/*-----*/
/* LINKAGE FROM CALLER */
/*-----*/
DCL NUMBER_ARGUMENTS      FIXED BIN(31);
DCL FAILURE_CODE          FIXED BIN(31);

DCL MESSAGE_BUFFER_PTR    PTR;
DCL 1 MESSAGE_BUFFER_STR  BASED(ADDR(MESSAGE_BUFFER_PTR)),
  4 MESSAGE_BUFFER        CHAR(255);
DCL MESSAGE_BUFFER_LENGTH FIXED BIN(31);

DCL TEXT_AREA_PTR         PTR;
DCL 1 TEXT_AREA_STR       BASED(ADDR(TEXT_AREA_PTR)),
  4 TEXT_AREA             CHAR(15);
DCL TEXT_AREA_LENGTH      FIXED BIN(31);

DCL NUMBER1               FIXED BIN(31);
DCL NUMBER1_LENGTH        FIXED BIN(31);
```

In the example program, two fields are passed to the program.

- The first field is called TEXT and has a maximum size of 15 bytes. Data can be moved from and to it using PL/I field TEXT-AREA. If it is not a fixed-length field, then the TEXT-AREA-LENGTH must be used to determine the actual length. If the program wants the length to change, it must store the required value in field TEXT-AREA-LENGTH.
- The second field is numeric, which in the Navigator is defined as NUM32. Data can be moved from and to it using field NUMBER1. It would be unusual to want to make use of field NUMBER1-LENGTH.

If the field is not nullable, the field contains a value of 4 on entry to the PL/I program.

If the NUMBER1-LENGTH field is NULL, it is set to 0. This situation might occur if the field was NULL before the program was called. If the program sets the field NULL, it also set NUMBER1-LENGTH to 0.

Testing User-Defined Fields in a Data Map

After you define a user-defined field in a record, run a database row test to test the operation of any PowerExchange or user-defined functions on the field.

Run a database row test on a table in the data map that includes a column based on the user-defined field.

To test a user-defined field in a data map:

1. In the data map, add a column to a table that uses the user-defined field that you want to test as its base field.
2. Save the data map.
3. Run a database row test on the table.

Note: Do not include the DISTINCT keyword in the SQL statement when you run a database row test for a data map that contains user-defined fields. If a query requires the DISTINCT keyword, create another table in the data map that contains the fields that are required for the query.

4. Review the output in the **Database Row Test Output** window.

For more information about using SQL for nonrelational sources, see the *PowerExchange Reference Manual*.

RELATED TOPICS:

- [“Adding a Column to a Table” on page 76](#)
- [“Testing a Data Map” on page 213](#)

APPENDIX B

User Access Method Programs

This appendix includes the following topics:

- [User Access Method Programs Overview, 317](#)
- [Parameter Structure Passed to the User Access Method Program, 319](#)
- [Return Codes from the User Access Method Program, 321](#)
- [User Access Method Programs - Examples, 321](#)

User Access Method Programs Overview

Create a user access method program written in Assembler, C, or PL/I to process data in a data map.

You call the user access method program by defining a data map with the USER access method and specifying the attributes of the user access method program, including the name of the compiled user access method DLL or shared library, parameters to pass to the program, and the full path and name of the associated data file.

A user access method program has a single entry point. PowerExchange calls the user access method program repeatedly to process the data in the associated data file until all the data is processed.

The following table lists the calls that PowerExchange makes to a user access method program and describes the actions that the program might take in response to a call:

Call	Possible User Access Method Program Actions
OPEN	Open the data file and perform initialization, as follows: <ul style="list-style-type: none"> - Allocate storage. - Validate and save any passed parameters. - Initialize counters. - verify that the data file exists. - Open the data file. - If the target is to be replaced, delete the output file. - Return a success or error code to the PowerExchange Navigator.
READ	Start or continue sequential reading, as follows: <ul style="list-style-type: none"> - Read the data file to get the next record, using any SQL statement passed from the PowerExchange Navigator to get the appropriate record. - Pass the data back to the PowerExchange Navigator by setting the record pointer to point to the data, and setting the value in the record_len field to the length of the data being returned. - Return a success, end-of-file, or error code to PowerExchange.
WRITE	Start or continue sequential writing, as follows: <ul style="list-style-type: none"> - Get the data to write by using the record pointer and the length set in the record_len field. - Write the specified data. - Return a success or error code to the PowerExchange Navigator.
CLOSE	Perform clean-up operations, as follows: <ul style="list-style-type: none"> - Close the data file. - Release allocated storage. - Commit the database. - Return a success or error code to the PowerExchange Navigator. - If an error occurs, place message text in the message buffer. The message buffer holds a 79-character message. - Pass control back to the calling program.

PowerExchange passes the user access method program a parameter structure with every call. The parameter structure includes the parameters that you define in the user access method data map.

The user access method passes a return code back to the PowerExchange Navigator to indicate the success or failure of the call, or to indicate that the end of the data file was reached.

RELATED TOPICS:

- [“Adding a Data Map” on page 62](#)

Parameter Structure Passed to the User Access Method Program

The PowerExchange Navigator provides a single parameter that locates the parameter structure.

The following table describes the parameter structure passed to the user access method program:

Variable Name	Description	Corresponding Field in the PowerExchange Navigator GUI	Type	Size
LGMODE	Used in connection with a value of 1 in the LGCODE variable, and indicates how the data file is opened, as follows: <ul style="list-style-type: none">- 1. Read.- 2. Write.- 3. Update.- 4. Append.	Not applicable	Binary	Word
LGCODE	Indicates the function to be performed by the user access method program, as follows: <ul style="list-style-type: none">- 1. Open.- 2. Close.- 3. Read.- 4. Write.	Not applicable	Binary	Word
LGUI1	User-defined parameter, which can be used to provide numeric information such as the record length or the number of rows to be retrieved.	Integer 1 value in the USER Access Method dialog box	Binary	Word
LGUI2	User-defined parameter, which can be used to provide numeric information such as the record length or the number of rows to be retrieved.	Integer 2 value in the USER Access Method dialog box	Binary	Word
LGREADCT	The number of records read from the data file.	Not applicable	Binary	Word
LGWRITCT	The number of records written to the data file.	Not applicable	Binary	Word
LGMLRECL	The record length of the retrieved data.	Set by the user access method program	Binary	Word

Variable Name	Description	Corresponding Field in the PowerExchange Navigator GUI	Type	Size
LGU1LEN	The length of the data entered in the String 1 box in the USER Access Method dialog box.	Derived from the String 1 value in the USER Access Method dialog box	Binary	Word
LGU2LEN	The length of the data entered in the String 2 box in the USER Access Method dialog box.	Derived from the String 2 value in the USER Access Method dialog box	Binary	Word
LGFNLEN	The length of the data file name entered in the File Name box in the USER Access Method dialog box.	Derived from the File Name value in the USER Access Method dialog box	Binary	Word
LGUIDLEN	The length of the data entered in the User ID box in the Data Map Remote Node dialog box.	Derived from the User ID value in the Data Map Remote Node dialog box	Binary	Word
LGPWDLEN	The length of the data entered in the Password box in the Data Map Remote Node dialog box.	Derived from the Password value in the Data Map Remote Node dialog box	Binary	Word
LGSQLEN	The length of the data entered in the SQL Statement box in the Database Row Test dialog box.	Derived from the SQL Statement value in the Database Row Test dialog box	Binary	Word
LGU1PTR	A pointer to the data stored for String 1.	String 1 value in the USER Access Method dialog box	Pointer	Word
LGU2PTR	A pointer to the data stored for String 2.	String 2 value in the USER Access Method dialog box	Pointer	Word
LGFNPTR	A pointer to the data stored for File Name.	File Name value in the USER Access Method dialog box	Pointer	Word
LGUIDPTR	A pointer to the data stored for User ID.	User ID value in the Data Map Remote Node dialog box	Pointer	Word
LGPWDPTR	A pointer to the data stored for Password.	Password value in the Data Map Remote Node dialog box	Pointer	Word
LGSQLPTR	A pointer to the data stored for SQL Statement.	SQL Statement value in the Database Row Test dialog box	Pointer	Word
LGPUWK1	A pointer to area of storage for user-defined data.	Set by the user access method program	Binary	Word
LGPUWK2	A pointer to area of storage for user-defined data.	Set by the user access method program	Binary	Word
LGOPENRC	Return code of the OPEN call, as follows : - 1. Success. - 0. Error.	Set by the user access method program	Binary	Word

Variable Name	Description	Corresponding Field in the PowerExchange Navigator GUI	Type	Size
LGRECLEN	Length of the retrieved record.	Set by the user access method program	Binary	Word
LGRECPTR	A pointer to the retrieved record data.	Set by the user access method program	Pointer	Word
LGMSGBUF	Message returned to PowerExchange from the user access method program.	Set by the user access method program	Character	80 bytes

RELATED TOPICS:

- [“Database Row Test Dialog Box” on page 219](#)
- [“Data Map Properties - USER Access Method” on page 343](#)
- [“Sending a Data Map to a Remote System” on page 90](#)

Return Codes from the User Access Method Program

The user access method passes a return code back to the PowerExchange Navigator.

The PowerExchange Navigator expects one of the following return codes from a user access method program:

0

Success. The OPEN, READ, WRITE, or CLOSE call succeeded.

255

End of file. No more records are available for read.

Other

Error. The OPEN, READ, WRITE, or CLOSE call failed.

User Access Method Programs - Examples

PowerExchange provides example user access method programs written in Assembler, C, and PL/I, with headers files and JCL.

These files, named uam*.* and dlgif*.*, are located in the examples directory in the PowerExchange Navigator installation directory. Edit and compile these example programs to customize them for your environment.

APPENDIX C

Application Groups and Applications

This appendix includes the following topics:

- [Application Groups and Applications Overview, 322](#)
- [Adding an Application Group, 323](#)
- [Viewing or Editing Properties for an Application Group, 324](#)
- [Resetting the Extraction Start Point for an Application, 326](#)
- [Deleting an Application Group, 326](#)

Application Groups and Applications Overview

If you use ODBC connections to extract change data, PowerExchange uses application groups and applications to maintain extraction information for change data extractions.

An application group defines the node location of the data source. The PowerExchange Navigator adds an application group when you add a registration group. You can also manually add an application group.

Applications appear under an application group. You do not add applications through the PowerExchange Navigator. Add applications dynamically when you run a CDC extraction with a new application name or run the DTLUAPPL utility. The application name is the name that you specify in the DBQUAL2 parameter of the ODBC connection or in the DTLAPP SQL override.

Note: Informatica recommends that you specify a unique application name for each extraction process to verify that restart occurs. If you use the same application name for multiple extraction processes, errors might occur, such as the wrong data being extracted and other unexpected results.

Each application displays the history of change data extractions in descending time stamp order. For each extraction, the history includes:

- Time stamp
- Audit trail
- Extraction status
- Capture registration name and table name

Additionally, each application stores the start point for the next extraction, which you can reset in the PowerExchange Navigator or by using the DTLUAPPL utility.

The CDEP file is the control file in which PowerExchange stores application information. Application information includes the application name, the start point for the next extraction, and information about the extraction processes that have run including their history. PowerExchange ODBC uses the restart tokens associated with application names in the CDEP file to determine the starting extraction point for a session. The application group is used to filter the application information in the CDEP file for display in the PowerExchange Navigator.

Note: If you use PowerExchange Client for PowerCenter (PWXPC) connections to extract change data as recommended, PowerExchange does not use application groups and applications. With PWXPC, PowerExchange maintains restart information for extractions in the following locations:

- For relational targets, in state tables
- For nonrelational targets, in the state file on the Integration Service

For more information, see the CDC guide for the operating system.

RELATED TOPICS:

- [“Registration Groups” on page 157](#)

Adding an Application Group

An application group is a named group of applications, which are used when you use ODBC connections.

An application group defines the node location of the source, and if required, a user ID and password to access the source. When you add a registration group, PowerExchange adds an application group with the same name. You can also manually add an application group by using this procedure.

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Application Group**.
2. In the **Add Application Group** dialog box, enter the following information:

Name

A user-defined name for the application group. Maximum length is 16 characters.

Location

The location of the source for which you want to extract change data. Select **local** if the source is on the PowerExchange Navigator system.

This field lists the locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.

Default is **local**.

UserID

For an i5/OS or z/OS source, enter an operating system user ID if PowerExchange security is enabled. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x).

For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the *PowerExchange Reference Manual*.

Password

A password for the specified user ID.

If the source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters:

- Uppercase and lowercase letters
- The numbers 0 to 9
- Spaces
- The following special characters:
' - ; # \ , . / ! % & * () _ + { } : @ | < > ?

Note: The first character is an apostrophe.

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

On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

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

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

If you specify a user ID but omit the password, when you open the application group, the PowerExchange Navigator prompts you for the password in the **Missing Password** dialog box.

3. Click **Finish**.

The **Application Group** tab in the **Resource Inspector** displays the properties for the application group.

If you use ODBC connections to extract change data, the **Application Group** tab in the **Resource Explorer** displays any applications for the node location associated with the application group.

RELATED TOPICS:

- ["Registration Groups" on page 157](#)

Viewing or Editing Properties for an Application Group

You can view or edit properties for an application group.

You can change the application group name, location, and user ID and password.

1. Open the application group.

The following table describes the properties that the **Application Group** tab in the **Resource Inspector** displays for the application group:

Property	Description
Name	A user-defined name for the application group.
Location	Location of the source for which you want to capture changes. Select local if the source is on the PowerExchange Navigator system. This field lists the locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.
UserID	A user ID that allows access to the source. The type of user ID depends on the source type and PowerExchange security settings: <ul style="list-style-type: none"> - If the source is on an i5/OS or z/OS system and PowerExchange security is enabled, enter an operating system user ID. PowerExchange security is enabled when the PowerExchange SECURITY statement in the DBMOVER member is set to (1,x) or 2.x). - For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID, if required. For more information, see the SECURITY Statement in the <i>PowerExchange Reference Manual</i>.
Password	A password for the specified user ID. If the source is on an i5/OS or z/OS system, you can enter a valid PowerExchange passphrase instead of a password. An i5/OS passphrase can be from 9 to 31 characters in length. A z/OS passphrase can be from 9 to 128 characters in length. A passphrase can contain the following characters: <ul style="list-style-type: none"> - Uppercase and lowercase letters - The numbers 0 to 9 - Spaces - The following special characters: ' - ; # \ , . / ! % & * () _ + { } : @ < > ? <p>Note: The first character is an apostrophe.</p> <p>Passphrases cannot include single quotation marks ('), double quotation marks ("), or currency symbols.</p> <p>On z/OS, the allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.</p> <p>Note: On z/OS, a valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.</p> <p>To use passphrases, ensure that the PowerExchange Listener runs with a security setting of SECURITY=(1,N) or higher in the DBMOVER member. For more information, see "SECURITY Statement" in the <i>PowerExchange Reference Manual</i>.</p>

2. To edit a property, click a property and enter or select a value.

After you edit a property, a red square appears next to the changed property.

Tip: To undo a change before you apply it, click **Reset**.

3. To apply a change, click **Apply**.

If you edit the application group name, you are prompted to confirm the change. If the new name matches the name of an existing application group, you are prompted to overwrite the existing group.

4. Click **Yes** to confirm changes.

The **Resource Inspector** updates any fields that you changed.

RELATED TOPICS:

- [“Adding an Application Group” on page 323](#)

Resetting the Extraction Start Point for an Application

An application displays the history of change data extractions for a capture registration and stores the start point for the next extraction.

You can reset the start point for the next extraction, displayed as the **Current Run** extraction, to the start point of another extraction.

After you reset the extraction start point, the application can either restart the extraction at the reset start point time, or have its extraction start point reset again. You cannot undo the reset of a start point.

Alternatively, after an initial extraction, you can use the DTLUAPPL utility to reset the extraction start point for an application.

To reset the extraction start point for an application:

1. Open the application group, and open the application.

The pane on the right side lists the extractions for the application in descending time stamp order.

The information for the current extraction, displayed as the **Current Run** extraction, is blank. All other extractions display a time stamp.

2. Select the extraction for which to reset the start point.

3. Right-click the application group name and click **Reset To New Start Point**.

The icon next to the application group name appears in red.

PowerExchange resets the restart point of the next extraction to the start point of the selected extraction.

When you open the application again, the **Current Run** extraction displays the run information from the selected extraction.

Deleting an Application Group

You can delete application groups.

When you delete an application group, PowerExchange deletes local group information on the PowerExchange Navigator system. PowerExchange does not delete the applications in the group or the associated registration group. If you add another application group with the same properties as a group that you deleted, the new group contains any existing applications for the deleted group.

Also, if PowerExchange generated the application group, PowerExchange does not delete the registration group that generated the creation of this application group.

To delete an application group:

1. If the application group is open, close it.
2. In the **Resources** window, delete one or more application groups.

-
-
3. In the message box that prompts you to confirm the deletion of the group or groups, click **Yes**. PowerExchange deletes the application group or groups.

APPENDIX D

Data Map Properties

This appendix includes the following topics:

- [Data Map Properties - ADABAS Access Method, 328](#)
- [Data Map Properties - DATACOM Access Method, 330](#)
- [Data Map Properties - DB2 Access Method, 330](#)
- [Data Map Properties - DB2UNLD Access Method, 331](#)
- [DB2 Table Filter Dialog Box, 332](#)
- [DB2 Table Filter - Available Tables Dialog Box, 334](#)
- [Data Map Properties - DL/I Batch or IMS ODBA Access Method, 334](#)
- [Data Map Properties - ESDS Access Method, 335](#)
- [Data Map Properties - IDMS Access Method, 336](#)
- [Data Map Properties - KSDS Access Method, 337](#)
- [Data Map Properties - MQSeries Access Method, 338](#)
- [Data Map Properties - RRDS Access Method, 339](#)
- [Data Map Properties - SEQ Access Method, 340](#)
- [Data Map Properties - TAPE Access Method, 342](#)
- [Data Map Properties - USER Access Method, 343](#)
- [Data Map Properties - Data Defaults, 345](#)
- [Data Map Properties - Data Checking, 346](#)
- [Data Map Properties - Record Level Exit, 348](#)
- [Data Map Properties - Keys, 348](#)

Data Map Properties - ADABAS Access Method

View or edit properties for an Adabas data map.

