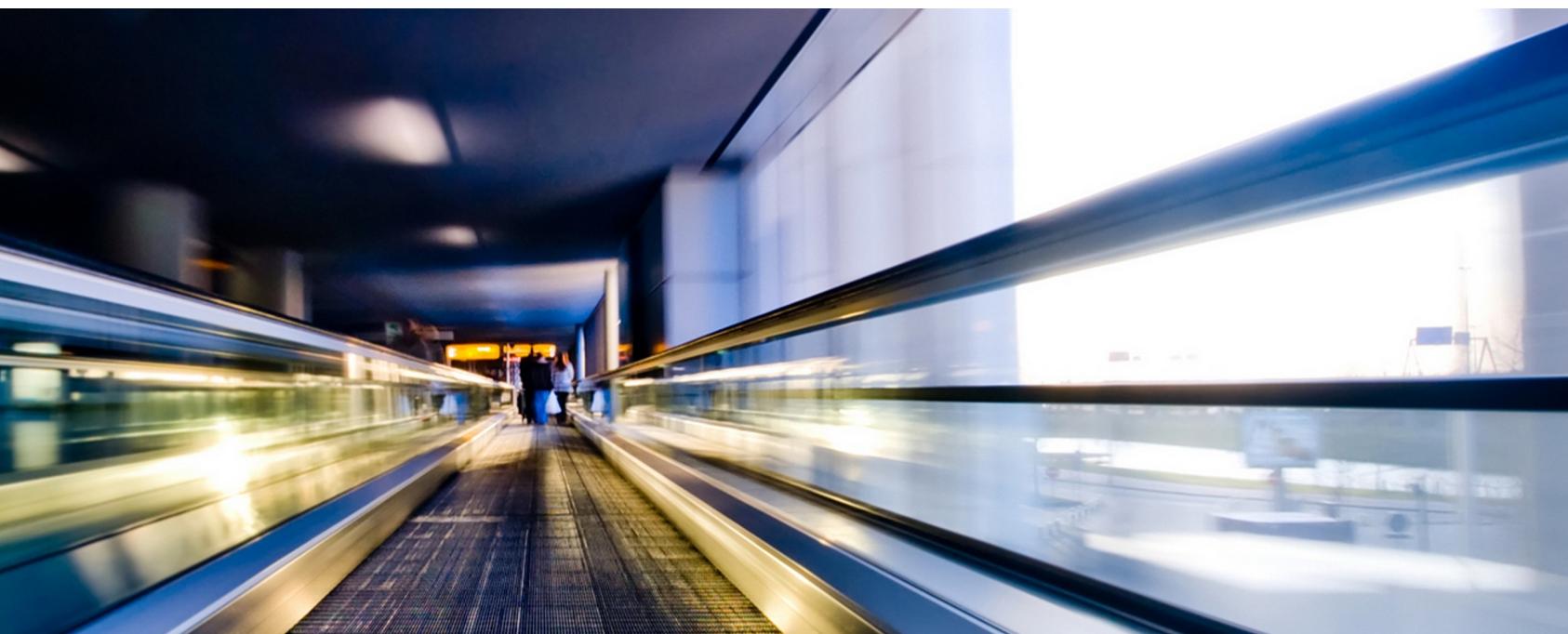




Informatica PowerExchange (Version 9.1.0 HotFix 2)

## Navigator User Guide



## Informatica PowerExchange Navigator User Guide

Version 9.1.0 HotFix 2  
September 2011

Copyright (c) 1998-2011 Informatica. All rights reserved.

This software and documentation contain proprietary information of Informatica Corporation and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica Corporation. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

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

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

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging and Informatica Master Data Management are trademarks or registered trademarks of Informatica Corporation in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Italio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright ej-technologies GmbH . All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved.

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

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

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

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

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

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

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

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

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

This product includes OSSP UUID software which is Copyright © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at [http://www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt).

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

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

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://www.asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqllicense.html>, <http://httputil.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt> , [http://www.gzip.org/zlib\\_license.html](http://www.gzip.org/zlib_license.html), <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>; <http://antlr.org/license.html>; <http://aopalliance.sourceforge.net/>; <http://www.bouncycastle.org/licence.html>; <http://www.jgraph.com/jgraphdownload.html> ; <http://www.jcraft.com/jsch/LICENSE.txt>; [http://jotm.objectweb.org/bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html); <http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>; <http://www.slf4j.org/license.html>; [http://developer.apple.com/library/mac/#samplecode/HelpHook/Listings/HelpHook\\_java.html](http://developer.apple.com/library/mac/#samplecode/HelpHook/Listings/HelpHook_java.html); <http://www.jcraft.com/jsch/LICENSE.txt>; <http://nanoxml.sourceforge.net/orig/copyright.html>; <http://www.json.org/license.html>; <http://forge.ow2.org/projects/javaservice/>; <http://www.postgresql.org/about/license.html>; <http://www.sqlite.org/copyright.html>; <http://www.tcl.tk/software/tcltk/license.html>; <http://www.jaxen.org/faq.html>; <http://www.jdom.org/docs/faq.html>; and <http://www.slf4j.org/license.html>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>) the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>) and the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>).

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

This product contains runtime modules of IBM DB2 Driver for JDBC and SQLJ (c) Copyright IBM Corporation 2006 All rights reserved.

This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178; 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,820,077; 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,254,590; 7,281,001; 7,421,458; 7,496,588; 7,523,121; 7,584,422, 7,720,842; 7,721,270; and 7,774,791 , international Patents and other Patents Pending.

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

#### NOTICES

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

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

Part Number: PWX-NUG-910-0003

# Table of Contents

<b>Preface .....</b>	<b>xi</b>
Informatica Resources . . . . .	xi
Informatica Customer Portal. . . . .	xi
Informatica Documentation. . . . .	xi
Informatica Web Site. . . . .	xii
Informatica How-To Library. . . . .	xii
Informatica Knowledge Base. . . . .	xii
Informatica Multimedia Knowledge Base. . . . .	xii
Informatica Global Customer Support. . . . .	xii
 <b>Chapter 1: PowerExchange Navigator Introduction.....</b>	<b>1</b>
PowerExchange Navigator Overview. . . . .	1
PowerExchange Navigator Interface. . . . .	2
Resource Explorer, Resources Window, Resource Inspector, and Other Windows. . . . .	3
Title Bar. . . . .	5
Menus and Commands. . . . .	5
Toolbar. . . . .	10
Shortcut Keys. . . . .	10
Status Bar. . . . .	12
PowerExchange Resources. . . . .	12
Opening a PowerExchange Resource. . . . .	12
Searching for a Resource. . . . .	13
Closing a PowerExchange Resource. . . . .	15
Deleting a PowerExchange Resource. . . . .	15
Resource Configuration. . . . .	15
Adding or Editing a Resource Configuration. . . . .	16
Creating a Shared Resource Configuration. . . . .	16
Selecting a Resource Configuration as the Current Configuration. . . . .	17
Deleting a Resource Configuration. . . . .	17
Data Map Preferences. . . . .	18
Defining Data Map Preferences. . . . .	18
Defining IDMS Data Map Preferences. . . . .	20
Encrypt Password Utility. . . . .	20
Generating an Encrypted Password. . . . .	21
 <b>Chapter 2: Data Maps.....</b>	<b>22</b>
Data Maps Overview. . . . .	22
Data Map Names. . . . .	23

Data Sources That Are Supported for Data Maps.....	24
Data Files and File-List Processing.....	25
Copybooks.....	25
Adabas Field Definition Tables and Adabas Keys.....	26
Data Map Records and Fields.....	26
Data Map Tables and Columns.....	27
Arrays.....	28
Code Pages.....	31
Data Checking.....	32
Exported Data Maps.....	33
Datacom Data Map Items.....	33
IDMS Data Map Items.....	33
IMS Data Map Items.....	34
User Access Method Programs.....	34
Adding a Data Map.....	34
Adding a Data Map Record.....	37
Adding a Field to a Record.....	37
Adding a User-Defined Field to a Record.....	38
Adding Items to a Datacom Record.....	39
Adding an Element to a Datacom Record.....	39
Adding a Key to a Datacom Record.....	39
Adding a Key Segment to a Datacom Record.....	41
Adding Items to an IDMS Record.....	41
Adding a CALC Element to an IDMS Record.....	42
Adding an Owner Record and Set to an IDMS Record.....	42
Adding Items to an IMS Record.....	44
Adding a CCK Field to an IMS Record.....	44
Adding Search Fields to an IMS Record.....	45
Adding a Data Map Table.....	46
Adding a Column to a Table.....	47
Viewing Records and Fields.....	47
Viewing Tables and Columns.....	49
Viewing a Data File.....	49
Sorting Data Map Records and Tables.....	51
Viewing or Editing Properties for a Data Map or Items in a Data Map.....	51
Viewing or Editing Properties for a Data Map.....	51
Viewing or Editing Properties for a Data Map Record.....	52
Viewing or Editing Properties for a Field in a Record.....	52
Viewing or Editing Properties for an Element in a Datacom Record.....	53
Viewing or Editing Properties for a Key in a Datacom Record.....	53
Viewing or Editing Properties for a Key Segment in a Datacom Record.....	53
Viewing or Editing Properties for a CALC Element in an IDMS Record.....	54
Viewing or Editing Properties for an Owner Record and Set in an IDMS Record.....	54

Viewing or Editing Properties for a CCK Field in an IMS Record.	55
Viewing or Editing Properties for a Search Field in an IMS Record.	55
Viewing or Editing Properties for a Data Map Table.	55
Viewing or Editing Properties for a Column in a Table.	56
Copying, Cutting, and Pasting Items in a Data Map.	56
Editing Logon Information for Accessing Data Maps and Data.	57
Sending a Data Map to a Remote System.	58
Importing Data Maps Sent to a Remote System.	59
Associating Multiple Data Files with a Data Map.	59
PowerExchange File-List Processing.	60
File-List File Syntax Rules.	60
Creating a File-List File and Associating It with a Data Map.	60
ODBC and File-List Processing.	61
File-List File Example for Linux, UNIX, and Windows.	61
File-List File Example for z/OS.	61
Printing a Data Map or Items in a Data Map.	61
Printing a Data Map.	62
Printing Data Map Records.	63
Printing Fields in a Record.	63
Printing Data Map Tables.	63
Printing Columns in a Table.	64
Deleting Data Maps or Items in a Data Map.	64
Deleting Data Maps.	64
Deleting Data Maps Exported to a Remote System.	64
Deleting Data Map Records.	65
Deleting Fields from a Record.	65
Deleting Data Map Tables.	66
Deleting Columns from a Table.	66
Deleting Elements from a Datacom Record.	66
Deleting Keys from a Datacom Record.	66
Deleting Key Segments from a Datacom Record.	67
Deleting CALC Elements from an IDMS Record.	67
Deleting Owner Records and Sets from an IDMS Record.	67
Deleting CCK Fields from an IMS Record.	67
Deleting Search Fields from an IMS Record.	67

<b>Chapter 3: Data Maps for Specific Data Sources.....</b>	<b>69</b>
Data Maps for Specific Data Sources Overview.	69
Adabas Data Maps.	70
Adding an Adabas Data Map.	71
Adding a Key to a Table in an Adabas Data Map.	72
Datacom Data Maps.	72
Adding a Datacom Data Map.	73

DB2 Catalog Data Maps . . . . .	74
Adding a DB2 Catalog Data Map . . . . .	74
Editing the Record Layout in a DB2 Data Map . . . . .	75
DB2 Unload File Data Maps . . . . .	75
Adding a DB2 Unload File Data Map . . . . .	76
Accessing DB2 Unload Data from Multiple Tables . . . . .	77
Adding a DB2 Unload File Data Map - Example . . . . .	77
IDMS Data Maps . . . . .	78
Adding an IDMS Data Map . . . . .	78
Editing IDMS Record Properties . . . . .	79
Adding Expressions to an IDMS Record . . . . .	80
IMS Data Maps . . . . .	80
Viewing IMS Data . . . . .	81
Viewing an IMS Hierarchy . . . . .	81
Prerequisites for Adding an IMS Data Map . . . . .	81
Configuring PowerExchange to Use DL/1 Batch or BMP to Access IMS Data . . . . .	82
Configuring PowerExchange to Use IMS ODBA to Access IMS Data . . . . .	82
Configuring the PowerExchange dbmover.cfg File on Windows . . . . .	83
Adding an IMS Data Map . . . . .	83
Adding Expressions to an IMS Record . . . . .	84
Getting the RBA for an IMS Record . . . . .	84
Editing Properties in an IMS Data Map . . . . .	85
Sending an IMS Data Map to a Remote System . . . . .	85
Using Lookup Transformations to Look up Data in IMS Databases . . . . .	85
Writing Data to an IMS Database . . . . .	85
Viewing or Editing IMS Options for a Table . . . . .	86
Testing an IMS Data Map . . . . .	87
IMS Data Map Examples . . . . .	87
VSAM Data Maps . . . . .	90
Adding a VSAM Data Map . . . . .	90
Improving Bulk Read Performance for VSAM Data Sets . . . . .	91
Getting the RRN or RBA for VSAM Data Set Records . . . . .	93

<b>Chapter 4: Copybooks . . . . .</b>	<b>95</b>
Copybooks Overview . . . . .	95
Copybook Types Supported in Data Maps . . . . .	96
Importing a Copybook into a Data Map . . . . .	97
Importing a Copybook into a Record or a Segment . . . . .	99
Importing an FDT into an Adabas Data Map . . . . .	100
Displaying and Refreshing the FDT for an Adabas Data Map . . . . .	101
Import Copybook - Source Details Dialog Box . . . . .	101
Import Copybook - Local Details Dialog Box . . . . .	104
Import Copybook - Local DB2 Catalog Details Dialog Box . . . . .	105

Import Copybook - Remote Details Dialog Box.	106
Import Copybook - Remote Datacom Details Dialog Box.	107
Import Copybook - Remote DB2 Catalog Details Dialog Box.	108
DB2 Column Selection Dialog Box.	109
IDMS Advanced Properties Dialog Box.	109
Import Copybook - Configuration Details Dialog Box.	110
Record Definition and Duplicate Record Definition Dialog Boxes.	110

## **Chapter 5: Registration Groups and Capture Registrations.....112**

Registration Group and Capture Registration Overview.	112
Registration Tag Names.	113
Registration Groups, Extraction Groups, and Application Groups.	115
Adding a Registration Group.	116
Adding a Capture Registration.	118
Status Options.	122
Condense Options.	123
Viewing a Capture Registration.	124
Viewing or Editing Properties for a Registration Group.	124
Viewing or Editing Properties for a Capture Registration.	125
Viewing Elements for a Datacom Capture Registration.	128
Viewing Paths for an IDMS Capture Registration.	129
Changing the Columns in a Capture Registration.	129
Deleting a Capture Registration.	129
Deleting a Registration Group.	130
Deleting All Capture Registrations in a Registration Group.	130

## **Chapter 6: Extraction Groups and Extraction Maps.....132**

Extraction Groups and Extraction Maps Overview.	132
Extraction Map Names.	133
Adding an Extraction Group.	135
Adding an Extraction Map.	137
Viewing an Extraction Map.	138
Adding or Removing Change Indicator and Before Image Columns for a Data Column.	139
Adding Columns to or Removing Columns from Extraction Processing.	140
Changing the Capture Registration Associated with an Extraction Map.	141
Merging a Data Map with an Extraction Map.	142
Viewing or Editing Properties for an Extraction Group.	143
Viewing Properties for an Extraction Map.	144
Exporting Metadata for an Extraction Map.	145
Creating a Data Map from an Extraction Map.	145
Deleting an Extraction Map.	146
Deleting an Extraction Group.	147
Deleting All Extraction Maps in an Extraction Group.	147

Extract Definition Window - Maps Tab . . . . .	147
Extract Definition Window - Extensions Tab . . . . .	148
PowerExchange-Generated Columns in Extraction Maps. . . . .	148
<b>Chapter 7: Personal Metadata.....</b>	<b>152</b>
Personal Metadata Overview. . . . .	152
Adding a Personal Metadata Profile. . . . .	152
Viewing Metadata for Tables in a Personal Metadata Profile. . . . .	155
Viewing Properties for a Table in a Personal Metadata Profile. . . . .	156
Viewing or Editing Properties for a Personal Metadata Profile. . . . .	156
Refreshing Personal Metadata. . . . .	157
Deleting a Personal Metadata Profile. . . . .	157
<b>Chapter 8: Database Row Test.....</b>	<b>158</b>
Database Row Test Overview. . . . .	158
Showing or Hiding the Database Row Test Output Window. . . . .	159
Database Row Test and Nonstandard Code Page Data. . . . .	159
Testing a Data Map. . . . .	159
Testing an Extraction Map. . . . .	160
Testing a Personal Metadata Profile. . . . .	161
Issuing PowerExchange Listener Commands in a Database Row Test. . . . .	162
Generating Restart Tokens for Change Data Capture. . . . .	163
Database Row Test Dialog Box. . . . .	164
CAPX Advanced Parameters Dialog Box. . . . .	171
CAPXRT Advanced Parameters Dialog Box. . . . .	172
DB2390IMG Advanced Parameters Dialog Box. . . . .	174
<b>Chapter 9: PowerExchange Navigator Examples.....</b>	<b>176</b>
PowerExchange Navigator Examples Overview. . . . .	176
Adding a Single-Record Data Map - Example. . . . .	177
Step 1. Add a Data Map. . . . .	177
Step 2. Add a Record to the Data Map. . . . .	178
Step 3. Add Fields to the Record in the Data Map. . . . .	178
Step 4. Add a Table to the Data Map. . . . .	179
Step 5. Test the Single-Record Data Map. . . . .	180
Adding a Single-Record Data Map with an Array - Example. . . . .	180
Step 1. Associate a Data File with a Data Map. . . . .	180
Step 2. Filter the Data in the Data Map. . . . .	181
Step 3. Create an Output Row for Each Element in an Array. . . . .	181
Step 4. Create One Output Row for All Elements in an Array. . . . .	182
Adding a Multiple-Record Data Map with a Complex Table - Example. . . . .	183
Step 1. Associate a Data File with a Multiple-Record Data Map. . . . .	183
Step 2. View Records in a Multiple-Record Data Map. . . . .	183

Step 3. Define a Hierarchy of Records in a Multiple-Record Data Map. . . . .	184
Step 4. Test a Multiple-Record Data Map. . . . .	184
Importing a COBOL Copybook into a Data Map - Example. . . . .	185
Step 1. Add a Data Map and Import a COBOL Copybook into the Data Map. . . . .	185
Step 2. Apply a Date Mask to a Field in the Data Map Record. . . . .	187
Step 3. Use a WHERE Clause to Filter an Empty Record from the Output. . . . .	188
Step 4. Display Multiple Output Rows for Items in an OCCURS DEPENDING ON Clause. . . . .	188
Importing a COBOL Copybook with REDEFINES Statements - Example. . . . .	189
Step 1. Add a Data Map and Import a COBOL Copybook with REDEFINES Statements into the Data Map. . . . .	189
Step 2. Apply a Date Mask to a GROUP Field in the Data Map. . . . .	191
Importing a PL/I Copybook - Example. . . . .	192
Step 1. Add a Data Map and Import a PL/I Copybook into the Data Map. . . . .	192
Step 2. Test the Data Map. . . . .	194
Defining Table Properties in a Data Map to Normalize an Array - Example . . . . .	194
Step 1. Add a Data Map that Contains an Array Defined with Multiple Occurrences. . . . .	195
Step 2. Add User-Defined Fields that Demonstrate the Normalization of an Array. . . . .	196
Step 3. Define Table Properties to Normalize an Array. . . . .	197
Step 4. Test the Normalization of the Array in the Data Map. . . . .	197
Calling a User Exit Program by Using the PowerExchange CallProg Function - Example . . . . .	198
Step 1. Add a Data Map. . . . .	199
Step 2. Create and Compile the User Exit Program. . . . .	201
Step 3. Add User-Defined Fields. . . . .	210
Step 4. Refresh Columns in the MASTER_REC Table . . . . .	214
Step 5. Test the Results of the User Exit Program. . . . .	214

## **Appendix A: PowerExchange Functions for User-Defined Fields. .... 216**

PowerExchange Functions for User-Defined Fields Overview. . . . .	216
PowerExchange Functions. . . . .	216
CallProg. . . . .	217
Check. . . . .	218
Concat. . . . .	219
CopyData. . . . .	219
Fragment. . . . .	220
GenVRowKey. . . . .	221
GetCurrentFileName. . . . .	222
GetDatabaseKey. . . . .	223
GetDataFlowType. . . . .	224
GetDbKey. . . . .	225
GetDbKeyOfOwner. . . . .	225
GetFullDbKey. . . . .	225
GetFullDbKeyOfOwner. . . . .	226
GetIMSRBAByLevel. . . . .	226

GetPageGroup. . . . .	227
GetPageGroupOfOwner. . . . .	227
GetPgGrpAndRdx. . . . .	228
GetPgGrpAndRdxOfOwner. . . . .	228
GetPgGrpOfOwner. . . . .	228
GetRadix. . . . .	229
GetRadixOfOwner. . . . .	229
GetSeqWithinLevel. . . . .	229
GetSeqWithinParent. . . . .	230
LengthOf. . . . .	230
LTrim. . . . .	231
RTrim. . . . .	231
SetBitA. . . . .	232
SetBitN. . . . .	232
Split. . . . .	233
Strip. . . . .	233
ToLower. . . . .	234
ToUpper. . . . .	234
Calling User-Defined Programs with the CallProg Function. . . . .	235
Load Errors for User-Defined Programs Called by the CallProg Function. . . . .	235
Parameter List Passed to User-Defined Programs. . . . .	236
User-Defined Programs Return Codes. . . . .	237
User-Defined Programs - Examples. . . . .	237
COBOL Program - Example. . . . .	238
Assembler Program - Example. . . . .	239
RPG Program - Example. . . . .	240
PL/I Program - Example. . . . .	241
Testing User-Defined Fields in a Data Map. . . . .	242
<b>Appendix B: User Access Method Programs. . . . .</b>	<b>243</b>
User Access Method Programs Overview. . . . .	243
Parameter Structure Passed to the User Access Method Program. . . . .	244
Return Codes from the User Access Method Program. . . . .	246
User Access Method Programs - Examples. . . . .	247
<b>Appendix C: Application Groups and Applications. . . . .</b>	<b>248</b>
Application Groups and Applications Overview. . . . .	248
Adding an Application Group. . . . .	249
Viewing or Editing Properties for an Application Group. . . . .	250
Resetting the Extraction Start Point for an Application. . . . .	250
Deleting an Application Group. . . . .	251

<b>Appendix D: Data Map Properties.....</b>	<b>252</b>
Data Map Properties - ADABAS Access Method . . . . .	252
Data Map Properties - DATACOM Access Method . . . . .	254
Data Map Properties - DB2 Access Method . . . . .	254
Data Map Properties - DB2UNLD Access Method . . . . .	255
DB2 Table Filter Dialog Box. . . . .	256
DB2 Table Filter - Available Tables Dialog Box. . . . .	257
Data Map Properties - DL/1 Batch or IMS ODBA Access Method . . . . .	257
Data Map Properties - ESDS Access Method . . . . .	258
Data Map Properties - IDMS Access Method . . . . .	259
Data Map Properties - ISAM Access Method . . . . .	260
Data Map Properties - KSDS Access Method . . . . .	260
Data Map Properties - MQSeries Access Method . . . . .	261
Data Map Properties - RRDS Access Method . . . . .	262
Data Map Properties - SEQ Access Method . . . . .	263
Data Map Properties - TAPE Access Method . . . . .	265
Data Map Properties - USER Access Method . . . . .	266
Data Map Properties - Data Defaults . . . . .	268
Data Map Properties - Data Checking . . . . .	268
Data Map Properties - Record Level Exit. . . . .	270
Data Map Properties - Keys . . . . .	271
<b>Appendix E: Record, Field, and Table Properties.....</b>	<b>272</b>
Record Properties - Name. . . . .	272
Record Properties - Name (Datacom). . . . .	273
Record Properties - Name (IDMS). . . . .	274
Sort Element Details Dialog Box (IDMS). . . . .	275
Record Properties - Import Details . . . . .	276
Field Properties - Name . . . . .	276
Field Properties - Array. . . . .	282
Non-Adabas Fields. . . . .	282
Adabas Fields. . . . .	283
Field Properties - Extra Properties . . . . .	283
Field Properties - Field Checking . . . . .	284
Field Properties - Code Page. . . . .	284
Field Properties - Variable Length. . . . .	285
Table Properties - Definition. . . . .	286
Table Properties - Filter. . . . .	288
Table Properties - IMS Options . . . . .	288
Table Properties - Keys . . . . .	289
Find Dialog Box. . . . .	290

<b>Index.....</b>	<b>291</b>
-------------------	------------

# Preface

This guide describes how to use the PowerExchange Navigator to define and manage capture registrations, extraction maps, data maps, and personal metadata profiles.

This guide applies to the following PowerExchange products:

- ◆ PowerExchange for Adabas®
- ◆ PowerExchange for CA Datacom®
- ◆ PowerExchange for CA IDMS™
- ◆ PowerExchange for DB2® for i5/OS® and Flat Files
- ◆ PowerExchange for DB2 for Linux®, UNIX®, and Windows®
- ◆ PowerExchange for DB2 for z/OS®
- ◆ PowerExchange for Flat Files on Linux, UNIX, and Windows
- ◆ PowerExchange for IMS™
- ◆ PowerExchange for Oracle®
- ◆ PowerExchange for SQL Server®
- ◆ PowerExchange for VSAM (including sequential data sets)

## Informatica Resources

### Informatica Customer Portal

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

### Informatica Documentation

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

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from <http://mysupport.informatica.com>.

## Informatica Web Site

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

## Informatica How-To Library

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

## Informatica Knowledge Base

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

## Informatica Multimedia Knowledge Base

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

## Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support. Online Support requires a user name and password. You can request a user name and password at <http://mysupport.informatica.com>.