Database ID

The Adabas database ID. Set to 0 to override the database at run time or if the actual database ID is 0. Overriding the database at run time makes migrating data maps from one Adabas environment to another easier.

To override the database at run time, use one of the following methods:

- ADABAS_DEFAULT_DBID statement in the DBMOVER configuration file
- Database Id Override option in the session properties in PowerCenter
- Override File Name in the ODBC DTLDSN escape sequence
- Override File Name in the ODBC DBQUAL1 parameter

Default is 1.

File Number

The file number in the Adabas database.

Default is 1.

Optimization Level

Specifies how PowerExchange uses Adabas keys. The options are:

- **OFF**. PowerExchange does not use Adabas keys.
- **Full**. PowerExchange uses Adabas keys when an SQL WHERE clause is present.
- **Ranges Only**. PowerExchange uses Adabas keys when the key value is in the specified range and when an SQL WHERE clause is present.

Default is **Ranges Only**.

Encoding

The encoding scheme. The options are:

- **Default**. PowerExchange uses the default encoding scheme for the operating system.
- **EBCDIC**. EBCDIC.
- **ASCII HIEND**. ASCII high endian.
- **ASCII LOEND**. ASCII little endian.

Default is **Default**.

Codepage

The code page that describes the character set for the character data in the database. For the **Default** encoding, you cannot select a code page. PowerExchange uses the default code page for the operating system.

Default is **Default**.

Wide Character Codepage

The code page for Adabas fields that use wide character datatypes.

Select **Default** to use the following code pages on the following operating systems:

- Linux, UNIX and Windows. UTF-8.
- z/OS. Default operating system code page.

Default is **Default**.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Adding a Data Map” on page 62](#)

- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - DATACOM Access Method

View or edit properties for a Datacom data map.

Database Name

Required. The CA Datacom/DB database name. Valid characters are ASCII characters and the pound and yen sign.

Database ID

Required. The ID number of the CA Datacom/DB database.

Note: When you define a CA Datacom/DB database in the data dictionary, you must assign a numeric ID to the database. This ID value is the DATACOM-ID attribute-value for the database. Certain IDs, such as 1 through 20, 1000 through 1020, 2000 through 2020, 3000 through 3020, 4000 through 4020, and 5000, are reserved for use by CA Datacom/DB.

Record Size

The record size for the table. If you are adding a data map and you did not select the **Import Record Definition** option, enter a record size. In a CA Datacom/DB table, the following limits apply to the size of a record:

- Minimum size is 1 byte.
- Maximum size is 32720 bytes.

Codepage

The code page that describes the character set to use for character data in the database. As the Default code page, PowerExchange uses IBM-037.

Note: Multibyte code pages are available for Datacom.

Default value is **Default**.

RELATED TOPICS:

- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - DB2 Access Method

View or edit properties for a DB2 data map.

DB Instance

Required. The DB2 database instance name. Valid characters are ASCII characters and the pound and yen sign.

Table Name

Required. The DB2 table name. Enter the table name or click the Browse button to enter filter criteria for the table.

Encoding

The encoding scheme. The options are:

- **Default.** PowerExchange uses the default encoding scheme for the operating system.
- **EBCDIC.** EBCDIC.
- **ASCII HIEND.** ASCII high endian.
- **ASCII LOEND.** ASCII little endian.

Default is **Default**.

Codepage

The code page that describes the character set for the character data in the database. For the **Default** encoding, you cannot select a code page. PowerExchange uses the default code page for the operating system.

Default is **Default**.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - DB2UNLD Access Method

View or edit properties for an DB2 unload file data map.

File Name

Required. The DB2 unload file name. Valid characters are ASCII characters and the pound and yen signs.

DB Instance

The DB2 database instance name. Valid characters are ASCII characters and the pound and yen signs.

Table Name

Required. The DB2 table name. Enter the table name or click the Browse button to enter filter criteria for the table.

Unload Type

The unload type. For bulk access to unload files, the following IBM and BMC formats are supported:

- REORG UNLOAD
- DSNTIAUL/BMC UNLOAD+
- DSNUTILB UNLOAD
- UNDEFINED. By default, this type is generated with null indicators, and count fields in the same position as in the REORG UNLOAD EXTERNAL type.

Instead of using these unload types, you can modify these statements to meet specific requirements. You can move or delete these statements to correspond to user-defined unload formats, but PowerExchange interprets them if they precede the data field to which they apply.

Encoding

The encoding scheme. The options are:

- **Default.** PowerExchange uses the default encoding scheme for the operating system.
- **EBCDIC.** EBCDIC.
- **ASCII HIEND.** ASCII high endian.
- **ASCII LOEND.** ASCII little endian.

Default is **Default**.

Codepage

The code page that describes the character set for the character data in the database. For the **Default** encoding, you cannot select a code page. PowerExchange uses the default code page for the operating system.

Default is **Default**.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

DB2 Table Filter Dialog Box

In the DB2 Table Filter dialog box, view or edit table filtering criteria for a DB2 catalog or DB2 unload file data map.

The dialog box contains the following fields:

DB Instance

The DB2 database instance name. Valid characters are ASCII characters and the pound and yen sign.

Location

The location of the data source. Select **local** if the data source is on the PowerExchange Navigator system.

This field lists the locations that are defined in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system.

Default is **local**.

UserID

A database user ID that can access the database, if required. For more information, see the SECURITY statement in the *PowerExchange Reference Manual*.

For a database on a supported Linux, UNIX, or Windows system, if you have enabled PowerExchange LDAP user authentication and, if applicable, disabled relational pass-through authentication, the user ID is the enterprise user name.

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

Password

A password for the specified user ID.

For access to z/OS, you can enter a valid PowerExchange passphrase instead of a password. A PowerExchange passphrase can be from 9 to 128 characters in length and can contain the following characters:

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

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

Note: The first character is an apostrophe.

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

The allowable characters in the IBM IRRPHREX exit do not affect the allowable characters in PowerExchange passphrases.

Note: A valid RACF passphrase can be up to 100 characters in length. PowerExchange truncates passphrases longer than 100 characters when passing them to RACF for validation.

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

Filter Data

Enter text in any of the following filter boxes:

- **Creator/Schema**
- **DBName/Definer**
- **Table**

For DB2 for i5/OS, the maximum length is 10 characters in the **Creator/Schema** box and 32 characters in the **Table** box.

For DB2 for z/OS, the maximum length is 128 characters in the **Creator/Schema** box and 128 characters in the **Table** box.

In any filter box, you can use the following wildcard characters:

- An asterisk (*) to represent one or more matching characters.
- A question mark (?) to represent a single matching character.

Respect Case

By default, text entered in any of the filter boxes is converted to the standard case for the DB2 data source. Select the **Respect Case** option for those filters for which you do not want PowerExchange to use the standard case for the DB2 data source.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

DB2 Table Filter - Available Tables Dialog Box

Select a table for a DB2 or DB2 unload file data map.

Available Tables

Lists the tables that meet the filter criteria that you entered in the **DB2 Table Filter** dialog box.

Select the DB2 table to map and click **Finish**.

RELATED TOPICS:

- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - DL/I Batch or IMS ODBA Access Method

View or edit properties for a data map that uses the **IMS DL/1 BATCH** or **IMS ODBA** access method.

IMS SSID

The IMS subsystem ID. Maximum length is four characters.

Depending on the access method, this property is a logical value or actual SSID:

- If you use the IMS ODBA access method, this value is the actual IMS SSID that is defined for the control region. This value must match the DBCTLIB value in the IMS job that configures ODBA access and match the SSID value in an IMSID statement in the DBMOVER configuration member.

For an IMS target, if you use the DTL.DBWRITE.IMS.*imsid* resource profile to control write access, verify that this IMS SSID value matches the *imsid* value in the resource profile.

- If you use DL/I or BMP access, this value is a logical SSID that matches the SSID that is specified in an IMSID statement in the DBMOVER configuration member and in the registration group. It is not used for DL/I access to the IMS database.

For an IMS target, if you use the DTL.DBWRITE.IMS resource profile to control write access, do *not* specify this IMS SSID value.

Note: You can override the IMS SSID value in the PowerCenter session properties for a bulk data movement source or target. You can also override the IMS SSID in the advanced parameters for a database row test.

DBD Name

The name of the IMS database description (DBD) for the IMS source. Specify this value if you plan to read IMS unload files.

PCB Number

Required for DL/I batch or BMP access by means of a netport job. Specify the relative PCB number for the database. If the PSB specifies COMPAT=YES, add 1 to the PCB number.

PSB Name

Required for IMS ODBA access. Specify the name of a PSB that contains the DBD for the database.

Maximum length is eight alphanumeric characters.

PCB Name

Required for IMS ODBA access. Enter the name of a PCB that is in the specified PSB and that references the named DBD. You can enter a label for the PCB or the PCB name parameter value.

Data Codepage

The code page that describes the character set for the character data in the database. For the **Default** code page, PowerExchange uses the default code page for the operating system.

Default is **Default**.

Skip First *n* Records from File

The number of header records in the data file to skip. For example, if the data file contains one or more header records that do not contain data, enter the number of header records to skip.

Valid values are from 0 through 2147483647. Default is 0.

Data Map Properties - ESDS Access Method

View or edit properties for a VSAM entry-sequenced data set (ESDS) data map.

File Name

The fully-qualified data set name of the VSAM entry-sequenced data set (ESDS).

Number of Data Buffers

The number of I/O buffers for data control intervals that PowerExchange requests VSAM to allocate.

Default is 2.

CI ACCESS

Accesses the entire contents of a control interval rather than individual data records when reading VSAM data sets. This option improves read performance.

Clear this option to access the contents of individual data records when reading VSAM data sets.

Note: You cannot use the **CI ACCESS** option if the data set is compressed.

Prefix record with XRBA value

Returns the extended relative byte address (XRBA) value for all records read from the VSAM data set. The XRBA value is returned in the first eight bytes of each record.

Note: You must have included a 8-byte binary field at the beginning of the record to contain the XRBA value. If you select this option and you do not include an 8-byte binary field at the beginning of the record, the data maps incorrectly when you get it.

File List Processing

Uses file list processing for the data map.

Data Codepage

The code page that describes the character set for the character data in the database. For the **Default** code page, PowerExchange uses the default code page for the operating system.

Default is **Default**.

Skip First *n* Records from File

The number of header records in the data file to skip. For example, if the data file contains one or more header records that do not contain data, enter the number of header records to skip.

Valid values are from 0 through 2147483647. Default is 0.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Associating Multiple Data Files with a Data Map” on page 93](#)
- [“Improving Bulk Read Performance for VSAM Data Sets” on page 129](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - IDMS Access Method

View or edit properties for an IDMS data map.

Note: The **Sub Schema Name** field is required. All other fields are optional, unless required by the installation configuration and standards.

Sub Schema Name

Required. The name of the subschema. Valid characters are ASCII characters and the pound and yen sign.

DBName

The name of the database. Valid characters are ASCII characters and the pound and yen sign.

Program Name

The program identification string. Valid characters are ASCII characters and the pound and yen sign.

DBNode

The Distributed Database System (DDS) node name. Valid characters are ASCII characters and the pound and yen sign.

Dictionary Name

The name of the dictionary. Valid characters are ASCII characters and the pound and yen sign.

Dictionary Node

The DDS dictionary node name. Valid characters are ASCII characters and the pound and yen sign.

Ready Mode

In the first list, select the protection mode. The options are:

- **Exclusive.** Prevents concurrent use by other users.
- **Protected.** Other processes cannot make updates until the run unit is complete.
- **Shared.** No locking or protection occurs.

Default is **Shared**.

In the second list, select the data mode. The options are:

- **Retrieval.** Data retrieval mode.
- **Update.** Data update mode.

Default is **Retrieval**.

Ready All

Indicates that all areas are readied.

RELATED TOPICS:

- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - KSDS Access Method

View or edit properties for a VSAM keyed-sequenced data set (KSDS) data map.

File Name

The fully-qualified data set name of the VSAM keyed sequenced data set (KSDS).

Number of Data Buffers

The number of I/O buffers for data control intervals that PowerExchange requests VSAM to allocate.
Default is 2.

Number of Index Buffers

The number of I/O buffers for index control intervals that PowerExchange requests VSAM to allocate.
Default is 1.

CI ACCESS

Accesses the entire contents of a control interval rather than individual data records when reading VSAM data sets. This option improves read performance.

Clear this option to access the contents of individual data records when reading VSAM data sets.

Note: You cannot use the **CI ACCESS** option if the data set is compressed.

Data Codepage

The code page that describes the character set for the character data in the database. For the **Default** code page, PowerExchange uses the default code page for the operating system.

Note: Multibyte code pages are available for KSDSs.

Default is **Default**.

Skip First *n* Records from File

The number of header records in the data file to skip. For example, if the data file contains one or more header records that do not contain data, enter the number of header records to skip.

Valid values are from 0 through 2147483647. Default is 0.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Improving Bulk Read Performance for VSAM Data Sets” on page 129](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - MQSeries Access Method

View or edit properties for an MQSeries data map.

Queue Manager

Required. The name of the MQ Series Queue Manager.

Valid characters are ASCII characters and the pound and yen sign.

Queue Name

Required. The name of the queue.

Valid characters are ASCII characters and the pound and yen sign.

Note: After you enter the queue manager name and the queue name, click **View Queue** to display the data file.

Field Separator

If the data file is a character-separated file, select this option to enter the character that is used to separate fields in the file.

Field Delimiter

Enter the character that is used to delimit data within a field. For example, the quote (") character might be used to enclose strings within a field.

Enabled if you select the **Field Separator** option.

Merge Adjacent Separators

Merges adjacent separators into a single separator.

Enabled if you select the **Field Separator** option.

Encoding

The encoding scheme. The options are:

- **Default.** PowerExchange uses the default encoding scheme for the operating system.

- **EBCDIC.** EBCDIC.
- **ASCII HIEND.** ASCII high endian.
- **ASCII LOEND.** ASCII little endian.

Default is **Default**.

Codepage

The code page that describes the character set for the character data in the database. For the **Default** code page, PowerExchange uses the default code page for the operating system. Default is **Default**.

View Queue

Displays the data file.

Enabled if you enter a queue manager and queue name.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - RRDS Access Method

View or edit properties for a VSAM relative record data set (RRDS) data map.

File Name

The fully-qualified data set name of the VSAM relative record data set (RRDS).

Number of Index Buffers

The number of I/O buffers for index control intervals that PowerExchange requests VSAM to allocate. Default is 1.

Number of Data Buffers

The number of I/O buffers for data control intervals that PowerExchange requests VSAM to allocate. Default is 2.

CI ACCESS

Accesses the entire contents of a control interval rather than individual data records when reading VSAM data sets. This option improves read performance.

Clear this option to access the contents of individual data records when reading VSAM data sets.

Note: You cannot use the **CI ACCESS** option if the data set is compressed.

Prefix record with RRN value

Returns the relative record number (RRN) value for all records read from the VSAM data set.

Note: You must include a 8-byte binary field at the beginning of the record to contain the RRN value.

Data Codepage

The code page that describes the character set for the character data in the database. For the **Default** code page, PowerExchange uses the default code page for the operating system. Default is **Default**.

Skip First *n* Records from File

The number of header records in the data file to skip. For example, if the data file. For example, if the data file contains one or more header records that do not contain data, enter the number of header records to skip.

Valid values are from 0 through 2147483647. Default is 0.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Improving Bulk Read Performance for VSAM Data Sets” on page 129](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - SEQ Access Method

View or edit properties for a data map defined for flat file or sequential data set.

Filename

The full path and file name of the sequential data set or flat file. If the file is located on the local system, click the Browse button to browse to the file.

Note: Do not enclose the name in single quotes. PowerExchange uses the specified name as the fully qualified file name and prevents z/OS from adding a user ID as a prefix.

Valid characters are ASCII characters and the pound and yen sign.

Fixed

For a fixed-length binary data file, select this option.

In the **Size** box, enter the record length.

A fixed-length binary file is exactly divisible by the specified record length.

Variable

For a variable-length data file, select this option.

In the **Variable** list, select one of the following variable-length data file types:

- **VB12.** Data length is 2-byte value, where length "l"ncludes 2-byte length.
- **VB14.** Data length is 4-byte value, where length "l"ncludes 4-byte length.
- **VBX2.** Data length is 2-byte value, where length e"X"cludes 2-byte length.
- **VBX4.** Data length is 4-byte value, where length e"X"cludes 4-byte length.
- **VS.** Variable-length stream data file. In the data file, no record boundaries exist. The records are field delimited and are not of fixed length.

You can use this option for data that you transferred between a UNIX or Windows system and an i5/OS or z/OS system.

For files that you transferred from UNIX or Windows to z/OS, verify that the LRECL value that you define for the data set is greater than the length of the longest record in the UNIX or Windows file. For files that you transferred from UNIX or Windows to i5/OS, verify that the record length that you specified on the Create Physical File (CRTPF) command is greater than the length of the longest record in the UNIX or Windows file.

If you select this option, the **Skip First *n* Records from File** data map property is not available. Also, you cannot use the **Skip First *n* Rows** option in the **Database Row Test** dialog box to skip a specific number of initial rows when performing a database row test.

- **VBP2.** VLI length value includes 4-byte prefix, 2-byte VLI, and two bytes of NULLS. Equates to z/OS variable unblocked.
- **VBM2.** VLI length value is "M"inus, includes 4-byte prefix, 2-byte VLI, two bytes of NULLS.
- **VBB2.** VLI length value includes 4-byte prefix and 4-byte Block Descriptor Word. Equates to z/OS Variable Blocked. Use for files that are transferred to an ASCII system to retain the Block and Record Descriptor Words.

Note: On Linux, Windows, and UNIX, you can process or create variable-length binary data by using these file types. With all these file types except VS, the data is preceded by one of the following integer-length fields:

- 16-bit short integer, which is two bytes
- 32-bit integer, which is four bytes

The length contained in this field provides the length of the actual data and might include the length of the prefix.

Default

If you want PowerExchange to assign the file type based on the operating system and file attributes, select this option. Use this option if you are reading mainframe data that did not originate in another system.

Default is selected by default.

Encoding

The encoding scheme. The options are:

- **Default.** PowerExchange uses the default encoding scheme for the operating system.
- **EBCDIC.** EBCDIC.
- **ASCII HIEND.** ASCII high endian.
- **ASCII LOEND.** ASCII little endian.

Note: The encoding scheme applies to both the record descriptors and the data in the record.

Default is **Default**.

Codepage

The code page that describes the character set for the character data in the database. For the **Default** code page, PowerExchange uses the default code page for the operating system.

Note: Multibyte code pages are available for sequential data sets or flat files.

Default is **Default**.

Field Separator

If the data file is a character-separated file, select this option to enter the character that is used to separate fields in the data file.

Valid characters are ASCII characters and the pound and yen sign.

If the field separator character is a hexadecimal combination, enter it in the following format:

x'nn'

Merge Adjacent Separators

Merges adjacent separators into a single separator.

Enabled if the **Field Separator** option is selected.

Field Delimiter

Enter the character that is used to delimit data within a field. For example, the quote (") character might be used to enclose strings within a field.

Enabled if the **Field Separator** option is selected.

Valid characters are ASCII characters and the pound and yen sign.

Skip First *n* Records from File

If the file has one or more header records that are not needed for PowerExchange operations, enter the number of header records to skip.

Valid values are 0 to 2147483647.

Default is 0, which causes no rows to be skipped.

File List Processing

Uses file list processing for the data map.

View File

Displays the data file.

Enabled after you enter a file name.

For more information about defining data map properties of sequential files for bulk data movement, see the chapter, "Sequential File Bulk Data Movement," in the *PowerExchange Bulk Data Movement Guide*.

RELATED TOPICS:

- ["Code Pages" on page 58](#)
- ["Associating Multiple Data Files with a Data Map" on page 93](#)
- ["Adding a Data Map" on page 62](#)
- ["Viewing or Editing Properties for a Data Map Table" on page 86](#)

Data Map Properties - TAPE Access Method

View or edit properties for a data map defined for a tape data set.

Filename

The full path and file name of the tape data set on z/OS. This name can be a GDG data set name.

Note: Do not enclose the name in single quotes. PowerExchange uses the specified name as the fully qualified file name and prevents z/OS from adding a user ID as a prefix.

Valid characters are ASCII characters and the pound and yen sign.

Field Separator

If the data file is a character-separated file, select this option to enter the character to use to separate fields in the data file.

Valid characters are ASCII characters and the pound and yen sign.

If the field separator character is a hexadecimal combination, enter it in the following format:

x'nn'

Merge Adjacent Separators

Merges adjacent separators into a single separator.

Enabled if you select the **Field Separator** option.

Position

The relative position to 1 of the file on tape.

This value must match the value of the RELPOS parameter in the DBMOVE configuration member.

Unit

This value must match the value of the UNIT parameter in the DBMOVE configuration member.

Volser

This value must match the value of the VOLSER parameter in the DBMOVE configuration member.

Skip First *n* Records from File

If the file has one or more header records, enter the number of header records to skip.

Valid values are from 0 through 2147483647.

Default is 0.

File List Processing

Uses file list processing for this data map.

RELATED TOPICS:

- [“Associating Multiple Data Files with a Data Map” on page 93](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - USER Access Method

View or edit properties for a data map defined for a user access method program.

Exit Program Name

The name of the user access method program to be called at run time. The program must be a valid DLL or shared library.

Calling Convention

The calling convention for the user access method program. The options are:

- **C.** Standard C calls.
- **ASM.** IBM Assembler type parameter lists.
- **PLI.** PL/I user programs.

File name

The full path and file name of the data file to be processed by the user access method program. If the file is located on the local system, click the Browse button to browse to the file.

Valid characters are ASCII characters and the pound and yen sign.

String 1

A string value to pass to the user access method program at run time.

Integer 1

An integer value to pass to the user access method program at run time.

String 2

A string value to pass to the user access method program at run time.

Integer 2

An integer value to pass to the user access method program at run time.

Encoding

The encoding scheme. The options are:

- **Default.** PowerExchange uses the default encoding scheme for the operating system.
- **EBCDIC.** EBCDIC.
- **ASCII HIEND.** ASCII high endian.
- **ASCII LOEND.** ASCII little endian.

Note: The encoding scheme applies to both the record descriptors and the data in the record.

Default is **Default**.

Codepage

The code page that describes the character set for the character data in the data file. For the **Default** code page, PowerExchange uses the default code page for the operating system. Default is **Default**.

Skip First *n* Records from File

If the data file has one or more header records that are not required, the number of header records to skip.

Valid values are from 0 through 2147483647. Default is 0.

RELATED TOPICS:

- [“Code Pages” on page 58](#)
- [“Adding a Data Map” on page 62](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - Data Defaults

View or edit data default properties for a data map.

Date Default Settings

Default date settings for century, year, month, and day. If you define a field mask and values in the mask are missing, the missing values default to these settings. For example, if you define a field mask as Y2-MM, and you provide input data of 99-04, the century and day values are missing. PowerExchange constructs the date field by using the default century and day values, which are 19 and 01 respectively, as follows:

```
19990401
```

The default date settings are:

- Century: 19
- Year: 80
- Month: 1
- Day: 1

Missing Century

Specify one of the following actions for a missing century value:

- **Default.** Complete the default action.
- **Fixed.** Check the year against the **Pivot** value. If the input year value is less than the **Pivot** value, the century is assumed to be 20. Otherwise, it is assumed to be 19.
- **Moveable.** Calculate a boundary value by subtracting the current year value from the input year. If the boundary is less than zero, one hundred is added to the boundary value. If the **Pivot** value is less than the boundary value, the century is assumed to be 20. Otherwise it is assumed to be 19.

Character Replacement

If data checking is enabled for CHAR fields and the data in the field fails data checking, the character in the **Value** box replaces either the first character or all characters in the field, as follows:

- **Single Char.** First character in the field.
- **All Chars.** All characters in the field.

Strict Signs in Keys

Returns data if it exactly matches the correct signing conventions of the appropriate datatype. For example, SPACKED.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - Data Checking

View or edit data checking properties for a data map.

How do you want to handle Record Mapping Errors?

Select one of the following options to specify how PowerExchange handles record mapping errors:

- **Skip row and write log file msg.** Skip the row, write a message to the log file, and continue extraction processing.
- **Terminate the data extract.** Stop extraction processing.

Default is **Terminate the data extract**.

How do you want to handle Bad Field Data?

Select one of the following options to specify how PowerExchange handle data that is not valid in fields:

- **Replace according to data defaults.** For CHAR or DATE fields, replace data by using the values on the **Data Defaults** tab in the **Data Map Properties** dialog box. You can specify whether to replace the first character or all characters in the field. PowerExchange sets values in numeric fields zeros.
- **Skip row and write log file msg.** Skip the row, write a message to the log file, and continue extraction processing.
- **Terminate the data extract.** Stop extraction processing.
- **Set to NULL.** Set the field to NULL.

Default is **Terminate the data extract**.

Check Field Types

Select the field types for which you want to perform validation. PowerExchange performs the following types of validation for the following field types:

CHAR

For EBCDIC and ASCII character sets, PowerExchange validates that the field contains only the following characters:

- 0 through 9
- A through Z
- a through z
- Greater than sign (>)
- Less than sign (<)
- Equals sign (=)
- Space ()

DATE

PowerExchange validates that the field uses the PowerExchange-defined DATE format.

NUMCHAR

PowerExchange validates that the field contains only the following characters:

- 0 through 9
- Asterisk (*)
- Comma (,)
- Decimal point (.)
- Dollar sign (\$)
- Forward slash (/)
- Minus sign (-)
- Plus sign (+)
- Space ()

PACKED

PowerExchange validates that the field contains only standard packed decimal format (S9 COMP-3) characters.

Caution: If you do not enable data checking for this field type, and any half byte of data other than the last half byte (which contains the sign) contains a value other than 0-9, PowerExchange interprets the field as having a NULL value. If the session writes to a nonrelational target, and the data map for the target does not allow NULL values, the session fails.

For example, if data checking of packed decimals is not enabled, and a 2-byte packed decimal field in the source data is defined as PIC S9(3) COMP-3, PowerExchange passes the following values to PowerCenter:

- X'FFFF' - NULL is passed to PowerCenter
- X'0000' - 0 is passed to PowerCenter
- X'4040' - 404 is passed to PowerCenter
- X'F00C' - NULL is passed to PowerCenter

To prevent the session from failing, enable data checking for packed decimal fields, and select the appropriate option for your situation at **How do you want to handle Bad Field data?**

TIME

PowerExchange validates that the field uses the PowerExchange-defined TIME format.

TIMESTAMP

PowerExchange validates that the field uses the PowerExchange-defined TIMESTAMP format.

ZONED

PowerExchange validates that the fields uses only zoned numeric, 0 through 9, and sign (-) characters.

Suppress Data Logging

Select this option to suppress data logging.

Default is cleared.

RELATED TOPICS:

- [“Data Checking” on page 60](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - Record Level Exit

View or edit properties for a record-level exit program defined for a data map.

Exit Program Name

The name of the record-level exit program to be called at run time. The program must be a valid DLL or shared library, depending on the operating system.

Calling Convention

The calling convention for the record-level exit program. The options are:

- **C.** Standard C call.
- **OS.** IBM Assembler type parameter list.
- **PLI.** PL/I user program.
- **COBOL.** COBOL user program.

Default is **C**.

String 1

String value passed to the record-level exit program at run time.

Integer 1

Integer value passed to the record-level exit program at run time.

String 2

String value passed to the record-level exit program at run time.

Integer 2

Integer value passed to the record-level exit program at run time.

RELATED TOPICS:

- [“Record-Level Exit Programs” on page 54](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Data Map Properties - Keys

View or edit key properties for an Adabas data map.

Descriptor

The descriptor for the Adabas key imported with the FDT.

Component Fields

The component field for the Adabas key imported with the FDT.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

APPENDIX E

Record, Field, and Table Properties

This appendix includes the following topics:

- [Record Properties - Name, 349](#)
- [Record Properties - Name \(Datacom\), 350](#)
- [Record Properties - Name \(IDMS\), 351](#)
- [Sort Element Details Dialog Box \(IDMS\), 353](#)
- [Record Properties - Import Details , 353](#)
- [Field Properties - Name , 354](#)
- [Field Properties - Array, 360](#)
- [Field Properties - Extra Properties , 362](#)
- [Field Properties - Field Checking , 362](#)
- [Field Properties - Code Page, 363](#)
- [Field Properties - Variable Length, 363](#)
- [Table Properties - Definition, 364](#)
- [Table Properties - Filter , 367](#)
- [Table Properties - IMS Options , 367](#)
- [Table Properties - Keys , 368](#)
- [Find Dialog Box, 369](#)

Record Properties - Name

View or edit properties for a record.

You can edit the record name. For IMS, SEQ, and VSAM records, you edit some additional properties.

Record Name

Required. A user-defined record name.

Maximum length is 256 characters.

Valid characters are ASCII characters and the pound and yen sign.

Segment Name

For IMS DL/1 batch or ODBA. Required. The name of the segment from the IMS DBD.

Variable Length

For IMS DL/1 batch or ODBA. Select this option if the specified segment is a variable-length segment.

Default is cleared.

Parent Segment Name

For IMS DL/1 batch or ODBA. The name of the parent segment.

Root Segment

For IMS DL/1 batch or ODBA. Positions the specified segment directly beneath the root segment.

Clear this option to enter a parent segment name in the **Parent Segment Name** box.

Default is cleared.

Parent Record Name

Optional for SEQ or VSAM. Select the parent record name. You cannot select the currently selected record, any children of the currently selected record, any records that redefine a base record, or any trailer or header records.

Redefines Record Name

Optional for SEQ or VSAM. Select the name of the base record that the current record redefines. You cannot select the currently selected record or any records that redefine a base record.

If the record is redefined by another record, this option is disabled.

Header/Trailer Record

Optional for SEQ or VSAM. Select **HEADER** or **TRAILER**. The caching algorithm that runs on the PowerCenter Integration Service machine uses this value to improve the performance of multiple-record write operations.

If a record is identified as being a parent, this option is disabled.

RELATED TOPICS:

- [“Adding a Data Map Record” on page 65](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Record Properties - Name (Datacom)

View or edit Datacom record properties.

Long Table Name

Required. The long name of the table. Maximum length is 256 characters. Valid characters are ASCII characters and the pound and yen sign.

Short Table Name

Required. The short name of the table. Maximum length is three characters. Valid characters are ASCII characters and the pound and yen sign.

Table ID

Required. The internal table identifier. You must enter at least 1. Default is 0.

Table Recovery

Uses the recovery facilities of CA DATACOM/DB for this table. Default is cleared.

Table Status

Required. Select the status of the data dictionary occurrence that defined this table, which is one of the following values:

- **HISTORY**
- **PROD**
- **TEST**

Default is none.

Table Version

The version of the table. You can enter a table version for tables with a status of **HISTORY** or **TEST**. Default is 1.

RELATED TOPICS:

- [“Adding a Data Map Record” on page 65](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Record Properties - Name (IDMS)

View or edit IDMS record properties.

Record Name

Required. A user-defined record name.

Maximum length is 256 characters. Valid characters are ASCII characters and the pound and yen sign.

IDMS Record Name

Required. The IDMS name for the record.

Area Name

Required. The name of the IDMS area in which the record is located.

Record ID

Optional. A unique record ID that identifies the record type within the schema.

Valid characters are ASCII characters and the pound and yen sign.

Location Mode

Optional. Select the method that IDMS uses to physically store occurrences of the record type.

The options are:

- **CALC**

- **DIRECT**
- **VIA**
- **VSAM**
- **VSAM CALC**

Duplicates

Enabled if you select **CALC** or **VSAM CALC** in the **LocationMode** list.

For **CALC**, the options are:

- **By DBKey**
- **First**
- **Last**
- **Not allowed**

For **VSAM CALC**, the options are:

- **Not allowed**
- **Unordered**

VSAM Type

Enabled if you select **VSAM CALC** in the **LocationMode** list. Identifies the record as a native VSAM record and indicates the file type of the file that contains the record.

The options are:

- **Fixed Spanned**
- **Fixed NonSpanned**
- **Variable Spanned**
- **Variable NonSpanned**

Page Group

The IDMS page group of the record.

Maximum value is 32767. Default is 0.

Radix

The IDMS radix for the record.

Valid values are from 0 through 12. Default is 8.

Minimum Root Length

The minimum root length for a compressed record. Default is 0.

Data Length

The data length of a fixed-length record for a compressed record. Default is 1.

Prefix Length

The length of the prefix for a compressed record. Default is 1.

Control Portion Length

The control portion length without the prefix for a compressed record. Default is 0.

Compressed

Select this option if IDMS stores the record in compressed format. Default is cleared.

Variable

Select this option if the record is a variable-length record. Default is cleared.

RELATED TOPICS:

- [“Adding a Data Map Record” on page 65](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Sort Element Details Dialog Box (IDMS)

View or edit sort element details for an IDMS owner record and set.

Name

The name of the field element.

Position

The position of the field element relative to the start of record. Default is 1.

Length

The length of the field element. Default is 1.

Order

The sort order. The options are:

- None.
- **ASC**. Ascending order.
- **DESC**. Descending order.

Default is none.

RELATED TOPICS:

- [“Adding a Data Map Record” on page 65](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Record Properties - Import Details

View or edit import details for a record.

Last Imported Copybook

Displays the full path and file name of the last copybook imported into the data map.

Last Imported DBD

IMS. Displays the full path and file name of the last DBD imported into the data map.

IMS Segment Processing Rules

IMS. Displays the value of the RULES keyword that you imported from the IMS DBD.

For example, if the DBD specifies RULES=(LAST), this field displays the value LAST.

RELATED TOPICS:

- [“Adding a Data Map Record” on page 65](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Field Properties - Name

View or edit field properties.

Field Name

A user-defined name for the field. The field name must be unique within the record.

When you add a column to a table based on this field, you reference this field name.

Valid characters are ASCII alphanumeric characters and the pound and yen sign. The first character must be alphabetic. Maximum length is 30 characters.

Adabas Name (Adabas only)

The Adabas name for the field.

Field Type, Precision, Scale, and Length

The field type that PowerExchange uses and the valid precision, scale, and length values for that field type. For some field types, you can edit the precision, scale, or length.

Tip: Do not edit field properties for DB2 10 for z/OS TIMESTAMP columns for which metadata was imported from a DB2 catalog or DB2 unload file.

The following table lists, for each field type, the values that you can enter for precision, scale, and length:

Field Type	Precision	Scale	Length
BIN	-	-	1 through 32767
BLOB	-	-	1 through 8000000
CHAR	-	-	1 through 32767
CLOB	-	-	1 through 8000000
DOUBLE	-	-	-
FLOAT	-	-	-
GROUP	-	-	-
NUM16	-	0 through 5	-

Field Type	Precision	Scale	Length
NUM16U	-	0 through 5	-
NUM32	-	0 through 10	-
NUM32U	-	0 through 10	-
NUM64	-	0 through 19	-
NUM64U	-	0 through 19	-
NUM8	-	0 through 3	-
NUM8U	-	0 through 3	-
NUMCHAR	1 through 100	0 through 50	1 through 100
PACKED	1 through 31	0 through 31	-
SPACKED	1 through 31	0 through 31	-
UNPACKED	1 through 31	0 through 31	-
UZONED	1 through 31	0 through 31	-
VARBIN	-	-	1 through 32767
VARCHAR	-	-	1 through 32767
ZONED	1 through 31	0 through 31	-

Note: You can specify a precision of up to 31 for certain field types, such as the PACKED field type, that are mapped to the Decimal transformation datatype in PowerCenter. However, PowerCenter supports a maximum precision of 28 for the Decimal datatype if you enable high precision in the session properties, or a maximum precision of 16 if you do not enable high precision. If you specify a precision greater than 28, PowerCenter uses a transformation datatype of Double, rather than Decimal, with a precision of 16.