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

North America / South America	Europe / Middle East / Africa	Asia / Australia
<b>Toll Free</b> Brazil: 0800 891 0202 Mexico: 001 888 209 8853 North America: +1 877 463 2435	<b>Toll Free</b> France: 0805 804632 Germany: 0800 5891281 Italy: 800 915 985 Netherlands: 0800 2300001 Portugal: 800 208 360 Spain: 900 813 166 Switzerland: 0800 463 200 United Kingdom: 0800 023 4632  <b>Standard Rate</b> Belgium: +31 30 6022 797 France: +33 1 4138 9226 Germany: +49 1805 702 702	<b>Toll Free</b> Australia: 1 800 151 830 New Zealand: 09 9 128 901  <b>Standard Rate</b> India: +91 80 4112 5738

North America / South America	Europe / Middle East / Africa	Asia / Australia
	Netherlands: +31 306 022 797 United Kingdom: +44 1628 511445	



## CHAPTER 1

# PowerExchange Navigator Introduction

This chapter includes the following topics:

- ◆ PowerExchange Navigator Overview, 1
- ◆ PowerExchange Navigator Interface, 2
- ◆ PowerExchange Resources, 12
- ◆ Resource Configuration, 15
- ◆ Data Map Preferences, 18
- ◆ Encrypt Password Utility, 20

## 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:

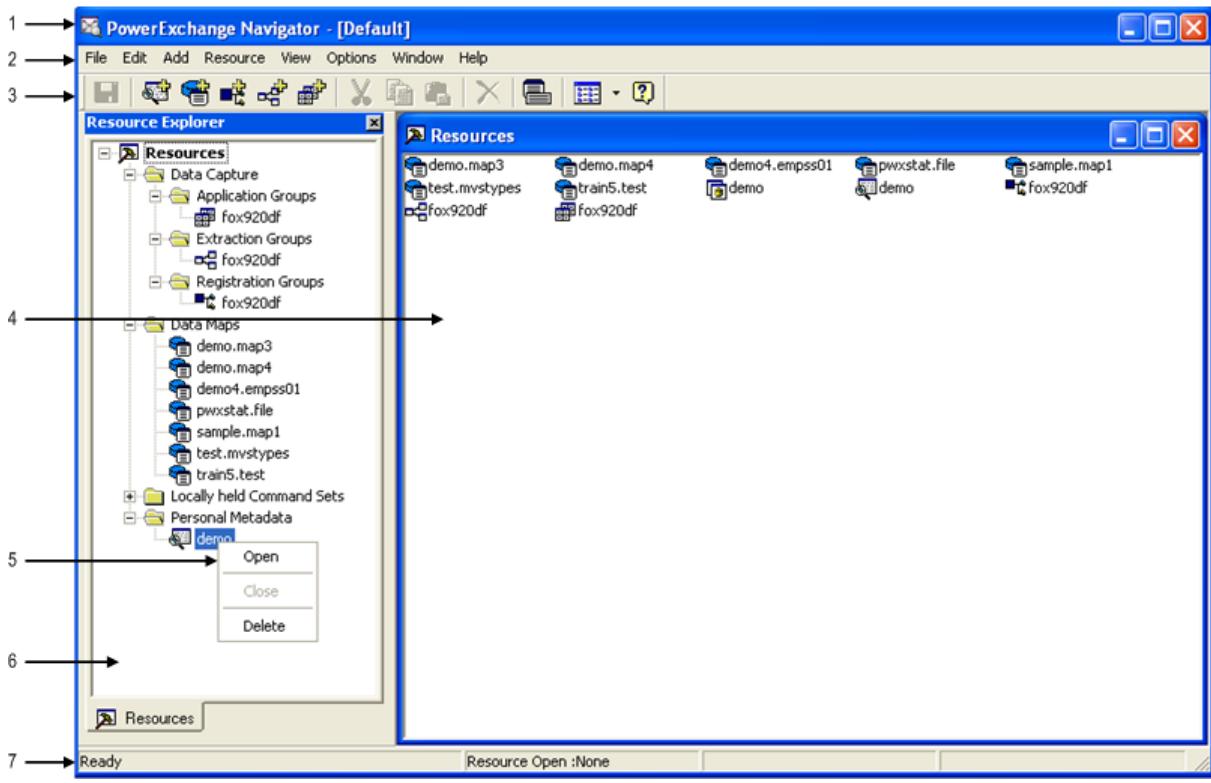
- ◆ Application groups and applications. If you use ODBC connections to extract change data, defines the extraction information for data source registered for capture. PowerExchange generates an application group when you define a registration group.
- Note:** Application groups and applications are not used if you use the PowerExchange Client for PowerCenter (PWXPC) 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.

## 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.

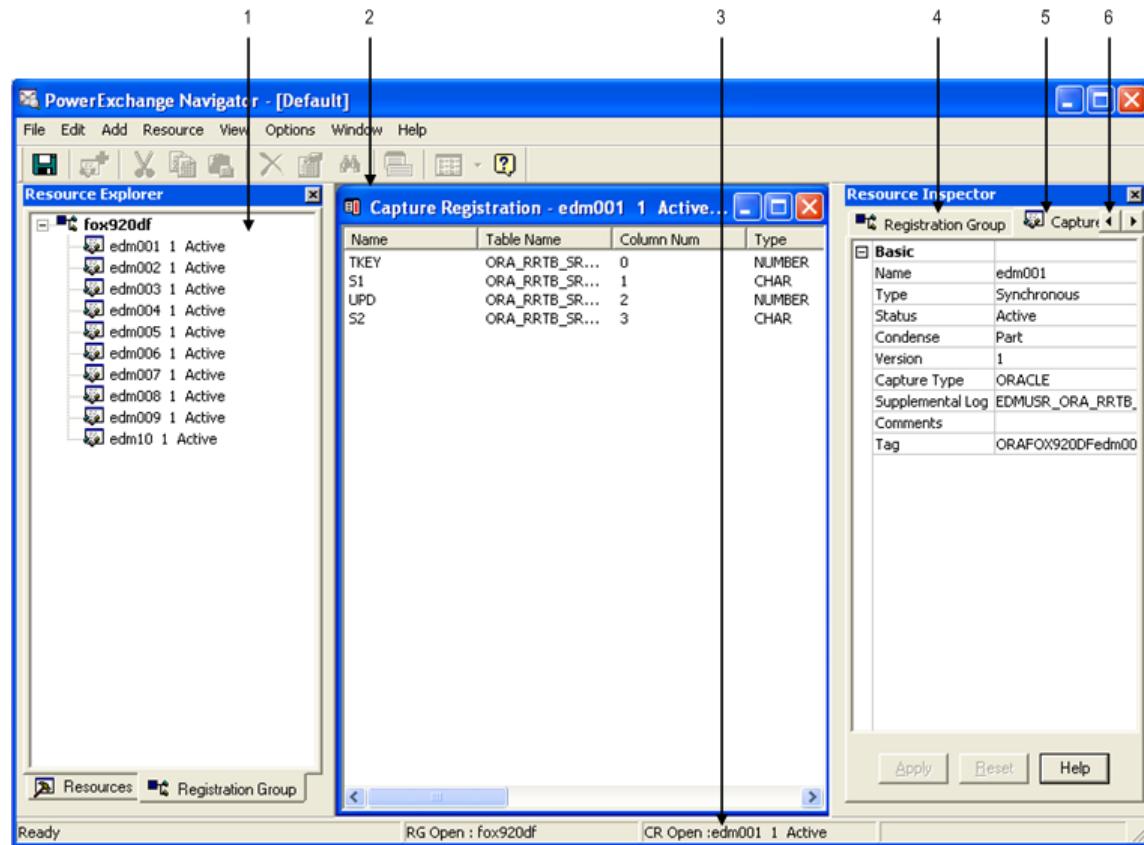
The following figure shows the **Resources** tab in the **Resource Explorer**, the **Resources** window, and the menus and toolbar that you use to define and manage resources.



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 following figure shows the following resource-specific elements in the PowerExchange Navigator GUI:

- ◆ 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**.



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 collasible 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 following table describes the commands in each menu and lists any shortcut keys that are available for the command.

Menu	Command	Description	Shortcut Key
File	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 <b>Database Row Test</b> dialog box.	
	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

Menu	Command	Description	Shortcut Key
	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 Configuration	Obsolete.	
	Exit	Exits the PowerExchange Navigator.	
Edit	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: - 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	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
	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.	
Resource	Extract Definition	Adds an extraction map to the selected extraction group.	
	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.	

Menu	Command	Description	Shortcut Key
	FDT	For an Adabas data map, displays the imported field definition table (FDT) in the <b>Adabas FDT</b> dialog box. <b>Note:</b> In the <b>Adabas FDT</b> dialog box, click <b>Refresh FDT</b> to refresh the FDT.	
Mode	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.	
View	Toolbar	Shows or hides the toolbar.	
	Status Bar	Shows or hides the status bar.	
	Resource Explorer	Shows or hides the <b>Resource Explorer</b> .	
	Output	Shows or hides the <b>Database Row Test Output</b> window.	
	Resource Inspector	Shows or hides the <b>Resource Inspector</b> .	
Options	Resource Configuration	Enables you to configure directories for PowerExchange resources.	
	Preferences	Enables you to configure preferences for data maps.	
Window	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.	
Help	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, **Application Groups**, **Extraction Groups**, and **Registration Groups**
  - **Data Maps**
  - **Personal Metadata**
- ◆ Resources in 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 in the menu bar or the shortcut menu or both. The following table describes the commands that are available from shortcut menus but not from the menu bar.

Command	Right-click	Use
<b>Application Groups and Applications</b>		
Reset to New Start Point	Open application group	Reset the extraction start point for an extraction tracked by an application.
<b>Capture Registration Groups and Capture Registrations</b>		
Amend Columns	Open capture registration	Add or remove columns from capture registrations for those sources that support selective-column capture.
View Elements	Open Datacom capture registration	View elements of a Datacom record such as offset and length. <b>Note:</b> This command is not valid for Datacom table-based CDC.
View Paths	Open IDMS capture registration	View the parent hierarchy above the IDMS base record.
<b>Data Files</b>		
Show Scale	Selected record in a data file view	Show a ruler that marks the number of characters in a record in a data file.
Hide Scale		Hide the ruler.
<b>Data Maps</b>		
Add Field After	Field name in a <b>Record</b> window	Add a field to a record after a selected field.

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

Command	Right-click	Use
Amend Change Indicator/ Before Images Extensions		Add or remove the change indicator and before image column for an extraction map.
Create Data Map		Create a data map from an extraction map.
Export Schema		Export the metadata for an extraction map in XML format. <b>Note:</b> The XML file is stored in the PowerExchange installation directory without an .xml extension.
Amend Change Indicator/ Before Images Extensions	Column in an extraction map	Add or remove the change indicator (CI) and before image (BI) columns for a column in an extraction map.
Show Auto Generated Columns		Show the PowerExchange-generated columns in an extraction map.
<b>Personal Metadata Profiles and Tables</b>		
Explore	Table in a personal metadata profile	View a table in a personal metadata profile in a <b>Table</b> window.
Refresh	Open personal metadata profile	Refresh the data that appears in a personal metadata profile.

## 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 <b>Record</b> or <b>Table</b> 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: - PowerExchange resource - Selected field in a <b>Record</b> window, or column in a <b>Table</b> 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: - 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 <b>Record</b> or <b>Table</b> window.
CTRL+Up arrow	Move a user-defined field up on the <b>Expr(n)</b> tab in a <b>Record</b> window.
CTRL+Down arrow	Move a user-defined field up on the <b>Expr(n)</b> tab in a <b>Record</b> window.
Arrow keys	Move up or down through fields and user-defined fields in a <b>Record</b> window or columns in a <b>Table</b> 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.

If you are starting PowerExchange Navigator for the first time, the **Resource Configuration - Current Configuration** dialog box appears. Click **OK** to create a resource configuration named **Default** in the following default directory:

c:/Informatica/PowerExchange.v.r.m/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.

## Adding or Editing a Resource Configuration

You can add or edit a resource configuration in the PowerExchange Navigator.

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

1. Click **Options > Resource Configuration**.
2. On the **Resource Configuration** dialog box, click the **Edit Configuration** tab.
3. Complete 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**.
4. In the **Edit Configuration** dialog box or **Add Configuration** dialog box, enter the following information:

Option	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.
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 <b>Shared Path</b> value must be different from the <b>Local Path</b> 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.
Set as current configuration	In the <b>Add Configuration</b> dialog box, sets the resource configuration as the current configuration. Default is selected.

5. Click **OK**.
6. 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:

PowerExchange Navigator - [Default]

## Creating a Shared Resource Configuration

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

The first user to access the shared data map has full access, but 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 **My Network Places**, and click **Map Network Drive**.
2. In the **Map Network Drive** dialog box, enter the following information:
  - ◆ In the **Drive** list, select the drive letter for the connection.
  - ◆ In the **Folder** box, enter or browse to the directory that contains the shared data maps.
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 **Edit Configuration** or **Add Configuration** dialog box, enter the following information:
  - ◆ In the **Shared Path** field, browse to the network drive that you mapped. 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:

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 a 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.

## 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. Click the **Data Map** tab on the **Preferences** dialog box and enter the following information:

Option	Description
Import Record Definitions on a new Data Map	Select this option to run the <b>Import Copybook</b> wizard when you add a data map. Clear this option to not run the <b>Import Copybook</b> wizard when you add a data map. Alternatively, clear the <b>Import Record Definitions</b> option on the <b>Name</b> dialog box in the <b>Add Data Map</b> wizard. Default is selected.
Import Key Fields/FDT (Adabas Only)	Select this option to run the <b>Adabas FDT Import</b> wizard when you add an Adabas data map. Clear this option to not run the <b>Adabas FDT Import</b> wizard when you add an Adabas data map. Alternatively, clear the <b>Import Key Fields/FDT</b> option on the <b>Name</b> dialog box in the <b>Add Data Map</b> 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 a map name with an appended table name, as follows: <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, a map name, and a table name, as follows: <code>select * from schema.map.table</code> <b>Note:</b> This option does not affect the naming conventions that the NRDB and NRDB2 data source types use, as follows: <ul style="list-style-type: none"><li>- NRDB uses the three-tier naming convention.</li><li>- NRDB2 uses the two-tier naming convention.</li></ul>

Option	Description
Network Operation Timeout	Enter the number of seconds that 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 keep open. The PowerExchange Navigator keeps up to this number of connections open to PowerExchange Listeners to speed up processing. Before you enter a Connection Pool Size greater than zero, verify that the value of MASTASKS in the DBMOVER file on the PowerExchange Listener machine is great 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 a column to a table: <ul style="list-style-type: none"> <li>- <b>Before current item</b></li> <li>- <b>After current item</b></li> <li>- <b>At End</b></li> </ul> The specified insertion point is used when you complete one of the following actions: <ul style="list-style-type: none"> <li>- To add a field to a record, click <b>Add &gt; Field</b>.</li> <li>- To add a column to a table, click <b>Add &gt; Column</b>.</li> </ul>
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 <b>Record</b> window: <ul style="list-style-type: none"> <li>- <b>Decimal</b>. Decimal format.</li> <li>- <b>Hex</b>. Hexadecimal format.</li> </ul>
Show Column SQL Type	Select this option to show the SQL datatypes of the columns that appear for a table in the <b>Table</b> window. Clear this option to hide the column datatypes.
Filter Column Names	Select this option to filter columns in data maps based on column name. The options are: <ul style="list-style-type: none"> <li>- <b>Include</b>. Includes columns that match the name pattern that you specify.</li> <li>- <b>Exclude</b>. Excludes the columns that match the name pattern.</li> </ul> Default is <b>Exclude</b> . Default is selected.
Column Name	Enter the name to use as filter criteria. In the filter criteria, use the following wildcard characters: <ul style="list-style-type: none"> <li>- Asterisk (*) represents one or more matching characters.</li> <li>- Question mark (?) represents a single matching character.</li> </ul> By default, PowerExchange excludes all columns with a name that begins with FILLER.
Number of Records	Enter the number of records that appear at a time in a <b>Data File</b> window. <b>Note:</b> If a data file is open in a <b>Data File</b> window, you must first close the data file and data map before editing this value for it to take effect. Default is 10 records.
Display Format	Select the display format for records in the data file view, which is one of the following values: <ul style="list-style-type: none"> <li>- <b>Char/Hex</b>. Displays records in character and hexadecimal format.</li> <li>- <b>Char</b>. Displays records in character format.</li> <li>- <b>Vertical</b>. Displays records vertically in character and hexadecimal format.</li> </ul> <b>Note:</b> If a data file is open in a <b>Data File</b> window, you must close the data file and data map before changes to this value take effect.

Option	Description
	Default is <b>Vertical</b> .

3. Click **OK**.

**RELATED TOPICS:**

- ◆ “Viewing a Data File” on page 49
- ◆ “Importing a Copybook into a Data Map” on page 97
- ◆ “Importing an FDT into an Adabas Data Map” on page 100

## 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 and enter the following information:

Option	Description
Dictionary Name	DDS dictionary name from which to retrieve subschema metadata.
DBName	IDMS database containing the dictionary.
Dictionary UserID	If required, user ID with access to the dictionary.
Dictionary Password	If required, password for the dictionary user ID.
Node Name	DDS node name.
Dictionary Node	DDS dictionary node name.

3. Click **OK**.

**RELATED TOPICS:**

- ◆ “IDMS Data Maps” on page 78

## Encrypt Password Utility

Use the encrypt password utility to encrypt a clear text password.

You can use the encrypted password to control access to source databases or to run pwxcmd commands, PowerExchange utilities, or the PowerExchange Logger for Linux, UNIX, and Windows. Copy the encrypted password and paste it into any of the following parameters that require an encrypted password:

Location	Parameter or Option	Reference
<b>Commands</b>		
pwxcmd commands	-epassword	<i>PowerExchange Command Reference</i>
<b>Configuration files</b>		
DBMOVER	EPWD on the UDB CAPI_CONNECTION statement	<i>PowerExchange Reference Manual</i>
pxxcl.cfg	CAPTURE_NODE_EPWD	<i>PowerExchange CDC Guide for Linux, UNIX, and Windows</i>
<b>Utilities</b>		
DTLREXE	-epwd	<i>PowerExchange Utilities Guide</i>
DTLUAPPL	EPWD	
DTLUCBRG	EPWD	
DTLUCUDB	EPWD	
DTLURDMO	EPWD	
DTLUTSK	EPWD	

## Generating an Encrypted Password

Some PowerExchange configuration statements, such as the EPWD statement in the DBMOVER configuration file, specify an encrypted password.

1. Click **File > Encrypt Password**.  
The **Generate Encrypted Password** dialog box appears.
2. In the **Password** box, type a clear text password.
3. Click **Generate Encrypted Password**.  
The encrypted password appears in the **Encrypted Password** box.
4. Copy the encrypted password.
5. Paste the encrypted password into the appropriate parameter in the configuration file, utility statement, or command option.

For example, you might copy the following encrypted password into the CAPTURE\_NODE\_EPWD parameter in the pxxcl.cfg configuration file:

```
CAPTURE_NODE_EPWD=F40B3B2742FB3A1F
```

## CHAPTER 2

# Data Maps

This chapter includes the following topics:

- ◆ Data Maps Overview, 22
- ◆ Adding a Data Map, 34
- ◆ Adding a Data Map Record, 37
- ◆ Adding a Field to a Record, 37
- ◆ Adding a User-Defined Field to a Record, 38
- ◆ Adding Items to a Datacom Record, 39
- ◆ Adding Items to an IDMS Record, 41
- ◆ Adding Items to an IMS Record, 44
- ◆ Adding a Data Map Table, 46
- ◆ Adding a Column to a Table, 47
- ◆ Viewing Records and Fields, 47
- ◆ Viewing Tables and Columns, 49
- ◆ Viewing a Data File, 49
- ◆ Sorting Data Map Records and Tables, 51
- ◆ Viewing or Editing Properties for a Data Map or Items in a Data Map, 51
- ◆ Copying, Cutting, and Pasting Items in a Data Map, 56
- ◆ Editing Logon Information for Accessing Data Maps and Data, 57
- ◆ Sending a Data Map to a Remote System, 58
- ◆ Importing Data Maps Sent to a Remote System, 59
- ◆ Associating Multiple Data Files with a Data Map, 59
- ◆ Printing a Data Map or Items in a Data Map, 61
- ◆ Deleting Data Maps or Items in a Data Map, 64

## 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 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 12.

#### RELATED TOPICS:

- ◆ “Testing a Data Map” on page 159

## 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 18

## 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	Supported Operating Systems			PowerExchange Access Method
	i5/OS	Linux, UNIX, and Windows	z/OS	
Adabas file		X	X	ADABAS
CA Datacom/DB file			X	DATACOM
CA IDMS/DB database			X	IDMS
C-ISAM data file		X		ISAM
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	DL1 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	Supported Operating Systems			PowerExchange Access Method
	i5/OS	Linux, UNIX, and Windows	z/OS	
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, including C-ISAM 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 59
- ◆ “Step 1. Associate a Data File with a Data Map” on page 180
- ◆ “Step 1. Associate a Data File with a Multiple-Record Data Map” on page 183
- ◆ “GetCurrentFileName” on page 222

## 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 95

## 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 100
- ◆ “Displaying and Refreshing the FDT for an Adabas Data Map” on page 101

## Data Map Records and Fields

In a data map, define, or import from a copybook, one or more records and fields, including user-defined fields and fields defined as arrays.

Data map records and fields define the layout of the data in the source or target.

**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.

Define a single record in a data map if the associated data file or data source contains one record type. In a single-record data map, you can define multiple tables.

Define multiple records in a data map if multiple record types exist in the associated data file or data source.

#### RELATED TOPICS:

- ◆ “Defining Array Properties in the Table Properties Dialog Box” on page 29

## 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 38

## Record-Level Exit Programs

Create a record-level exit program written 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 to decompress compressed data in a record before data retrieval.

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

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

Define a record-level exit program by specifying the attributes of the record-level exit program on the **Record Level Exit** tab in the **Data Map Properties** dialog box, including 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 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. Edit and compile these example programs to customize them for your environment.

## RELATED TOPICS:

- ◆ “Data Map Properties - Record Level Exit” on page 270
- ◆ “User Access Method Programs” on page 243

## 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.

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 or CA IDMS/DB database, data set, 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. The following table lists the data sources, with the associated access methods, that support complex tables:

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

#### RELATED TOPICS:

- ◆ “Adding a Data Map Table” on page 46

## 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.

Option	Description
	For a record definition imported through a COBOL copybook, this value corresponds to the <code>field-y</code> value specified in the <code>OCCURS x DEPENDING ON field-y</code> clause that defines the array.

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

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

Then, the **Array** tab in the **Field Properties** dialog box shows the following options:

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"> <li>- <b>Apply array format changes.</b> Only changes made to array elements are reflected in the table.</li> <li>- <b>Refresh with missing columns.</b> Only new columns are added to the table.</li> <li>- <b>Reset to defaults.</b> A new set of columns are generated with the default names derived from the field names. Any changes that you have made are discarded.</li> <li>- <b>Remove Hidden Columns.</b> DB2UNL only. A new set of columns are generated by using the default values of the <b>Hide from Table</b> option for each field.</li> </ul>
Fields	<p>Controls how elements in an array or group field are displayed:</p> <ul style="list-style-type: none"> <li>- To display each element in an array or group field in a single row, select the field.</li> <li>- To display each element in an array or group field in a separate row, clear the field.</li> </ul> <p><b>Note:</b> To display only fields defined as a group field or as an array in the <b>Fields</b> list, select the <b>Groups and Arrays only</b> option. To display all fields in the record in the <b>Fields</b> list, clear this option. This option controls the fields that are displayed in the <b>Fields</b> list and in the <b>Table Properties</b> dialog box, but not the fields that are displayed in the <b>Database Row Test Output</b> window.</p>

Option	Description
How do you want to handle multiple instances of selected records?	Select one of the following options: - <b>New Row</b> . For every element in the array, a new row is displayed or written to the target. - <b>Ignore</b> . Second and subsequent elements in the array are not displayed or written to the target. - <b>Array</b> . The number of elements specified in the <b>Array</b> 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 <b>New Row on Overflow</b> option.
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. For more information about this option, see "Multiple Arrays in a Single Input Row" on page 30. This option is enabled only for a table with an imported COPYLIB with multiple OCCURS clauses.

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 180.

#### RELATED TOPICS:

- ◆ "Adding a Single-Record Data Map with an Array - Example" on page 180
- ◆ "Viewing or Editing Properties for a Data Map Table" on page 55

### 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

Output Row	Fld1	Fld2
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:

- ◆ 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 UNIX and Windows systems, the ISO-8859 code page is the default code page.

The following table describes the PowerExchange code pages.

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

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

Code Page	Description
IBM-285*	United Kingdom
IBM-297*	France
IBM-424*	Modern Hebrew
IBM-500*	International
ISO-8859	UNIX and Windows

## 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 USRCP $nn$ , 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 skip the row, write a message to the log file, and continue the extraction, or stop the extraction.
 

For example, if PowerExchange detects a record mapping error such as a value in an OCCURS DEPENDING ON clause that is greater than the maximum allowed value, you can specify that PowerExchange stop the extraction.
- ◆ Data errors in fields. You can specify that PowerExchange completes one of the following actions:
  - For CHAR or DATE fields, replace data by using 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. Values in numeric fields are set to zeros.
  - Skip the row, write a message to the log file, and continue the extraction.
  - Stop the extraction.
  - Set the field to NULL.

When you define data checking at the data map level, select the datatypes on which to perform data checking.

You can define data checking at the data map or field level. Field-level data checking definitions override the data map-level data checking definitions.

**Note:** Performing data checking at the field level uses additional resources.

### RELATED TOPICS:

- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55
- ◆ “Viewing or Editing Properties for a Field in a Record” on page 52

## 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 58
- ◆ “[Importing Data Maps Sent to a Remote System](#)” on page 59

## 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 39
- ◆ “[Datacom Data Maps](#)” on page 72
- ◆ “[Adding a Key to a Datacom Record](#)” on page 39
- ◆ “[Adding a Key Segment to a Datacom Record](#)” on page 41

## 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 42
- ◆ “[IDMS Data Maps](#)” on page 78
- ◆ “[Adding an Owner Record and Set to an IDMS Record](#)” on page 42

## 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 44
- ◆ “IMS Data Maps” on page 80
- ◆ “Adding Search Fields to an IMS Record” on page 45

## 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 243
- ◆ “Adding a Data Map” on page 34

## Adding a Data Map

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

For data maps for flat files or sequential data sets, associate the data map with a data file.

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

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Data Map**.
2. In the **Name** dialog box, enter the following information:

Option	Description
Schema Name	Enter a user-defined schema name. PowerExchange uses this name as the first part of the data map name, as follows: <code>schema_name.data_map_name</code> Valid values are alphanumeric characters.

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

3. For the access method that you choose, enter information in the dialog box listed in the following table:

Access Method	Dialog Box	Reference
ADABAS	ADABAS Access Method	"Data Map Properties - ADABAS Access Method " on page 252
DATACom	DATACom Access Method	"Data Map Properties - DATACom Access Method " on page 254
DB2	DB2 Access Method	"Data Map Properties - DB2 Access Method " on page 254
DB2UNLD	DB2UNLD Access Method	"Data Map Properties - DB2UNLD Access Method " on page 255
DL/1 BATCH	DL/1 Batch Access Method	"Data Map Properties - DL/1 Batch or IMS ODBA Access Method " on page 257
ESDS	ESDS Access Method	"Data Map Properties - ESDS Access Method " on page 258
IDMS	IDMS Access Method	"Data Map Properties - IDMS Access Method " on page 259

Access Method	Dialog Box	Reference
IMS ODBA	IMS ODBA Access Method	"Data Map Properties - DL/1 Batch or IMS ODBA Access Method " on page 257
ISAM	ISAM Access Method	"Data Map Properties - ISAM Access Method " on page 260
KSDS	KSDS Access Method	"Data Map Properties - KSDS Access Method " on page 260
MQSERIES	MQSeries Access Method	"Data Map Properties - MQSeries Access Method " on page 261
RRDS	RRDS Access Method	"Data Map Properties - RRDS Access Method " on page 262
SEQ	SEQ Access Method	"Data Map Properties - SEQ Access Method " on page 263
TAPE	TAPE Access Method	"Data Map Properties - TAPE Access Method " on page 265
USER	USER Access Method	"Data Map Properties - USER Access Method " on page 266

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

Option	Description
UserID	For i5/OS or z/OS, enter an operating system user ID, which is required if security is enabled.
Password	Enter the password for the user ID. If you specify a user ID and omit the password, you are prompted for the password in the <b>Missing Password</b> dialog box.
Location	Select the location of the FDT. The options are: - <b>Local</b> . The FDT is located on the PowerExchange Navigator system. - <b>Remote</b> . The FDT is located on a remote system. The locations displayed in the <b>Location</b> list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is <b>Local</b> .
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.

5. If you selected the **Import Records Definitions** option, 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 72
- ◆ "Importing a Copybook into a Data Map" on page 97

# 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.

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

The **Add Record** dialog box appears.

3. On the **Name** tab, define record properties. The properties that you can define depend on the data source type.
4. 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 39
- ◆ “Adding a Key to a Datacom Record” on page 39
- ◆ “Adding a Key Segment to a Datacom Record” on page 41
- ◆ “Adding a CALC Element to an IDMS Record” on page 42
- ◆ “Adding an Owner Record and Set to an IDMS Record” on page 42
- ◆ “Adding a CCK Field to an IMS Record” on page 44
- ◆ “Adding Search Fields to an IMS Record” on page 45
- ◆ “Record, Field, and Table Properties” on page 272

# 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 276

## 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.

Enter the following information 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 <b>Expression</b> 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"><li>- A text or numeric constant such as <code>My text</code> or <code>1234</code></li><li>- A PowerExchange function such as <code>Concat</code> or <code>Split</code></li><li>- The PowerExchange CallProg function that specifies a user-defined Assembler, C, COBOL, or PL/I program and its parameters</li></ul>
Validate	Validates the expression in the <b>Expression</b> box. If PowerExchange detects an error, an error message displays in the <b>Validation</b> 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 159
- ◆ “PowerExchange Functions for User-Defined Fields Overview” on page 216

# 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, enter the following information to define element properties:

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 33
- ◆ “Viewing or Editing Properties for an Element in a Datacom Record” on page 53

## 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**.

On the **Keys** tab in the **Record Properties** dialog box, the following information appears for any defined keys:

Option	Description
Long Key Name	The long name of the key.

Option	Description
	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, enter the following information to define the key:

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.
Nillable	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 33
- ◆ “Viewing or Editing Properties for a Key in a Datacom Record” on page 53

## 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, enter the following information to define the key segment:

Option	Description
Name	Enter the name of the key segment. <b>Note:</b> This field is read-only on the <b>Segment Details</b> tab in the <b>Key Properties</b> 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: - <b>ASC.</b> Ascending order. - <b>DESC.</b> Descending order.
Sensitivity	Select one of the following options to specify the sensitivity of the key segment: - <b>ON.</b> Sensitivity is enabled. - <b>OFF.</b> 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 33
- ◆ “Viewing or Editing Properties for a Key Segment in a Datacom Record” on page 53

## 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.
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 **CALC Element Names** tab, click the **Add New** toolbar button.  
The **New CALC Element** dialog box appears.
4. In the **New CALC Element** dialog box, enter the following information to define the CALC element:

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.

5. 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 42.

### RELATED TOPICS:

- ◆ “Viewing or Editing Properties for a CALC Element in an IDMS Record” on page 54

## 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:

1. Open the IDMS 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 **Owner Record and Set Names** tab, click the **Add New** toolbar button.  
The **Owner Record and Set Properties** dialog box appears.

4. On the **Owner Record and Set Details** tab, enter the following information to define the owner record and set properties:

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

Option	Description
	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 <b>Linked to Owner</b> 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 33
- ◆ “Viewing or Editing Properties for an Owner Record and Set in an IDMS Record” on page 54

## 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, enter the following information to define the CCK field properties:

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 SEGMENT.

Option	Description
	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.

- 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 34
- ◆ “Viewing or Editing Properties for a CCK Field in an IMS Record” on page 55
- ◆ “Table Properties - Definition” on page 286

## 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.

- Open the 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 **Search Fields** tab, click the **Add New** toolbar button.  
The **New Search Field** dialog box appears.
- In the **New Search Field** dialog box, enter the following information to define the search field:

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 SEGMENT. 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.

Option	Description
	Valid values are from 1 through 32767. Default is 1.
Key	Select one of the following key types: - Blank. - U. Unique key type. - M. Non-unique key type. - S. The sequence field is supplied in the DBD definition.

- Click **OK**.

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

#### RELATED TOPICS:

- “IMS Data Map Items” on page 34
- “Viewing or Editing Properties for a Search Field in an IMS Record” on page 55

## 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.

- Open the data map.
- On the **Data Map** tab, right-click the data map and click **Add Table**.
- Depending on the data source for the data map, view or define the following properties on some or 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.
- 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 72
- “Adding a Multiple-Record Data Map with a Complex Table - Example” on page 183

- ◆ “Table Properties - Definition” on page 286
- ◆ “Table Properties - Filter ” on page 288
- ◆ “Table Properties - Keys ” on page 289
- ◆ “Table Properties - IMS Options ” on page 288

## 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 the following column 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 56

## 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.

- 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 **Expr(n)** tab displays the following columns for each field:

Column	Description
Field status indicator	Indicates the status of the field. The options are: <ul style="list-style-type: none"> <li><b>Check mark.</b> The field is valid.</li> <li><b>X.</b> The field is not valid. The comment in the Validation column describes the type of error that the field contains..</li> <li><b>Arrow.</b> A new field is selected, after selecting one of the Add Field menu commands.</li> </ul>
Name	The user-defined name for the field.
Type	The datatype of the field.
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: <ul style="list-style-type: none"> <li><b>R.</b> Read.</li> <li><b>W.</b> Write.</li> <li><b>RW.</b> Read and write.</li> </ul>
Expression	Any expression defined for the field.
Validation	If you click <b>Validate</b> in the <b>Expression Editor</b> dialog box, the <b>Validation</b> column displays the validation message.

- 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 38

## 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 it in a **Data File** window.

You can display a specified number of additional records until all records in the data file appear, and you can select a display format to change the format in which data appears. Also, you can display a ruler above a record to show the number of characters in the record.

**Note:** Define the default display format and the number of records initially displayed in the **Data Map Preferences** dialog box.

To view a data file:

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 the following connection information for the system on which the data file is located and click **OK**:

Option	Description
User ID	If a user ID is required to access the remote system, enter a user ID.
Password	Enter a password for the user ID, if required.
Location	Select the location of the system where the data file to view is located. The locations displayed in the <b>Location</b> list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is <b>local</b> .
File Password	If a password is required to access the data file, enter a password.
Save User ID and Password(s) for session	Saves the information that you enter for the duration of the session. Default is selected.

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

Column	Description
Rec No	The record number.
Offset	The data offset of the hexadecimal data. Displayed 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"><li>- <b>Char/Hex</b>. Displays records in character and hexadecimal format.</li><li>- <b>Char</b>. Displays records in character format.</li><li>- <b>Vertical</b>. Displays records vertically in character and hexadecimal format.</li></ul> Default is <b>Vertical</b> .

By default, ten records are displayed.

4. To display other records in the data file, enter the number of record to display in the box next to the **Get Records** toolbar button and click the **Get Records** toolbar button. When all records are retrieved, a message box indicates that the end of the file is reached.
5. To change the display format or show or hide the scale for a record, complete 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 18

## 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, NUMCHAR, PACKED, SPACKED, UPACKED, 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.
  - ◆ **Adabas Length and Offset.** Obsolete.

#### **RELATED TOPICS:**

- ◆ “Field Properties - Name” on page 276
- ◆ “Field Properties - Array” on page 282
- ◆ “Field Properties - Variable Length” on page 285
- ◆ “Field Properties - Extra Properties” on page 283
- ◆ “Field Properties - Field Checking” on page 284
- ◆ “Field Properties - Code Page” on page 284

## **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 39
- ◆ “Datacom Data Map Items” on page 33

## **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 39
- ◆ “Datacom Data Map Items” on page 33

## **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 41
- ◆ “Datacom Data Map Items” on page 33

## **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 42
- ◆ “IDMS Data Map Items” on page 33

## **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** tab in the **Owner Record and Set Properties** dialog box displays the following information 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 42
- ◆ “IDMS Data Map Items” on page 33

## **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 44
- ◆ “IMS Data Map Items” on page 34

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

#### **RELATED TOPICS:**

- ◆ “Adding Search Fields to an IMS Record” on page 45
- ◆ “IMS Data Map Items” on page 34

## **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 34
- ◆ “Adding an Element to a Datacom Record” on page 39
- ◆ “Adding a Key to a Datacom Record” on page 39
- ◆ “Adding a Key Segment to a Datacom Record” on page 41
- ◆ “Adding a CALC Element to an IDMS Record” on page 42
- ◆ “Adding an Owner Record and Set to an IDMS Record” on page 42
- ◆ “Adding a CCK Field to an IMS Record” on page 44
- ◆ “Adding Search Fields to an IMS Record” on page 45
- ◆ “Data Map Properties” on page 252
- ◆ “Table Properties - Definition” on page 286
- ◆ “Table Properties - Keys ” on page 289
- ◆ “Table Properties - IMS Options ” on page 288

## **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 47

## **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 credentials used to log in to a remote system to access a data map or data.

To edit logon information for accessing data maps and data:

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

Option	Description
Location	Select the location of the data source. Select <b>local</b> if the data source is on the PowerExchange Navigator system. The locations displayed in the <b>Location</b> list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is <b>local</b> .
Data Map Logon	Enter the following credentials for accessing the data map: <ul style="list-style-type: none"><li>- <b>UserID</b>. Enter the database user ID to access the data source, if required.</li><li>- <b>Password</b>. Enter the password for the user ID.</li><li>- <b>File Password</b>. For Adabas, if the FDT file is password-protected, enter the password to access the file.</li></ul>
Data Access Logon	Enter the following credentials for accessing the data source: <ul style="list-style-type: none"><li>- <b>UserID</b>. For z/OS, enter the database user ID to access the data source, if required.</li><li>- <b>Password</b>. For z/OS, required. Enter the password for the user ID.</li></ul>

Option	Description
	<b>Note:</b> You must enter both sets of user IDs and passwords even if they are the same. On z/OS, one user might own the data map and another user might own the actual data.
Allow logons to be changed next time they are required	Select this option to prompt for credentials again before proceeding with the next requested function.

## Sending a Data Map to a Remote System

If the data source for a data map is on a remote system, such as i5/OS or z/OS, 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 data extraction during a bulk data movement operation. The original data map remains on the PowerExchange Navigator system. If you change the local data map, send the data map to the remote system again to update the data map on the remote system.

To send a data map to a remote system:

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

Option	Description
User ID	Enter the user ID to access the remote system, if required.
Password	Enter the password for the user ID, if required.
Location	Select the location of the system to which to send the data map. The <b>Location</b> list displays the locations that you define in NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is <b>local</b> .
Save User ID and Password(s) for session	Select this option to save the information that you enter for the duration of the session. Default is selected.

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 33

## Importing Data Maps Sent to a Remote System

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.

To import data maps sent 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 the following connection information for the system from which you are importing data maps:

Option	Description
User ID	If a user ID is required to access the remote system, enter a user ID.
Password	Enter a password for the user ID, if required.
Location	Select the location of system from which to import the data map. The locations displayed in the <b>Location</b> list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is <b>local</b> .
File Password	For Adabas, 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 information that you enter for the duration of the session. Default is selected.

4. In the **Exported Data Maps** dialog box, import one or more exported data maps.

A message box displays the status of the import operation.

5. Click **OK**.

The data map or data maps appear on the **Resources** tab in the **Resource Explorer**.

#### RELATED TOPICS:

- ♦ “Exported Data Maps” on page 33

## 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 25

## 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:\pxx\data\filelist1.dat
c:\pxx\data\filelist2.dat
c:\pxx\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**.
3. In the **Print** dialog box, view or edit the following print options:

Option	Description
Name	Select the printer. Click <b>Properties</b> 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.

4. Click **OK**.

- PowerExchange prints the data map on the selected printer unless you selected the **Print to file** option.
5. 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:

1. 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 62.

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 62.

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 62.

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 62.  
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 58

## 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, 69
- ◆ Adabas Data Maps, 70
- ◆ Datacom Data Maps, 72
- ◆ DB2 Catalog Data Maps, 74
- ◆ DB2 Unload File Data Maps, 75
- ◆ IDMS Data Maps, 78
- ◆ IMS Data Maps, 80
- ◆ VSAM Data Maps, 90

## 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 the following data sources:

Data Source	PowerExchange Access Method	Reference
Adabas file	ADABAS	"Adabas Data Maps" on page 70
CA Datacom/DB file	DATACOM	"Datacom Data Maps" on page 72
CA IDMS/DB database	IDMS	"IDMS Data Maps" on page 78
DB2 catalog	DB2	"DB2 Catalog Data Maps" on page 74
DB2 unload file	DB2UNLD	"DB2 Unload File Data Maps" on page 75
IMS database	DL/I BATCH	"IMS Data Maps" on page 80

Data Source	PowerExchange Access Method	Reference
IMS database accessed through a PowerExchange Listener	IMS ODBA	
Sequential data set or flat file	SEQ	See the following topics: - “Adding a Single-Record Data Map - Example” on page 177 - “Adding a Single-Record Data Map with an Array - Example” on page 180 - “Adding a Multiple-Record Data Map with a Complex Table - Example” on page 183
User-defined user access method program	USER	Appendix B, “User Access Method Programs” on page 243
VSAM entry-sequenced data set (ESDS)	ESDS	“VSAM Data Maps” on page 90
VSAM key-sequenced data set (KSDS)	KSDS	
VSAM relative record data set (RRDS)	RRDS	

#### RELATED TOPICS:

- ◆ “Data Maps” on page 22
- ◆ “Copybooks” on page 95
- ◆ “Database Row Test” on page 158

## 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.

- ◆ Natural Data Definition Module (DDM) for UNIX and Windows systems. To import a DDM, you must enter a Natural user library path.
- ◆ 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**, click **Add > Data Map**.
2. Enter the following information in the **Name** dialog box:
  - ◆ Select the **ADABAS** access method.
  - ◆ Select the **Import Record Definitions** and **Import Key Fields/FDT** options.
3. Click **Next**.
4. In the **Adabas Access Method** dialog box, define data source information and click **Finish**.
5. In the **Adabas FDT Import** dialog box, select the node name for the Adabas source system. If the source system is z/OS and PowerExchange is configured for security, enter a valid z/OS user ID and password. If you use Adabas file security, enter the file password.
6. Click **OK**.
7. In the **Import Copybook - Source Details** dialog box, enter information about the copybook. Select one of the following source types:
  - ◆ **ADACMP**
  - ◆ **COBOL**
  - ◆ **DDM**. Also enter FDIC database ID and file number values.
  - ◆ **DDM OPEN SYS**. Also enter the path for the Natural user library that contains the DDM for the database.
  - ◆ **FDT**
  - ◆ **PL/1**
  - ◆ **PREDICT**. Also enter FDIC database ID and file number values.
  - ◆ **TEXT**. Available if **Local** is selected as the source.
8. Click **Next**.

9. In the **Import Copybook - Remote Details** dialog box enter information for the remote copybook. If you selected the **DDM**, **DMM OPEN SYS**, or **PREDICT** source type, enter a name in the **Name** box or click the **Browse** button to browse to the file. Click **Next**.
  10. 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 97.
  11. In the **Import Copybook Information** window, review the information for the import operation and click **OK**.
  12. 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 97.
- 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 100
- ◆ “Data Map Properties - ADABAS Access Method ” on page 252
- ◆ “Import Copybook - Source Details Dialog Box” on page 101
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106

## **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 33

## 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, 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 DBMOVER 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 71.
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 97.

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 39
- ◆ “Adding a Key to a Datacom Record” on page 39
- ◆ “Adding a Key Segment to a Datacom Record” on page 41

- ◆ “Data Map Properties - DATACOM Access Method ” on page 254
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106
- ◆ “Import Copybook - Remote Datacom Details Dialog Box” on page 107

## DB2 Catalog Data Maps

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

However, some DB2 tables use single DB2 columns to store an array of fields in a format that might not be consistent with the column type. For example, a CHAR column might contain a multiple packed data fields. You can define a DB2 data map to split the data fields in the CHAR column into separate fields.

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

You can import a copybook when you add an DB2 data map or later.

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

### 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, 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 97.
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 97.  
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 254
- ◆ “Import Copybook - Remote DB2 Catalog Details Dialog Box” on page 108

## Editing the Record Layout in a DB2 Data Map

In a DB2 data map, you can edit the record layout of a DB2 table in the data map.

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

In this example, the MISC\_1 field is defined as a PACKED field with a precision of 9 and a 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\_1 field and click **Properties**.  
The **Field Properties** dialog box appears.
4. In the **Field Properties** dialog box, enter the following information:
  - ◆ In the **Field Name** box, enter MISC\_DATA.
  - ◆ In the **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 97.
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 97.
- 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 255
- ◆ “Import Copybook - Remote DB2 Catalog Details Dialog Box” on page 108

## 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 33

## 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, 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/I**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 109.

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. IDMS log-based change data capture works at a single-record level.

In the **Import Copybook - Configuration Details** dialog box, enter the following information:

- ◆ If the data map is to be used for CDC, select the **Create Table on Each Record Imported** option.
- ◆ If the data map is to be used for bulk data movement and a relational representation of the IDMS data is required, select the **Create Tables for IDMS Hierarchical Paths** option.

For more information, see “Importing a Copybook into a Data Map” on page 97.

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 42
- ◆ “Adding an Owner Record and Set to an IDMS Record” on page 42
- ◆ “Sending a Data Map to a Remote System” on page 58
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106
- ◆ “IDMS Advanced Properties Dialog Box” on page 109
- ◆ “Data Map Properties - IDMS Access Method ” on page 259

## 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 42
- ◆ “Adding an Owner Record and Set to an IDMS Record” on page 42

- ◆ “IDMS Data Map Items” on page 33
- ◆ “Record Properties - Name (IDMS)” on page 274

## 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
- ◆ GetDbKeyOfOwner
- ◆ GetFullDbKey
- ◆ GetFullDbKeyOfOwner
- ◆ GetPageGroup
- ◆ GetPageGroupOfOwner
- ◆ GetPgGrpAndRdx
- ◆ GetPgGrpAndRdxOfOwner
- ◆ GetPgGrpOfOwner
- ◆ GetRadix
- ◆ GetRadixOfOwner
- ◆ GetSeqWithinLevel
- ◆ GetSeqWithinParent

You can also use the CallProg function in an expression to invoke a user-defined program to process source data.

Use expressions to create relational structures, such as data warehouses, from CA IDMS/DB source data.

### RELATED TOPICS:

- ◆ “Adding a User-Defined Field to a Record” on page 38
- ◆ “GetDataFlowType” on page 224
- ◆ “GetDbKeyOfOwner” on page 225
- ◆ “Adding Expressions to an IMS Record” on page 84
- ◆ “PowerExchange Functions for User-Defined Fields” on page 216

## IMS Data Maps

To use an IMS database as a data source, define an IMS data map, which creates SQL statements for relational-type access to IMS data during extraction processing.

Add one or more segments, and fields in segments, to the data map to define the layout of actual physical data.

**Note:** For IMS data sources, *record* refers to a root segment and its children.

A data map maps IMS segments to table views of the data. The segment and fields definitions provide the nonrelational view of the data, and the tables and columns, which are based on the segment and field definitions, provide the relational source view of the data.

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.

Use data maps to align fields, convert dates, filter, and enhance source data to improve data accuracy and reduce data volume.

You can use data maps for Lookup transformations to look up related values and determine if rows exist in an IMS target. When you create a data map for IMS, you can also use IMS as a source or target database.

To create IMS data maps, perform the following tasks:

1. Add an IMS data map. When you define the data map, use 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 in the data map.
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, PowerExchange maps the IMS segments and fields to tables, which creates the relational format of source data that PowerExchange requires.
3. Into each IMS segment in the data map, import a COBOL copybook to overlay the segment with its COPYLIB.
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 a 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 IMSDEMONS file that includes a example IMS DBD source that you can use to create a example IMS data map.

## 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.

## 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 IMS information:

- ◆ **IMS database to access.** IMS database and IMS database data set name.
- ◆ **PSB.** The PSB to use, and which PCB within the PSB to use.
- **For IMS non-ODBA access.** Determine which PCB within a PSB to use. Count the number of PCBs in the PSB. If the PSB has COMPAT=YES, add one to that number.
- **For IMS ODBA access.** Determine which PCB to use within a PSB. Note the PCB name because you must enter it when you create a data map.
- **PROCOPT of the PCB.** To reduce the number of locking conflicts, set PROCOPT to read-only for the PCB for the IMS database.
- **PSB definition in the IMS SYSGEN.** If the PSB is used with the IMS ODBA interface or a Netport BMP job, define the PSB in the IMS SYSGEN.
- ◆ **IMS subsystem ID.**
- ◆ **IMS segments.** The segments to use, and which segments, if any, are of variable length. Additionally, specifies the hierarchical sequence as defined in the DBD source.
- ◆ **IMS DBD source.** Data set location.
- ◆ **Update the PowerExchange configuration on z/OS.** Update the DBMOVER configuration member in the RUNLIB library.
- ◆ **Update the PowerExchange configuration on Windows.** Update the dbmover.cfg configuration file on the PowerExchange Navigator system.
- ◆ **Verify that the data map represents a complete IMS segment.** If you plan to perform an INSERT or UPDATE to 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.

**Note:** To avoid this issue, add a FILLER definition to the COPYLIB before you import it to PowerExchange.

## Configuring PowerExchange to Use DL/1 Batch or BMP to Access IMS Data

To access IMS data by using an IMS DL/1 batch data map, configure the DBMOVER member on z/OS.

1. Update the DBMOVER configuration member in the RUNLIB library, as follows:
  - ◆ Verify that the PSB name in the LISTENER or NETPORT statement is correct.
  - ◆ If you use more than one PSB, include additional LISTENER or NETPORT statements.
2. Update the IMSJCL member in the RUNLIB library so that it works as either a DL/1 batch or BMP job.
3. Restart the PowerExchange Listener to implement the configuration changes.

## Configuring PowerExchange to Use IMS ODBA to Access IMS Data

To access IMS data through an IMS ODBA data map, complete configuration steps on the z/OS system.

1. Include the RESLIB in the STEPLIB library for the PowerExchange Listener, or verify that it is accessible through the LNKLIST concatenation on the z/OS system.
2. Use a PSB with defined PCB NAMES.
3. Restart the PowerExchange Listener to implement the configuration changes.

## 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

You can import a COBOL, DBD, or PL/I copybook when you add an IMS data map or after you add it.

Use a database test to test the tables in the data map.

When you add a data map, specify a schema name and a data map name, and select one of the following access methods to indicate how the source data is accessed:

- ◆ **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.

To add an 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 the name of the schema.
  - ◆ In the **Data Map Name** box, enter the name of the data map.
  - ◆ In the **Access Method** list, select the the **DL/1 BATCH**, **IMS ODBA**, or **SEQ** access method.
  - ◆ Select the **Import Record Definitions** option.
3. Click **Next**.
4. In the **Access Method** dialog box for the access method, enter information for the access method and click **Finish**.
5. If you selected the **Import Record Definitions** option, import an IMS DBD into the data map.

#### **RELATED TOPICS:**

- ◆ “Importing a Copybook into a Data Map” on page 97

## **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
- ◆ GetIMSRBAByLevel
- ◆ 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 80
- ◆ “PowerExchange Functions for User-Defined Fields” on page 216

## **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 GetIMSRBAByLevel function in an expression.

To add the segment RBA to segment data in an IMS data map, Informatica recommends that you use the GetIMSRBAByLevel function rather than the GetDatabaseKey function. The GetIMSRBAByLevel function enables you to get the RBA of an unkeyed or non-unique keyed parent segment.

Use the GetIMSRBAByLevel function in IMS data maps used for bulk data movement operations or IMS synchronous CDC.