For z/OS data sources, the field types correspond to COBOL PIC clauses. The following table shows the COBOL PIC clause for each field type and the precision, scale, and fixed length based on that PIC clause:

Field Type	COBOL PIC Clause	Maximum Precision	Scale	Fixed Length
BIN	X(nn)	-	-	32767
CHAR	X(nn)	-	-	32767
DOUBLE	COMP-2	-	-	8
FLOAT	COMP-1	-	-	4
GROUP	X(nn)	-	-	-

Field Type	COBOL PIC Clause	Maximum Precision	Scale	Fixed Length
NUM16	S9(4) COMP	4	0 through 5	2
NUM16	S9(4) COMP-5	4	0 through 5	2
NUM16U	9(4) COMP	4	0 through 5	2
NUM16U	9(4) COMP-5	4	0 through 5	2
NUM32	S9(9) COMP	9	0 through 10	4
NUM32	S9(9) COMP-5	9	0 through 10	4
NUM32U	9(9) COMP	9	0 through 10	4
NUM32U	9(9) COMP-5	9	0 through 10	4
NUM64	S9(18) COMP	19	0 through 19	8
NUM64	S9(18) COMP-5	19	0 through 19	8
NUM64U	9(18) COMP	20	0 through 19	8
NUM64U	9(18) COMP-5	20	0 through 19	8
NUMCHAR	9(nn)	100	0 through 50	50
PACKED ¹	S9(nn) COMP-3	31	0 through 31	-
SPACKED ²	-	31	0 through 31	-
UPACKED ³	9(nn) COMP-3	31	0 through 31	-
UZONED	9(nn)	31	0 through 31	-
VARBIN	X(nn) set by 2-byte length field	-	-	32767
VARCHAR	X(nn) set by 2-byte length field	-	-	32767
VARCHAR	X(nn) set by 2-byte length field	-	-	32767
ZONED	9(nn)	31	0 through 31	-

1. A PACKED field is a standard MVS binary decimal packed field. It contains a C, A, or F nibble to indicate a positive value or a B or D nibble to indicate a negative value. It can have an odd precision such as X'12345C' or an even precision such as X'01234C'.

2. An SPACKED field is a short number without a sign nibble. It is used to store information such as dates. It can have an odd precision such as X'012345' or an even precision such as X'1234'.

3. A UNPACKED field is an unsigned PACKED field. However, it still contains the F nibble. It can have an odd precision such as X'012345F' or an even precision such as X'01234F'.

Implied Decimal Point

Select this option for a NUMCHAR field that implies, but does not actually contain, a decimal point. By selecting this option, PowerExchange aligns the data in the field. Default is cleared.

Field Format

For CHAR, VARCHAR, ZONED, UZONED, PACKED, SPACKED, or UNPACKED field types, provides a character mask that defines the format of the input data and generates the appropriate output field. You can select a date, time, or time stamp format.

Date formats are:

- D2-MM-Y2
- D2-MM-Y4
- MM-D2-Y2
- MM-D2-Y4
- Y2-MM-D2
- Y4-MM-D2

Time formats are:

- HH-MI-SA
- HH-MI-SS

Time stamp formats are:

- Y4-MM-D2-HH.MI.SS.N6
- Y4-MM-D2-HH.MI.SS.N9
- Y4-MM-D2-HH.MI.SS.N0 to Y4-MM-D2-HH.MI.SS.N12 (DB2 10 for z/OS extended-precision time stamps)
- Y4MMD2HHMISSN6
- Y4MMD2HHMISSN9
- Y4MMD2HHMISSN0 to Y4MMD2HHMISSN12 (DB2 10 for z/OS extended-precision time stamps)

Note: Do not edit DB2 V10 for z/OS extended-precision time stamps based on a DB2 catalog or unload file that you imported for a data map.

Click the arrow to the right of the **Field Format** list to insert specific mask characters.

The following table describes the character masks in the field formats. The table also lists additional character masks that you can add to a field format:

Character Mask	Description	Valid Input Values
D2	Days.	1 through 31
D3	Julian days.	1 through 366 For example, 001 is January 1 and 032 is February 1.
HH	Hours.	-
HH12	Hours in 12-hour format.	1 through 12

Character Mask	Description	Valid Input Values
HH24	Hours in 24-hour format.	0 through 23
LA	The before noon or after noon indicator, in long format.	a.m. or p.m.
LM	Long month.	January, February, and so on
MI	Minutes.	0 through 60
MM	Months.	1 through 12
NATDATE	Adabas natural date.	-
NATTIME	Adabas natural time.	-
Nn	Number of fractional seconds.	The <i>n</i> value is a number from 1 through 6.
SA	The before noon or after noon indicator, in short format.	am or pm
SM	Short month.	The first three characters of a long month name, such as APR, SEP, and so on
SS	Seconds.	0 through 60
Y2	A 2-digit year.	00 through 99
Y4	A 4-digit year that includes the century and the year.	0000 through 9999
YYY	A year and century indicator (th or st, such as 20th or 21st) for PACKED fields.	Examples: 990823 for 23rd Aug 1999 1000914 for 14th Sep 2000 1010129 for 29th Jan 2001

The following table shows example field formats with their corresponding input and output values:

Example Format	Input	Output
D2-MM-Y2 HH:MI/N3	02-07-09 02:45/123	20090702024500123000
Y2-SM-D2	09-APR-01	20090401
HH:MI:SS LA	01:30:23 p.m.	133023
Y2D3	96366	19990401

Note: Use the following guidelines when entering character masks:

- Use alphanumeric characters in the mask if the field is nonnumeric.
- In the character mask, match the spacing and characters in the input.

- If the mask is longer than the input field, PowerExchange adds leading zeros to the input field to pad it to the length of the mask. For example, if the mask six characters long, such as D2MMY2, and the input field is five bytes long, PowerExchange adds a leading zero to pad the field to six characters.
- If any characters are missing from the mask, PowerExchange uses values from the **Data Defaults** tab on the **Data Map Properties** dialog box to fill in the missing values.

Record ID Values

Assigns one or more record ID values to a field, which PowerExchange uses in conjunction with an operator to filter the records at run time.

In the **Record ID Values** list, select the = or <> button to select the operators for filtering the data:

- Click the = button to display records with a record ID value equal to any of the specified record ID values.
- Click the <> button to display records with a record ID value that is greater than or less than (but not equal to) any of the specified record ID values.

Note: The data map stores the record IDs but not the data. Record ID fields might not appear in the column view of the data. You cannot use record ID filtering on WRITE.

To enter a hexadecimal record ID value of '0000', enter the following value for the record ID:

```
x'0000'
```

Optional Field

For a batch session with a nonrelational source and nonrelational target, select this option to allow the NRDB mapping process to handle records of varying sizes. Selecting **Optional Field** has the following effect:

- When the mapping process reads a short record with a missing optional field, the corresponding relational column is set to null in the result set.
- When a corresponding relational column is null, the mapping process sets the record length so that the optional field is not written to the target.
- For nonrelational data types that do not allow empty records, the first field cannot be an optional field. If **Optional Field** is selected, it is ignored.
- Optional fields can be defined at the end of a variable-length record.
- If a field is marked as optional, any fields that follow it in the same record should also be marked as optional

By default, the option is cleared. The NRDB mapping process expects the field to always be present in the record.

- When reading, the NRDB mapping process fails if the record length is too small to hold the non-optional field.
- When writing, the mapping process sets the record length large enough to hold the field. Default values are written when the corresponding relational column is null.

No fields can be defined as optional if the record length is defined as fixed in the data map, or if the map properties use the default file type for the operating system and that file type has a fixed length, such as a RECFM=FB file on z/OS.

An optional field must either be completely present or completely absent from the variable-length record. Mapping fails if a record boundary falls within a fixed-length field type such as ZONED.

Nullable Field

This option controls whether the relational columns that are used to populate fields on NRDB records can receive null values when the mapping process is writing the record. This option has no effect when the mapping process is reading a record.

Select this option to allow a NULL value for a field when writing to a target that uses column values from a source such as PowerCenter or Publisher. The option has the following effect:

- If the field is marked as an **Optional Field**, the incoming null value data causes a short record to be written that does not contain the optional field.
- If the field is not marked as optional, the incoming null value data causes the field to be populated with a *not available* value according to the data type, such as spaces or binary zeros.

By default, the option is cleared. The process that populates the field must always provide a value.

An attempt to write null data in a column marked as **not nullable** fails in the relational data validation process, so that the row is not sent to the NRDB writing engine.

The following properties appear for a field in a DB2 unload file data map:

Null Field Control

Enables the **Field Name** box. Default is cleared.

Field Name

If you select the **Null Field Control** option, the **Field Name** box is displayed or enables you to enter the control field that contains the null indicator for the field.

Hide from Table

Hides any columns based on this field in the **Table** window. Default is cleared.

RELATED TOPICS:

- [“Adding a Field to a Record” on page 66](#)
- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)

Field Properties - Array

View or edit field properties, which enable you to define a field as an array.

Available properties depend on the data source for the data map.

Non-Adabas Fields

View or edit the following properties for a field in a non-Adabas data map.

Array

Select this option if the field is an array. Arrays are defined in COBOL by using the OCCURS statement. Default is cleared.

If you select this option, also select one of the following options:

Fixed

Select this option if the array is a fixed-length array (OCCURS *x*). Also enter the array size in the **Array Size** box.

Variable

Select this option if the number of elements in the array varies depending on another field in the record (OCCURS *x* DEPENDING ON *field-y*). Also enter the array size in the **Array Size** box and the count size in the **Count Field** box.

Note: For VARCHAR fields, select the **Optional Field** check box on the **Name** tab in the **Field Properties** dialog box.

Default is **Fixed**.

Array Size

The size of the array. Required if the **Array** option is selected.

Default is 1.

Count Size

The count size. Required for a variable-length array.

RELATED TOPICS:

- [“Adding a Field to a Record” on page 66](#)
- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)

Adabas Fields

View or edit the following properties for a field in an Adabas data map.

Periodic Group/Multiple Value Field

Select this option if the field is an array, which is a PE or MU field.

Default is cleared.

If you select this option, also select one of the following options:

Pad Unused Values

Pads any unused values in the array with the appropriate character. For example, in a numeric field, unused values are padded with zeros.

Set Unused Values to Null

Sets any unused values in the array to NULL.

Default is **Set Unused Values to Null**.

Array Size

The size of the array. Default is 1.

Treat Array Overflow as Mapping Error

Generates an error if the actual array is larger than the value specified in the **Array Size** box.

Clear this option to return the number of elements specified in the **Array Size** box. Default is cleared.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)

Field Properties - Extra Properties

View extra properties for a field.

Usage

The field usage, such as COMP or DISPLAY for a field as specified in the imported COBOL copybook.

Level

The indent level of the field in the imported COBOL copybook.

Original Name

The original name of the field as specified in the imported COBOL copybook.

Picture

For fields in data maps for all data sources except DB2. The COBOL Picture format of the field.

DB2 Data Type

For fields in data maps for DB2 data sources. Displays the datatype of the field.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)
- [“Adding a Field to a Record” on page 66](#)

Field Properties - Field Checking

View or edit field checking properties for a field.

Select Field for checking?

Select one of the following options to specify how PowerExchange complete data checking for the field:

- **Use Data Map rules.** Use the data checking properties defined for the data map.
- **Yes.** Use field checking.
- **No.** Do not use field checking.

Default is **Use Data Map rules**.

How do you want to handle a Field error?

Select one of the following options to specify how PowerExchange handles errors in field data:

- **Use Data Map rules.** Use the data checking properties defined for the data map.

- **Replace with default values.** For CHAR fields, replace data by using the values on the **Data Defaults** tab in the **Data Map Properties** dialog box. You can specify whether to replace the first character or all characters in the field. PowerExchange sets values in numeric fields to zeros.
- **Skip field and write log file.** Skip the field, write a message to the log file, and continue extraction processing.
- **Terminate the data extract.** Stop extraction processing.
- **Set to NULL.** Set the field to NULL.

Default is **Use Data Map rules**.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)
- [“Data Checking” on page 60](#)
- [“Adding a Field to a Record” on page 66](#)

Field Properties - Code Page

View or change the code page that is selected for a field. The code page describes the character set that is used for character data in the field.

Use Data Map Default Codepage

Select this option to have PowerExchange use the default operating system code page for the field.

By default, this option is selected.

Use Selected Codepage

Select a code page for the field.

Note: Multibyte code pages are available for Datacom, DB2, VSAM KSDSs, sequential data sets, and flat files.

RELATED TOPICS:

- [“Viewing or Editing Properties for a Field in a Record” on page 83](#)
- [“Code Pages” on page 58](#)
- [“Adding a Field to a Record” on page 66](#)

Field Properties - Variable Length

View or edit properties for a variable length field.

Count Field

Enables you to specify the name of the field that contains the length value for this field. Default is cleared.

Name

The name of the field that contains the length value for the field. Enabled if you select the **Count Field** option.

Delimiter

Defines the field as a delimited field. Default is selected.

Inclusive

Includes the delimiter value in the field data. Enabled if you select the **Delimiter** option.

Exclusive

Excludes the delimiter value from the end of the field data. Default is selected. Enabled if you select the **Delimiter** option.

End of Record

If the field is the last field in the record, defines an end-of-record marker when mapping variable-length records. Enabled if you select the **Delimiter** option and the field is the last field in the record. If you select this option, the **Value** box is disabled.

Value

The delimiter value for a delimited field. Valid values are:

- A text string
- A hexadecimal value specified as:

`x'nn'`

Default value is:

`x'00'`

Enabled if you select the **Delimiter** option.

Padded

Data files for some data sources, such as DB2, might have fixed-length fields that contain variable-length data. In this case, the fields are padded. When you import this type of data, select this option to verify that PowerExchange handles padded fields correctly. For example, select this option when you import DB2 data that is sequentially accessed from an i5/OS system. Clear this option if the field is a variable-length field that is not padded. Default is cleared.

Table Properties - Definition

View or edit table definition properties.

Table Name

The table name, which is used in the SQL at extraction time.

Maximum length is 128 characters.

Column Generation

Indicates how PowerExchange is to refresh columns in a table after you change field definitions in the record on which the table is based. When you first define a table, PowerExchange derives column names from the field names in the record on which the table is based. However, PowerExchange uses a special naming convention for records that contain fields defined as arrays.

Select one of the following options:

- **Apply array format changes.** If the record on which the table is based contains changed fields that are defined as arrays, those changes are reflected in the corresponding columns in the table.
- **Refresh with missing columns.** If the record on which the table is based contains new fields, corresponding columns are added to the table.
- **Reset to defaults.** PowerExchange resets column names to the corresponding fields names in the record on which the table is based. PowerExchange discards any changed column names in the table.
- **Remove Hidden Columns.** For DB2UNLD. PowerExchange generates a new set of columns with the default values of **Hide from Table** for each field.

Generate CCK Columns

For IMS. If the record on which the table is based contains concatenated key (CCK) fields, controls whether the table displays columns for the CCK fields.

- To display CCK columns in the table, select this option.
To run a database row test on a table that includes CCK columns in the following types of IMS data maps, you must APF-authorize the following libraries:
 - If you use IMS DL/1 batch data maps with BMP for the netport job, you must APF-authorize all libraries in the BMP STEPLIB concatenation. Otherwise, PowerExchange issues message PWX-02204 with return code 12.
 - If you use IMS ODBA data maps, you must APF-authorize all libraries in the STEPLIB concatenation of the PowerExchange Listener. Otherwise, PowerExchange abends.
- To hide CCK columns from the table, clear this option. If the record on which the table is based contains CCK fields, you are prompted to confirm that you want to hide CCK columns.

Available Records

Lists the records in the data map that are not in the complex table.

To add a record to the **Record Dependencies** list, in the **Available Records** list, right-click a record and click **Add Record**.

To add a child record to a parent record:

- In the **Record Dependencies** list, select the parent record.
- In the **Available Records** list, right-click a record and click **Add Record as Child**. The record moves to the **Record Dependencies** list as a child record of the parent record.

For IDMS, you can select the system index to use for record retrieval. Right-click a record and click **Use System Index**. Then, select the index.

For Datacom, to add a record to the **Record Dependencies** list, right-click a record in the **Available Records** list, and select one of the following commands:

- **Add Record Using Table Read Sequence > Physical Sequential.** Adds a record to the data map. To retrieve data, the access method calls the Datacom GETPS routine, which retrieves records by physical sequence. However, for queries that include a WHERE clause with a key, the access method uses index sequential access.
- **Add Record Using Table Read Sequence > Index Sequential.** Adds a record to the data map. To retrieve data, the access method calls the Datacom GETIT routine, which retrieves records by using an index.

Default is **Physical Sequential**.

Record Dependencies

Lists the records that are in the complex table, with any defined hierarchical dependencies.

To remove a record dependency, right-click a record and click **Delete**.

For IDMS, to reverse the direction of the area read, right-click a record and click **Reverse Area Read**. To reverse the direction of the set read, right-click the record and click **Reverse Set Read**.

For Datacom, to change the table read sequence, right-click a record in the **Available Records** list, and select one of the following commands:

- **Table Read Sequence > Physical Sequential**
- **Table Read Sequence > Index Sequential**

How do you want to handle multiple instances of selected records?

Controls how rows appear in the database row output or are written to the target.

Select one of the following options:

- **New Row**. A new row appears or is written to the target for every instance of the record or segment.
- **Ignore**. Second and subsequent instances of a record or segment appear or are written to the target.
- **Array**. The number of records or segments specified in the **Array** list appear or are written to the target in a single row of output.

PowerExchange populates the output row until it is full, and then completes one of the following actions:

- If you clear the **New Row on Overflow** option, PowerExchange ignores subsequent records or segments.
- If you select the **New Row on Overflow** option, PowerExchange displays a new row with the overflow records or segments.

For example, for a record with five instances, if you enter 3 in the **Array** list, PowerExchange builds two output rows. The first row contains an array of three instances, and the second row contains an array of two instances.

Note: If you set a parent record or segment to **Array**, you must set all child records or segments to **Ignore**.

Fields

Controls how elements in an array or group field appear in the database row output or are written to the target.

Select one of the following options:

- To display each element in an array or group field in a single row, select the field.
- To display each element in an array or group field in a separate row, clear the field.

Note: To display fields defined as a group field or as an array in the **Fields** list, select the **Groups and Arrays only** option. To display all fields in the record in the Fields list, clear this option. This option controls the fields that appear in the **Fields** list and in the **Table Properties** dialog box, but not the fields that appear in the **Database Row Test Output** window.

Multiple Arrays in a Single Input Row

Generates multiple output rows from a single record that contains multiple arrays, or OCCURS clauses. PowerExchange sets the output fields to NULL when the data in the record is exhausted.

Enabled for tables with an imported COPYLIB with multiple OCCURS clauses.

RELATED TOPICS:

- [“Adding a Data Map Table” on page 75](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)
- [“Adding a Single-Record Data Map with an Array - Example” on page 244](#)
- [“Adding a CCK Field to an IMS Record” on page 73](#)

Table Properties - Filter

View or edit filtering criteria for a table.

Filter Column Names

Filters columns in data maps based on column name. The options are:

- **Include.** Includes columns that match the name pattern you specify.
- **Exclude.** Excludes the columns that match the name pattern.

Default is **Exclude**.

Column Name

Enter the name to use as filter criteria.

In the filter criteria, you can use the following wildcard characters:

- Asterisk (*) represents one or more matching characters.
- Question mark (?) represents a single matching character.

By default, PowerExchange excludes all columns with a name that begins with `FILLER`.

RELATED TOPICS:

- [“Adding a Data Map Table” on page 75](#)
- [“Defining Data Map Preferences” on page 43](#)
- [“Viewing or Editing Properties for a Data Map Table” on page 86](#)

Table Properties - IMS Options

View or edit IMS options for a table.

Update/Insert Options

Select one of the following options:

- **Lowest segment only.** Applies the action to the lowest segment.
- **All possible levels.** Applies the action to segment levels.

Default is **Lowest segment only**.

Update all matching segments

Updates all non-unique matching segments.

Note: This option is disabled for complex tables.

Default is cleared.

Delete Options

Select one of the following options:

- **Lowest segment only.** Applies the delete action to the lowest segment.
- **All childless segments in hierarchy.** Applies the delete action to all childless segment levels.

Warning: Do not edit the default value of the Delete Options option unless you are familiar with the IMS hierarchy. Changing this setting can result in loss of IMS data.

Default is **Lowest segment only.**

Delete all matching segments

Deletes all non-unique matching segments.

Default is cleared.

Ignore record not found error on delete

Ignores all "Record not found" errors that are generated by the delete action.

Default is cleared.

RELATED TOPICS:

- ["Adding a Data Map Table" on page 75](#)
- ["Viewing or Editing Properties for a Data Map Table" on page 86](#)
- ["Viewing or Editing IMS Options for a Table" on page 122](#)

Table Properties - Keys

View or edit key properties for an Adabas or Datacom table.

Primary Key

Select a primary key for the table.

Key Details

Displays the sequence number and column for the selected primary key.

RELATED TOPICS:

- ["Adding a Data Map Table" on page 75](#)
- ["Viewing or Editing Properties for a Data Map Table" on page 86](#)
- ["Viewing or Editing Properties for a Column in a Table" on page 87](#)

Find Dialog Box

Enter search criteria to find fields in data map records, tables in personal metadata profiles, or columns in data map tables, personal metadata profile tables, extraction maps, or capture registrations.

You can include wildcards for the search. The default wildcards are:

- An asterisk (*) represents one or more matching characters.
- A question mark (?) represents one matching character.

Tip: If the table, field, or column name contains wildcard characters in its name, precede the wildcard characters in the name with the escape character. By default, the escape character is the tilde (~).

Click the arrow to the right of any search criteria box to insert wildcards or the escape character into the search criteria.

Find what

To find a field in a data map record or a column in a data map table, personal metadata profile table, extraction map, or capture registration, enter the name of the field or column to find.

Schema

To find a table in a personal metadata profile, enter the schema name.

Mapname

To find a table in a personal metadata profile, enter the map name.

Table

To find a table in a personal metadata profile, enter the table name.

Advanced

Click **Advanced** to change the wildcards or escape character for the search.

RELATED TOPICS:

- [“Viewing Tables and Columns” on page 78](#)
- [“Searching for a Table in a Personal Metadata Profile” on page 31](#)
- [“Searching for a Column in a Table in a Personal Metadata Profile” on page 32](#)

APPENDIX F

DTL__CAPXTIMESTAMP Time Stamps

This appendix includes the following topic:

- [Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source, 370](#)

Time Stamps That Are Reported in the DTL__CAPXTIMESTAMP Field by Data Source

The time stamp that PowerExchange reports in the generated DTL__CAPXTIMESTAMP field in change records depends on the data source type and on certain parameter settings.

For PowerExchange data sources on z/OS, the `TIMESTAMP` parameter in the `UOWC CAPI_CONNECTION` controls the type of time stamp that PowerExchange reports in the `DTL__CAPXTIMESTAMP` field. If you set the `TIMESTAMP` parameter to `COMMIT`, PowerExchange reports the time stamp of the transaction commit on the source for all changes in the transaction. If you use the default parameter value of `LOG`, PowerExchange retrieves the time stamp from the source database logs. In this case, the time stamp type depends on the source type.

The following table describes the time stamps that PowerExchange reports when you use the default value of `LOG` for the `TIMESTAMP` parameter:

Data Source Type	Time Stamp Type
Adabas	The <code>HDDATE</code> time stamp from the <code>PLOG</code> block header, which indicates when the block was written. Note: In Adabas environments with a low level of update activity, the same time stamp might be reported for multiple updates that occurred at different times.
Datacom table-based CDC	The Coordinated Universal Time (UTC) time or local time when the change record was written to the Datacom <code>LXX</code> log. The <code>LOCAL_TIME</code> parameter in the <code>ECCR</code> configuration member, <code>ECCRDCMP</code> , controls whether the UTC or local time is used.
Db2 for i (i5/OS)	An i5/OS journal time stamp that reflects when the change was recorded in the journal.
Db2 for z/OS	The time at which the DB2 <code>ECCR</code> captured the change data record. Each record in a <code>UOW</code> has a different time stamp. Usually, this time stamp is a UTC value that reflects the time zone of the DB2 for z/OS system.

Data Source Type	Time Stamp Type
IDMS	The time at which the change data record was written to the IDMS log file. This time stamp is equivalent to the storeclock (STCK) time stamp. It does not reflect the local time zone.
IMS log-based CDC	The time at which the change was recorded in the IMS logs.
IMS synchronous CDC	The time at which the change occurred.
Batch VSAM and CICS/VSAM	The time at which the change record was captured. Each record in a UOW has a different time stamp. Usually, this time stamp is a UTC value.

For other data sources that do not use the UOWC CAPI_CONNECTION statement, PowerExchange determines the appropriate time stamp to report in the DTL__CAPXTIMESTAMP field. For PowerExchange Express CDC for Oracle sources, the TIME_STAMP_MODE parameter in the OPTIONS statement of the Express CDC configuration file controls the time stamp type.

The following table describes the time stamp types that PowerExchange reports for these data sources:

Data Source Type	Time Stamp Type
Db2 on Linux, UNIX, or Windows	The time stamp of the transaction commit. This time stamp is an ascending virtual time stamp (VTS) of the Db2 system, which usually corresponds to the UTC value.
Microsoft SQL Server	The time at which the change was written to the distribution database.
MySQL	The time stamp of the change event that MySQL recorded in the binary log.
PowerExchange Express CDC for Oracle	<p>The time stamp type is controlled by the TIME_STAMP_MODE parameter setting in the OPTIONS statement of the Express CDC configuration file.</p> <ul style="list-style-type: none"> - If you use the default value of LOGTIME, PowerExchange reports the time stamp of the change on source database, as recorded in the redo logs. This time stamp reflects the local time zone. - If you specify COMMITTIME, PowerExchange reports the time stamp of the transaction commit on the source database. - If you specify BEGINTIME, PowerExchange reports the time stamp of the begin UOW log record.
PostgreSQL	The time of the transaction commit.

APPENDIX G

Trace for Creating a Memory Map When Importing a COBOL Copybook

When you import a COBOL copybook into a data map from the PowerExchange Navigator, the Navigator can generate trace output that shows an enhanced data memory map.

In comparison to the data division map that the IBM IGYWC compile procedure produces, the trace memory map provides additional information and uses a format in which FILLER and REDEFINES groups are easy to identify. Use the trace when you need to perform any of the following tasks:

- Troubleshoot mapping errors from the PowerExchange Navigator machine instead of from the z/OS system
- Verify that the memory map is consistent with the data map
- Check the offsets of copybook items in storage
- Check FILLER and REDEFINES items to ensure that they're correctly mapped
- Ensure that any FILLER_REDEF_ *nn* fields that PowerExchange adds for redefinitions are consistent with padding values
- Identify any fields that do not need to be mapped
- View map information that is not available from the IBM IGYWC compiler, including COBOL levels, locations of FILLER and REDEFINES items, padding, and OCCURS values

To configure and run the trace, perform the following steps:

1. In the `dbmover.cfg` configuration file on the Windows system where the PowerExchange Navigator runs, specify the following TRACE statement:

```
TRACE=(COBDTL,0,99)
```
2. In the PowerExchange Navigator, import the COBOL copybook into a data map. PowerExchange Navigator generates the trace output in the detail log, or in an alternative log file if you specified the TRACING statement in the `dbmover.cfg` file.
3. Compare the generated trace output to your copybook to ensure that data storage is correctly mapped.

Example Trace Output for a COBOL Memory Map

If you specify the TRACE=(COBDTL,0,99) statement in the DBMOVER configuration file, the PowerExchange Navigator can generate trace output that describes the memory map for the source data record when you import a COBOL copybook into a data map in the PowerExchange Navigator.

For example, assume that you are importing the following COBOL copybook:

```

01 TESTREC1.
  02 REC1-1.
    05 ITEM-1.
      10 ITEM-1-1          PIC X(4) .
      10 ITEM-1-2          PIC X(4) .
    05 ITEM-2.
      10 ITEM-2-1          PIC X(4) .
    05 ITEM-3.
      10 ITEM-3-1          OCCURS 4 TIMES.
        15 ITEM-3-1-1      PIC XX.
    05 FILLER REDEFINES ITEM-3.
      15 ITEM-4-1-1        PIC XX.
      15 ITEM-4-1-2        PIC XX.
      15 ITEM-4-1-3        PIC XX.
      15 ITEM-4-1-4        PIC XX.
...

```

Based on this COBOL copybook, the PowerExchange Navigator generates the following trace output:

COBOL Memory Map for TESTREC1

```

1 01 TESTREC1          0CL121      pad 0      type G occurs 0      00000000 ( 0) last NEW_2
... 2 02 REC1_1        0CL42      pad 0      type G occurs 0      00000000 ( 0) last ITEM_8_1_2
..... 3 05 ITEM_1      0CL8       pad 0      type G occurs 0      00000000 ( 0) last ITEM_1_2_2
..... 4 10 ITEM_1_1     CL4        pad 0      type D occurs 0      00000000 ( 0)
..... 4 10 ITEM_1_2     CL4        pad 0      type D occurs 0      00000004 ( 4)
..... 3 05 ITEM_2      0CL4       pad 0      type G occurs 0      00000008 ( 8) last ITEM_2_1
..... 4 10 ITEM_2_1     CL4        pad 0      type D occurs 0      00000008 ( 8)
***** 3 05 ITEM_3     0CL8       pad 22     type R occurs 0      0000000C ( 12) last ITEM_3_1_1
..... 4 10 ITEM_3_1     0CL8       pad 0      type O occurs 4      0000000C ( 12) last ITEM_3_1_1
..... 5 15 ITEM_3_1_1   CL2        pad 0      type D occurs 0      0000000C ( 12)
===== 3 05 FILLER    0CL8       pad 22     type R occurs 0      0000000C ( 12) last ITEM_4_1_4
..... 4 15 ITEM_4_1_1   CL2        pad 0      type D occurs 0      0000000C ( 12)
..... 4 15 ITEM_4_1_2   CL2        pad 0      type D occurs 0      0000000E ( 14)
..... 4 15 ITEM_4_1_3   CL2        pad 0      type D occurs 0      00000010 ( 16)
..... 4 15 ITEM_4_1_4   CL2        pad 0      type D occurs 0      00000012 ( 18)
...

```

This trace output includes the following information:

- Hierarchical position followed by COBOL level. For example, in the following line, "1" is the hierarchical position and "01" is the COBOL level:


```
1 01 TESTREC1
```
- Record name. For example: TESTREC1.
- Item or group length. For example: OCL8 for a group, and CL4 for an item.
- Padding. The pad values reflect any FILLER_REDEF_ *nn* fields that PowerExchange adds to the data map when a REDEFINE has multiple redefinitions of varying lengths. The padding and offsets should be consistent.
- Type of item: G for group, D for display, or R for a REDEFINES group.

Note: The first group of redefinitions is preceded by a series of asterisks (*). Subsequent redefinition groups are preceded by equals (=) signs.
- OCCURS value, or 0 if no OCCURS clause is defined.
- Offset in hexadecimal and decimal formats. The decimal offset is enclosed in parentheses.
- Last dependent item in a group.

Compare the trace output to the following COBOL compiler output:

Source LineID	Hierarchy and Data Name	Base Locator	Displacement Structure	Asmblr Data Definition	Data Type	Data Def Attributes
2	PROGRAM-ID UAMC-----*					
17	1 TESTREC1		000000000	DS 0CL121	Group	
18	2 REC1-1		000000000	DS 0CL42	Group	
19	3 ITEM-1		000000000	DS 0CL8	Group	
20	4 ITEM-1-1		000000000	DS 4C	Display	
21	4 ITEM-1-2		000000004	DS 4C	Display	
22	3 ITEM-2		000000008	DS 0CL4	Group	
23	4 ITEM-2-1		000000008	DS 4C	Display	
24	3 ITEM-3		00000000C	DS 0CL8	Group	
25	4 ITEM-3-1		00000000C	DS 0CL2	Group	O
26	5 ITEM-3-1-1		00000000C	DS 2C	Display	
27	3 FILLER		00000000C	DS 0CL8	Group	R
28	4 ITEM-4-1-1		00000000C	DS 2C	Display	
29	4 ITEM-4-1-2		00000000E	DS 2C	Display	
30	4 ITEM-4-1-3		000000010	DS 2C	Display	
31	4 ITEM-4-1-4		000000012	DS 2C	Display	

Notice that the trace provides the following additional information: COBOL level, padding, OCCURS value, offset in decimal format, and name of the last item in a group. It also clearly identifies redefinition items.