You cannot use the GetIMSRBAByLevel 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 GetIMSRBAByLevel 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 38
- ◆ “GetDatabaseKey” on page 223

## **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 55
- ◆ “Importing a Copybook into a Data Map” on page 97
- ◆ “Importing a Copybook into a Record or a Segment” on page 99

## **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 58

## **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, use 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.

To write data to an IMS database:

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
```

 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 288

## 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 **IMSUNL**.
- ♦ **IMS database**. In the **DB Type** list, select **NRDB** or **NRDB2**.

## RELATED TOPICS:

- ♦ “Testing a Data Map” on page 159

## 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=IMSDEMO,ACCESS=(HDAM,VSAM),X
      RMNAME=(DFSHDC40,50,80),X
      EXIT=(*,KEY,PATH,(CASCADE,KEY,PATH),LOG)
DATASET DDI=IMSDEMO,DEVICE=3380,SIZE=8192,SCAN=3
SEGMENT NAME=IMSDemos,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 97.
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 99.
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 97.
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 81
- ◆ “Table Properties - Definition” on page 286

## 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 basic format, extended-format, or compressed data sets.

Run a database row test to test a VSAM data map.

### Adding a VSAM Data Map

When you add a VSAM data map, you can import a copybook into the data map.

To add a VSAM 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 **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.
3. Click **Next**.
4. In the **Access Method** dialog box for the access method, 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.

5. In the **Import Copybook - Source Details** dialog box, select the source type, and select one of the following copybooks types:
  - ◆ **COBOL**
  - ◆ **PL/1**

Default is **COBOL**.
6. Click **Next**.
7. In the **Import Copybook - Remote Details** dialog box for the copybook type, enter information for the copybook and click **Next**.
8. 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 97.
9. In the **Import Copybook Information** window, review the information for the import and click **OK**.
10. 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 97.

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:

- ◆ “Data Map Properties - ESDS Access Method ” on page 258
- ◆ “Data Map Properties - KSDS Access Method ” on page 260
- ◆ “Data Map Properties - RRDS Access Method ” on page 262

## 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, 95
- ◆ Importing a Copybook into a Data Map, 97
- ◆ Importing a Copybook into a Record or a Segment, 99
- ◆ Importing an FDT into an Adabas Data Map, 100
- ◆ Displaying and Refreshing the FDT for an Adabas Data Map, 101
- ◆ Import Copybook - Source Details Dialog Box, 101
- ◆ Import Copybook - Local Details Dialog Box, 104
- ◆ Import Copybook - Local DB2 Catalog Details Dialog Box, 105
- ◆ Import Copybook - Remote Details Dialog Box, 106
- ◆ Import Copybook - Remote Datacom Details Dialog Box, 107
- ◆ Import Copybook - Remote DB2 Catalog Details Dialog Box, 108
- ◆ DB2 Column Selection Dialog Box, 109
- ◆ IDMS Advanced Properties Dialog Box, 109
- ◆ Import Copybook - Configuration Details Dialog Box, 110
- ◆ Record Definition and Duplicate Record Definition Dialog Boxes, 110

## 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 185
- ◆ “Importing a COBOL Copybook with REDEFINES Statements - Example” on page 189
- ◆ “Importing a PL/I Copybook - Example” on page 192

## 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.
  - **DDM OPEN SYS.** A Natural DDM for Adabas for UNIX and Windows sources.
  - **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/I.** 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

Import a copybook when you add a data map or after you add it.

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 101.

**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 106.
- ◆ **Import Copybook - Local Details** dialog box. For more information, see “Import Copybook - Local Details Dialog Box” on page 104.

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 110.

8. Click **Finish**.

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 110.
  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.
- 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 55.
  - ◆ 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 99.
  - ◆ 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 104
- ◆ “Import Copybook - Local DB2 Catalog Details Dialog Box” on page 105
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106
- ◆ “Import Copybook - Remote Datacom Details Dialog Box” on page 107
- ◆ “Import Copybook - Remote DB2 Catalog Details Dialog Box” on page 108

# 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 101.  
**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 106.
  - ◆ **Import Copybook - Local Details** dialog box. For more information, see “Import Copybook - Local Details Dialog Box” on page 104.  
**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 110.
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 110.
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 101
- ◆ “Import Copybook - Local Details Dialog Box” on page 104
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106

## 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 34.
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 72

# 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 100.
  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.

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:

Option	Description	Import into Data Maps Defined with Access Methods	Corresponding Data Source
ADACMP	File created by using the ADACMP utility.	ADABAS	Adabas file
COBOL or PL/I	COBOL or PL/I copybook.	ADABAS	Adabas file
		DATACom	CA Datacom/DB file
		DB2UNLD	DB2 unload file
		DL/I BATCH	IMS database
		ESDS	VSAM ESDS
		IMDS	CA IDMS/DB database
		IMS ODBA	IMS database accessed through a PowerExchange Listener
		ISAM	C-ISAM data file
		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
DATACom	Datacom copybook. Provides combined CA Datacom/DB schema and COBOL copybook information.	DATACom	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
		DB2UNLD	DB2 unload file
DBD	Database description block (DBD). Adds segment, key field, search field, and CCK field definitions from	DL/I BATCH	IMS database

Option	Description	Import into Data Maps Defined with Access Methods	Corresponding Data Source
	the IMS database, and adds tables to the data map. Also defines the hierarchical sequence of segments.	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
DDM OPEN SYS	Natural DDM.	ADABAS	Adabas file on Linux, UNIX, or Windows
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
PL/I	PL/I copybook.	See the COBOL type in this table.	
PREDICT	Predict data dictionary. Provides long name and scale information.	ADABAS	Adabas file on z/OS
TEXT	Text file, which is a delimited format file that uses an Informatica internal format.	ADABAS	Adabas file

#### Column Range

Enter 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.

Option	Description
	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

For DDM or PREDICT copybook types, enter both of the following values:

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.

#### Natural User Library Path

For the DDM OPEN SYS copybook type, enter the natural user library path.

Click the Browse button to browse to the path.

#### RELATED TOPICS:

- ◆ “Importing a Copybook into a Data Map” on page 97

## 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 97

# 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 the location of the DB2 catalog. The options are:

- ♦ **Local**. The catalog is located on the PowerExchange Navigator system.
- ♦ **Remote**. The catalog 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.

### 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 the user ID that has authority to access the database.

### Password

Enter a password for the 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 97
- ♦ “DB2 Column Selection Dialog Box” on page 109

# 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)*

## Location

Location from which to import the copybook. The options are:

Option	Description
Local	The copybook is located on the PowerExchange Navigator system.
Remote	The copybook 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.

## 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 user ID.

## 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 97
- ◆ “IDMS Advanced Properties Dialog Box” on page 109

# 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. The options are:

- ◆ **Local**. The copybook is located on the PowerExchange Navigator system.
- ◆ **Remote**. The copybook 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.

## 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 user ID.

## 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**

Default is cleared and **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 97

# Import Copybook - Remote DB2 Catalog Details Dialog Box

Enter information for the remote DB2 copybook.

### DB Instance

Enter the instance name of the DB2 database.

### Location

Location from which to import the DB2 catalog. The options are:

- ◆ **Local**. The catalog is located on the PowerExchange Navigator system.
- ◆ **Remote**. The catalog 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.

### Table Name

Enter the table name, or click the Browse button to browse to enter filter criteria for a table.

### UserID

Enter the user ID to access the database.

### Password

Enter a password for the user ID.

### 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 97
- ◆ “DB2 Column Selection Dialog Box” on page 109

# 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 100

# 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. The 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 containing the following dictionary:

//SYSIDMS

## Dictionary Node

Enter the name of the node containing 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.

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.
Create table on each record imported	Select this option to create a table based on the record for each imported record. Default is selected.
Create tables for DL1 hierarchical paths	For a DBD, select this option to create all tables for all DL1 hierarchical paths. Default is cleared.
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	Specifies the action to take if PowerExchange finds a duplicate record, field, or table during the import operation.
Action on duplicate field	The options are: <ul style="list-style-type: none"><li>- <b>PROMPT</b>. Prompts for the action to take.</li><li>- <b>UNIQUE NAME</b>. Imports the record or segment using a unique name.</li><li>- <b>OVERWRITE</b>. Overwrites the record or segment.</li><li>- <b>SKIP</b>. Skips the record or segment.</li><li>- <b>APPEND</b>. Appends the record or segment from the copybook to the record or segment in the data map.</li></ul> Default is <b>PROMPT</b> .
Action on duplicate table	

# 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: <ul style="list-style-type: none"> <li>- <b>Fields Only</b></li> <li>- <b>Records</b></li> </ul> 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, 112
- ◆ Registration Tag Names, 113
- ◆ Registration Groups, Extraction Groups, and Application Groups, 115
- ◆ Adding a Registration Group, 116
- ◆ Adding a Capture Registration, 118
- ◆ Viewing a Capture Registration, 124
- ◆ Viewing or Editing Properties for a Registration Group, 124
- ◆ Viewing or Editing Properties for a Capture Registration, 125
- ◆ Viewing Elements for a Datacom Capture Registration, 128
- ◆ Viewing Paths for an IDMS Capture Registration, 129
- ◆ Changing the Columns in a Capture Registration, 129
- ◆ Deleting a Capture Registration, 129
- ◆ Deleting a Registration Group, 130
- ◆ Deleting All Capture Registrations in a Registration Group, 130

## 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.

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, Extraction Groups, and Application Groups” on page 115

## Registration Tag Names

A registration tag name is a unique identifier that PowerExchange generates and assigns to a data source registered for capture.

PowerExchange uses these tags to identify the change records of interest for CDC processing. The PowerExchange Logger for z/OS, PowerExchange Condense, and the PowerExchange Logger for Linux, UNIX, and Windows store the registration tag with each change record in their files.

**Note:** With the z/OS ECCRs and the PowerExchange Logger for z/OS, the registration tag name is referred to as the EDMNAME.

When you create a capture registration, PowerExchange assigns the same registration tag name to both the capture registration and the PowerExchange-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. The following list describes the format of the registration tag name for each source type:

### Adabas

```
ADAinstanceDBIDnnnnnFILEIDnnnnn
```

In the registration tag name format, the variables are as follows:

- ♦ *instance*. The collection identifier value specified when the registration group was created.
- ♦ *DBIDnnnnn*. The Adabas database ID (DBID) number.
- ♦ *FILEIDnnnnn*. The Adabas file number.

### CA Datacom/DB table-based or CA Datacom/DB synchronous

```
DCMmuf_namereg_namen
```

In the registration tag name format, the variables are as follows:

- ♦ *muf\_name*. The CA Datacom/DB MUF name value specified when the registration group was created.
- ♦ *reg\_name*. The name assigned when the capture registration was created.

- ◆ *n*. Version number.

#### **DB2 for Linux, UNIX, and Windows**

`UDBdatabasereg_namen`

In the registration tag name format, the variables are as follows:

- ◆ *database*. The database name value specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ *n*. Version number.

#### **DB2 for z/OS**

`DB2instancereg_namen`

In the registration tag name format, the variables are as follows:

- ◆ *instance*. The database instance value specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ *n*. Version number.

#### **DB2 for i5/OS**

`AS4instancereg_namen`

In the registration tag name format, the variables are as follows:

- ◆ *instance*. The collection identifier value specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ *n*. Version number.

#### **IDMS log-based**

`IDLlogsidreg_namen`

In the registration tag name format, the variables are as follows:

- ◆ *logsid*. The Logsid value specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ *n*. Version number.

#### **IMS log-based**

`IMLrecon_idreg_name100000`

In the registration tag name format, the variables are as follows:

- ◆ *recon\_id*. The RECON identifier value specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ 100000. Static version number.

#### **IMS synchronous**

`IMS.database_name.segment_name`

In the registration tag name format, the variables are as follows:

- ◆ *database\_name*. The database name value specified when the data map was created.
- ◆ *segment\_name*. The segment name value specified when the data map was created.

#### **Microsoft SQL Server**

`MSSinstancereg_namen`

In the registration tag name format, the variables are as follows:

- ◆ *instance*. The value PowerExchange generates from the database name value that was specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ *n*. Version number.

#### Oracle

```
ORAINSTANCEREG_NAMEN
```

In the registration tag name format, the variables are as follows:

- ◆ *instance*. The collection identifier value specified when the registration group was created.
- ◆ *reg\_name*. The name assigned when the capture registration was created.
- ◆ *n*. Version number.

#### VSAM

The registration tag name 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 long, the tag has the following format:

```
VSAMDATASET_NAME
```

If the VSAM data set name is equal to or greater than 29 characters long, the data set name in the tag is truncated to 23 characters long, and a sequentially increasing numeric value of *nnnnn* is appended to the tag, as follows:

```
VSAMDATASET_NAMENNNNN
```

For example, a data set name of USERID1.VSAM.EDMSRCV.MAKEITA is 28 characters long. Consequently, it is assigned the following tag:

```
VSAMUSERID1.VSAM.EDMSRCV.MAKEITA
```

However, a data set name of USERID1.VSAM.EDMSRCV.MAKEITAB is 29 characters long and is assigned the following tag:

```
VSAMUSERID1.VSAM.EDMSRCV.MA00001
```

Subsequent data sets with names that are longer than 28 characters where the first 23 characters match USERID1.VSAM.EDMSRCV.MA are assigned a tag with an *nnnnn* value that is incremented by 1, as follows:

```
VSAMUSERID1.VSAM.EDMSRCV.MA00001+1
```

For example:

```
VSAMUSERID1.VSAM.EDMSRCV.MA00002
```

# Registration Groups, Extraction Groups, and Application 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 adds an extraction group and application group with the same name. You can also manually add an extraction group or an 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 116
- ◆ “Adding an Application Group” on page 249
- ◆ “Adding an Extraction Group” on page 135

## Adding a Registration Group

A registration group is a named group of capture registrations of the same source type and instance.

Before you can add capture registrations, you must add a registration group. You can add capture registrations to the registration group when you create the group or later.

To add a registration group:

1. On the **Resources** tab in the **Resource Explorer**, click **Add > Registration Group**.
2. In the **Add Registration Group** dialog box, enter the following information:

#### Name

User-defined name for the registration group. The maximum length is 16 characters.

#### Location

Location of the source for which you want to capture changes. Select **local** if the source is on the PowerExchange Navigator 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**.

#### Type

Data source type.

You must enter associated information for each data source, such as an instance name.

Select a data source type and enter the associated information, as follows:

- ◆ **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.

In the **MUF Name** box, enter the name of 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.** IDMS tables registered for IDMS synchronous CDC. Obsolete.

- ◆ **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, enter the IMS subsystem ID in the **Recon Identifier** box.

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**

If PowerExchange security is enabled for the source, enter the appropriate user ID, as follows:

- ◆ For a source on i5/OS or z/OS, enter an operating system user ID.

- ◆ For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID.

#### **Password**

Password for the user ID.

If you specify a user ID and omit the password, when you open the registration group, the PowerExchange Navigator prompts you for the password in the **Missing Password** dialog box.

#### **Add Registration**

Adds capture registrations to the group.

Clear this option to add only the registration group.

Default is selected.

3. Complete one of the following actions:
  - ◆ If you selected the **Add Registration** option, click **Next** and proceed to “Adding a Capture Registration” on page 118 in the add a capture registration procedure.
  - ◆ 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, Extraction Groups, and Application Groups” on page 115
- ◆ “Adding a Capture Registration” on page 118

## Adding a Capture Registration

A capture registration defines a source segment, table, database, or data set for which PowerExchange captures changes.

PowerExchange stores capture registrations in the CCT file on the source system that is specified by the registration group to which the capture registration belongs.

You can add one capture registration for a source in a registration group. In a capture registration, select all columns for which you want to capture change data. Edit an extraction map to extract data for a subset of the captured columns. You cannot extract data for any columns that you do not select in the capture registration.

**Note:** You cannot register a source that has column names that are greater than 128 characters long.

When selecting columns for change capture, use the following guidelines:

- ◆ Because you can add one capture registration for a source, select all columns for which you want to capture change data. Edit an extraction map to extract data for a subset of the captured columns. You cannot extract data for any columns that you do not select for change capture.
- ◆ 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.

To add a capture registration:

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 the following information:

#### Name

A user-defined name for the capture registration. From one to eight lowercase alphanumeric characters. Must begin with an alphabetic character.

#### Table Filter

Based on the data source type, optionally enter filters in the following filter boxes with or without wildcards:

- ◆ For the **ADABAS**, **DATACOM**, **IDMS**, or **ORACLE** data source type, enter filters in the **Schema** and **Table** boxes. The maximum length is 128 characters in the **Schema** box and 128 characters in the **Table** box.
- ◆ For the **DB2** data source type, enter filters in the **Creator/Schema**, **DBName/Definer**, and **Table** boxes.  
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.
- ◆ For the **DB2UDB** data source type, enter filters in the **Creator/Schema**, **DBName/Definer**, and **Table** boxes.
- ◆ For the **IMS** data source type, enter filters in the **Schema**, **MapName**, and **Table** boxes. In the **MapName** box, enter the name of a PowerExchange IMS data map that you defined. The maximum length is 10 characters in the **Schema** box, 10 characters in the **MapName** box, and 128 characters in the **Table** box.
- ◆ For the **MSSql** data source type, enter filters in the **Owner** and **Table** boxes. The maximum length is 128 characters in the **Owner** box and 128 characters in the **Table** box.
- ◆ For the **VSAM** data source type, enter filters in the **Schema**, **MapName**, and **Table** boxes. In the **MapName** box, enter the name of a PowerExchange VSAM data map that you defined. The maximum length is 10 characters in the **Schema** box, 10 characters in the **MapName** box, and 128 characters in the **Table** box.

In any filter box, you can use the following wildcard characters:

- ◆ Asterisk (\*) represents one or more matching characters.
- ◆ Question mark (?) represents a single matching character.

#### Respect Case

By default, PowerExchange converts text that you enter in any of the filter boxes 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** option for those filters for which you do not want PowerExchange to use the standard case for the data source.

#### Escape Character

Enter the escape character to use in the filter boxes to delimit an asterisk (\*) or question mark (?) that is a literal value instead of a wildcard.

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.

4. Click **Next**.

The **Add Capture Registration - Tables and Columns** dialog box appears.

If you entered filters, the **Tables** list displays the tables that match the filters. Otherwise, the **Tables** list displays all tables.

The following columns can appear in the **Tables** list, based on the data source type:

Column	Description
Creator Name	The table creator.
Schema	
Name	The table name.
Table	
Type	The object type, such as TABLE.
DataSet Name	The data set name.

5. In the **Tables** list, double-click a table to register it for change capture.

**Note:** If the table contains column names that are greater than 128 characters long, an error message appears and you cannot register the table.

The **Columns** list displays the following information for each column in the selected table:

Field	Description
Select Column	Enables you to select a column to register it for change capture. If an index column is a primary key, it is selected by default in the <b>Columns</b> list.
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 individual columns, select the check box for each column. PowerExchange captures changes for the selected columns.

To select all columns, click one of the following options:

Option	Description
Select All Columns	Available for source types that support selective column capture. If you select this option, all columns in the <b>Columns</b> list are selected and PowerExchange captures changes for all columns. 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.
Select all and notify changes	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. 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.

Option	Description
	<p>For an Oracle source, Oracle CDC fails and logs an error message in the following situations:</p> <ul style="list-style-type: none"> <li>- If a change record for a table that you registered for capture contains a column that you did not register for capture</li> <li>- If a change record does not contain a column that you registered for capture</li> </ul> <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><b>Note:</b> If you clear the <b>Select all and notify changes</b> option after selecting it, cleared check boxes appear for the columns. You must select check boxes to register the columns again.</p>

7. Click **Next**.
8. In the **Add Capture Registration - Type** dialog box, enter the following information:

#### Status

The status of the capture registration. The options are:

- ♦ **I.** Inactive. PowerExchange does not capture changes for the source.
- ♦ **A.** Active. PowerExchange captures changes for the source.

Default is **Inactive**.

**Note:** After you add a capture registration, you can change its status from **Inactive** to **Active**, or from **Active** to **History**, which disables the change capture but retains the capture registration for audit purposes.

#### Condense

The type of condense processing for which the registration is eligible. The options are:

- ♦ **None.** The capture registration is not eligible for condense processing.
- ♦ **Part.** The capture registration is eligible for partial condense processing. Available for PowerExchange Condense or for the PowerExchange Logger for Linux, UNIX, and Windows.
- ♦ **Full.** The capture registration is eligible for full condense processing. Available for PowerExchange Condense on i5/OS or z/OS.

#### Database ID

For an Adabas source, enter the DBID of the database that contains the file from which changes are captured.

#### Primary Dataset Name

For an IMS source with a type of **Synchronous**, enter the data set file name to access.

#### File Number

For an Adabas source, enter the file number in the database identified by the database ID from which changes are captured.

#### Supplement Log Group Name

For an Oracle source, enter the name of the supplemental log group to create. 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.

**Comment**

User-defined comment about the capture registration.

**Execute DDL now**

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 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.

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 ALTER TABLE SQL to file window, enter a file name for the SQL in the **File Name** box and click **Save**.

PowerExchange completes the following actions:

- ◆ For an Oracle source, PowerExchange saves the SQL to the specified file on the local system.
- ◆ For all sources, 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**. Additionally, the following windows and the status bar display the following information for the capture registration:

- ◆ 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 toggle 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 124
- ◆ “Status Options” on page 122
- ◆ “Condense Options” on page 123

## Status Options

A capture registration has a status option that indicates whether changes are captured for the registered source.

When you add a capture registration, set its status to inactive or active. When you edit the properties for a capture registration, you can change its status from inactive to active, or from active to history.

The status options are:

#### Inactive

To prevent data from being captured for a source before the target is materialized, set the status for the capture registration to inactive.

#### Active

To capture changes for a source, set the status for the capture registration to active.

#### History

To stop capturing change capture for a source, but to retain the capture registration for audit purposes, set the status for the capture registration to history.

**Notes:** After you set the status of a registration to history:

- ◆ You cannot use that same registration name again. You must delete and redefine the registration.
- ◆ You cannot make the registration active again. To capture changes for the source again, add a new capture registration for it.

## Condense Options

The condense option for a capture registration defines the type of the condense processing that PowerExchange Condense or the PowerExchange Logger for Linux, UNIX, and Windows performs for the registered source.

The condense options are:

#### None

The capture registration is not eligible for condense processing.

#### Part

The capture registration is eligible for partial condense processing.

This condense type is available for PowerExchange Condense or for the PowerExchange Logger for Linux, UNIX, and Windows. With this condense type, changes in UOWs that complete successfully on registered sources are written to PowerExchange Condense condense files or PowerExchange Logger log files in chronological order based on UOW end time. PowerExchange writes all changes for the columns of interest, not just the latest changes. This condense type maintains transactional consistency.

#### Full

The capture registration is eligible for full condense processing.

This condense type is available for PowerExchange Condense on i5/OS or z/OS. With this condense type, change data is accumulated in keyed condense files such that later changes supersede earlier changes. This condense type 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 for tables with primary keys or for DDS files 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.

**Note:** Select this option if the table 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. If you update unique keys, specify KEY\_CHANGE\_ALW=Y in the PowerExchange Condense configuration file.

# Viewing a Capture Registration

You can view the columns in a capture registration.

To view a capture registration:

1. Open the registration group, and open the capture registration.

The **Capture Registration** window displays the following information for each registered column:

Field	Description
Name	The name of the column.
Table Name	The name of the table.
Column Num	The column number of the column.
Type	The datatype.
Length	The length of the field.
Scale	The data scale.
Key	The value 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	The value YES or NO to indicate whether the column is nullable.

2. To sort the columns in alphabetical order by column name, click the Name column heading in a **Capture Registration** window. To sort the column in reverse alphabetical order, click the Name column heading again.

A small arrow appears in the Name column heading that indicates the sort order, as follows:

- ◆ A down arrow indicates alphabetical order.
- ◆ An up arrow indicates reverse alphabetical order.

# Viewing or Editing Properties for a Registration Group

You can view or edit properties for a registration group.

You might edit the properties for a registration group to change its name, location, or the user ID and password used to access the source. You cannot change the database type for a registration group.

To view or edit properties for a registration group:

1. Open the registration group.

- The **Registration Group** tab in the **Resource Inspector** displays the following properties for the registration group, depending the data source type:

Property	Description
Name	User-defined name for the registration group.
Access Method	Read-only. Data source type.
Location	Location of the source for which you want to capture changes.
UserID	User ID to access the source or the operating system, depending on the access method.
Password	Password for the user ID.
Instance	For an Adabas, DB2 for i5/OS, DB2 for z/OS, Oracle, or VSAM source, the database instance.
MUF Name	For a CA Datacom/DB source, the name of the Multi-User Facility (MUF).
Database	For a DB2 for Linux, UNIX, and Windows source, the name of the 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 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**.

- 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 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.

- If prompted, click **Yes** to confirm any changes.

The **Resource Inspector** updates any fields that you edited.

#### RELATED TOPICS:

- ♦ “Adding a Registration Group” on page 116

## Viewing or Editing Properties for a Capture Registration

After you register a source for change capture, you can edit its capture registration to reflect changes to the data source or change capture processing.

For a DB2 data source, you can add columns to and remove columns from a capture registration.

You can edit the following properties for a capture registration:

- ◆ **Status**. For all sources, you can change the status for a capture registration from inactive to active, or from active to history.
- ◆ **Condense**. For all sources, you can change the condense option for a capture registration from none to partial, or from partial to none. For i5/OS and z/OS data sources with keys, you can change the condense option to none, partial, or full.
- ◆ **Comments**. For all sources, you can change the comments for the capture registration.

To change any other properties for a capture registration, you must add a new capture registration.

**Note:** Making changes to an active capture registration has specific implications.

To view or edit the properties for a capture registration:

1. Open the registration group, and open the capture registration.
2. The **Capture Registration** tab in the **Resource Inspector** displays the following properties for the capture registration, depending the data source type:

Property	Description
<b>All Data Source Types</b>	
Name	Read-only. User-defined name for the capture registration.
Type	Read-only. The capture type for the data source, which is either: <ul style="list-style-type: none"><li>- <b>Log-Based</b>. PowerExchange captures changes from the data source logs.</li><li>- <b>Synchronous</b>. PowerExchange captures changes in a synchronous manner.</li></ul>
Status	The status of the capture registration. The options are: <ul style="list-style-type: none"><li>- <b>Inactive</b>. PowerExchange does not capture changes for the source.</li><li>- <b>Active</b>. PowerExchange captures changes for the source.</li><li>- <b>History</b>. PowerExchange stops capturing changes for the source, but retains the capture registration for audit purposes. After you set the status to history, you cannot make the registration active again.</li></ul>
Condense	The type of condense processing that PowerExchange is to perform for the source. The options are: <ul style="list-style-type: none"><li>- <b>None</b>. PowerExchange does not perform condense processing.</li><li>- <b>Part</b>. PowerExchange performs partial condense processing. Available for PowerExchange Condense or for the PowerExchange Logger for Linux, UNIX, and Windows.</li><li>- <b>Full</b>. PowerExchange performs full condense processing. Available for PowerExchange Condense on i5/OS or z/OS.</li></ul>
Version	Read-only. The version number.
Capture Type	Read-only. The data source type.
Comments	User-defined comment for the capture registration.
Tag	Read-only. PowerExchange-defined name assigned to the registered source to identify the change records of interest for CDC processing.
<b>Adabas</b>	
Database ID	The DBID of the database.
File Number	The file number of the database.

Property	Description
<b>CA Datacom/DB</b>	
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	The value YES or NO to indicate whether recovery facilities of CA DATACOM/DB are used for this table.
Element Count	The number of elements.
<b>DB2 for i5/OS</b>	
File Name	The file name.
<b>IDMS Log-Based</b>	
SubSchema Type	The name of the subschema. Valid characters are ASCII characters and the pound and yen sign.
DB Name	The name of the database. Valid characters are ASCII characters and the pound and yen sign.
Base Record	The base record.
Compressed	The value YES or NO to indicate whether the data is compressed.
Variable	The value YES or NO to indicate the data is of a variable-length datatype.
Page Group	The page group number.
Radix	The radix number.
Min Root Length	The minimum root length.
Data Length	The data length.
Prefix Length	The prefix length.
Ctrl Portion Len	The control portion length.
<b>IMS Synchronous</b>	
Database Name	Read-only. The database name.
Primary Dataset Name	Read-only. The data set file name.
<b>Microsoft SQL Server</b>	
Server	The server where the source is located.

Property	Description
Database Name	The database name.
Dist DBName	The SQL Server distribution database name.
Article Id	The SQL Server article ID.
<b>Oracle</b>	
Supplemental Log	Read-only. The name of the supplemental log group.
<b>VSAM</b>	
Data Set Name	The VSAM ESDS, KSDS, or RRDS data set name.

3. 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**.

4. To apply a change, click **Apply**.

If you change the condense option of an active capture registration, you are notified that the original version of the capture registration is active and that a new version of the registration will be created with a status of inactive. Click **Yes** to confirm this action.

Alternatively, you can change the status of an active registration to inactive before changing its condense option.

#### RELATED TOPICS:

- ◆ “Adding a Capture Registration” on page 118
- ◆ “Changing the Columns in a Capture Registration” on page 129
- ◆ “Registration Tag Names” on page 113
- ◆ “Status Options” on page 122
- ◆ “Condense Options” on page 123

## 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 Columns in a Capture Registration

For a DB2 data source, you can add columns to and remove columns from a capture registration.

**Note:** Making changes to an active capture registration has specific implications.

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 **Amend Columns**.
3. The **Capture Registration - Columns** dialog box appears.
4. For a DB2 source, you can complete one of the following actions:
  - ◆ If the **Select all and notify changes** option is selected, clear this option to enable changes to the columns.
  - ◆ If the **Select all and notify changes** option is cleared, select this option to capture changes for all columns and be notified of any change to the schema for the table.
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 intend to use.

You can delete one capture registration at a time.

When you delete a capture registration, PowerExchange deletes the registration from the CCT file, which is located on the system where the data source is located. To capture new changes for the source registered by the deleted registration, you must add the capture registration again.

PowerExchange does not delete the extraction map associated with the deleted capture registration. You must manually delete the associated extraction map, if desired. To use the extraction map again, you can add the associated capture registration again.

To delete a capture registration:

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.
  3. In the message box that prompts you to confirm the deletion of the registration, click **Yes**.
  4. If the registration is active, a second message box prompts you to confirm the deletion of the registration. Click **Yes** to confirm the deletion of the group. Otherwise, click **No** to cancel the deletion and set the status to **History**.
- PowerExchange deletes the capture registration on the source system. Changes are no longer captured for the registration.

#### RELATED TOPICS:

- ◆ “Viewing or Editing Properties for a Capture Registration” on page 125

## 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 130

## 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 their 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.

To delete all capture registrations in a registration group:

1. Delete each capture registration in the registration group.

2. Because you cannot open or use an extraction map that is associated with a deleted capture registration, complete one of the following actions:
  - ◆ Delete the extraction maps and their extraction group.
  - ◆ Change the capture registrations associated with the extraction maps to active capture registrations.
3. Delete the registration group.

#### **RELATED TOPICS:**

- ◆ “Deleting a Registration Group” on page 130
- ◆ “Deleting a Capture Registration” on page 129
- ◆ “Changing the Capture Registration Associated with an Extraction Map” on page 141
- ◆ “Deleting an Extraction Map” on page 146
- ◆ “Deleting an Extraction Group” on page 147

## CHAPTER 6

# Extraction Groups and Extraction Maps

This chapter includes the following topics:

- ◆ Extraction Groups and Extraction Maps Overview, 132
- ◆ Extraction Map Names, 133
- ◆ Adding an Extraction Group, 135
- ◆ Adding an Extraction Map, 137
- ◆ Viewing an Extraction Map, 138
- ◆ Adding or Removing Change Indicator and Before Image Columns for a Data Column, 139
- ◆ Adding Columns to or Removing Columns from Extraction Processing, 140
- ◆ Changing the Capture Registration Associated with an Extraction Map, 141
- ◆ Merging a Data Map with an Extraction Map, 142
- ◆ Viewing or Editing Properties for an Extraction Group, 143
- ◆ Viewing Properties for an Extraction Map, 144
- ◆ Exporting Metadata for an Extraction Map, 145
- ◆ Creating a Data Map from an Extraction Map, 145
- ◆ Deleting an Extraction Map, 146
- ◆ Deleting an Extraction Group, 147
- ◆ Deleting All Extraction Maps in an Extraction Group, 147
- ◆ Extract Definition Window - Maps Tab, 147
- ◆ Extract Definition Window - Extensions Tab, 148
- ◆ PowerExchange-Generated Columns in Extraction Maps, 148

## 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 160
- ◆ “Registration Tag Names” on page 113
- ◆ “Registration Groups, Extraction Groups, and Application Groups” on page 115

## Extraction Map Names

An extraction map name has a specific format.

An extraction map name has the following format:

`xnninstance.mapname`

In extraction maps, the `xnninstance` variable string is called the *schema*.

When you extract changes using an extraction map, the table name is appended to the extraction map name, as follows:

`xnninstance.mapname_tablename`

The variables in the extraction map name are:

`x`

A character that defines the type of extraction map, which can be one of the following values:

- ◆ **d.** Default extraction map.
- ◆ **u.** User-defined extraction map.

*nn*

A one- or two-digit value that defines the source type, as follows:

- ◆ 1. DB2 for z/OS.
- ◆ 2. IMS.
- ◆ 3. DB2 for i5/OS.
- ◆ 4. Adabas.
- ◆ 5. IDMS synchronous.
- ◆ 6. VSAM.
- ◆ 7. CA Datacom/DB.
- ◆ 8. Oracle.
- ◆ 10. Microsoft SQL Server.
- ◆ B. DB2 for Linux, UNIX, and Windows.
- ◆ D. IDMS log-based.

*instance*

The instance value specified 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 name specified when the capture registration was created.
- ◆ For a user-defined extraction map, the map name specified when the extraction map was created.

*tablename*

The table name of the relational source table or from the source data map.

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 116
- ◆ “Adding a Capture Registration” on page 118

- ◆ “Adding an Extraction Map” on page 137

## 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 by completing the following steps. You can add extraction maps to the extraction group when you add the group or later.

To add an extraction group:

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

User-defined name for the extraction group. The maximum length is 16 characters.

### Location

Location of the source for which you want to extract change data. Select **local** if the source is on the PowerExchange Navigator 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**.

### Type

Data source type.

You must enter associated information for each data source, such as an instance name. This information must match the information in the corresponding registration group.

Select a data source type and enter the associated information, as follows:

- ◆ **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 or synchronous 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.** IDMS tables registered for IDMS synchronous CDC. Obsolete.
- ◆ **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**

If PowerExchange security is enabled for the source, enter the appropriate user ID, as follows:

- ◆ For a source on i5/OS or z/OS, enter an operating system user ID.
- ◆ For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID.

#### **Password**

Password for the user ID.

If you specify a user ID and 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 137 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, Extraction Groups, and Application Groups” on page 115

- ◆ “Adding an Extraction Map” on page 137

# Adding an Extraction Map

You can manually add an extraction map to an extraction group, if necessary.

PowerExchange creates an extraction map when you define a capture registration. However, you might want to add an additional extraction map for a capture registration.

To add an extraction map:

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:

Option	Description
Map Name	A unique user-defined name for the extraction map. Maximum length is eight characters.
Table Name	User-defined name for the table portion of the extraction map name. Maximum length is 20 characters.

In the **Add Extract Definition - Name** dialog box, the **Schema Name** field is read-only and displays the name that PowerExchange generates for the schema portion of the extraction map name. The schema name has the following format:

*unninstance*

Where:

- ◆ *u* indicates that the extraction map is user-defined.
- ◆ *nn* is a one- to two-digit value that indicates the source type. For example, 10 represents Microsoft SQL Server.
- ◆ *instance* is the instance value specified when the registration group was created, which varies by source type.

4. Click **Next**.

The **Active Capture Registrations** list in the **Add Extract Definition - Capture Registrations** dialog box displays the capture registrations available for mapping. In the **Active Capture Registrations** list, the following columns appear:

## Name

The name of the capture registration.

## Version

The version number of the capture registration.

## Condense

The type of condense processing for which the registration is eligible. The options are:

- ◆ **None**. The capture registration is not eligible for condense processing.
- ◆ **Part**. The capture registration is eligible for partial condense processing. Available for PowerExchange Condense or for the PowerExchange Logger for Linux, UNIX, and Windows.
- ◆ **Full**. The capture registration is eligible for full condense processing. Available for PowerExchange Condense on i5/OS or z/OS.

#### Status

The status of the capture registration. The options are:

- ◆ I. Inactive. PowerExchange does not capture changes for the source.
- ◆ A. Active. PowerExchange captures changes for the source.
- ◆ H. History. PowerExchange does not capture changes for the source, but retains the capture registration for audit purposes.

5. To map a capture registration, in the **Active Capture Registrations** list, select a capture registration and click **Add**. The selected capture registration appears in the **Current Capture Registration** list.

If you need to change a selection, double-click a registration in the **Active Capture Registrations** list. The registration in the **Current Capture Registration** list moves to the **Active Capture Registrations** list and the new registration moves to the **Current Capture Registration** list.

If you need to remove a capture registration, select it in the **Current Capture Registration** list and click **Remove**. The registration moves from the **Current Capture Registration** list to the **Active Capture Registrations** list.

6. Click **Next**.

PowerExchange creates the extraction map on the source system in the DTLCAMAP file on a z/OS system, or in the CAMAPS file on a non-z/OS system, and names it as follows:

*schema.mapname*

Where:

- ◆ *schema* is the schema name generated by PowerExchange.
- ◆ *mapname* is the map name that you specified in “Adding an Extraction Map” on page 137 in this procedure.

In the PowerExchange Navigator, the new extraction map appears in the list of extraction maps on the **Extraction Group** tab in the **Resource Explorer**. Additionally, the following windows and the status bar display the following information for the extraction map:

- ◆ 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 toggle 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 133
- ◆ “Adding a Registration Group” on page 116
- ◆ “Status Options” on page 122
- ◆ “Condense Options” on page 123

## Viewing an Extraction Map

You can view the columns in an extraction map, including the columns that are generated by PowerExchange for CDC processing.

To view an extraction map:

1. Open the extraction group, and open the extraction map.

The following tabs in the **Extract Definition** window display extraction map information:

- ◆ The **Maps** tab displays the metadata for the columns for which PowerExchange captures changes and any CI or BI columns that you added. The **Maps** tab also displays the metadata for the PowerExchange-generated columns if you select to show these columns.
- ◆ The **Extensions** tab displays any defined CI and BI columns for the extraction map columns.

2. To show the PowerExchange-generated columns, right-click anywhere on the **Maps** tab and click **Show Auto Generated Columns**.

The metadata for the PowerExchange-generated columns appears.

To hide the PowerExchange-generated columns, right-click anywhere on the **Maps** tab and click **Hide Auto Generated Columns**.

3. To sort the columns on either tab in alphabetical order by a column type, click a column heading in an **Extract Definition** window. To sort the column in reverse alphabetical order, click the column heading again.

A small arrow appears in the column heading that indicates the sort order, as follows:

- ◆ A down arrow indicates alphabetical order.
- ◆ An up arrow indicates reverse alphabetical order.

#### RELATED TOPICS:

- ◆ “Extract Definition Window - Maps Tab” on page 147
- ◆ “Extract Definition Window - Extensions Tab” on page 148

## 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 the table.

These columns store the following information:

- ◆ A CI column indicates whether an UPDATE on the associated data column occurred.
- ◆ A BI column contains the before image for an UPDATE on the data column.

PowerExchange populates a CI column with one of the following values when a change occurs in the associated data column on the source:

- ◆ **Y.** Indicates that an UPDATE changed the data column.
- ◆ **N.** Indicates that an UPDATE did not change the data column.
- ◆ **Null.** Indicates an INSERT or DELETE occurred on the data 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.

A BI column stores the before image for an UPDATE of a data column. If your source database allows changes to primary keys, you can use BI fields with a PowerCenter Flexible Target Key Custom transformation to trigger updates to primary keys on the target when source keys change.

For more information about uses of CI and BI columns, see the PowerExchange CDC guide for your operating system.

**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 BI or CI columns. With the AI setting, the PowerExchange Logger stores only after images. Consequently, PowerExchange cannot provide before images to PWXPC for downstream processing in PowerCenter. Also, if you enter BA for the CAPT\_IMAGE parameter, the PowerExchange Logger stores both before and after images for UPDATEs. Any INSERT or DELETE operations on the source result in Null values in the CI columns.

If necessary, you can edit an extraction map to remove CI and BI columns. In the extraction map, the indicator fields appear as DTL\_CI\_column\_name and DTL\_BI\_column\_name.

To add CI or BI columns to or remove CI or BI columns from a data column:

1. Open the extraction group, and open the extraction map.
2. Right-click the extraction map and click **Amend Change Indicator/Before Image Extensions**.  
The **Extract Definition - Amend Columns** dialog box appears. The **Change Indicators** tab is active.
3. To add 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. To add BI columns, click the **Before Images** tab. In the **Available Columns** list, select one or more columns and click **Add**.  
The selected columns appear in the **Selected Columns** list.
5. To remove CI or BI columns, in the **Selected Columns** list, select the columns and click **Remove**.  
The selected columns appear in the **Available Columns** list.
6. Click **OK**.  
If you added CI or BI columns, the columns appear on both the **Maps** and **Extensions** tabs in the **Extract Definition** window.
7. To verify the changes to the columns, run a database row test on the extraction map.

#### RELATED TOPICS:

- ◆ “Testing an Extraction Map” on page 160

## 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.

After you remove columns, PowerExchange stops extracting change data for those columns.

**Note:** 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 CI or BI columns with columns in the extraction map to capture the following information:

- ◆ **Change indicator (CI) column.** Indicates whether the contents of the column changed in a change record.
- ◆ **Before image (BI) column.** Contains the contents of the column from before the last update operation.  
PowerExchange can save before images as part of the change data stream.

To add columns to or remove columns from extraction processing:

1. Open the extraction group, and open the extraction map.

The following tabs in the **Extract Definition** window display the following information:

- ◆ The **Maps** tab displays the metadata for the columns for which PowerExchange captures changes and any CI or BI columns that you added. The **Maps** tab also displays the metadata for the PowerExchange-generated columns if you select to show these columns.
- ◆ The **Extensions** tab displays CI and BI columns defined for the extraction map columns.

2. To show the PowerExchange-generated columns, right-click anywhere in the **Maps** tab and click **Show Auto Generated 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. To verify changes to the extraction map, run a database row test.

#### RELATED TOPICS:

- ◆ “Adding or Removing Change Indicator and Before Image Columns for a Data Column” on page 139
- ◆ “Testing an Extraction Map” on page 160
- ◆ “Extract Definition Window - Maps Tab” on page 147
- ◆ “Extract Definition Window - Extensions Tab” on page 148

## 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.

To change the capture registration associated with an extraction map:

1. Open the extraction group, and open the extraction map.
2. 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 that are available for selection. Each list displays the following columns for the registrations:

Column	Description
Name	User-defined name of the capture registration.
Version	The version number of the capture registration.

Column	Description
Condense	The type of condense processing for which the registration is eligible. The options are: <ul style="list-style-type: none"> <li>- <b>None.</b> The capture registration is not eligible for condense processing.</li> <li>- <b>Part.</b> The capture registration is eligible for partial condense processing. Available for PowerExchange Condense or for the PowerExchange Logger for Linux, UNIX, and Windows.</li> <li>- <b>Full.</b> The capture registration is eligible for full condense processing. Available for PowerExchange Condense on i5/OS or z/OS.</li> </ul>
Status	The status of the capture registration. The options are: <ul style="list-style-type: none"> <li>- <b>Inactive.</b> PowerExchange does not capture changes for the source.</li> <li>- <b>Active.</b> PowerExchange captures changes for the source.</li> <li>- <b>History.</b> PowerExchange has stopped capturing changes for the source, but retains the capture registration for audit purposes. After you set the status to history, you cannot make the registration active again.</li> </ul>

3. In the **Current Capture Registration** list, select the current registration and click **Remove**.
4. 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.
5. Click **OK**.
6. To verify the change, run a database row test on an extraction map.

#### RELATED TOPICS:

- ◆ “Testing an Extraction Map” on page 160
- ◆ “Status Options” on page 122
- ◆ “Condense Options” on page 123

## 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.

To merge an extraction map with a data map:

1. Open the extraction group, and open the extraction map.
2. On the **Extraction Group** tab in the **Resource Explorer**, right-click the extraction map and select **Amend Capture Registrations**.
3. In the **Extract Definition - Amend Capture Registrations** dialog box, select **Merge**.
4. In the **Bulk Datamap Name** box, enter a data map name or select a data map, as follows:
  - ◆ 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**.
5. 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 74

## 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.

To view or edit properties for an extraction group:

1. Open the extraction group.
2. The **Extraction Group** tab in the **Resource Inspector** displays the following properties for the extraction group, depending the data source type:

Property	Description
Name	User-defined name for the extraction group.
Access Method	Read-only. Data source type.
Location	Location of the source for which you want to capture changes.
UserID	User ID to access the source or the operating system, depending on the access method.
Password	Password for the user ID.
Instance	For an Adabas, DB2 for i5/OS, DB2 for z/OS, Oracle, or VSAM source, the database instance.
MUF Name	For a CA Datacom/DB source, the name of the Multi-User Facility (MUF).
Database	For a DB2 for Linux, UNIX, and Windows source, the name of the 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.

3. 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**.

4. 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.

5. If prompted, click **Yes** to confirm any changes.

The **Resource Inspector** updates any fields that you edited.

#### RELATED TOPICS:

- ◆ “Adding an Extraction Group” on page 135

## Viewing Properties for an Extraction Map

You can view but not change the properties of an extraction map.

To view the properties for an extraction map:

- Open the extraction group, and open the extraction map.

The **Extract Definition** tab in the **Resource Inspector** displays the following properties for the extraction map:

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 135
- ◆ “Registration Tag Names” on page 113

# Exporting Metadata for an Extraction Map

For relational sources, you can export the metadata for an extraction map in XML format.

**Note:** By default, the XML file is stored in the PowerExchange installation directory on the system that is the target of the export without an .xml extension. You must edit the file name to add the .xml extension if required.

To export the metadata for an extraction map:

1. Open the extraction group, and open the extraction map.
2. Right-click the extraction map and click **Export Schema**.
3. In the **Export Schema** dialog box, enter the following information:

Option	Description
Location	Select the target location to which to export the metadata for the extraction map. The XML file is stored in the PowerExchange installation directory on the target system. The locations displayed in the <b>Location</b> list are defined by NODE statements in the dbmover.cfg configuration file on the PowerExchange Navigator system. Default is <b>local</b> .
UserID	If required, enter the user ID to access the export target system.
Password	Password for the user ID.
Export Schema As	Enter a name for the exported XML document or use the default value, which does not include the .xml extension. Add .xml to the default value to store the document as an XML document. <b>Note:</b> By default, the XML file is stored in the PowerExchange installation directory on the target system without an .xml extension. You must edit the file name to add the .xml extension, if required.
Document Enveloping	Uses XML document enveloping. Default is cleared.

4. Click **OK**.
5. In the confirmation dialog box, click **OK**.

PowerExchange stores the XML file in the PowerExchange installation directory.

# 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 129

## 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 146
- ◆ “Deleting an Extraction Group” on page 147
- ◆ “Deleting a Capture Registration” on page 129
- ◆ “Deleting a Registration Group” on page 130

## Extract Definition Window - Maps Tab

The **Maps** tab of an **Extract Definition** window displays information for each column in an extraction map.

The **Maps** tab displays the following information:

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"><li>- <b>Real</b>. An original column of the extraction map.</li><li>- <b>BI</b>. A before image column that was added through the <b>Amend Change Indicator/ Before Image Extensions</b> command.</li><li>- <b>CI</b>. A change indicator image column that was added through the <b>Amend Change Indicator/ Before Image Extensions</b> command.</li><li>- <b>Auto</b>. A PowerExchange-generated column.</li></ul>
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 of an **Extract Definition** window displays the following information for each column in an extraction map:

Column	Description
Name	The name of the column.
CI Column	The associated change indicator (CI) column.
BI Column	The associated before image (BI) column.
Real Column	The actual column for which the CI and BI extensions were added.

**Note:** If you added a CI or BI column for a data column, 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 139

## PowerExchange-Generated Columns in Extraction Maps

Besides the table columns defined in capture registrations, extraction maps include columns that PowerExchange generates.

These PowerExchange-generated columns contain CDC-related information, such as the change type and time stamp.

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

When you run a database row test on an extraction map, the PowerExchange Navigator displays the PowerExchange-generated columns in the results. By default, the PowerExchange Navigator hides these columns from view when you open the extraction map. To display these columns, open the extraction map, right-click anywhere within the **Extract Definition** window, and select **Show Auto Generated Columns**.

**Note:** By default, all columns except the DTL\_\_*columnname*\_CNT and DTL\_\_*columnname*\_IND columns are selected in an extraction map. To select these columns, you must edit the extraction map.

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

Column	Description	Datatype	Length
DTL__CAPXRESTART1	A binary value that represents the position of the end of the UOW for that change record followed by the position of the change record itself. The length of a sequence token varies by data source type, except on z/OS where sequence tokens for all data source types have the same length. The value of DTL__CAPXRESTART1 is also known as the <i>sequence token</i> , which when combined with the <i>restart token</i> comprises the restart token pair. A sequence token for a change record is a strictly ascending and repeatable value.	VARBIN	255
DTL__CAPXRESTART2	A binary value that represents a position in the change stream that can be used to reconstruct the UOW state for the change record, with the following exceptions: <ul style="list-style-type: none"> <li>- Microsoft SQL Server CDC. A binary value that contains the DBID of the distribution database and the name of the distribution server.</li> <li>- 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.</li> </ul> The length of a restart token varies by data source type. On z/OS, restart tokens for all data source types have the same length, except for change data extracted from full condense files. The value of DTL__CAPXRESTART2 is also known as the <i>restart token</i> , which when combined with the <i>sequence token</i> comprises the restart token pair.	VARBIN	255
DTL_CAPXRRN	For DB2 on i5/OS only, the relative record number.	DECIMAL	10
DTL__CAPXUOW	A binary value that represents the position in the change stream of the start of the UOW for the change record.	VARBIN	255

Column	Description	Datatype	Length
DTL__CAPXUSER	<p>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"> <li>- For Datacom synchronous CDC, this value is the JOB card name for the job that made the change to the data source.</li> <li>- For Datacom table-based CDC, this value is the MUF name.</li> <li>- For DB2 for i5/OS, this value depends on the LIBASUSER parameter on 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.</li> <li>- For DB2 for z/OS, if you do not specify the UIDFMT parameter on the LRAP CAPI_CONNECTION, this value is the user ID of the user who made the change. Otherwise, this value is based on the UIDFMT parameter.</li> <li>- For Microsoft SQL Server, this value is null because Microsoft SQL Server does not record this information in the distribution database.</li> <li>- For Oracle, this value is the user ID that Oracle provides, if known. Otherwise, this value is null.</li> </ul>	VARCHAR	255
DTL__CAPXTIMESTAMP	<p>The time stamp for when the change was made to the data source, as recorded by the source DBMS in the following format:  <i>YYYYMMDDhhmmssnnnnnn</i>  Where:</p> <ul style="list-style-type: none"> <li>- <i>YYYYMMDD</i> is the date in year (<i>YYYY</i>), month (<i>MM</i>), and day (<i>DD</i>) format.</li> <li>- <i>hhmmssnnnnnn</i> is the time in hours (<i>hh</i>), minutes (<i>mm</i>), seconds (<i>ss</i>), and microseconds (<i>nnnnnn</i>) format.</li> </ul> <p><b>Note:</b> Oracle does not support microseconds in the time stamp.</p>	CHAR	20
DTL__CAPXACTION	<p>A single character that indicates the type of change record that PowerExchange passes to the target during extraction processing. A DTL__CAPXACTION value corresponds to the type of change operation on the source database: insert, delete, or update.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> <li>- I. Insert.</li> <li>- D. Delete.</li> <li>- U. After image of an update operation.</li> <li>- T. Before image of an update operation. (ODBC connections only)</li> </ul> <p>If you specify an <b>Image Type of BA</b> 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 D value. In the insert record, the DTL__CAPXACTION column contains the I value.</p>	CHAR	1

Column	Description	Datatype	Length
	If you specify an <b>Image Type</b> of <b>AI</b> on the connection for a CDC session, PowerExchange generates one record for an update. In this record, the DTL__CAPXACTION column contains the U value. If you use an ODBC connection to write change data to a staging table and either set the ODBC driver CAPXIMAGETYPE parameter to TU or enter the SQL escape sequence DTLIMTYPE=TU in PowerCenter, this column can contain a value of T or U. For each source update, PowerExchange delivers two records to the staging table: one for the before image and another for the after image. In the before image record, the DTL__CAPXACTION column contains the T value. In the after image record, The DTL__CAPXACTION column contains the U value.		
DTL__CAPXCASDELIND	For DB2 for z/OS sources only, a single character that indicates whether DB2 has deleted the row because the table specifies the ON DELETE CASCADE clause. Valid values are: <ul style="list-style-type: none"> <li>- Y. Indicates that DB2 deleted this row because of a cascade delete rule.</li> <li>- N. Indicates that DB2 did not delete this row because of a cascade delete rule.</li> </ul>	CHAR	1
DTL__BI_columnname	For UPDATE operations, the value of the before image of the selected column in the change record.	Datatype of the source column	Length of the source column
DTL__CI_columnname	For UPDATE operations, a single character that indicates whether the selected column was changed. Valid values are: <ul style="list-style-type: none"> <li>- Y. Indicates that the column changed.</li> <li>- N. Indicates that the column did not changed.</li> <li>- Null value. Indicates an INSERT or DELETE operation.</li> </ul>	CHAR	1
DTL__columnname_CNT	Binary count column. PowerExchange generates this column for variable length columns of types VARCHAR and VARBIN to determine the length of the column during change data extraction processing. <b>Note:</b> By default, binary count columns are not selected in an extraction map. You must edit an extraction map to select these columns.	NUM32U	0
DTL__columnname_IND	Null indicator column. PowerExchange generates this column for nullable columns to indicate the nullable value for the column. <b>Note:</b> By default, null indicator columns are not selected in an extraction map. You must edit an extraction map to select these columns.	BIN	1

## CHAPTER 7

# Personal Metadata

This chapter includes the following topics:

- ◆ Personal Metadata Overview, 152
- ◆ Adding a Personal Metadata Profile, 152
- ◆ Viewing Metadata for Tables in a Personal Metadata Profile, 155
- ◆ Viewing Properties for a Table in a Personal Metadata Profile, 156
- ◆ Viewing or Editing Properties for a Personal Metadata Profile, 156
- ◆ Refreshing Personal Metadata, 157
- ◆ Deleting a Personal Metadata Profile, 157

## 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 12 and “Closing a PowerExchange Resource” on page 15.

For information about searching for tables and columns in tables in personal metadata profiles, see “Searching for a Resource” on page 13.

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

A personal metadata profile enables you to connect to a source or target.

A profile can include filters to limit the tables that it contains.

To add a personal metadata profile:

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**

User-defined name for the personal metadata profile.

**Location**

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.

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**.

**Type**

Data source type for the source or target. Select one of the following types and enter the associated information:

- ◆ **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** box. Default is the default database.
  - For DB2 for z/OS, enter the SSID in the **SSID or DBName** box.
- ◆ **DB2400C.** DB2 Call Level Interface (CLI) for i5/OS sources. Enter the database name in the **SSID or DBName** box.
- ◆ **DB2UDB.** DB2 for Linux, UNIX, and Windows sources or targets. Optionally enter a database name in the **DBName** box.
- ◆ **MSSQL.** Microsoft SQL Server sources or targets. Enter the SQL Server instance name in the **DSN** box. Optionally enter a database name to override the database for the connection in the **Database** box.
- ◆ **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** box.

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** box.
- ◆ **SYBASE.** Sybase sources or targets. Enter the server instance in the **Server** box. Optionally enter a database within the instance in the **Database** box.

**Note:** The following data source types are deprecated and are not valid for personal metadata profiles:

- ◆ADAUNLD
- ◆AS4\_META
- ◆CAPX

- ◆CAPXRT
- ◆DB2390IMG
- ◆IMSUMLD
- ◆INFORMIX
- ◆ODBC
- ◆TASK\_CTNL

#### **UserID**

For an i5/OS or z/OS source, enter an operating system user ID, which is required if security is enabled.

For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID, if required.

#### **Password**

Password for the user ID.

If you specify a user ID and omit the password, when you open the personal metadata profile, the PowerExchange Navigator prompts you for the password in the **Missing Password** dialog box.

#### **Description**

User-defined description for the personal metadata profile.

3. Click **Next**.
4. On the **Personal Metadata - Filters** dialog box, enter the following information to limit the tables that the profile contains:

#### **Filter Data**

Depending on the data source type, optionally enter filters in the following filter boxes, with or without wildcards:

- ◆ For the DB2, DB2400C, or DB2UDB data source type, enter filters in the **Creator/Schema**, **DBName/Definer**, and **Table** boxes.
- ◆ For the ORACLE data source type, enter filters in the **Schema** and **Table** boxes.
- ◆ For the MSSQL or SYBASE data source type, enter filters in the **Owner** and **Table** boxes.
- ◆ For the NRDB or NRDB2 data source type, enter filters in the **Schema**, **Mapname**, and **Table** boxes.

In any filter box, you can use the following wildcard characters:

- ◆ Asterisk (\*) represents one or more matching characters.
- ◆ Question mark (?) represents 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** option for those filters for which you do not want PowerExchange to use the standard case for the data source.

#### **Escape Character**

Enter the escape character to use to delimit an asterisk (\*) or question mark (?) that is used as a literal value rather than as a wildcard in the 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.

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 161

## 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 152

## 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, 158
- ◆ Testing a Data Map, 159
- ◆ Testing an Extraction Map, 160
- ◆ Testing a Personal Metadata Profile, 161
- ◆ Issuing PowerExchange Listener Commands in a Database Row Test, 162
- ◆ Generating Restart Tokens for Change Data Capture, 163
- ◆ Database Row Test Dialog Box, 164
- ◆ CAPX Advanced Parameters Dialog Box, 171
- ◆ CAPXRT Advanced Parameters Dialog Box, 172
- ◆ DB2390IMG Advanced Parameters Dialog Box, 174

## 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 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:** Selecting specific columns in the table does not affect SQL DTLDESCRIBE requests.

4. Click **File > Database Row Test**.

The **Data Map Remote Node** dialog box appears.

5. In the **Data Map Remote Node** dialog box, enter connection information for the remote system and click **OK**.  
The **Database Row Test** dialog box appears.

6. In the **DB Type** list in the **Database Row Test** dialog box, click the data source type. Then, enter any associated information for the data source type.  
**Note:** Use the **AS4\_META** data source type 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 metadata for a table or a schema.  
Depending on the option that you select in the **Fetch** list, an SQL statement is generated in the **SQL Statement** box, as follows:
  - ◆ If you select **Data** in the **Fetch** list, an SQL SELECT statement is generated.
  - ◆ If you select a metadata option in the **Fetch** list, an SQL DTLDESCRIBE statement.
8. Optionally, edit the SQL SELECT or DTLDESCRIBE statement in the **SQL Statement** box.
9. For the **CAPX** or **CAPXRT** data source type, enter an application name in the **Application** box.
10. For the **CAPX**, **CAPXRT**, or **DB2390IMG** data source types, click **Advanced** to display the **Advanced Parameters** dialog box and enter information to override settings in the PowerExchange configuration file.
11. In the **Get Rows** list, select the number of rows to return in the output. You can display up to 999 rows of data at a time. By default, PowerExchange displays ten rows of data.
12. Click **Go**.

The **Database Row Test Output** window displays the results of the database row test.

#### RELATED TOPICS:

- ◆ “Issuing PowerExchange Listener Commands in a Database Row Test” on page 162
- ◆ “Sending a Data Map to a Remote System” on page 58
- ◆ “Database Row Test Dialog Box” on page 164
- ◆ “CAPX Advanced Parameters Dialog Box” on page 171
- ◆ “CAPXRT Advanced Parameters Dialog Box” on page 172

## 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 in the **Database Row Test** dialog box, click the data source type. Then, enter any associated information for the data source type.  
**Note:** Use the **AS4\_META** data source type 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 metadata for a table or a schema.

Depending on the option that you select in the **Fetch** list, an SQL statement is generated in the **SQL Statement** box, as follows:

- ◆ If you select **Data** in the **Fetch** list, an SQL SELECT statement is generated.
  - ◆ If you select a metadata option in the **Fetch** list, an SQL DTLDESCRIBE statement.
6. Optionally, edit the SQL SELECT or DTLDESCRIBE statement in the **SQL Statement** box.
  7. For the **CAPX** or **CAPXRT** data source type, enter an application name in the **Application** box.
  8. For the **CAPX**, **CAPXRT**, or **DB2390IMG** data source types, click **Advanced** to display the **Advanced Parameters** dialog box and enter information to override settings in the PowerExchange configuration file.
  9. In the **Get Rows** list, select the number of rows to return in the output. You can display up to 999 rows of data at a time. By default, PowerExchange displays ten rows of data.
  10. 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 139

## 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 in the **Database Row Test** dialog box, click the data source type. Then, enter any associated information for the data source type.

**Note:** Use the **AS4\_META** data source type 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. For the **CAPX** or **CAPXRT** data source type, enter an application name in the **Application** box.
9. For the **CAPX**, **CAPXRT**, or **DB2390IMG** data source types, click **Advanced** to display the **Advanced Parameters** dialog box and enter information to override settings in the PowerExchange configuration file.
10. In the **Get Rows** list, select the number of rows to return in the output. You can display up to 999 rows of data at a time. By default, PowerExchange displays ten rows of data.

11. 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 run a database row test to issue a LISTTASK or STOPTASK command to a PowerExchange Listener running on any system.

You can also issue a LISTLOCATIONS command, which returns a listing of the defined NODE statements in the DBMOVER configuration file.

To issue PowerExchange Listener commands in a database row test:

1. Open a data map defined for the system where the PowerExchange Listener to which you want to issue commands is running.

**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 in the **Resource Explorer**, click a table and click **File > Database Row Test**.
3. In the **Data Map Remote Node** dialog box, enter connection information for the system on which 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 NODE statements that are defined in the DBMOVER configuration file on the system where the PowerExchange Listener is running.
  - ◆ **List Task.** Displays information about each active PowerExchange Listener task, including the TCP/IP address, port number, application name, access type, and status.
  - ◆ **Stop Task.** Stops an individual PowerExchange Listener task based on the application name that you specify.
6. Optionally, if you are issuing a STOPTASK command, you can enter an application name in the **Application** box. The application name is the name of the active extraction process to stop. This name is included in the LISTTASK command output.
7. If you are issuing a STOPTASK command, add a task ID to the following command in the **SQL Statement** box:

```
stoptask taskid=
```

The task IDs for active tasks are included in the LISTTASK command output.

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 58

# 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.

To generate restart tokens for change data capture:

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. A valid value is from 1 through 20 characters long.
6. To generate restart tokens, enter the following PowerExchange SQL statement in the **SQL Statement** box:

```
SELECT CURRENT_RESTART [WHERE {CONNAME=conn_name|CONTTYPE=conn_type}]
```

This SQL statement has the following parameters:

**SELECT CURRENT\_RESTART**

Generates current restart tokens.

**CONNAME=conn\_name**

Specifies the CAPI\_CONNECTION name at the source location that PowerExchange uses to determine the data source type.

**Note:** If you do not specify either the CONNAME or CONTTYPE parameter, PowerExchange determines the data source type by using the value in the CAPI\_CONN\_NAME statement for the PowerExchange Listener in the source location.

**CONTYPE=conn\_type**

Specifies the connection type that PowerExchange uses to determine the data source type.

**Note:** If you do not specify either the CONNAME or CONTYPE parameter, PowerExchange determines the data source type by using the value in the CAPI\_CONN\_NAME statement for the PowerExchange Listener in the source location.

Restart token formats vary by source type. Specify one of the following values for the *conn\_type* value:

- ◆ **ADA.** Adabas sources (z/OS).
- ◆ **AS4.** DB2 for i5/OS sources.
- ◆ **DCM.** CA Datacom/DB sources (z/OS).
- ◆ **DB2.** DB2 for z/OS sources (z/OS).
- ◆ **IDL.** CA IDMS/DB log-based sources (z/OS).
- ◆ **IDM.** CA IDMS/DB synchronous sources (z/OS).
- ◆ **IML.** IMS log-based sources (z/OS).
- ◆ **IMS.** IMS synchronous sources (z/OS).
- ◆ **ORA.** Oracle sources.
- ◆ **MSS.** Microsoft SQL Server sources.
- ◆ **UDB.** 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. Restart tokens for z/OS sources represent locations within the PowerExchange Logger logs.

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

Specify options for a database row test.

### DB Type

The data source type, and for some data sources, how the data source is accessed. For extraction maps, default is **CAPXRT**.

Select a data source type and enter the associated data source information, as follows:

- ◆ **ADAUNLD.** Nonrelational Adabas unload file accessed through a data map.
  - In the **File Name** box, enter the file name for data retrieval.
  - In the **Database** box, enter the database name.
- ◆ **AS4\_META.** Use at the direction of Informatica Global Customer Support.
- ◆ **CAPX.** Reads changes from condense files created by PowerExchange Condense.

- In the **Override File Name** box, enter the name of the file to override the default file.
- ◆ **CAPXRT**. Reads changes from the change stream.
- In the **Override File Name** box, enter the name of the file to override the default file.
- ◆ **DB2**. DB2 for i5/OS or DB2 for z/OS database.
- For DB2 for z/OS, enter the DB2 subsystem ID or the DB2 data-sharing group name in the **SSID or DBName** box.
- ◆ **DB2390IMG**. Reads data from a DB2 image copy for uncompressed table spaces from a remote system.
- In the **Override File Name** box, enter the name of the file to override the default file.
- In the **SSID** box, enter the SSID.
- ◆ **DB2400C**. Uses the Call Level Interface (CLI) access method to access DB2 data, which provides faster access to DB2 data.
- For DB2 for z/OS, enter the SSID in the **SSID or DBName** box.
- ◆ **DB2UDB**. DB2 for Linux, UNIX, and Windows database.
- Optionally, enter the database name in the **SSID or DBName** box. If you omit the database name, the default database name is used.
- ◆ **IMSUNLD**. IMS unload file.
- In the **File Name** box, enter the IMS unload file name.
- ◆ **INFORMIX**. Informix database.
- In the **Database** box, enter the Informix database name.
- In the **Server** box, enter the Informix database server name.
- ◆ **MSSQL**. Microsoft SQL Server database.
- In the **DSN** box, enter the SQL Server instance name.
- Optionally enter a database name to override the database for the connection in the **Database** box.
- ◆ **NRDB**. Nonrelational data source such as a VSAM data set or a flat file.
- Note:** This data source type uses a three-tier naming convention in the generated SQL.
- Optionally enter the name of the file to override the default file in the **Override File Name** box.
- In the **File Password** box, enter the password for the file, if required.
- ◆ **NRDB2**. Nonrelational data source.
- Note:** This data source type uses a two-tier naming convention in the generated SQL.
- Optionally enter the name of the file to override the default file in the **Override File Name** box.
- In the **File Password** box, enter the password for the file, if required.
- ◆ **ODBC**. ODBC data source.
- In the **DSN** box, enter the data set name.
- In the **Database** box, enter the database name.
- ◆ **ORACLE**. Oracle database.
- Optionally enter a TNS name for the Oracle instance in the **SQL\*Net Name** box.
- ◆ **Simulator**. Simulator data source.
- In the **DB Qual1** box, enter a value.
- ◆ **SYBASE**. Sybase database.
- In the **Server** box, enter the name of the Sybase database server instance.

Optionally enter the name of the database in the database server instance in the **Database** box.

- ◆ **TASK\_CTRL**. 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 162.

#### Location

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.

#### User ID

User ID to access the remote system, if required.

#### Password

Password for the user ID, if required.

#### Fetch

The type of data to access and display in the database row test, or for the TASK\_CTRL data source type, the type of command to issue to a PowerExchange Listener.

**Note:** For personal metadata profiles, the **Fetch** list is disabled, and the default option 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		Displays data rows.
Foreign keys		Displays metadata for foreign keys.
Primary keys		Displays metadata for primary keys.
Procedure Cols	- DB2 - DB2400C - Simulator	Displays metadata for specific stored procedures.
Procedures		Displays metadata for the available stored procedures.
Records	- NRDB - NRDB2	Displays metadata for NRDB and NRDB2 records.

Option	Available for Data Source Types	Description
Schemas	<ul style="list-style-type: none"> <li>- ADAUNLD</li> <li>- CAPX</li> <li>- CAPXRT</li> <li>- DB2</li> <li>- DB2390IMG</li> <li>- DB2400C</li> <li>- DB2UDB</li> <li>- IMSUNLD</li> <li>- INFORMIX</li> <li>- MSSQL</li> <li>- NRDB</li> <li>- NRDB2</li> <li>- ODBC</li> <li>- ORACLE</li> <li>- SYBASE</li> </ul>	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		Displays information about each active PowerExchange Listener task, including the TCP/IP address, port number, application name, access type, and status.
Stop Task		Stops an individual PowerExchange Listener task based on the application name that you specify.

To preview data, select **Data**.

Default is **Data**.

#### Access Methods

If you select **Tables** or **Columns** in the **Fetch** list for the **NRDB** or **NRDB2** data source type, or **Tables** is selected for the **CAPX** or **CAPXRT** data source type, the **Access Methods** list is enabled.

To return information for specific access methods, select one or more 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.

**Note:** To clear all selections in the **Access Methods** list, right-click in the **Access Methods** list and click **Deselect All**.

The following table describes the options in the **Access Methods** list:

Access Method	Description	DTLDESCRIBE Keyword
DM_ADABAS	Adabas	A
DM_CAPX	<ul style="list-style-type: none"> <li>- CAPX</li> <li>- CAPXRT</li> </ul>	C

Access Method	Description	DTLDESCRIBE Keyword
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_ISAM	C-ISAM	M
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

The following table describes the extraction map access methods that are available for the CAPX or CAPXRT data source type:

Access Method	Description	DTLDESCRIBE Keyword
DM_XDB2	DB2 extraction map	B
DM_XDB2UDB	DB2 for Linux, UNIX, and Windows extraction map	V
DM_XMSSQL	Microsoft SQL Server extraction map	L
DM_XORACLE	Oracle extraction map	P

#### 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

If you select a metadata option in the **Fetch** list, you can optionally enter filters in the following metadata filter boxes, with or without wildcards:

- ◆ For the **ADAUNLD**, **CAPX**, **CAPXRT**, **DB2390IMG**, **IMSUNLD**, **NRDB2**, or **ORACLE** data source type, enter filters in the **Schema** and **Table** boxes. The maximum length is 128 characters in the **Schema** box and 128 characters in the **Table** box.
- ◆ For the **Simulator** data source type, enter a filter in the **Table** box.
- ◆ For the **DB2**, **DB2400C**, or **DB2UDB** data source type, enter filters in the **Creator/Schema**, **DBName/Definer**, and **Table** boxes.

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.

- ◆ For the **INFORMIX**, **MSSQL**, **ODBC**, or **SYBASE** data source type, enter filters in the **Owner** and **Table** boxes. The maximum length is 128 characters in the **Owner** box and 128 characters in the **Table** box.

- ◆ For the **NRDB** data source type, enter filters in the **Schema**, **Mapname**, and **Table** boxes. In the **Mapname** box, enter the name of a PowerExchange data map that you defined. The maximum length is 10 characters in the **Schema** box, 10 characters in the **Mapname** box, and 128 characters in the **Table** box.
- In any filter box, you can use the following wildcard characters:
- ◆ Asterisk (\*) represents one or more matching characters.
  - ◆ Question mark (?) represents 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** option for those filters for which you do not want PowerExchange to use the standard case for the data source.

#### Comments

Comments for the table.

#### Extensions

Returns extended metadata, if appropriate for the select data source type.

#### Advanced

For the **CAPX**, **CAPXRT**, or **DB2390IMG** data source types.

Click **Advanced** to display the **Advanced Parameters** dialog box and enter information to override settings in the PowerExchange configuration file.

#### Application

Required for the **CAPX** and **CAPXRT** data source types.

The application name.

A valid value is a 1- to 20-character string.

#### 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 Next n Rows

Indicates how many rows to get on the next request.

**Note:** Retrieving large amounts of data can slow performance.

Valid values are from 1 through 999. Default is 10.

#### RELATED TOPICS:

- ◆ “**CAPX Advanced Parameters Dialog Box**” on page 171
- ◆ “**CAPXRT Advanced Parameters Dialog Box**” on page 172

- ♦ “Data Map Names” on page 23

## CAPX Advanced Parameters Dialog Box

Enter information to define CAPX advanced parameters in a database row test.

### Extract Type

The extraction type. The options are:

- ♦ **SL**. Starts the extraction from the point where the last extraction ended, either by restart tokens for z/OS, or by checkpoint timestamps for i5/OS.
- ♦ **RS**. Restarts the extraction from the start point of the last extraction. This option enables you to run the same extraction multiple times.

Default is **RS**.

### Image Type

The capture image type. The options are:

- ♦ **BA**. Captures the before image data, before the latest change, and the after image data, 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 in the format of an insert operation.
- ♦ **AI**. Captures the after image data.
- ♦ **TU**. Captures the before and after image data, as it occurs. The before image data enables the application 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 change can change the target record rather than delete it and insert a new record, as occurs with BA.

Default is **AI**.

### Timeout

The maximum time in seconds to wait for data before returning end-of-file (EOF).

Valid values are from 0 through 86400.

The options are:

- ♦ 0. The EOF is returned 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. The number of seconds to wait for data before returning EOF.
- ♦ 86400. The EOF is never returned. The job waits forever.

Default is 15.

### No Progress Update

Indicates whether to add database row test output to the PowerExchange CDC audit trail for the application.

The options are:

- ♦ Select this option to omit database row test output from the PowerExchange CDC audit trail.
- ♦ Clear this option to include database row test output in the PowerExchange CDC audit trail.

Default is selected.

#### **AS400 Instance**

Overrides the instance name of the DB2 for i5/OS database.

#### **Extraction Schema**

Overrides the schema specified in the extraction map.

## CAPXRT Advanced Parameters Dialog Box

Define CAPXRT advanced parameters.

#### **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 where the last extraction ended, either by restart tokens for z/OS, or by checkpoint timestamps for i5/OS.

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, before the latest change. Also, captures the after image data, 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 in the format of an insert operation.

- ◆ **TU.** Captures the before and after image data, as it occurs. The before image data enables the application 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 change can change the target record rather than delete it and insert a new record, as occurs with BA.

Default is **AI**.

#### **Timeout**

The maximum time in seconds to wait for data before returning end-of-file (EOF).

Valid values are from 0 through 86400.

Select one of the following options:

- ◆ 0. The EOF is returned 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. The number of seconds to wait for data before returning EOF.
- ◆ 86400. The EOF is never returned. The job waits forever.

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 option to omit database row test output from the PowerExchange CDC audit trail.

Clear this option to include database row test output in the PowerExchange CDC audit trail.

By default, this option is selected.

#### **AS400 Library/Journal**

Overrides the fully qualified library and journal name that you specify in the PowerExchange capture registration.

For example:

```
STQA/NEWJOURNAL
```

#### **Oracle Instance**

Overrides the Oracle instance value in the second positional parameter in the ORACLEID statement in the dbmover.cfg configuration file. For example:

```
ORACLEID=(coll_id,oracle_sid,connect_string,cap_connect_string)
```

Used in conjunction with the **Oracle Connection String** value to enable the use of a single set of capture registrations to capture data from multiple Oracle instances.

If you specify an **Oracle Instance** value but no **Oracle Connection String** value, Oracle capture 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 in the ORACLEID statement in the dbmover.cfg configuration file. For example:

```
ORACLEID=(coll_id,oracle_sid,connect_string,cap_connect_string)
```

Used in conjunction with the **Oracle Instance** value to enable the use of a single set of capture registrations to capture data from multiple Oracle instances.

If you specify an **Oracle Connection String** value but no **Oracle Instance** value, Oracle capture uses the Oracle connection string value that you define in the ORACLEID statement in the dbmover.cfg file.

#### **Oracle Schema**

Overrides the Oracle schema value in the PowerExchange registration group to enable the use of a single set of capture registrations to capture data from multiple schemas that might exist in an Oracle instance.

#### **IDMS LOGSID**

Not in use.

#### **DB2 UDB Database**

For DB2 for Linux, UNIX, and Windows. Overrides the database connection value that you specify in the PowerExchange extraction map.

#### **CAPI Connection Name**

Overrides the default database connection that you define 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 Schema**

Overrides the extraction schema in the extraction map.

#### **ADABAS File Password**

The Adabas FDT password.

Required if the Adabas FDT is password protected in Adabas.

#### **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.

#### **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 you select 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.

#### **Password**

A clear text password that is used with the user ID that you specify in the **UserID** box to authorize PowerExchange Logger for Linux, UNIX, and Windows access to extractions maps at the remote location.

Usually, this user ID is the one that you define in the CAPTURE\_NODE\_PWD parameter in the pwxcl.cfg configuration file.

## **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.

## CHAPTER 9

# PowerExchange Navigator Examples

This chapter includes the following topics:

- ◆ PowerExchange Navigator Examples Overview, 176
- ◆ Adding a Single-Record Data Map - Example, 177
- ◆ Adding a Single-Record Data Map with an Array - Example, 180
- ◆ Adding a Multiple-Record Data Map with a Complex Table - Example, 183
- ◆ Importing a COBOL Copybook into a Data Map - Example, 185
- ◆ Importing a COBOL Copybook with REDEFINES Statements - Example, 189
- ◆ Importing a PL/I Copybook - Example, 192
- ◆ Defining Table Properties in a Data Map to Normalize an Array - Example , 194
- ◆ Calling a User Exit Program by Using the PowerExchange CallProg Function - Example , 198

## 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\PowerExchangen.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 following information:

Option	Description
Schema Name	Enter <code>demo</code> .
Data Map Name	Enter <code>map1</code> .
Access Method	Select <b>SEQ</b> .
Import Record Definitions	Clear this option. You specify record definitions later.

3. Click **Next**.
4. In the **SEQ Access Method** dialog box, enter the following information:

Option	Description
File Name	Click the Browse button to browse to the demo1.dat file in the examples directory.

Option	Description
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 34

## 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 37

## 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 the following information.

Option	Description
Field Name	Enter <code>seqno</code> .
Field Type	Select <b>NUMCHAR</b> .
Precision	Enter 2.
Scale	Enter 0.
Length	Enter 2. The <b>Length</b> 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 the following information.

Option	Description
Field Name	Enter name.
Field Type	Select <b>CHAR</b> .
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 the following information.

Option	Description
Field Name	Enter gender.
Field Type	Select <b>CHAR</b> .
Length	Enter 1.

12. Click **OK**.
- The **Record** window displays the gender field.

#### RELATED TOPICS:

- ◆ “Adding a Field to a Record” on page 37

## 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 the following information:

Option	Description
Table Name	Enter table1.
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.

Option	Description
How do you want to handle multiple instances of selected records	Select <b>New Row</b> . 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 46

## 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 159

## 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 25

## 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 the following options:

Option	Description
Fields	Select <b>Groups and Arrays only</b> . 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.

Option	Description
How do you want to handle multiple instances of selected records	Select <b>New Row</b> . 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 159
- ◆ “Defining Array Properties in the Table Properties Dialog Box” on page 29

## 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, enter the following information:

Option	Description
Fields	Clear the <b>items</b> field.
How do you want to handle multiple instances of selected records	Select <b>Array</b> .
Array	Select <b>1</b> .
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 159
- ◆ “Defining Array Properties in the Table Properties Dialog Box” on page 29

# 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 25

## 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.  
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.
3. On the **Data Map** tab in the **Resource Explorer**, click the **rec2** record.  
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 159

# 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 the following information:

Option	Description
Schema Name	Enter COBOL.
Data Map Name	Enter map2.
Access Method	Select <b>SEQ</b> .
Import Record Definitions	Select this option.

3. Click **Next**.
4. In the **SEQ Access Method** dialog box, enter the following information:

Option	Description
File Name	Click the Browse button to browse to the train3.dat file in the examples directory.
Fixed	Select this option.
Size	Enter 60.

5. Click **Finish**.
6. In the **Import Copybook - Source Details** dialog box, enter the following information:

Option	Description
Source	Select <b>Local</b> .
Type	Select <b>COBOL</b> .
Start	Enter 7.
End	Enter 72.

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 110.
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 110.  
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**.

- In the **Add Table** dialog box, enter the following information:

Option	Description
Table Name	Enter <b>both</b> .
Available Records	Right-click the <b>NAME_REC</b> record and click <b>Add Record</b> . Then, right-click the <b>ACCOUNT_REC</b> record and click <b>Add Record as Child</b> .

- Click **OK**.
- 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**.
- In the **Database Row Test** dialog box, enter the following information:

Option	Description
DB Type	Select <b>NRDB</b> .
Location	Select <b>local</b> .

- Click **Go**.  
The **Database Row Test Output** window displays the output.
- Close the **Database Row Test Output** and the **Database Row Test** windows.
- Close the **Database Row Test Output** and **Database Row Test** windows.

#### RELATED TOPICS:

- ◆ “Testing a Data Map” on page 159
- ◆ “Import Copybook - Local Details Dialog Box” on page 104
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106

## 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.

- 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.
- On the **Data Map** tab in the **Resource Explorer**, click the **ACCOUNT\_REC** record.
- In the **Record** window, right-click the **POLICY\_DATE** field and click **Properties**.
- In the **Field Format** list, select **Y2-MM-D2**.
- Click **OK**.
- On the **Data Map** tab in the **Resource Explorer**, click the **both** table.
- Click **File > Database Row Test**.

- On the **Database Row Test** dialog box, enter the following information:

Option	Description
DB Type	Select NRDB.
Location	Select local.

- 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 188
- “Testing a Data Map” on page 159

## 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.

- In the **Database Row Test** dialog box, enter the following statement in the **SQL Statement** box:  