INDEX

- 2-tier names
 - data maps [50](#)
- 3-tier names
 - data maps [50](#)

A

- access method for IMS
 - configuration tasks for DL/I batch or BMP access to IMS data [116](#)
 - configuration tasks for ODBA access to IMS data [117](#)
- access method types
 - by data source [51](#), [102](#)
 - choosing for data maps [51](#)
- accessing
 - DB2 unload data from multiple tables [111](#)
- ADABAS access method
 - data map properties [328](#)
- Adabas data maps
 - adding [104](#)
 - adding keys to tables in [105](#)
 - importing field definition tables (FDTs) into [53](#), [139](#)
 - importing keys into [53](#)
 - overview [103](#)
- Adabas databases
 - data maps for [328](#)
- Adabas field definition tables (FDTs)
 - displaying in data maps [139](#)
 - importing into data maps [53](#), [139](#)
 - refreshing in data maps [139](#)
- Adabas files
 - data maps for [51](#)
- Adabas keys
 - adding to data map tables [75](#), [105](#)
 - importing into data maps [53](#)
- ADACMP copybooks
 - data map types supported in [134](#)
- adding
 - application groups [323](#)
 - before image (BI) columns to extraction map columns [189](#)
 - CALC elements to data map records (IDMS) [70](#)
 - capture registrations [163](#)
 - change indicator columns (CI) to extraction map columns [189](#)
 - columns to capture registrations [178](#)
 - columns to data map tables [76](#)
 - columns to extraction processing [190](#)
 - data map records [65](#)
 - data maps [62](#), [240](#)
 - data maps (Adabas) [104](#)
 - data maps (Datacom) [106](#)
 - data maps (DB2 catalog) [108](#)
 - data maps (DB2 unload file) [110](#), [111](#)
 - data maps (IDMS) [112](#)
 - data maps (IMS) [123](#), [125](#)
 - data maps (SEQ) [126](#)
 - data maps (VSAM) [128](#)
- adding (*continued*)
 - data maps with complex tables [246](#)
 - elements to data map records (Datacom) [67](#)
 - expressions to data map records (IDMS) [114](#)
 - expressions to data map records (IMS) [120](#)
 - extraction groups [184](#)
 - extraction maps [186](#)
 - fields to data map records [66](#), [241](#)
 - key segments to data map records (Datacom) [69](#)
 - keys to data map records (Datacom) [68](#)
 - keys to data map tables (Adabas) [75](#), [105](#)
 - owner record and sets to data map records (IDMS) [70](#)
 - owner records and sets to data map records (IDMS) [71](#)
 - personal metadata profiles [206](#)
 - registration groups [158](#)
 - resource configurations [34](#)
 - search fields to data map records (IMS) [74](#)
 - tables to data maps [75](#), [243](#)
 - user-defined fields to data map records [66](#)
- application groups
 - adding [323](#)
 - deleting [326](#)
 - editing properties for [324](#)
 - overview [322](#)
 - viewing properties for [324](#)
- Application window
 - overview [19](#)
- applications
 - overview [322](#)
 - resetting extraction start point for [326](#)
- arranging windows
 - PowerExchange Navigator [20](#)
- arrays
 - defining a new row for each element in [245](#)
 - defining fields in data map records as [360](#)
 - defining in data maps [56](#)
 - defining single row for all elements in [246](#)
 - overview [57](#)
- arrays with multiple occurrences
 - adding user-defined fields to demonstrate the normalization of [261](#)
 - defining data map table properties for [259](#)
- Assembler user-defined programs
 - example [313](#)
 - linkage [313](#)
 - operating system support [313](#)
- associating
 - capture registrations with extraction maps [191](#)
 - data files with data maps [52](#), [244](#), [247](#)
 - file-list files with data maps [94](#)
 - multiple data files with data maps [52](#), [93](#)

B

- before image (BI) columns
 - adding to extraction map columns [189](#)

- before image (BI) columns (*continued*)
 - viewing for extraction maps [198](#)
- bit settings at offset in fields
 - getting for data map records [304](#)
- bulk read performance
 - improving for VSAM data sets [129](#)

C

- CALC elements in data map records (IDMS)
 - adding [70](#)
 - deleting [101](#)
 - editing properties for [85](#)
 - overview [61](#)
 - viewing properties for [85](#)
- CallProg function
 - calling in user-defined fields in data maps [263](#)
 - in an expression in a user-defined field [277](#)
- CAMAPS file [181](#)
- Capture Registration window
 - overview [19](#)
- capture registrations
 - adding [163](#)
 - adding columns to [178](#)
 - changing for extraction maps [191](#)
 - condense options [169](#)
 - deleting [179](#)
 - deleting all in a registration group [180](#)
 - editing properties [173](#)
 - editing to change columns in [178](#)
 - finding columns in [369](#)
 - overview [153](#)
 - removing columns from [178](#)
 - searching for columns in [31](#)
 - status options [168](#)
 - tag names for [154](#)
 - viewing columns in a registration [170](#)
 - viewing properties [173](#)
- capture registrations (Datacom)
 - viewing elements in [178](#)
- capture registrations (IDMS)
 - viewing parent hierarchy for base records in [178](#)
- CAPTURE_NODE_EPWD parameter
 - generating encrypted password for [46](#)
- CAPX advanced parameters
 - database row tests for an i5/OS source [230](#)
 - general parameters for database row tests [228](#)
- CAPXRT advanced parameters
 - Adabas parameters for database row tests [234](#)
 - AS400 parameters for database row tests [234](#)
 - DB2 for Linux, UNIX, and Windows parameters for database row tests [235](#)
 - general parameters for database row tests [230](#)
 - Microsoft SQL Server parameters for database row tests [235](#)
 - Oracle parameters for database row tests [235](#)
- CCK fields
 - adding to data map records (IMS) [73](#)
 - deleting from data map records (IMS DL/1 batch) [101](#)
 - deleting from data map records (IMS ODBA) [101](#)
 - editing in data map records (IMS) [73](#), [86](#)
 - viewing in data map records (IMS) [86](#)
- CCK fields in data map records (IMS DL/1 batch)
 - overview [62](#)
- CCK fields in data map records (IMS ODBA)
 - overview [62](#)
- change data capture
 - generating restart tokens for [218](#)

- change indicator (CI) columns
 - adding to extraction map columns [189](#)
 - viewing for extraction maps [198](#)
- changing
 - capture registrations for extraction maps [191](#)
 - columns in capture registrations [178](#)
 - view type for Resources window [20](#)
- Check function
 - example [285](#)
- CheckNumData function [285](#)
- choosing
 - access methods for data maps [51](#)
- CL user-defined programs
 - example [314](#)
 - linkage [314](#)
- closing
 - PowerExchange resources [32](#)
- COBOL copybooks
 - data map types supported in [134](#)
 - importing into data maps [249](#), [253](#), [259](#), [264](#)
- COBOL user-defined programs
 - operating system support [312](#)
- code pages
 - for fields in data map records [143](#), [363](#)
 - overview [58](#)
 - supported for PowerExchange [58](#)
 - supported in database row tests [213](#)
 - user-defined [59](#)
- columns in capture registrations
 - adding [178](#)
 - removing [178](#)
 - searching for [31](#)
- columns in data map tables
 - adding [76](#)
 - copying [87](#)
 - deleting [100](#)
 - editing properties for [87](#)
 - overview [55](#)
 - printing [97](#)
 - searching for [31](#)
 - selecting (DB2 catalog) [149](#)
 - viewing [78](#)
 - viewing properties for [87](#)
- columns in extraction maps
 - adding [190](#)
 - adding before image (BI) columns to [189](#)
 - adding change indicator (CI) columns to [189](#)
 - removing [190](#)
 - removing before image (BI) columns from [189](#)
 - removing change indicator (CI) columns from [189](#)
 - searching for [31](#)
 - viewing [188](#)
- columns in personal metadata profile tables
 - searching for [32](#)
- commands
 - issuing in a database row test [217](#)
- compiling
 - user exit programs [267](#)
- complex tables
 - adding to data maps [75](#)
 - data sources that support [55](#)
 - defining [125](#)
 - defining in data maps [246](#)
 - in data maps [55](#)
- Concat function
 - example [286](#)
- concatenating
 - fields in data map records [286](#)

- condense options
 - capture registrations [169](#)
- configuring
 - PowerExchange dbmover.cfg file on Windows to access IMS data [118](#)
- control interval access
 - for VSAM data sets [130](#)
- converting
 - characters to lowercase in fields in data map records [307](#)
 - characters to uppercase in fields in data map records [307](#)
- copybook types
 - supported in data maps [134](#)
- copybooks
 - ADACMP [134](#)
 - COBOL [134](#), [249](#), [253](#), [259](#), [264](#)
 - Datacom [146](#)
 - DATACOM [134](#)
 - DB2 catalog [134](#), [144](#), [148](#)
 - DBD [134](#)
 - field definition table (FDT) [134](#)
 - IDMS [134](#), [150](#)
 - importing into data map records [137](#)
 - importing into data map segments [137](#)
 - importing into data maps [135](#), [145](#)
 - importing into data maps (IMS) [124](#)
 - local details for [143](#)
 - Natural data definition module (DDM) [134](#)
 - overview [52](#), [133](#)
 - PL/I [134](#)
 - PREDICT [134](#)
 - remote details [145](#)
 - source details for importing [140](#)
 - TEXT [134](#)
- CopyData function
 - in an expression in a user-defined field [277](#)
- copying
 - field data in data map records [287](#)
 - items in data maps [87](#)
- creating
 - data maps from extraction maps [195](#)
 - file-list files [94](#)
 - shared resource configurations [35](#)
 - user exit programs [267](#)
- current configuration
 - selecting [36](#)
- cutting and pasting
 - columns in data map tables [87](#)
 - data map records [87](#)
 - data map tables [87](#)
 - fields in data map records [87](#)

D

- data
 - logon credentials for [88](#)
- data buffers
 - for VSAM data sets [129](#)
- data checking
 - data map properties [346](#)
 - for fields in data map records [362](#)
 - overview [60](#)
- data defaults
 - defining for data maps [345](#)
- data description block (DBD) copybooks
 - data map types supported in [134](#)
- data files
 - associating multiple data files with data maps [93](#)

- data files (*continued*)
 - associating multiple with data maps [52](#)
 - associating with data maps [244](#), [247](#)
 - display modes for [79](#)
 - file-list processing for [52](#)
 - viewing [79](#)
- data flow information
 - getting for data map records (IDMS) [292](#)
- data map access methods
 - overview [102](#)
 - types [51](#)
- data map properties
 - using to get RBA for VSAM data sets [131](#)
 - using to get RRN for VSAM data sets [131](#)
- data map records
 - adding [65](#), [241](#)
 - adding fields to [66](#), [241](#)
 - arrays in [246](#)
 - calling user-defined programs in [283](#)
 - concatenating fields in [286](#)
 - copying [87](#)
 - copying data from one field to another [287](#)
 - defining hierarchy of records in [247](#)
 - deleting [99](#)
 - editing properties for [82](#), [349](#), [353](#)
 - filtering data in [244](#)
 - finding fields in [369](#)
 - getting bit setting at specified offset in a field [304](#)
 - getting database keys for [291](#)
 - getting source file name for [289](#)
 - importing copybooks into [137](#)
 - printing [96](#)
 - record IDs in [247](#)
 - searching for fields in [30](#)
 - splitting data in fields in [287](#)
 - viewing properties for [82](#), [349](#)
- data map records (Datacom)
 - adding elements to [67](#)
 - properties [350](#)
 - viewing properties for elements in [83](#)
- data map records (IDMS)
 - adding expressions to [114](#)
 - getting data flow information for [292](#)
 - getting fully qualified database key for [295](#)
 - getting page group and radix for [300](#)
 - getting page groups for [299](#)
 - getting radix for [301](#)
 - getting sequence numbers for [302](#)
- data map records (IMS DL/1 batch)
 - deleting CCK fields from [101](#)
- data map records (IMS ODBA)
 - deleting CCK fields from [101](#)
- data map records (IMS)
 - adding CCK fields [73](#)
 - adding CCK fields to [73](#)
 - adding expressions to [120](#)
 - adding search fields [73](#)
 - adding search fields to [74](#)
 - CCK fields in [86](#)
 - deleting search fields from [101](#)
 - editing CCK fields in [73](#)
 - editing properties for search fields in [86](#)
 - editing search fields in [73](#)
 - getting sequence numbers for [302](#)
 - getting the RBA in [120](#)
 - viewing properties for search fields in [86](#)
- data map segments
 - importing copybooks into [137](#)

- data map table properties
 - using to normalize arrays with multiple occurrences [259](#)
- data map tables
 - adding [75](#), [243](#)
 - adding columns to [76](#)
 - complex [55](#), [75](#)
 - copying [87](#)
 - deleting [99](#)
 - deleting columns from [100](#)
 - editing properties for [86](#), [364](#)
 - filtering columns in [367](#)
 - finding columns in [369](#)
 - printing [97](#)
 - properties [261](#), [280](#)
 - refreshing with missing columns [66](#)
 - searching for columns in [31](#)
 - table definition properties [364](#)
 - viewing [78](#)
 - viewing properties for [86](#), [364](#)
- data map tables (Adabas)
 - adding keys to [75](#), [105](#)
- data map tables (DB2 catalog)
 - selecting columns in [149](#)
- data maps
 - 2-tier names [50](#)
 - 3-tier names [50](#)
 - access method types [51](#)
 - adding [62](#), [240](#)
 - adding fields to records in [66](#)
 - adding records to [65](#), [241](#)
 - adding user-defined fields to records in [66](#)
 - adding with an array [259](#), [264](#)
 - applying date mask to GROUP fields [255](#)
 - associating data files with [244](#)
 - associating file-list files with [52](#), [94](#)
 - copybook types supported in [134](#)
 - creating from extraction maps [195](#)
 - data checking [346](#)
 - data defaults [345](#)
 - data sources that are supported for [51](#)
 - defining arrays in [56](#)
 - defining preferences for [43](#)
 - deleting [98](#)
 - deleting fields from records in [99](#)
 - deleting fields from segments in [99](#)
 - deleting records from [99](#)
 - deleting segments from [99](#)
 - deleting tables from [99](#)
 - editing properties for [82](#)
 - for Adabas databases [328](#)
 - for Datacom files [330](#)
 - for DB2 databases [330](#)
 - for flat files [340](#)
 - for IDMS databases [336](#)
 - for IMS databases [334](#)
 - for MQSeries message queues [338](#)
 - for sequential data sets [340](#)
 - for specific data sources [102](#)
 - for tape data sets [342](#)
 - for user access method programs [343](#)
 - for VSAM entry-sequenced data sets (ESDSs) [335](#)
 - for VSAM keyed-sequenced data sets (KSDSs) [337](#)
 - for VSAM relative record data sets (RRDSs) [339](#)
 - importing COBOL copybooks into [249](#), [253](#), [259](#), [264](#)
 - importing copybooks into [135](#)
 - importing from remote systems [92](#)
 - importing PL/I copybooks into [256](#)
 - logon credentials for [88](#)
- data maps (*continued*)
 - merging with extraction maps [192](#)
 - names for [50](#), [62](#)
 - overview [49](#), [50](#)
 - printing [95](#)
 - record-level exit programs defined for [348](#)
 - records in [53](#)
 - segments in [53](#)
 - sending to remote systems [60](#), [90](#)
 - single-record [240](#)
 - sorting records in [81](#)
 - sorting tables in [81](#)
 - testing [213](#)
 - testing user-defined fields in [316](#)
 - USER access method [317](#)
 - viewing properties for [82](#)
- data maps (Adabas)
 - adding [104](#)
 - adding keys to tables in [105](#)
 - importing field definition tables (FDTs) [139](#)
 - importing keys into [53](#)
 - overview [103](#)
- data maps (Datacom)
 - adding [106](#)
 - adding elements to records in [67](#)
 - adding key segments to records in [67](#), [69](#)
 - adding keys to records in [67](#), [68](#)
 - deleting elements from records in [100](#)
 - deleting key segments from records in [100](#)
 - deleting keys from records in [100](#)
 - elements in records in [61](#)
 - keys in records in [61](#)
 - overview [105](#)
- data maps (DB2 catalog)
 - adding [108](#)
 - overview [107](#)
- data maps (DB2 unload file)
 - adding [110](#), [111](#)
 - overview [109](#)
- data maps (DB2)
 - editing record layout in [109](#)
 - filtering tables in [332](#), [334](#)
- data maps (IDMS)
 - adding [112](#)
 - adding CALC elements to records in [70](#)
 - adding owner records and sets to records in [71](#)
 - CALC elements in [61](#)
 - deleting CALC elements from records in [101](#)
 - deleting owner records and sets from records in [101](#)
 - editing properties for records in [351](#)
 - overview [112](#)
 - owner records and sets in [61](#)
 - sort element details for owner records and sets (IDMS) [353](#)
 - viewing properties for records in [351](#)
- data maps (IMS)
 - adding [118](#), [123](#), [125](#)
 - editing properties for [121](#)
 - editing properties for tables in [122](#), [367](#)
 - examples [123](#)
 - importing copybooks into [124](#)
 - overview [115](#)
 - sending to remote systems [121](#)
 - testing [122](#)
 - viewing hierarchy created by DBD source import [119](#)
 - viewing properties for tables in [122](#), [367](#)
 - viewing records in [119](#)
 - viewing tables in [119](#)

- data maps (SEQ)
 - adding [126](#)
 - viewing hierarchy [127](#)
- data maps (VSAM)
 - adding [128](#)
 - overview [126](#), [127](#)
 - viewing hierarchy [129](#)
- data maps, exported
 - deleting [98](#)
- data sources
 - supported for data maps [51](#), [102](#)
- database keys
 - getting for data map records [291](#)
 - getting for data map records (IDMS) [295](#)
 - getting for owner records and sets in data map records (IDMS) [294](#), [297](#)
- database row test
 - to test normalization of an array [262](#)
 - to test results of user exit program in data map record [280](#)
- Database Row Test dialog box
 - description [219](#)
- Database Row Test Output window
 - hiding [213](#)
 - showing [213](#)
- database row tests
 - CAPX advanced parameters for i5/OS source [230](#)
 - CAPX general advanced parameters [228](#)
 - CAPXRT advanced parameters on Adabas tab [234](#)
 - CAPXRT advanced parameters on AS400 tab [234](#)
 - CAPXRT advanced parameters on DB2UDB tab [235](#)
 - CAPXRT advanced parameters on General tab [230](#)
 - CAPXRT advanced parameters on MSSQL tab [235](#)
 - CAPXRT advanced parameters on Oracle tab [235](#)
 - Database Row Test dialog box [219](#)
 - DB2390IMG advanced parameters in [236](#)
 - generating restart tokens in [218](#)
 - IMSUNLD and NRDB advanced parameters in [237](#)
 - issuing PowerExchange Listener commands [217](#)
 - multibyte data in [213](#)
 - nonstandard code page data in [213](#)
 - overview [212](#)
 - running [212](#)
 - running for user-defined fields [316](#)
- DATACOM access method
 - data map properties [330](#)
- Datacom capture registrations
 - viewing elements in [178](#)
- DATACOM copybooks
 - data map types supported in [134](#)
- Datacom data map records
 - adding elements to [67](#)
 - properties [350](#)
- Datacom data maps
 - adding [106](#)
 - adding elements to records in [67](#)
 - adding key segments to records in [67](#), [69](#)
 - adding keys to records in [67](#), [68](#)
 - deleting elements from records in [100](#)
 - deleting key segments from records in [100](#)
 - deleting keys from records in [100](#)
 - elements in records in [61](#)
 - key segments in records in [61](#)
 - keys in records in [61](#)
 - overview [105](#)
- Datacom files
 - data maps for [51](#), [330](#)
- date masks
 - applying to fields in data map records [251](#)
- date masks (*continued*)
 - applying to GROUP fields in data map records [255](#)
- DB2 access method
 - data map properties [330](#)
- DB2 catalog copybooks
 - data map types supported in [134](#)
 - local details [144](#)
- DB2 catalog data maps
 - adding [108](#)
 - overview [107](#)
- DB2 data maps
 - editing record layout in [109](#)
 - filtering tables in [334](#)
- DB2 databases
 - data maps for [51](#), [330](#)
- DB2 unload data from multiple tables
 - accessing [111](#)
- DB2 unload file data maps
 - adding [110](#), [111](#)
 - overview [109](#)
- DB2 unload files
 - data maps for [51](#)
- DB2_BIN_CODEPAGE statement
 - DBMOVER configuration file [213](#)
- DB2390 IMG advanced parameters
 - in database row tests [236](#)
- DB2CODEPAGE statement
 - DBMOVER configuration file [213](#)
- DB2UNLD access method
 - data map properties [331](#)
 - DBMOVER configuration file
 - generating encrypted passwords for [46](#)
 - DBMOVER configuration statements
 - DB2_BIN_CODEPAGE [213](#)
 - DB2CODEPAGE [213](#)
- dbmover.cfg file
 - configuring on Windows to access IMS data [118](#)
- defining
 - complex tables [125](#)
 - data map preferences [43](#)
 - IDMS data map preferences [46](#)
 - layout of data in source or target [53](#)
 - resource configurations [33](#)
- deleting
 - all registrations in a registration group [180](#)
 - application groups [326](#)
 - CALC elements from data map records (IDMS) [101](#)
 - capture registrations [179](#)
 - columns from data map tables [100](#)
 - data map records [99](#)
 - data map segments [99](#)
 - data map tables [99](#)
 - data maps [98](#)
 - data maps, exported [98](#)
 - elements from data map records (Datacom) [100](#)
 - extraction groups [197](#)
 - extraction maps [196](#)
 - extraction maps in an extraction group [197](#)
 - fields from data map records [99](#)
 - fields from data map segments [99](#)
 - items in data maps [98](#)
 - key segments from data map records (Datacom) [100](#)
 - keys from data map records (Datacom) [100](#)
 - owner records and sets from data map records (IDMS) [101](#)
 - personal metadata profiles [211](#)
 - PowerExchange resources [32](#)
 - registration groups [180](#)
 - resource configurations [36](#)

- deleting (*continued*)
 - search fields from data map records (IMS) [101](#)
- display modes
 - for data files [79](#)
- displaying
 - Adabas field definition tables (FDTs) [139](#)
 - multiple output rows for items in OCCURS DEPENDING ON clauses [252](#)
- DL/1 Batch access method
 - data map properties [334](#)
- DL/I or BMP access to IMS
 - configuration tasks for ODBA access to IMS data [116](#)
- DTL__BI__columnname column
 - described [199](#)
- DTL__CAPXACTION
 - described [199](#)
- DTL__CAPXCASDELIND
 - described [199](#)
- DTL__CAPXRESTART1 column
 - described [199](#)
- DTL__CAPXRESTART2 column
 - described [199](#)
- DTL__CAPXROWID column
 - described [199](#)
- DTL__CAPXRRN column
 - described [199](#)
- DTL__CAPXTIMESTAMP column
 - described [199](#)
- DTL__CAPXTIMESTAMP field
 - types of reported time stamps by data source [370](#)
- DTL__CAPXUOW column
 - described [199](#)
- DTL__CAPXUSER column
 - described [199](#)
- DTL__CI__columnname column
 - described [199](#)
- DTL__ST column
 - described [199](#)
- DTLCAMAP file [181](#)
- DTLDSN SQL escape sequence [94](#)
- DTLFILELIST SQL escape sequence [94](#)
- DTLREXE utility -epwd option
 - generating encrypted password for [46](#)
- DTLUAPPL utility EPWD option
 - generating encrypted password for [46](#)
- DTLUCBRG utility EPWD option
 - generating encrypted password for [46](#)
- DTLUCUDB utility EPWD option
 - generating encrypted password for [46](#)
- DTLURDMO utility EPWD option
 - generating encrypted password for [46](#)
- DTLUTSK utility EPWD option
 - generating encrypted password for [46](#)

E

- editing
 - capture registrations to change columns in [178](#)
 - CCK fields in data map records (IMS) [86](#)
 - properties for search fields in data map records (IMS) [86](#)
 - record layout in data maps (DB2) [109](#)
 - resource configurations [34](#)
- editing properties
 - application groups [324](#)
 - CALC elements in data map records (IDMS) [85](#)
 - capture registrations [173](#)
 - columns in data map tables [87](#)

- editing properties (*continued*)
 - data map records [82](#), [349](#)
 - data map records (IDMS) [113](#), [351](#)
 - data map tables [86](#)
 - data map tables (IMS) [122](#), [367](#)
 - data maps [82](#)
 - data maps (IMS) [121](#)
 - definitions for data map tables [364](#)
 - extraction groups [192](#)
 - fields in data map records [83](#)
 - key segments in data map records (Datacom) [84](#)
 - keys in data map records (Datacom) [84](#)
 - keys in data map tables (Adabas) [368](#)
 - keys in data map tables (Datacom) [368](#)
 - owner records and sets in data map records (IDMS) [85](#)
 - personal metadata profiles [210](#)
 - registration groups [171](#)
- elements
 - viewing properties for in data map records (Datacom) [83](#)
- elements in data map records (Datacom)
 - adding [67](#)
 - deleting [100](#)
 - overview [61](#)
- elements in owner records and sets (IDMS)
 - properties [353](#)
- encrypted passwords
 - generating [46](#), [47](#)
- errors
 - user-defined programs [308](#)
- ESDS access method
 - data map properties [335](#)
- examples
 - adding data maps (DB2 unload file) [111](#)
 - adding data maps (single-record) [240](#)
 - adding fields to single-record data maps [241](#)
 - adding multiple-record data maps with complex tables [246](#)
 - adding multiple-segment data maps (IMS) [125](#)
 - adding records to single-record data maps [241](#)
 - adding single-record data maps [240](#)
 - adding single-segment data maps (IMS) [123](#)
 - adding tables to single-record data maps [243](#)
 - Check function [285](#)
 - Concat function [286](#)
 - file-list files (Linux) [95](#)
 - file-list files (UNIX) [95](#)
 - file-list files (Windows) [95](#)
 - file-list files (z/OS) [95](#)
 - Fragment function [288](#)
 - GenVRowKey function [289](#)
 - GetSeqWithinLevel function [302](#)
 - GetSeqWithinParent function [302](#)
 - importing COBOL copybooks into data maps [248](#)
 - importing COBOL copybooks with redefines [253](#)
 - importing PL/I copybooks into data maps [256](#)
 - IMS data maps [123](#)
 - LTrim function [303](#)
 - PowerExchange Navigator [239](#)
 - RTrim function [304](#)
 - single-record data map with an array [244](#)
 - Split function [306](#)
 - Strip function [307](#)
 - user access method programs [321](#)
 - user-defined programs [311](#)
 - user-defined programs (Assembler language) [313](#)
 - user-defined programs (C language) [311](#)
 - user-defined programs (CL language) [314](#)
 - user-defined programs (COBOL language) [312](#)
 - user-defined programs (PL/I language) [315](#)

- examples (*continued*)
 - user-defined programs (RPG language) [314](#)
- exported data maps
 - deleting [98](#)
 - importing [92](#)
 - overview [60](#)
- Expr(n) tab
 - Record window [77](#)
- expressions
 - adding to data map records (IDMS) [114](#)
 - adding to data map records (IMS) [120](#)
 - PowerExchange functions for [283](#)
- Extensions tab
 - Extract Definition window [198](#)
- extra properties
 - for fields in data map records [362](#)
- Extract Definition window
 - Extensions tab [198](#)
 - Map tab [198](#)
 - overview [19](#)
- extraction groups
 - adding [184](#)
 - deleting [197](#)
 - deleting all extraction maps in [197](#)
 - editing properties for [192](#)
 - overview [181](#)
 - viewing properties for [192](#)
- extraction map columns, PowerExchange-generated
 - DTL__BI_columname [199](#)
 - DTL__CAPXACTION [199](#)
 - DTL__CAPXCASDELIND [199](#)
 - DTL__CAPXRESTART1 [199](#)
 - DTL__CAPXRESTART2 [199](#)
 - DTL__CAPXROWID [199](#)
 - DTL__CAPXRRN [199](#)
 - DTL__CAPXTIMESTAMP [199](#)
 - DTL__CAPXUOW [199](#)
 - DTL__CAPXUSER [199](#)
 - DTL__CI_columname [199](#)
 - DTL__ST [199](#)
- extraction maps
 - adding before image (BI) columns to data columns [189](#)
 - adding change indicator (CI) columns to data columns [189](#)
 - adding extraction maps [186](#)
 - change data capture [181](#)
 - changing capture registrations for [191](#)
 - creating data maps from [195](#)
 - Database Row Test dialog box [215](#)
 - Database Row Test Output window [215](#)
 - deleting [196](#)
 - deleting all in an extraction group [197](#)
 - finding columns in [369](#)
 - merging with data maps [192](#)
 - names for [182](#)
 - overview [181](#)
 - PowerExchange-generated columns [199](#)
 - searching for columns in [31](#)
 - testing [215](#)
 - viewing [188](#)
 - viewing BI columns for [198](#)
 - viewing CI columns for [198](#)
 - viewing properties for [195](#)
- extraction processing
 - adding columns to [190](#)
 - removing columns from [190](#)
- extraction start point
 - resetting for applications [326](#)

F

- field definition table (FDT) copybooks
 - data map types supported in [134](#)
- field definition tables (FDTs)
 - displaying for data maps (Adabas) [139](#)
 - importing into data maps (Adabas) [53](#), [139](#)
 - refreshing for data maps (Adabas) [139](#)
- fields in data map records
 - adding [66](#), [241](#)
 - applying date mask to [255](#)
 - applying date masks to [251](#)
 - code pages for [363](#)
 - converting characters to lowercase [307](#)
 - converting characters to uppercase [307](#)
 - copying [87](#)
 - data checking properties for [362](#)
 - defining as arrays [360](#)
 - deleting [99](#)
 - editing properties [354](#), [360](#), [362](#)
 - editing properties for [83](#)
 - generating multiple table rows for [289](#)
 - getting contents of [285](#), [305](#)
 - getting length of [303](#)
 - getting sequence number in hierarchy [301](#)
 - overview [53](#)
 - printing [97](#)
 - searching for [30](#)
 - stripping characters from [306](#)
 - trimming characters from [303](#)
 - variable length properties for [363](#)
 - viewing [77](#)
 - viewing properties for [83](#)
- fields in data map records (Adabas)
 - editing properties [361](#)
- fields in data map records (IDMS)
 - getting sequence number in hierarchy [301](#)
- fields in data map records (IMS)
 - getting sequence number in hierarchy [301](#)
- fields in data map segments
 - deleting [99](#)
 - overview [53](#)
- fields in data map segments (IMS)
 - getting RBA values for [298](#)
- Fields tab
 - Record window [77](#)
- file-list files
 - associating with data maps [52](#), [93](#)
 - creating [94](#)
 - example (Linux) [95](#)
 - example (UNIX) [95](#)
 - example (Windows) [95](#)
 - example (z/OS) [95](#)
 - ODBC [94](#)
 - processing for [93](#)
 - syntax rules for [94](#)
- filtering
 - columns in data map tables [367](#)
 - data in data map records [244](#)
 - tables in data maps (DB2) [332](#), [334](#)
- flat files
 - data maps for [51](#), [340](#)
- Fragment function
 - example [288](#)
- functions
 - CallProg [263](#)

G

- generating
 - encrypted passwords [46, 47](#)
 - multiple table rows for fields in data map records [289](#)
- GenVRowKey function
 - example [289](#)
- GetCurrentFileName function
 - examples [290](#)
- GetDatabaseKey function
 - alias for [293](#)
 - to get the RBA [132](#)
 - to get the RRN [132](#)
- GetDataFlowType function [292](#)
- GetDbKey function [293](#)
- GetDbKeyOfFirstMember function [293](#)
- GetDbKeyOfLastMember function [293](#)
- GetDbKeyOfNextMember function [294](#)
- GetDbKeyOfOwner function [294](#)
- GetDbKeyOfPriorMember function [295](#)
- GetFullDbKey function [295](#)
- GetFullDbKeyOfFirstMember function [296](#)
- GetFullDbKeyOfLastMember function [296](#)
- GetFullDbKeyOfNextMember function [297](#)
- GetFullDbKeyOfOwner function [297](#)
- GetFullDbKeyOfPriorMember function [298](#)
- GetIMSRBByLevel function [298](#)
- GetPageGroup function [299](#)
- GetPageGroupOfOwner function [299](#)
- GetPgGrpAndRdx function [300](#)
- GetPgGrpAndRdxOfOwner function [300](#)
- GetPgGrpOfOwner function [300](#)
- GetRadix function [301](#)
- GetRadixOfOwner function [301](#)
- GetSeqWithinLevel function
 - example [302](#)
- GetSeqWithinParent function
 - example [302](#)
- getting
 - length of fields in data map records [303](#)
 - RBA values for fields in data map segments (IMS) [298](#)
 - sequence number in hierarchy for fields in data map records (IDMS) [301](#)
 - sequence number in hierarchy for fields in data map records (IMS) [301](#)

H

- hiding
 - Database Row Test Output window [213](#)
 - Resource Explorer [20](#)
 - status bar [29](#)
 - toolbar [27](#)
- hierarchy
 - defining for multiple-record data maps [247](#)
 - viewing for SEQ data maps [127](#)
 - viewing for VSAM data maps [129](#)
- hierarchy created by DBD source import
 - viewing in data maps (IMS) [119](#)

I

- IBM MQSeries message queues
 - data maps for [51](#)
- IDMS access method
 - data map properties [336](#)

- IDMS capture registrations
 - viewing parent hierarchy for base records in [178](#)
- IDMS copybooks
 - advanced properties for [150](#)
 - data map types supported in [134](#)
- IDMS data map records
 - editing properties for [113](#)
- IDMS data maps
 - adding [112](#)
 - adding CALC elements to records in [70](#)
 - adding owner record and sets to records in [70](#)
 - adding owner records and sets to records in [71](#)
 - CALC elements in [61](#)
 - defining preferences for [46](#)
 - deleting CALC elements from records in [101](#)
 - deleting owner records and sets from records in [101](#)
 - editing properties for records in [351](#)
 - overview [112](#)
 - owner records and sets in [61](#)
 - viewing properties for records in [351](#)
- IDMS databases
 - data maps for [51, 336](#)
- import details
 - for data map records [353](#)
- importing
 - COBOL copybooks with REDEFINES statements into data maps [253](#)
 - copybooks into data map records [137](#)
 - copybooks into data map segments [137](#)
 - copybooks into data maps [135](#)
 - copybooks into IMS data maps [124](#)
 - exported data maps [92](#)
 - FDTs into data maps (Adabas) [53](#)
 - field definition tables (FDTs) into data maps (Adabas) [139](#)
 - keys into data maps (Adabas) [53](#)
 - PL/I copybook into data maps [256](#)
 - PL/I copybooks into data maps [256](#)
- improving
 - bulk read performance for VSAM data sets [129](#)
- IMS access method
 - configuration tasks for DL/I batch or BMP access to IMS data [116](#)
 - configuration tasks for ODBA access to IMS data [117](#)
- IMS data maps
 - adding [118, 123, 125](#)
 - editing properties for [121](#)
 - examples [123](#)
 - importing copybooks into [124](#)
 - overview [115](#)
 - prerequisites for adding [116](#)
 - sending to remote systems [121](#)
 - testing [122](#)
 - viewing records in [119](#)
- IMS databases
 - data maps for [51, 334](#)
 - maintaining hierarchical metadata in data maps [137](#)
 - using lookup transformations to look up data in [121](#)
 - writing data to [121](#)
- IMS DL/1 batch data map records
 - CCK fields in [62](#)
 - search fields in [62](#)
- IMS hierarchy created by DBD source import
 - viewing in data maps (IMS) [119](#)
- IMS ODBA access method
 - data map properties [334](#)
- IMS ODBA data map records
 - CCK fields in [62](#)
 - search fields in [62](#)
- IMS options
 - editing for data map tables [122](#)

- IMS RBA
 - getting in data map records (IMS) [120](#)
- IMSUNLD and NRDB advanced parameters
 - in database row tests [237](#)
- index buffers
 - for VSAM data sets [130](#)
- introduction
 - PowerExchange Navigator [15](#)
- items in data maps
 - printing [95](#)

K

- key segments in data map records (Datacom)
 - adding [67](#), [69](#)
 - deleting [100](#)
 - editing properties for [84](#)
 - overview [61](#)
 - viewing properties for [84](#)
- keys
 - adding to data map tables (Adabas) [75](#), [105](#)
 - for VSAM data sets [130](#)
 - importing into data maps (Adabas) [53](#)
- keys in data map records (Datacom)
 - adding [67](#), [68](#)
 - deleting [100](#)
 - editing properties for [84](#)
 - overview [61](#)
 - viewing properties for [84](#)
- keys in data map tables (Adabas)
 - editing properties for [368](#)
 - viewing properties for [368](#)
- keys in data map tables (Datacom)
 - editing properties for [368](#)
 - viewing properties for [368](#)
- keys in data maps (Adabas)
 - properties [348](#)
- KSDS access method
 - data map properties [337](#)

L

- layout of data
 - in data source or target [53](#)
- LengthOf function [303](#)
- LISTTASK command
 - issuing in a database row test [217](#)
- local DB2 catalog details
 - for importing copybooks [144](#)
- logon credentials
 - to access data [88](#)
 - to access data maps [88](#)
- lookup transformations
 - using to look up data in IMS databases [121](#)
- LTrim function
 - example [303](#)

M

- Map tab
 - Extract Definition window [198](#)
- menu bar
 - overview [21](#)
- menu commands
 - overview [21](#)

- menu commands (*continued*)
 - PowerExchange Navigator [21](#)
- menus
 - on menu bar [21](#)
- merging
 - data maps with extraction maps [192](#)
- metadata
 - refreshing in personal metadata profiles [211](#)
- MQSeries access method
 - data map properties [338](#)
- MQSeries message queues
 - data maps for [51](#), [338](#)
- multibyte data
 - in database row tests [213](#)
- multiple-record data maps
 - defined [53](#)
 - testing [248](#)
- multiple-segment data maps
 - adding [125](#)

N

- names
 - for extraction maps [182](#)
- naming
 - data maps [50](#), [62](#)
- Natural data definition module (DDM) copybooks
 - data map types supported in [134](#)
- normalization of an array
 - testing [262](#)

O

- OCCURS clauses
 - defining arrays with [57](#)
- OCCURS DEPENDING ON clause
 - displaying multiple output rows for [252](#)
- ODBA access to IMS
 - configuration tasks for ODBA access to IMS data [117](#)
- ODBC
 - file-list files [94](#)
- opening
 - PowerExchange resources [30](#)
- overview
 - application groups [322](#)
 - applications [322](#)
 - arrays [57](#)
 - capture registrations [153](#)
 - code pages [58](#)
 - copybooks [52](#), [133](#)
 - data checking [60](#)
 - data maps [49](#), [50](#)
 - data maps (Adabas) [103](#)
 - data maps (Datacom) [105](#)
 - data maps (DB2 catalog) [107](#)
 - data maps (DB2 unload file) [109](#)
 - data maps (IDMS) [112](#)
 - data maps (IMS) [115](#)
 - data maps for specific data sources [102](#)
 - database row test [212](#)
 - database row tests [212](#)
 - exported data maps [60](#)
 - extraction groups [181](#)
 - extraction maps [181](#)
 - menu commands [21](#)
 - personal metadata profiles [205](#)

- overview (*continued*)
 - PowerExchange Navigator [15](#)
 - record-level exit programs [54](#)
 - registration groups [153](#)
 - shortcut keys [21](#)
 - user access method programs [62](#), [317](#)
 - user-defined fields [54](#)
- owner record and sets in data map records (IDMS)
 - adding [70](#)
 - sort element details for [353](#)
- owner records and sets for data map records (IDMS)
 - getting page groups for [300](#)
 - getting radix for [301](#)
- owner records and sets in data map records (IDMS)
 - adding [71](#)
 - deleting [101](#)
 - editing properties for [85](#)
 - getting database key for [294](#)
 - getting fully qualified database keys for [297](#)
 - getting page group and radix for [300](#)
 - getting page groups for [299](#)
 - overview [61](#)
 - viewing properties for [85](#)

P

- page groups
 - getting for data map records (IDMS) [299](#), [300](#)
 - getting for owner records and sets in data map records (IDMS) [299](#), [300](#)
- parameter list
 - passed to user-defined programs [309](#)
- parameter structure
 - passed to user access method programs [319](#)
- personal metadata profile tables
 - searching for columns in [32](#)
- personal metadata profiles
 - adding [206](#)
 - Database Row Test dialog box [216](#)
 - Database Row Test Output window [216](#)
 - deleting [211](#)
 - editing properties for [210](#)
 - finding tables in [369](#)
 - overview [205](#)
 - refreshing metadata in [211](#)
 - searching for a table in [31](#)
 - testing [216](#)
 - viewing metadata for tables in [209](#)
 - viewing properties for [210](#)
 - viewing properties for tables in [210](#)
- personal metadata tables
 - viewing properties for [209](#)
- PIC G fields in COBOL copybooks
 - code pages for [143](#)
- PL/I copybooks
 - data map types supported in [134](#)
 - importing into data maps [256](#)
- PL/I user-defined programs
 - operating system support [315](#)
- PowerExchange
 - code pages supported [58](#)
 - resources [29](#)
- PowerExchange functions
 - for expressions in user-defined fields [283](#)
 - for user-defined fields [282](#)
- PowerExchange Listener commands
 - issuing in a database row test [217](#)

- PowerExchange Navigator
 - Application window [19](#)
 - arranging windows [20](#)
 - Capture Registration window [19](#)
 - examples [239](#)
 - Extract Definition window [19](#)
 - interface [17](#)
 - introduction [15](#)
 - menu bar [21](#)
 - overview [15](#)
 - Record window [19](#)
 - Resource Explorer [19](#)
 - Resource Inspector [19](#)
 - Resources window [19](#)
 - shortcut keys [28](#)
 - shortcut menus [25](#)
 - status bar [29](#)
 - Table window [19](#)
 - title bar [21](#)
 - toolbar [27](#)
- PowerExchange resources
 - closing [32](#)
 - deleting [32](#)
 - opening [30](#)
 - searching for [30](#)
- PowerExchange-generated extraction map columns
 - DTL__BI_columnname [199](#)
 - DTL__CAPXACTION [199](#)
 - DTL__CAPXCASDELIND [199](#)
 - DTL__CAPXRESTART1 [199](#)
 - DTL__CAPXRESTART2 [199](#)
 - DTL__CAPXROWID [199](#)
 - DTL__CAPXRRN [199](#)
 - DTL__CAPXTIMESTAMP [199](#)
 - DTL__CAPXUOW [199](#)
 - DTL__CAPXUSER [199](#)
 - DTL__CI_columnname [199](#)
 - DTL__ST [199](#)
- PREDICT copybooks
 - data map types supported in [134](#)
- preferences
 - defining for data maps [43](#)
- Preferences - IDMS Properties dialog box [46](#)
- prerequisites for adding data maps
 - IMS data maps [116](#)
- printing
 - columns in data map tables [97](#)
 - data map records [96](#)
 - data map tables [97](#)
 - data maps [95](#)
 - fields in data map records [97](#)
- properties
 - application groups [324](#)
 - CALC elements in data map records (IDMS) [85](#)
 - capture registrations [173](#)
 - columns in data map tables [87](#)
 - data checking for data maps [346](#)
 - data defaults for data maps [345](#)
 - data map records [82](#), [349](#), [353](#)
 - data map records (Datacom) [350](#)
 - data map records (IDMS) [113](#), [351](#)
 - data map tables [86](#)
 - data map tables (IMS) [367](#)
 - data maps [82](#)
 - data maps (ADABAS access method) [328](#)
 - data maps (DATACOM access method) [330](#)
 - data maps (DB2 access method) [330](#)
 - data maps (DB2 unload file) [331](#)

- properties (*continued*)
 - data maps (DL/1 Batch access method) [334](#)
 - data maps (ESDS access method) [335](#)
 - data maps (flat files) [340](#)
 - data maps (IDMS access method) [336](#)
 - data maps (IMS ODBA access method) [334](#)
 - data maps (KSDS access method) [337](#)
 - data maps (MQSeries access method) [338](#)
 - data maps (RRDS access method) [339](#)
 - data maps (sequential data sets) [340](#)
 - data maps (tape data set) [342](#)
 - data maps (USER access method) [343](#)
 - definitions for data map tables [364](#)
 - elements in owner records and sets (IDMS) [353](#)
 - extraction groups [192](#)
 - extraction maps [195](#)
 - fields in data map records [83](#), [354](#), [360](#), [362](#), [363](#)
 - fields in data map records (Adabas) [361](#)
 - filtering criteria for columns in data map tables [367](#)
 - key segments in data map records (Datacom) [84](#)
 - keys in data map records (Datacom) [84](#)
 - keys in data map tables (Adabas) [368](#)
 - keys in data maps (Adabas) [348](#)
 - owner records and sets in data map records (IDMS) [85](#)
 - personal metadata profile tables [210](#)
 - personal metadata profiles [210](#)
 - record-level exit programs defined for data maps [348](#)
 - registration groups [171](#)
- pwxcmd -epassword option
 - generating encrypted password for [46](#)

R

- radix
 - getting for data map records (IDMS) [300](#), [301](#)
 - getting for owner records and sets in data map records (IDMS) [300](#), [301](#)
- record IDs
 - in data map records [247](#)
- record layout
 - editing in data maps (DB2) [109](#)
- Record window
 - Expr(n) tab [77](#)
 - fields in [77](#)
 - Fields tab [77](#)
 - overview [19](#)
 - records in [77](#)
- record-level exit programs
 - overview [54](#)
 - properties [348](#)
- records in data maps
 - adding [65](#)
 - adding fields to [66](#)
 - deleting [99](#)
 - deleting fields from [99](#)
 - editing properties for [82](#)
 - editing properties for fields in [83](#)
 - importing copybooks into [137](#)
 - overview [53](#)
 - printing [96](#)
 - printing fields in [97](#)
 - record-level exit programs for [54](#)
 - sorting [81](#)
 - user-defined fields in [66](#)
 - viewing [77](#)
 - viewing fields in [77](#)
 - viewing properties for [82](#)

- records in data maps (*continued*)
 - viewing properties for fields in [83](#)
- records in data maps (Datacom)
 - adding elements to [67](#)
 - adding items to [67](#)
 - adding key segments to [69](#)
 - adding keys to [68](#)
 - deleting elements from [100](#)
 - deleting key segments from [100](#)
 - deleting keys from [100](#)
 - editing properties for key segments in [84](#)
 - editing properties for keys in [84](#)
 - viewing properties for key segments in [84](#)
 - viewing properties for keys in [84](#)
- records in data maps (IDMS)
 - adding CALC elements to [70](#)
 - adding items to [70](#)
 - deleting CALC elements from [101](#)
 - editing properties for CALC elements in [85](#)
 - viewing properties for CALC elements in [85](#)
- records in data maps (IMS)
 - viewing [119](#)
- REDEFINES statements
 - in COBOL copybooks [253](#)
- references affected by the current delete list [99](#)
- refreshing
 - Adabas field definition tables (FDTs) [139](#)
 - data map tables with missing columns [66](#)
 - metadata in personal metadata profiles [211](#)
- registration groups
 - adding [158](#)
 - adding application groups with [157](#)
 - adding extraction groups with [157](#)
 - deleting [180](#)
 - deleting all capture registrations in [180](#)
 - overview [153](#)
 - viewing properties [171](#)
- registration tag names [154](#)
- registrations groups
 - editing properties [171](#)
- relative byte address (RBA)
 - GetDatabaseKey function [132](#)
 - getting for data set records (VSAM) [131](#)
- relative record number (RRN)
 - GetDatabaseKey function [132](#)
 - getting for data set records (VSAM) [131](#)
- remote Datacom details
 - copybooks [146](#)
- remote DB2 catalog details
 - for importing copybooks [148](#)
- remote details
 - copybooks [145](#)
- remote systems
 - sending data maps to [90](#)
- removing
 - columns from capture registrations [178](#)
 - columns from extraction processing [190](#)
- resetting
 - extraction start point for applications [326](#)
- resource configurations
 - adding [34](#)
 - creating shared [35](#)
 - defining [33](#)
 - deleting [36](#)
 - editing [34](#)
 - selecting [36](#)
- Resource Explorer
 - hiding [20](#)

- Resource Explorer (*continued*)
 - overview [17, 19](#)
 - showing [20](#)
- Resource Inspector
 - overview [17, 19](#)
- resources
 - PowerExchange [29](#)
- Resources window
 - changing view type for [20](#)
 - overview [17, 19](#)
- restart tokens
 - generating for change data capture [218](#)
- return codes
 - for user-defined programs [310](#)
 - identifying for programs called with CallProg function [284](#)
 - user access method programs [321](#)
- RPG programs
 - operating system support [314](#)
- RRDS access method
 - data map properties [339](#)
- RTrim function
 - example [304](#)
- running
 - database row tests [212](#)

S

- search criteria
 - to find columns in capture registrations [369](#)
 - to find columns in data map tables [369](#)
 - to find columns in extraction maps [369](#)
 - to find columns in personal metadata profile tables [369](#)
 - to find fields in data map records [369](#)
 - to find tables in personal metadata profiles [369](#)
 - wildcards in [30](#)
- search fields
 - adding to data map records (IMS) [73, 74](#)
 - deleting from data map records (IMS) [101](#)
 - editing in data map records (IMS) [73](#)
 - editing properties for in data map records (IMS) [86](#)
 - viewing properties for in data map records (IMS) [86](#)
- search fields in data map records (IMS DL/1 batch)
 - overview [62](#)
- search fields in data map records (IMS ODBA)
 - overview [62](#)
- searching
 - for columns in capture registrations [31](#)
 - for columns in data map tables [31](#)
 - for columns in extraction maps [31](#)
 - for columns in personal metadata tables [32](#)
 - for fields in data map records [30](#)
 - for PowerExchange resources [30](#)
 - for tables in personal metadata profiles [31](#)
- segments in data maps
 - deleting [99](#)
 - deleting fields from [99](#)
 - importing copybooks into [137](#)
 - overview [53](#)
- selecting
 - columns in data map tables (DB2 catalog) [149](#)
 - resource configurations [36](#)
- sending
 - data maps to remote systems [60, 90](#)
- SEQ access method
 - data map properties [340](#)
- SEQ data maps
 - adding [126](#)

- sequence numbers
 - getting for fields in data map records (IDMS) [302](#)
 - getting for fields in data map records (IMS) [302](#)
- sequential data sets
 - data maps for [51, 340](#)
- SetBitA function [304](#)
- SetBitN function [304](#)
- shared resource configurations
 - creating [35](#)
- shortcut keys
 - overview [21](#)
 - PowerExchange Navigator [28](#)
- shortcut menus
 - PowerExchange Navigator [21, 25](#)
- showing
 - Database Row Test Output window [213](#)
 - Resource Explorer [20](#)
 - status bar [29](#)
 - toolbar [27](#)
- simple tables
 - in data maps [55](#)
- single-record data maps
 - adding [240](#)
 - adding fields to records in [241](#)
 - adding records to [241](#)
 - defined [53](#)
 - testing [243](#)
- single-segment data maps (IMS)
 - adding [123](#)
- sort element details
 - for owner records and sets (IDMS) [353](#)
- sorting
 - records in data maps [81](#)
 - tables in data maps [81](#)
- source details
 - for importing copybooks [140](#)
- source files
 - getting name of for data map records [289](#)
- Split function
 - example [306](#)
- splitting
 - field data in data map records [287](#)
- SQL escape sequences
 - DTLDSN [94](#)
 - DTLFILELIST [94](#)
- status bar
 - hiding [29](#)
 - PowerExchange Navigator [29](#)
 - showing [29](#)
- STOPTASK command
 - issuing in database row test [217](#)
- storing
 - data maps (IMS) [121](#)
- Strip function
 - example [307](#)
- syntax rules
 - for file-list files [94](#)

T

- Table window
 - columns in [78](#)
 - overview [19](#)
 - tables in [78](#)
- tables
 - adding to data maps [243](#)

- tables in data maps
 - adding [75](#)
 - adding Adabas keys to [75](#)
 - adding columns to [76](#)
 - deleting [99](#)
 - deleting columns from [100](#)
 - editing properties for [86, 364](#)
 - editing properties for columns in [87](#)
 - finding columns in [369](#)
 - overview [55](#)
 - printing [97](#)
 - printing columns in [97](#)
 - properties [261, 280](#)
 - searching for columns in [31](#)
 - sorting [81](#)
 - viewing [78](#)
 - viewing properties for [86, 364](#)
 - viewing properties for columns in [87](#)
- tables in data maps (Adabas)
 - adding keys to [105](#)
- tables in data maps (DB2)
 - filtering [332](#)
- tables in data maps (IMS)
 - editing properties for [122, 367](#)
 - viewing [119](#)
 - viewing properties for [122, 367](#)
- tables in personal metadata profiles
 - finding columns in [369](#)
 - searching for [31](#)
 - searching for columns in [32](#)
 - viewing metadata for columns in [209](#)
 - viewing properties for [210](#)
- tag names
 - for capture registrations [154](#)
- TAPE access method
 - data map properties [342](#)
- tape data sets
 - data maps for [51, 342](#)
- testing
 - data maps [213](#)
 - data maps (IMS) [122](#)
 - extraction maps [215](#)
 - multiple-record data map [248](#)
 - personal metadata profiles [216](#)
 - single-record data map [243](#)
- title bar
 - PowerExchange Navigator [21](#)
- ToLower function [307](#)
- toolbar
 - hiding [27](#)
 - PowerExchange Navigator [27](#)
 - showing [27](#)
- ToUpper function [307](#)
- trimming
 - characters from fields in data map records [303](#)

U

- UDB_CAPI_CONNECTION statement EPWD option
 - generating encrypted password for [46](#)
- USER access method
 - data map properties [343](#)
 - data maps [317](#)
- user access method programs
 - data maps for [343](#)
 - examples [321](#)
 - overview [62, 317](#)
- user access method programs (*continued*)
 - parameter structure passed to [319](#)
 - return codes [321](#)
- user exit program
 - testing [280](#)
- user exit programs
 - compiling [267](#)
 - creating [267](#)
 - creating and calling in data map records [263](#)
 - sample [267](#)
- user-defined code pages [59](#)
- user-defined fields
 - adding to data map records [66](#)
 - functions for expressions in [283](#)
 - overview [54](#)
 - PowerExchange functions for [282](#)
 - testing in data maps [316](#)
 - to demonstrate the normalization of an array [261](#)
 - with expressions [277](#)
- user-defined functions
 - CallProg [54, 283](#)
 - Check [284](#)
 - CheckNumData [285](#)
 - Concat [286](#)
 - CopyData [287](#)
 - Fragment [287](#)
 - GenVRowKey [289](#)
 - GetCurrentFileName [289, 290](#)
 - GetDatabaseKey [291](#)
 - GetDataFlowType [292](#)
 - GetDbKey [293](#)
 - GetDbKeyOfFirstMember [293](#)
 - GetDbKeyOfLastMember [293](#)
 - GetDbKeyOfNextMember [294](#)
 - GetDbKeyOfOwner [294](#)
 - GetDbKeyOfPriorMember [295](#)
 - GetFullDbKey [295](#)
 - GetFullDbKeyOfFirstMember [296](#)
 - GetFullDbKeyOfLastMember [296](#)
 - GetFullDbKeyOfNextMember [297](#)
 - GetFullDbKeyOfOwner [297](#)
 - GetFullDbKeyOfPriorMember [298](#)
 - GetIMSRBABYLevel [298](#)
 - GetPageGroup [299](#)
 - GetPageGroupOfOwner [299](#)
 - GetPgGrpAndRdx [300](#)
 - GetPgGrpAndRdxOfOwner [300](#)
 - GetPgGrpOfOwner [300](#)
 - GetRadix [301](#)
 - GetRadixOfOwner [301](#)
 - GetSeqWithinLevel [301](#)
 - GetSeqWithinParent [302](#)
 - LengthOf [303](#)
 - LTrim [303](#)
 - RTrim [303](#)
 - SetBitA [304](#)
 - SetBitN [304](#)
 - Split [305](#)
 - Strip [306](#)
 - ToLower [307](#)
 - ToUpper [307](#)
- user-defined programs
 - calling from data map records [283](#)
 - errors [308](#)
 - example C programs [311](#)
 - example (Assembler language) [313](#)
 - example (COBOL language) [312](#)
 - example (PL/I language) [314](#)

user-defined programs (*continued*)
example (RPG language) [314](#)
parameter list passed to [309](#)
return codes for [310](#)

V

variable length properties
for fields in data map records [363](#)
variable-length stream data file [340](#)
viewing
CCK fields in data map records (IMS) [86](#)
columns in data map tables [78](#)
data files [79](#)
data map records [77](#), [247](#)
data map records (IMS) [119](#)
data map tables [78](#)
elements for capture registrations (Datacom) [178](#)
extraction maps [188](#)
fields in data map records [77](#)
hierarchy created by DBD source import in data maps (IMS) [119](#)
hierarchy for SEQ data maps [127](#)
hierarchy for VSAM data maps [129](#)
parent hierarchy for base records for capture registrations (IDMS) [178](#)
properties for elements in data map records (Datacom) [83](#)
properties for search fields in data map records (IMS) [86](#)
viewing properties
application groups [324](#)
CALC elements in data map records (IDMS) [85](#)
capture registrations [173](#)
columns in data map tables [87](#)
data map records [82](#), [349](#)
data map records (IDMS) [351](#)
data map tables [86](#)
data map tables (IMS) [122](#), [367](#)
data maps [82](#)
definitions for data map tables [364](#)
extraction groups [192](#)
extraction maps [195](#)
fields in data map records [83](#)
key segments in data map records (Datacom) [84](#)
keys in data map records (Datacom) [84](#)
keys in data map tables (Datacom) [368](#)
owner records and sets in data map records (IDMS) [85](#)

viewing properties (*continued*)
personal metadata profile tables [209](#), [210](#)
personal metadata profiles [210](#)
registration groups [171](#)
VSAM data maps
adding [128](#)
overview [126](#), [127](#)
VSAM data set records
getting the RBA in [131](#), [132](#)
getting the relative byte address (RBA) in [131](#)
getting the relative record number (RRN) in [131](#)
getting the RRN in [131](#), [132](#)
VSAM data sets
control interval access [130](#)
data buffers for [129](#)
improving bulk read performance for [129](#)
index buffers for [130](#)
keys for [130](#)
VSAM entry-sequenced data sets (ESDSs)
data maps for [51](#), [335](#)
VSAM key-sequenced data sets (KSDSs)
data maps for [51](#)
VSAM keyed-sequenced data sets (KSDSs)
data maps for [337](#)
VSAM relative record data sets (RRDSs)
data maps for [51](#), [339](#)

W

wildcards
in search criteria for PowerExchange resources [30](#)
windows
Application [19](#)
Capture Registration [19](#)
Database Row Test Output [213](#), [215](#), [216](#)
Extract Definition [19](#), [188](#), [198](#)
Record [19](#), [77](#)
Resource Explorer [17](#)
Resource Inspector [17](#)
Resources [17](#), [19](#), [20](#)
Table [19](#), [78](#)
writing data
to IMS databases [121](#)