```
select * from COBOL.map2.both where itemct > 0
```
- 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.
- Close the **Database Row Test Output** and **Database Row Test** windows.

#### RELATED TOPICS:

- “Testing a Data Map” on page 159

## 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.

- On the **Data Map** tab in the **Resource Explorer**, right-click the **both** table and click **Properties**.
- In the **Table Properties** dialog box, select the **ITEMS** check box in the **Fields** list.
- Click **OK**.  
In the **Table** window, the icon for the ITEMS\_L column appears in turquoise.
- On the **Data Map** tab in the **Resource Explorer**, click the **both** table.
- Click **File > Database Row Test**.
- 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 159

# 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 the following information:

Option	Description
Schema Name	Enter COBOL.
Data Map Name	Enter map3.
Access Method	Select <b>SEQ</b> .
Import Record Definitions	Select this option.

3. Click **Next**.

4. In the **SEQ Access Method** dialog box, enter the following information:

Option	Description
File Name	Click the Browse button to browse to the train5.dat file in the examples directory.

Option	Description
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.

5. Click **Finish**.
6. In the **Import Copybook - Source Details** dialog box, enter the following information:

Option	Description
Source	Select Local.
Type	Select COBOL.
Start	Enter 7.
End	Enter 72.

7. Click **Next**.
8. 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**.
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 110.
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 110.

The **Copybook Redefines** window notifies you that the copybook contains REDEFINES statements.

The **Cobol Import** window appears.

13. To point to the line that contains the PIC S9(8) field, click **Redefinition > Next** or **Redefinition > Previous**.
14. 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 159
- ◆ “Import Copybook - Source Details Dialog Box” on page 101
- ◆ “Import Copybook - Local Details Dialog Box” on page 104
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106

## 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 the following information:

Option	Description
Field Type	Select <b>CHAR</b> .
Length	Enter 6.
Field Format	Select <b>Y2-MM-D2</b> . 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 the following information:

Option	Description
Name	Enter <b>new_date</b> .
Base Field	Select <b>MASTER_REC:MASTER_DATE</b> .

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 47
- ◆ “Viewing or Editing Properties for a Field in a Record” on page 52
- ◆ “Testing a Data Map” on page 159

## 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 the following information:

Option	Description
Schema Name	Enter <b>PLI</b> .
Data Map Name	Enter <b>map5</b> .
Access Method	Select <b>SEQ</b> .
Import Record Definitions	Select this option.

3. Click **Next**.

- In the **SEQ Access Method** dialog box, enter the following information:

Option	Description
File Name	Click the Browse button to browse to the train5.dat file in the examples directory.
Fixed	Select this option.
Size	Enter 57.
Default	Select this option.
Encoding	Select <b>EBCDIC</b> .
Skip First	Enter 1.

After you set the encoding, select the **Fixed** option again.

- Click **Finish**.
- In the **Import Copybook - Source Details** dialog box, enter the following information:

Option	Description
Source	Select <b>Local</b> .
Type	Select <b>PL/I</b> .
Start	Enter 1.
End	Enter 72.

- Click **Next**.
- In the **Import Copybook - Local PL/I 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**.
- 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 110.
- 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 110.

The **Copybook Redefines** message box notifies you that the copybook contains two definitions of the BIN\_NO field.

The **PL/I Import** window appears.

- To point to line 6, which contains the FIXED BINARY(16) field, click **Redefinition > Next** or **Redefinition > Previous**.
- Click **Import > Resume**.

The **PL/I Import** window displays the imported copybook.

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

15. Close the **PL/I Import** window.

#### RELATED TOPICS:

- ◆ “Testing a Data Map” on page 159
- ◆ “Import Copybook - Source Details Dialog Box” on page 101
- ◆ “Import Copybook - Local Details Dialog Box” on page 104
- ◆ “Import Copybook - Remote Details Dialog Box” on page 106

## 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 of 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 221 and “GetSeqWithinLevel” on page 229.

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 `r1genrow_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 `r1genseq`.
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 `GenSeqWithinLevel` function.
17. In the **Expression List** list, enter `()` at the end of the `GenSeqWithinLevel` 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 286.

## 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 displays the following results:

The screenshot shows a Windows application window titled "Database Row Test Output". The window contains a table with several rows of data. The columns are labeled: Row Number, RE..., NAME, SEX, ITE..., ITEM\_L, r1genrow\_idx, and rigenseq. The data includes names like Mark Jones, Shirley Wong, John Jackson, Donald Leary, David Wu, Jean Connor, Ronald Rose, and Betsy Martin, along with various items like apple, orange, pear, raspberry, pansy, daisy, fox, dog, cat, rabbit, horse, pony, and wolf. Two specific columns are highlighted with circles: "r1genrow\_idx" and "rigenseq". The "r1genrow\_idx" column has circled values 1, 2, and 3. The "rigenseq" column has circled values 1, 2, 3, 4, 5, 6, 7, and 8. A legend at the bottom left of the window identifies these circled numbers: "1. Results of the GenVRowKey function" and "2. Results of the GetSeqWithinLevel function".

Row Number	RE...	NAME	SEX	ITE...	ITEM_L	r1genrow_idx	rigenseq
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 r1genrow\_idx column displays the results of the GenVRowKey function, which is the number of the occurrence in the array. The rigenseq 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 user exit program returns one of the following class types:

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. Save the DLL in 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.
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 the following data set members that ship with PowerExchange on the z/OS system:

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 the following properties for the data map:

Property	Value
Schema Name	demo
Data Map Name	userexit
Access Method	SEQ

Also, select the **Import Record Definitions** option.

3. Click **Next**.

- In the **SEQ Access Method** dialog box, define the following properties for the data map:

Property	Value	Notes
File Name	<i>PWX_installation_dataset.DTLDEMO (KSDSDAT)</i>	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 (KSDSDAT)</i>
Record Format	Default	
Skip First	0	

Also, verify that the **File List Processing** option is disabled. By default, this option is disabled.

- Click **Finish**.
- In the **Import Copybook - Source Details** dialog box, define the following properties for the copybook:

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 the following properties for the copybook:

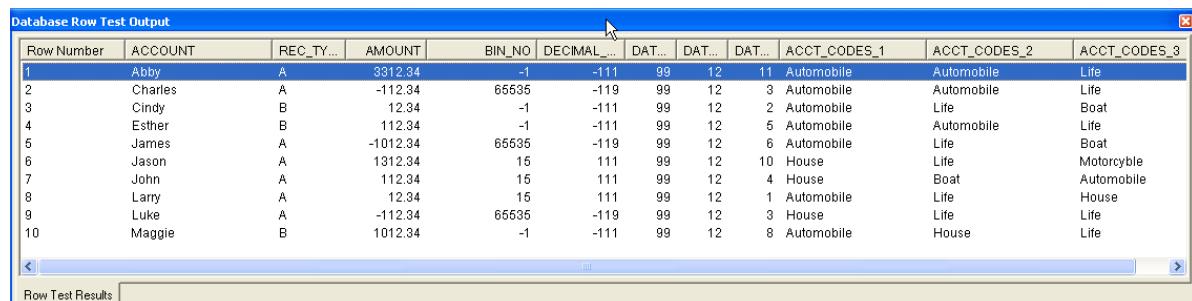
Property	Value	Notes
File Name	<i>PWX_installation_dataset.DTLDEMO (KSDSCOB)</i>	Where <i>PWX_installation_dataset</i> is the PowerExchange installation data set. For example, you might enter the following value for the file name: <i>PWX.V901.DTLDEMO (KSDSCOB)</i>
Location	Node for the z/OS system	
UserID	The user ID for the z/OS system	
Password	The password for the user ID on the z/OS system	
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.

- 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 the following results for the database row test:



The screenshot shows a Windows application window titled "Database Row Test Output". The window contains a data grid with 10 rows of data. The columns are labeled: Row Number, ACCOUNT, REC\_TY..., AMOUNT, BIN\_NO, DECIMAL\_..., DAT..., DAT..., DAT..., ACCT\_CODES\_1, ACCT\_CODES\_2, and ACCT\_CODES\_3. The data is as follows:

Row Number	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	66535	-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	66535	-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	66535	-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 UCPEP user exit program and modify it. Then, compile the user exit program and save the DLL in the PowerExchange Listener LOADLIB library.

PowerExchange ships sample user exit programs, including the UCPEP 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 UCPEP 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 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 save the DLL in 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 217.

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. Define the following properties 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. Define the following properties for the field:

Property	Value	Notes
Name	rc_rec_type	
Type	NUM32	
Precision	0	
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 <b>Expression</b> column and click the Browse button. The <b>Expression Editor</b> dialog box appears. 2. In the <b>Function List</b> list in the <b>Expression Editor</b> dialog box, double- click the CallProg function. 3. In the <b>Expression List</b> list, enter ('UCPGCLSC','UCPGCLSC','COBOL',R

Property	Value	Notes
		<p>EC_TYPE,classtype_rec_type) at the end of the CallProg function name.</p> <p>4. Click <b>Validate</b>. In the <b>Validate</b> box, the <b>No Errors</b> message appears.</p> <p>5. Click <b>OK</b>.</p>

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. Define the following properties 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. Define the following properties for the field:

Property	Value	Notes
Name	rc_bin_no	
Type	NUM32	
Precision	0	
Scale	0	
Length	0	
Phase	RW	Indicates that the operation is read or write.
Expression	CallProg('UCPGCLSC', 'UCPGCLSC','COBOL', BIN_NO,classtype_bin_no)	<p>To enter the expression for the field, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Click in the cell in the <b>Expression</b> column and click the Browse button. The <b>Expression Editor</b> dialog box appears.</li> <li>2. In the <b>Function List</b> list in the <b>Expression Editor</b> dialog box, double-click the CallProg function.</li> <li>3. In the <b>Expression List</b> list, enter ('UCPGCLSC', 'UCPGCLSC', 'COBOL', B</li> </ol>

Property	Value	Notes
		<p>IN_NO,classtype_bin_no) at the end of the CallProg function name.</p> <p>4. Click <b>Validate</b>. In the <b>Validate</b> box, the <b>No Errors</b> message appears.</p> <p>5. Click <b>OK</b>.</p>

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. Define the following properties 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. Define the following properties for the field:

Property	Value	Notes
Name	rc_decimal_no	
Type	NUM32	
Precision	0	
Scale	0	
Length	0	
Phase	RW	Indicates that the operation is read or write.
Expression	CallProg('UCPGCLSC', 'UCPGCLSC','COBOL', DECIMAL_NO,classtype_dec_no)	<p>To enter the expression for the field, complete the following steps:</p> <ol style="list-style-type: none"> <li>1. Click in the cell in the <b>Expression</b> column and click the Browse button. The <b>Expression Editor</b> dialog box appears.</li> <li>2. In the <b>Function List</b> list in the <b>Expression Editor</b> dialog box, double-click the CallProg function.</li> <li>3. In the <b>Expression List</b> list, enter ('UCPGCLSC', 'UCPGCLSC', 'COBOL', D</li> </ol>

Property	Value	Notes
		<p>DECIMAL_10,classe_type_dec_no) at the end of the CallProg function name.</p> <p>4. Click <b>Validate</b>. In the <b>Validate</b> box, the <b>No Errors</b> message appears.</p> <p>5. Click <b>OK</b>.</p>

## 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 286.

## 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, password, and the node for 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 the following user-defined fields:

Database Row Test Output								
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	Lite	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 user-defined fields display the following information:

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	0	Indicates that the UCPGCLSC user exit program ran successfully for all rows.
rc_decimal_no		
rc_rec_type		

## APPENDIX A

# PowerExchange Functions for User-Defined Fields

This appendix includes the following topics:

- ◆ PowerExchange Functions for User-Defined Fields Overview, 216
- ◆ PowerExchange Functions, 216
- ◆ Calling User-Defined Programs with the CallProg Function, 235
- ◆ Testing User-Defined Fields in a Data Map, 242

## 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.

### RELATED TOPICS:

- ◆ “Testing User-Defined Fields in a Data Map” on page 242

## PowerExchange Functions

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 linkage type is one of the following values:

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		Program return code

Linkage Type	Supported Operating Systems	Arguments	Returns
OS400	i5/OS		Address of failure code integer
PLI	z/OS		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 235

## 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
message	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.	
comparison	The type of comparison to be performed. The options are: - 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.

## 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.

### 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 datatype, 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 datatype, 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 the source field to a specified target field or to the return field.

### Syntax:

```
[result=]CopyData(source_field,target_field)
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Optional. User-defined field populated with the copied source data. Must be large enough to contain the copied data.	Any
<i>source_field</i>	Field to be copied.	A real or user-defined field of any datatype, or a value enclosed in single quotes
<i>target_field</i>	Target field into which to copy the source field. Specify a target field for a user-defined source field.	A user-defined field of any datatype <b>Note:</b> The source and target fields must be of the same datatype, length, and so on.

## 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"><li>- 0. The source field was fragmented successfully to the target field or fields.</li><li>- 1. All data in the source field was not distributed to target fields.</li><li>- 2. The last target field is of a variable datatype and was not populated with data to its maximum length.</li><li>- 3. Some target fields were not populated with data and were set to null.</li></ul>	
<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.

## GetVRowKey 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 25

## 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 RSRS data map.

### Syntax:

```
result=GetDatabaseKey()
```

Alternatively, use the following alias:

```
GetDbKey()
```

### Returns:

The following objects 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

Access Method	Data Source	Database Key Object Returned	Datatype
DL/1 BATCH	IMS	Relative byte address (RBA)	BIN 8
IMS ODBA	IMS		
SEQ	i5/OS files that are accessed sequentially	Relative record number (RRN)	NUM32
RRDS	VSAM relative record data set (RRDS) on z/OS		

**Note:** Although an IMS RBA is four 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 GetIMSRBAByLevel function rather than the GetDatabaseKey function. The GetIMSRBAByLevel function enables you to get the RBA of an unkeyed or non-unique keyed parent segment.

Use the GetIMSRBAByLevel function in IMS data maps used for bulk data movement operations or IMS synchronous CDC.

You cannot use the GetIMSRBAByLevel 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

#### RELATED TOPICS:

- ◆ “Getting the RRN or RBA for VSAM Data Set Records” on page 93
- ◆ “GetIMSRBAByLevel” on page 226

## 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 223

## GetDbKeyOfOwner

IDMS. Gets the database key of an IDMS set name. Enables you to identify the owner record without physically reading the record.

Use this function when duplicate database keys do not exist because this function does not return the IDMS page group or radix. If duplicate database keys might exist, use the GetFullDbKeyOfOwner function.

**Syntax:**

```
result=GetDbKeyOfOwner('IDMS_set_name')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the database key of the specified IDMS set name.	NUM32U or BIN4
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes. <b>Note:</b> Do not use an IDMS set name that represents a system index.	

**RELATED TOPICS:**

- ◆ “GetFullDbKeyOfOwner” on page 226

## GetFullDbKey

IDMS. Gets 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:**

```
result=GetFullDbKey()
```

The *result* parameter returns the fully qualified database key of the IDMS record, which includes the four-byte concatenated page group and radix identifier prefix. BIN8 datatype.

## GetFullDbKeyOfOwner

IDMS. Gets the fully qualified database key of an IDMS owner 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:**

```
result=GetFullDbKeyOfOwner('IDMS_set_name')
```

The following table describes the parameters:

Parameter	Description	Datatype
<i>result</i>	Returns the fully qualified database key of the IDMS owner record, which includes the four-byte concatenated page group and radix identifier prefix.	BIN8
<i>IDMS_set_name</i>	IDMS set name. Enclose this value in single quotes. <b>Note:</b> Do not use an IDMS set name for a system index.	

### RELATED TOPICS:

- ◆ “GetDbKeyOfOwner” on page 225

## GetIMSRBAByLevel

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 GetIMSRBAByLevel function rather than the GetDatabaseKey function. The GetIMSRBAByLevel function enables you to get the RBA of an unkeyed or non-unique keyed parent segment.

Use the GetIMSRBAByLevel function in IMS data maps used for bulk data movement operations or IMS synchronous CDC.

You cannot use the GetIMSRBAByLevel 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 GetIMSRBAByLevel 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=GetIMSRBAByLevel([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.	BIN 8
<i>integer</i>	Optional. The type of request , which is one of the following: - 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.	NUM32

**Note:** Although an IMS RBA is four bytes long, the GetIMSRBAByLevel 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 223

## 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:**

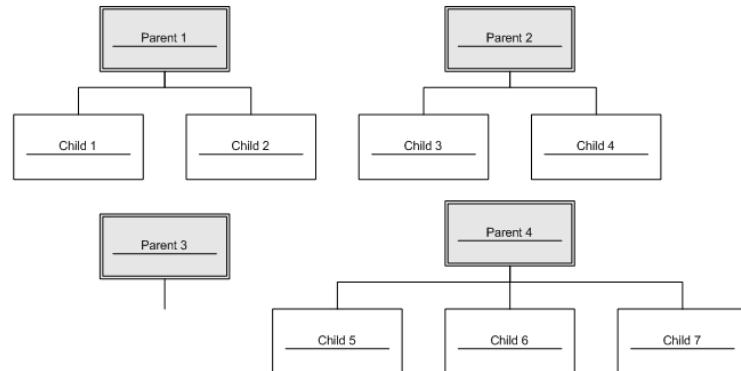
```
result=GetSeqWithinLevel()
```

**Returns:**

NUM32. The record sequence number.

## GetSeqWithinLevel Example

For the hierarchy shown in the following figure:



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 229

## LengthOf

Gets the length of a field.

### Syntax:

```
result=LengthOf(field)
```

The following table describes the parameters:

Parameter	Description	Datatype
result	Returns the length of the field.	NUM32
field	The field for which to return the length.	Any

## LTrim

Trims a specified character from the left side of a field.

### 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.

### 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: <ul style="list-style-type: none"><li>- Y. On.</li><li>- N. Off.</li></ul>	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: <ul style="list-style-type: none"><li>- 1. On.</li><li>- 0. Off.</li></ul>	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:** 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
<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 219

## 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"><li>- L. Left side.</li><li>- R. Right side.</li><li>- B. Both sides.</li></ul> 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.

### 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 program was not one of the following program types:

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. This error might indicate one of the following problems:

Problem	Linkage Type	Operating System
The symbol for the routine was not exported.	C	i5/OS
	OS400	
The routine was not defined with a #pragma fetchable.	C	z/OS

Problem	Linkage Type	Operating System
	PLI	
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:  <code>CallProg('program','subroutine','linkage',arg1,arg2)</code>  PowerExchange sets the NbrFlds parameter to 2.</p> <p><b>Note:</b> 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"> <li>- 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.</li> <li>- 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.</li> </ul> <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>
pMsgBuffer	Pointer to a character string	Output	<p>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.</p> <p><b>Note:</b> When returning error messages, do not exceed the memory buffer size allocated by the CallProg routine, which is 128 bytes.</p>
pMsgBufferSize	Integer pointer	Input and output	<p>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:</p> <pre>strncpy(pMsgBuffer, "Error in program FRED", (*pMsgBufferSize) -1);</pre>

Parameter	Datatype	Use	Description
			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: <code>char Name[50]; ... strcpy(Name, ppData[1]);</code>

## 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 218

## 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. Specify as the third argument in the CallProg function call.	Exported routines in a service program
Linux UNIX				Exported routines in an executable shared object
Windows	ucpe.c	The examples directory in the PowerExchange installation directory		Exported routines in the CDECL convention in a DLL
z/OS	UCPE	<i>hlq</i> .SRCLIB library		<ul style="list-style-type: none"> <li>- Must be fetchable from a load module.</li> <li>- 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.</li> </ul>

#### RELATED TOPICS:

- ♦ “Parameter List Passed to User-Defined Programs” on page 236

## 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,AMSGBFF          = address of message buffer
      L      R3,8(R1)           get address of argument 3
      ST    R3,AMSGBFSZ         = 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,ANUMBLEN         = length of number (always 4)

...
AMSGBFF  DS  F
AMSGBFSZ DS  F
ATEXT    DS  F
ATEXTLEN DS  F
ANUMBER  DS  F
ANUMBLEN 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 UCPERPLE 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 an non-C language programs. 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 47
- ◆ “Testing a Data Map” on page 159

## APPENDIX B

# User Access Method Programs

This appendix includes the following topics:

- ◆ User Access Method Programs Overview, 243
- ◆ Parameter Structure Passed to the User Access Method Program, 244
- ◆ Return Codes from the User Access Method Program, 246
- ◆ User Access Method Programs - Examples, 247

## 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.

PowerExchange makes the following calls to a user access method program. The user access method program can perform some or all of the following actions in response to these calls:

Call	Possible User Access Method Program Actions
OPEN	<p>Open the data file and perform initialization, as follows:</p> <ul style="list-style-type: none"><li>- Allocate storage.</li><li>- Validate and save any passed parameters.</li><li>- Initialize counters.</li><li>- verify that the data file exists.</li><li>- Open the data file.</li><li>- If the target is to be replaced, delete the output file.</li><li>- Return a success or error code to the PowerExchange Navigator.</li></ul>
READ	<p>Start or continue sequential reading, as follows:</p> <ul style="list-style-type: none"><li>- Read the data file to get the next record, using any SQL statement passed from the PowerExchange Navigator to get the appropriate record.</li><li>- 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.</li><li>- Return a success, end-of-file, or error code to PowerExchange.</li></ul>

Call	Possible User Access Method Program Actions
WRITE	<p>Start or continue sequential writing, as follows:</p> <ul style="list-style-type: none"> <li>- Get the data to write by using the record pointer and the length set in the record_len field.</li> <li>- Write the specified data.</li> <li>- Return a success or error code to the PowerExchange Navigator.</li> </ul>
CLOSE	<p>Perform clean-up operations, as follows:</p> <ul style="list-style-type: none"> <li>- Close the data file.</li> <li>- Release allocated storage.</li> <li>- Commit the database.</li> <li>- Return a success or error code to the PowerExchange Navigator.</li> <li>- If an error occurs, place message text in the message buffer. The message buffer holds a 79-character message.</li> <li>- Pass control back to the calling program.</li> </ul>

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 34

## 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	<p>Used in connection with a value of 1 in the LGCODE variable, and indicates how the data file is opened, as follows:</p> <ul style="list-style-type: none"> <li>- 1. Read.</li> <li>- 2. Write.</li> <li>- 3. Update.</li> <li>- 4. Append.</li> </ul>	Not applicable	Binary	Word
LPCODE	<p>Indicates the function to be performed by the user access method program, as follows:</p> <ul style="list-style-type: none"> <li>- 1. Open.</li> <li>- 2. Close.</li> <li>- 3. Read.</li> <li>- 4. Write.</li> </ul>	Not applicable	Binary	Word

Variable Name	Description	Corresponding Field in the PowerExchange Navigator GUI	Type	Size
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 <b>USER Access Method</b> 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 <b>USER Access Method</b> 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
LGU1LEN	The length of the data entered in the <b>String 1</b> box in the <b>USER Access Method</b> dialog box.	Derived from the <b>String 1</b> value in the <b>USER Access Method</b> dialog box	Binary	Word
LGU2LEN	The length of the data entered in the <b>String 2</b> box in the <b>USER Access Method</b> dialog box.	Derived from the <b>String 2</b> value in the <b>USER Access Method</b> dialog box	Binary	Word
LGFNLEN	The length of the data file name entered in the <b>File Name</b> box in the <b>USER Access Method</b> dialog box.	Derived from the <b>File Name</b> value in the <b>USER Access Method</b> dialog box	Binary	Word
LGUIDLEN	The length of the data entered in the <b>User ID</b> box in the <b>Data Map Remote Node</b> dialog box.	Derived from the <b>User ID</b> value in the <b>Data Map Remote Node</b> dialog box	Binary	Word
LGPWDLEN	The length of the data entered in the <b>Password</b> box in the <b>Data Map Remote Node</b> dialog box.	Derived from the <b>Password</b> value in the <b>Data Map Remote Node</b> dialog box	Binary	Word
LGSQLEN	The length of the data entered in the <b>SQL Statement</b> box in the <b>Database Row Test</b> dialog box.	Derived from the <b>SQL Statement</b> value in the <b>Database Row Test</b> dialog box	Binary	Word
LGU1PTR	A pointer to the data stored for String 1.	<b>String 1</b> value in the <b>USER Access Method</b> dialog box	Pointer	Word
LGU2PTR	A pointer to the data stored for String 2.	<b>String 2</b> value in the <b>USER Access Method</b> dialog box	Pointer	Word
LGFNPTR	A pointer to the data stored for File Name.	<b>File Name</b> value in the <b>USER Access Method</b> dialog box	Pointer	Word

Variable Name	Description	Corresponding Field in the PowerExchange Navigator GUI	Type	Size
LGUIDPTR	A pointer to the data stored for User ID.	User ID value in the <b>Data Map Remote Node</b> dialog box	Pointer	Word
LGPWDPTR	A pointer to the data stored for Password.	Password value in the <b>Data Map Remote Node</b> dialog box	Pointer	Word
LGSQLPTR	A pointer to the data stored for SQL Statement.	SQL Statement value in the <b>Database Row Test</b> 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
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 164
- ◆ “Data Map Properties - USER Access Method ” on page 266
- ◆ “Sending a Data Map to a Remote System” on page 58

## 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, 248
- ◆ Adding an Application Group, 249
- ◆ Viewing or Editing Properties for an Application Group, 250
- ◆ Resetting the Extraction Start Point for an Application, 250
- ◆ Deleting an Application Group, 251

## 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, Extraction Groups, and Application Groups” on page 115

## Adding an Application Group

An application group is a named group of applications.

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 completing the following steps.

To add an application group:

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

User-defined name for the application group. Maximum length is 16 characters.

#### Location

Location of the source for which you want to extract change data. Select **local** if the source is on the PowerExchange Navigator 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**.

#### UserID

For an i5/OS or z/OS source, enter an operating system user ID, which is required if security is enabled.

For a DB2 for Linux, UNIX, and Windows, Microsoft SQL Server, or Oracle source, enter a valid database user ID, if required.

#### Password

Password for the user ID.

If you specify a user ID and 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, Extraction Groups, and Application Groups” on page 115

## Viewing or Editing Properties for an Application Group

You can view or edit properties for an application group.

You can edit the properties for an application group to change its name, location, or the user ID and password used to access the source.

To view or edit properties for a application group:

1. Open the application group.

The **Application Group** tab in the **Resource Inspector** displays the following properties for the application group:

Option	Description
Name	User-defined name for the application group.
Location	Location of the source for which you want to capture changes.
UserID	User ID to access the source or the operating system, depending on the access method.
Password	Password for the user ID.

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. If prompted, click **Yes** to confirm any changes.

The **Resource Inspector** updates any fields that you changed.

#### RELATED TOPICS:

- ♦ “Adding an Application Group” on page 249

## 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 , 252
- ◆ Data Map Properties - DATACOM Access Method , 254
- ◆ Data Map Properties - DB2 Access Method , 254
- ◆ Data Map Properties - DB2UNLD Access Method , 255
- ◆ DB2 Table Filter Dialog Box, 256
- ◆ DB2 Table Filter - Available Tables Dialog Box, 257
- ◆ Data Map Properties - DL/1 Batch or IMS ODBA Access Method , 257
- ◆ Data Map Properties - ESDS Access Method , 258
- ◆ Data Map Properties - IDMS Access Method , 259
- ◆ Data Map Properties - ISAM Access Method , 260
- ◆ Data Map Properties - KSDS Access Method , 260
- ◆ Data Map Properties - MQSeries Access Method , 261
- ◆ Data Map Properties - RRDS Access Method , 262
- ◆ Data Map Properties - SEQ Access Method , 263
- ◆ Data Map Properties - TAPE Access Method , 265
- ◆ Data Map Properties - USER Access Method , 266
- ◆ Data Map Properties - Data Defaults , 268
- ◆ Data Map Properties - Data Checking , 268
- ◆ Data Map Properties - Record Level Exit, 270
- ◆ Data Map Properties - Keys , 271

## 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 31
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

# 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.

## RELATED TOPICS:

- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

# 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 31
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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 sign.

#### **DB Instance**

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.

#### **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 31
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## DB2 Table Filter Dialog Box

View or edit table filtering criteria for a DB2 catalog or DB2 unload file data map.

#### **DB Instance**

The DB2 database instance name. Valid characters are ASCII characters and the pound and yen sign.

#### **Location**

Location of the data source. Select **local** if the data source is on the PowerExchange Navigator 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**.

#### **UserID**

The database user ID to access the database, if required.

#### **Password**

Password for the user ID.

#### **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:

- ◆ Asterisk (\*) represents one or more matching characters.
- ◆ Question mark (?) represents 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 31
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## Data Map Properties - DL/1 Batch or IMS ODBA Access Method

View or edit properties for an IMS DL/1 batch or IMS ODBA data map.

#### IMS SSID

The IMS subsystem ID.

- ◆ For IMS ODBA, specify the IMS ID for the control region. Also, this value must match the value of the IMSID parameter in the DBMOVER configuration member.
- ◆ For DL/1 batch, this value must match the value of the IMSID parameter in the DBMOVER configuration member if you plan to read IMS unload data sets.

**Note:** For DL/1 batch, the DTL.DBWRITE.IMS resource profile that controls write access to databases accessed through DL/1 batch does not use this IMS SSID value.

For IMS ODBA, the DTL.DBWRITE.IMS.imsid controls write access to databases accessed through ODBA. Verify that the imsid value matches the IMS SSID value for this field.

Maximum length is four characters.

**DBD Name**

The name of the IMS database description (DBD) for the IMS source. Specify this value if you plan to read IMS unload data sets.

**PCB Number**

For DL/1 batch, the relative PCB number for the database. If the PSB specifies COMPAT=YES, add one to the PCB number.

Specify this value if you plan to read the IMS database by using a netport job that runs either DL/1 or BMP.

**PSB Name**

For IMS ODBA, specify the name of a PSB that contains the DBD to access this database.

Valid characters are alphanumeric characters. Maximum length is eight characters.

**PCB Name**

For IMS ODBA, for the specified PSB, enter the named PCB that references the specified DBD. Enter either a label for the PCB or the PCB name parameter in the **PCB Name** box.

**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 include 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 31
- ◆ “Associating Multiple Data Files with a Data Map” on page 59
- ◆ “Improving Bulk Read Performance for VSAM Data Sets” on page 91
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **Data Map Properties - ISAM Access Method**

View or edit properties for a C-ISAM data map.

#### **File Name**

The full path and file name of the C-ISAM data file.

#### **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:**

- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 31
- ◆ “Improving Bulk Read Performance for VSAM Data Sets” on page 91
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 31
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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 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 31
- ◆ “Improving Bulk Read Performance for VSAM Data Sets” on page 91
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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:

- ◆ **VBI2**. Data length is 2-byte value, where length "I"ncludes 2-byte length.
- ◆ **VBI4**. Data length is 4-byte value, where length "I"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**. In the data file, no record boundaries exist and the records are not fixed length.
- ◆ **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. 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

For a data file, select this option.

Optionally, select an encoding in the **Encoding** list and a code page in the **Codepage** list.

Default is **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 required, 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 the data map.

#### **View File**

Displays the data file.

Enabled after you enter a file name.

### **RELATED TOPICS:**

- ◆ “Code Pages” on page 31
- ◆ “Associating Multiple Data Files with a Data Map” on page 59
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 DBMOVER configuration member.

#### **Unit**

This value must match the value of the UNIT parameter in the DBMOVER configuration member.

#### **Volser**

This value must match the value of the VOLSER parameter in the DBMOVER 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 59
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 31
- ◆ “Adding a Data Map” on page 34
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

# 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:

1990401

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 55

# 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.

#### **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 32
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 27
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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 55

## APPENDIX E

# Record, Field, and Table Properties

This appendix includes the following topics:

- ◆ Record Properties - Name, 272
- ◆ Record Properties - Name (Datacom), 273
- ◆ Record Properties - Name (IDMS), 274
- ◆ Sort Element Details Dialog Box (IDMS), 275
- ◆ Record Properties - Import Details , 276
- ◆ Field Properties - Name , 276
- ◆ Field Properties - Array, 282
- ◆ Field Properties - Extra Properties , 283
- ◆ Field Properties - Field Checking , 284
- ◆ Field Properties - Code Page, 284
- ◆ Field Properties - Variable Length, 285
- ◆ Table Properties - Definition, 286
- ◆ Table Properties - Filter , 288
- ◆ Table Properties - IMS Options , 288
- ◆ Table Properties - Keys , 289
- ◆ Find Dialog Box, 290

## Record Properties - Name

View or edit a record name.

For IMS records, view or edit 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.

### **RELATED TOPICS:**

- ◆ “Adding a Data Map Record” on page 37
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 37
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

# 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 37
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 37
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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 37
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## 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	N/A	N/A	1 through 32767
CHAR	N/A	N/A	1 through 32767
DOUBLE	N/A	N/A	N/A
FLOAT	N/A	N/A	N/A
GROUP	N/A	N/A	N/A
NUM16	N/A	0 through 5	N/A
NUM16U	N/A	0 through 5	N/A
NUM32	N/A	0 through 10	N/A
NUM32U	N/A	0 through 10	N/A
NUM64	N/A	0 through 19	N/A
NUM64U	N/A	0 through 19	N/A
NUM8	N/A	0 through 3	N/A
NUM8U	N/A	0 through 3	N/A
NUMCHAR	1 through 100	0 through 50	1 through 100
PACKED	1 through 31	0 through 31	N/A
SPACKED	1 through 31	0 through 31	N/A
UPACKED	1 through 31	0 through 31	N/A
UZONED	1 through 31	0 through 31	N/A
VARBIN	N/A	N/A	1 through 32767

Field Type	Precision	Scale	Length
VARCHAR	N/A	N/A	1 through 32767
ZONED	1 through 31	0 through 31	N/A

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)	N/A	N/A	32767
CHAR	X(nn)	N/A	N/A	32767
DOUBLE	COMP-2	N/A	N/A	8
FLOAT	COMP-1	N/A	N/A	4
GROUP	X(nn)	N/A	N/A	N/A
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 <sup>1</sup>	S9(nn) COMP-3	31	0 through 31	N/A
SPACKED <sup>2</sup>	N/A	31	0 through 31	N/A
UPACKED <sup>3</sup>	9(nn) COMP-3	31	0 through 31	N/A

Field Type	COBOL PIC Clause	Maximum Precision	Scale	Fixed Length
UZONED	9(nn)	31	0 through 31	N/A
VARBIN	X(nn) set by 2-byte length field	N/A	N/A	32767
VARCHAR	X(nn) set by 2-byte length field	N/A	N/A	32767
VARCHAR	X(nn) set by 2-byte length field	N/A	N/A	32767
ZONED	9(nn)	31	0 through 31	N/A
<p>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'.</p> <p>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'.</p> <p>3. A UPACKED 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'.</p>				

#### 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 UPACKED 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 Y4MMD2HHISSN12 (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
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.	<b>Examples:</b> 990823 for 23rd Aug 1999 1000914 for 14th Sep 2000

Character Mask	Description	Valid Input Values
		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

Allows empty input fields. If the input field has no data, the output column is empty and marked as NULL. You must select this option for VARCHAR fields that you define as arrays.

Clear this option to prohibit input fields without data. Default is cleared.

#### Nullable Field

Allows a NULL value for a field.

Clear this option to use the characters that the data source uses to indicate a NULL value. Default is cleared.

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 37
- ◆ “Viewing or Editing Properties for a Field in a Record” on page 52

## 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 37
- ◆ “Viewing or Editing Properties for a Field in a Record” on page 52

## 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 52

## 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 52
- ◆ “Adding a Field to a Record” on page 37

# 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 52
- ◆ “Data Checking” on page 32
- ◆ “Adding a Field to a Record” on page 37

# Field Properties - Code Page

View or edit code page properties for a field.

#### **Use Datamap Default Codepage**

PowerExchange uses the default code page for the operating system.

Default is selected.

#### **Use Selected Codepage**

Enables selection of a code page. The code page describes the character set for the character data in the database.

**Note:** Multibyte code pages are available for DB2, VSAM KSDSs, sequential data sets, and flat files.

#### **RELATED TOPICS:**

- ◆ “Viewing or Editing Properties for a Field in a Record” on page 52
- ◆ “Code Pages” on page 31
- ◆ “Adding a Field to a Record” on page 37

## 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 192 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.

## **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**.

## **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 46
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55
- ◆ “Adding a Single-Record Data Map with an Array - Example” on page 180
- ◆ “Adding a CCK Field to an IMS Record” on page 44

## **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 46
- ◆ “Defining Data Map Preferences” on page 18
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55

## **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 46
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55
- ◆ “Viewing or Editing IMS Options for a Table” on page 86

## **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 46
- ◆ “Viewing or Editing Properties for a Data Map Table” on page 55
- ◆ “Viewing or Editing Properties for a Column in a Table” on page 56

# 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 49
- ◆ “Searching for a Table in a Personal Metadata Profile” on page 14
- ◆ “Searching for a Column in a Table in a Personal Metadata Profile” on page 14

# INDEX

## 2-tier names

  data maps 23

## 3-tier names

  data maps 23

## A

### access method types

  by data source 24, 69  
  choosing for data maps 24

### accessing

  DB2 unload data from multiple tables 77  
  IMS data through DL/1 batch data maps 82  
  IMS data through IMS ODBA data maps 82

### ADABAS access method

  data map properties 252

### Adabas data maps

  adding 71  
  adding keys to tables in 72  
  importing field definition tables (FDTs) into 26, 100  
  importing keys into 26  
  overview 70

### Adabas databases

  data maps for 252

### Adabas field definition tables (FDTs)

  displaying in data maps 101  
  importing into data maps 26, 100  
  refreshing in data maps 101

### Adabas files

  data maps for 24

### Adabas keys

  adding to data map tables 46, 72  
  importing into data maps 26

### ADACMP copybooks

  data map types supported in 96

### adding

  application groups 249  
  before image (BI)columns to extraction map columns 139  
  CALC elements to data map records (IDMS) 41, 42  
  capture registrations 118  
  change indicator columns (CI) to extraction map columns 139  
  columns to capture registrations 129  
  columns to data map tables 47  
  columns to extraction processing 140  
  data map records 37  
  data maps 34, 177  
  data maps (Adabas) 71  
  data maps (Datacom) 73  
  data maps (DB2 catalog) 74  
  data maps (DB2 unload file) 76, 77  
  data maps (IDMS) 78  
  data maps (IMS) 83, 87, 89  
  data maps (VSAM) 90  
  data maps with complex tables 183  
  elements to data map records (Datacom) 39

expressions to data map records (IDMS) 80

expressions to data map records (IMS) 84

extraction groups 135

extraction maps 137

fields to data map records 37, 178

key segments to data map records (Datacom) 41

keys to data map records (Datacom) 39

keys to data map tables (Adabas) 46, 72

owner record and sets to data map records (IDMS) 41

owner records and sets to data map records (IDMS) 42

personal metadata profiles 153

registration groups 116

resource configurations 16

search fields to data map records (IMS) 45

tables to data maps 46, 179

user-defined fields to data map records 38

### application groups

  adding 249

  deleting 251

  editing properties for 250

  overview 248

  viewing properties for 250

### Application window

  overview 3

### applications

  overview 248

  resetting extraction start point for 250

### arranging windows

  PowerExchange Navigator 5

### arrays

  defining a new row for each element in 181

  defining fields in data map records as 282

  defining in data maps 28

  defining single row for all elements in 182

  overview 29

### arrays with multiple occurrences

  adding user-defined fields to demonstrate the normalization of 196

  defining data map table properties for 194

### Assembler user-defined programs

  example 240

  linkage 239

  operating system support 239

### associating

  capture registrations with extraction maps 141

  data files with data maps 25, 181, 183

  file-list files with data maps 60

  multiple data files with data maps 25, 60

## B

### before image (BI) columns

  adding to extraction map columns 139

  viewing for extraction maps 148

### bit settings at offset in fields

  getting for data map records 232

bulk read performance  
improving for VSAM data sets 91

## C

C-ISAM files  
  data maps for 24, 260

CALC elements in data map records (IDMS)  
  adding 41, 42  
  deleting 67  
  editing properties for 54  
  overview 33  
  viewing properties for 54

CallProg function  
  calling in user-defined fields in data maps 198  
  in an expression in a user-defined field 211

CAMAPS file 132

Capture Registration window  
  overview 3

capture registrations  
  adding 118  
  adding columns to 129  
  changing for extraction maps 141  
  condense options for 123  
  deleting 129  
  deleting all in a registration group 130  
  editing properties for 125  
  editing to change columns in 129  
  finding columns in 290  
  overview 112  
  removing columns from 129  
  searching for columns in 14  
  status options for 122  
  tag names for 113  
  viewing 124  
  viewing properties for 125

capture registrations (Datacom)  
  viewing elements in 128

capture registrations (IDMS)  
  viewing parent hierarchy for base records in 129

CAPTURE\_NODE\_EPWD parameter  
  generating encrypted password for 21

CAPX advanced parameters  
  in database row tests 171

CAPXRT advanced parameters  
  in database row tests 172

CCK fields  
  adding to data map records (IMS) 44  
  deleting from data map records (IMS DL/1 batch) 67  
  deleting from data map records (IMS ODBA) 67  
  editing in data map records (IMS) 44, 55  
  viewing in data map records (IMS) 55

CCK fields in data map records (IMS DL/1 batch)  
  overview 34

CCK fields in data map records (IMS ODBA)  
  overview 34

change data capture  
  generating restart tokens for 163

change indicator (CI) columns  
  adding to extraction map columns 139  
  viewing for extraction maps 148

changing  
  capture registrations for extraction maps 141  
  columns in capture registrations 129  
  view type for Resources window 4

Check function  
  example 218

choosing  
  access methods for data maps 24

CL user-defined programs  
  example 240  
  linkage 240

closing  
  PowerExchange resources 15

COBOL copybooks  
  data map types supported in 96  
  importing into data maps 185, 189, 195, 199

COBOL user-defined programs  
  operating system support 238

code pages  
  for fields in data map records 285  
  overview 31  
  supported for PowerExchange 31  
  supported in database row tests 159  
  user-defined 32

columns in capture registrations  
  adding 129  
  removing 129  
  searching for 14

columns in data map tables  
  adding 47  
  copying 56  
  deleting 66  
  editing properties for 56  
  overview 27  
  printing 64  
  searching for 13  
  selecting (DB2 catalog) 109  
  viewing 49  
  viewing properties for 56

columns in extraction maps  
  adding 140  
  adding before image (BI) columns to 139  
  adding change indicator (CI) columns to 139  
  removing 140  
  removing before image (BI) columns from 139  
  removing change indicator (CI) columns from 139  
  searching for 14  
  viewing 139

columns in personal metadata profile tables  
  searching for 14

commands  
  issuing in a database row test 162

compiling  
  user exit programs 201

complex tables  
  adding to data maps 46  
  data sources that support 27  
  defining 89  
  defining in data maps 183  
  in data maps 27

Concat function  
  example 219

concatenating  
  fields in data map records 219

condense options  
  for capture registrations 123

configuring  
  PowerExchange DBMOVER member on z/OS to access IMS data 82  
  PowerExchange dbmover.cfg file on Windows to access IMS data 83

control interval access  
  for VSAM data sets 92

converting  
  characters to lowercase in fields in data map records 234  
  characters to uppercase in fields in data map records 234

copybook types  
     supported in data maps 96

copybooks  
     ADACMP 96  
     COBOL 96, 185, 189, 195, 199  
     Datacom 107  
     DATACOM 96  
     DB2 catalog 96, 105, 108  
     DBD 96  
     field definition table (FDT) 96  
     IDMS 96, 109  
     importing into data map records 99  
     importing into data map segments 99  
     importing into data maps 97, 106  
     importing into data maps (IMS) 89  
     local details for 104  
     Natural data definition module (DDM) 96  
     overview 25, 95  
     PL/I 96  
     PREDICT 96  
     remote details 106  
     source details for importing 101  
     TEXT 96

CopyData function  
     in an expression in a user-defined field 211

copying  
     field data in data map records 219  
     items in data maps 56

creating  
     data maps from extraction maps 145  
     file-list files 60  
     shared resource configurations 17  
     user exit programs 201

current configuration  
     selecting 17

cutting and pasting  
     columns in data map tables 56  
     data map records 56  
     data map tables 56  
     fields in data map records 56

**D**

data  
     logon credentials for 57

data buffers  
     for VSAM data sets 91

data checking  
     data map properties 269  
     for fields in data map records 284  
     overview 32

data defaults  
     defining for data maps 268

data description block (DBD) copybooks  
     data map types supported in 96

data files  
     associating multiple data files with data maps 60  
     associating multiple with data maps 25  
     associating with data maps 181, 183  
     display modes for 49  
     file-list processing for 25  
     viewing 49

data flow information  
     getting for data map records (IDMS) 224

data map access methods  
     overview 69  
     types 24

data map properties  
     using to get RBA for VSAM data sets 93  
     using to get RRN for VSAM data sets 93

data map records  
     adding 37, 178  
     adding fields to 37, 178  
     arrays in 182  
     calling user-defined programs in 217  
     concatenating fields in 219  
     copying 56  
     copying data from one field to another 219  
     defining hierarchy of records in 184  
     deleting 65  
     editing properties for 52, 272, 276  
     filtering data in 181  
     finding fields in 290  
     getting bit setting at specified offset in a field 232  
     getting database keys for 223  
     getting source file name for 222  
     importing copybooks into 99  
     printing 63  
     record IDs in 183  
     searching for fields in 13  
     splitting data in fields in 220  
     viewing properties for 52, 272

data map records (Datacom)  
     adding elements to 39  
     properties 273  
     viewing properties for elements in 53

data map records (IDMS)  
     adding expressions to 80  
     getting data flow information for 224  
     getting fully qualified database key for 225  
     getting page group and radix for 228  
     getting page groups for 227  
     getting radix for 229  
     getting sequence numbers for 230

data map records (IMS DL/1 batch)  
     deleting CCK fields from 67

data map records (IMS ODBA)  
     deleting CCK fields from 67

data map records (IMS)  
     adding CCK fields 44  
     adding CCK fields to 44  
     adding expressions to 84  
     adding search fields 44  
     adding search fields to 45  
     CCK fields in 55  
     deleting search fields from 68  
     editing CCK fields in 44  
     editing properties for search fields in 55  
     editing search fields in 44  
     getting sequence numbers for 230  
     getting the RBA in 84  
     viewing properties for search fields in 55

data map segments  
     importing copybooks into 99

data map table properties  
     using to normalize arrays with multiple occurrences 194

data map tables  
     adding 46, 179  
     adding columns to 47  
     complex 27, 46  
     copying 56  
     deleting 66  
     deleting columns from 66  
     editing properties for 55, 286  
     filtering columns in 288

finding columns in 290  
 overview 27  
 printing 63  
 properties 197, 214  
 refreshing with missing columns 38  
 searching for columns in 13  
 simple 27  
 table definition properties 286  
 viewing 49  
 viewing properties for 55, 286  
**data map tables (Adabas)**  
 adding keys to 46, 72  
**data map tables (DB2 catalog)**  
 selecting columns in 109  
**data maps**  
 2-tier names 23  
 3-tier names 23  
 access method types 24  
 adding 34, 177  
 adding fields to records in 37  
 adding records to 37, 178  
 adding user-defined fields to records in 38  
 adding with an array 195, 199  
 applying date mask to GROUP fields 191  
 associating data files with 181  
 associating file-list files with 25, 60  
 copybook types supported in 96  
 creating from extraction maps 145  
 data checking 269  
 data defaults 268  
 data sources that are supported for 24  
 defining arrays in 28  
 defining preferences for 18  
 deleting 64  
 deleting fields from records in 65  
 deleting fields from segments in 65  
 deleting records from 65  
 deleting segments from 65  
 deleting tables from 66  
 editing properties for 51  
 for Adabas databases 252  
 for C-ISAM files 260  
 for Datacom files 254  
 for DB2 databases 254  
 for flat files 263  
 for IDMS databases 259  
 for IMS databases 257  
 for MQSeries message queues 261  
 for sequential data sets 263  
 for specific data sources 69  
 for tape data sets 265  
 for user access method programs 266  
 for VSAM entry-sequenced data sets (ESDSs) 258  
 for VSAM keyed-sequenced data sets (KSDSs) 261  
 for VSAM relative record data sets (RRDSs) 262  
 importing COBOL copybooks into 185, 189, 195, 199  
 importing copybooks into 97  
 importing from remote systems 59  
 importing PL/I copybooks into 192  
 logon credentials for 57  
 merging with extraction maps 142  
 names for 23, 34  
 overview 22, 23  
 printing 61, 62  
 record-level exit programs defined for 270  
 records in 26  
 segments in 26  
 sending to remote systems 33, 58  
 single-record 177  
 sorting records in 51  
 sorting tables in 51  
 testing 159  
 testing user-defined fields in 242  
 USER access method 243  
 viewing properties for 51  
**data maps (Adabas)**  
 adding 71  
 adding keys to tables in 72  
 importing field definition tables (FDTs) 100  
 importing keys into 26  
 overview 70  
**data maps (Datacom)**  
 adding 73  
 adding elements to records in 39  
 adding key segments to records in 39, 41  
 adding keys to records in 39  
 deleting elements from records in 66  
 deleting key segments from records in 67  
 deleting keys from records in 66  
 elements in records in 33  
 keys in records in 33  
 overview 72  
**data maps (DB2 catalog)**  
 adding 74  
 overview 74  
**data maps (DB2 unload file)**  
 adding 76, 77  
 overview 75  
**data maps (DB2)**  
 editing record layout in 75  
 filtering tables in 256, 257  
**data maps (IDMS)**  
 adding 78  
 adding CALC elements to records in 41, 42  
 adding owner records and sets to records in 42  
 CALC elements in 33  
 deleting CALC elements from records in 67  
 deleting owner records and sets from records in 67  
 editing properties for records in 274  
 overview 78  
 owner records and sets in 33  
 sort element details for owner records and sets (IDMS) 275  
 viewing properties for records in 274  
**data maps (IMS)**  
 adding 83, 87, 89  
 editing properties for 85  
 editing properties for tables in 86, 289  
 examples 87  
 importing copybooks into 89  
 overview 80  
 sending to remote systems 85  
 testing 87  
 viewing hierarchy created by DBD source import 81  
 viewing properties for tables in 86, 289  
 viewing records in 81  
 viewing tables in 81  
**data maps (VSAM)**  
 adding 90  
 overview 90  
**data maps, exported**  
 deleting 64  
**data sources**  
 supported for data maps 24, 69  
 that support complex tables 27  
**database keys**  
 getting for data map records 223

getting for data map records (IDMS) 225  
 getting for owner records and sets in data map records (IDMS) 225, 226  
**database row test**  
     to test normalization of an array 197  
     to test results of user exit program in data map record 214  
**Database Row Test dialog box**  
     description 164  
**Database Row Test Output window**  
     hiding 159  
     showing 159  
**database row tests**  
     CAPX advanced parameters in 171  
     CAPXRT advanced parameters in 172  
     Database Row Test dialog box 164  
     DB2390IMG advanced parameters in 174  
     generating restart tokens in 163  
     issuing PowerExchange Listener commands 162  
     multibyte data in 159  
     nonstandard code page data in 159  
     overview 158  
     running 158  
         running for user-defined fields 242  
**DATACOM access method**  
     data map properties 254  
**Datacom capture registrations**  
     viewing elements in 128  
**DATACOM copybooks**  
     data map types supported in 96  
**Datacom data map records**  
     adding elements to 39  
     properties 273  
**Datacom data maps**  
     adding 73  
     adding elements to records in 39  
     adding key segments to records in 39, 41  
     adding keys to records in 39  
     deleting elements from records in 66  
     deleting key segments from records in 67  
     deleting keys from records in 66  
     elements in records in 33  
     key segments in records in 33  
     keys in records in 33  
     overview 72  
**Datacom files**  
     data maps for 24, 254  
**date masks**  
     applying to fields in data map records 187  
     applying to GROUP fields in data map records 191  
**DB2 access method**  
     data map properties 254  
**DB2 catalog copybooks**  
     data map types supported in 96  
     local details 105  
**DB2 catalog data maps**  
     adding 74  
     overview 74  
**DB2 data maps**  
     editing record layout in 75  
     filtering tables in 257  
**DB2 databases**  
     data maps for 24, 254  
**DB2 unload data from multiple tables**  
     accessing 77  
**DB2 unload file data maps**  
     adding 76, 77  
     overview 75  
**DB2 unload files**  
     data maps for 24  
**DB2\_BIN\_CODEPAGE statement**  
     DBMOVER configuration file 159  
**DB2390 IMG advanced parameters**  
     in database row tests 174  
**DB2CODEPAGE statement**  
     DBMOVER configuration file 159  
**DB2UNLD access method**  
     data map properties 255  
**DBMOVER configuration file**  
     generating encrypted passwords for 21  
**DBMOVER configuration statements**  
     DB2\_BIN\_CODEPAGE 159  
     DB2CODEPAGE 159  
**dbmover.cfg file**  
     configuring on Windows to access IMS data 83  
**defining**  
     complex tables 89  
     condense options for capture registrations 123  
     data map preferences 18  
     IDMS data map preferences 20  
     layout of data in source or target 26  
     resource configurations 16  
**deleting**  
     all registrations in a registration group 130  
     application groups 251  
     CALC elements from data map records (IDMS) 67  
     capture registrations 129  
     columns from data map tables 66  
     data map records 65  
     data map segments 65  
     data map tables 66  
     data maps 64  
     data maps, exported 64  
     elements from data map records (Datacom) 66  
     extraction groups 147  
     extraction maps 146  
     extraction maps in an extraction group 147  
     fields from data map records 65  
     fields from data map segments 65  
     items in data maps 64  
     key segments from data map records (Datacom) 67  
     keys from data map records (Datacom) 66  
     owner records and sets from data map records (IDMS) 67  
     personal metadata profiles 157  
     PowerExchange resources 15  
     registration groups 130  
     resource configurations 17  
     search fields from data map records (IMS) 68  
**display modes**  
     for data files 49  
**displaying**  
     Adabas field definition tables (FDTs) 101  
     multiple output rows for items in OCCURS DEPENDING ON clauses 188  
**DL/1 Batch access method**  
     data map properties 257  
**DTL\_\_BI\_columnname column**  
     described 149  
**DTL\_\_CAPXACTION**  
     described 149  
**DTL\_\_CAPXCASDELIND**  
     described 149  
**DTL\_\_CAPXRESTART1 column**  
     described 149  
**DTL\_\_CAPXRESTART2 column**  
     described 149

DTL\_\_CAPXRRN column  
described 149  
DTL\_\_CAPXTIMESTAMP column  
described 149  
DTL\_\_CAPXUOW column  
described 149  
DTL\_\_CAPXUSER column  
described 149  
DTL\_\_CI\_columnname column  
described 149  
DTLCAMAP file 132  
DTLDSN SQL escape sequence 61  
DTLFILELIST SQL escape sequence 61  
DTLREXE utility -epwd option  
generating encrypted password for 21  
DTLUAPPL utility EPWD option  
generating encrypted password for 21  
DTLUCBRG utility EPWD option  
generating encrypted password for 21  
DTLUCUDB utility EPWD option  
generating encrypted password for 21  
DTLURDMO utility EPWD option  
generating encrypted password for 21  
DTLUTSK utility EPWD option  
generating encrypted password for 21

## E

editing  
capture registrations to change columns in 129  
CCK fields in data map records (IMS) 55  
properties for search fields in data map records (IMS) 55  
record layout in data maps (DB2) 75  
resource configurations 16  
editing properties  
application groups 250  
CALC elements in data map records (IDMS) 54  
capture registrations 125  
columns in data map tables 56  
data map records 52, 272  
data map records (IDMS) 79, 274  
data map tables 55  
data map tables (IMS) 86, 289  
data maps 51  
data maps (IMS) 85  
definitions for data map tables 286  
extraction groups 143  
fields in data map records 52  
key segments in data map records (Datacom) 53  
keys in data map records (Datacom) 53  
keys in data map tables (Adabas) 289  
keys in data map tables (Datacom) 289  
owner records and sets in data map records (IDMS) 54  
personal metadata profiles 156  
registration groups 124  
elements  
viewing properties for in data map records (Datacom) 53  
elements in data map records (Datacom)  
adding 39  
deleting 66  
overview 33  
elements in owner records and sets (IDMS)  
properties 275  
encrypted passwords  
generating 21  
errors  
user-defined programs 235

ESDS access method  
data map properties 258  
examples  
adding data maps (DB2 unload file) 77  
adding data maps (single-record) 177  
adding fields to single-record data maps 178  
adding multiple-record data maps with complex tables 183  
adding multiple-segment data maps (IMS) 89  
adding records to single-record data maps 178  
adding single-record data maps 177  
adding single-segment data maps (IMS) 87  
adding tables to single-record data maps 179  
Check function 218  
Concat function 219  
file-list files (Linux) 61  
file-list files (UNIX) 61  
file-list files (Windows) 61  
file-list files (z/OS) 61  
Fragment function 221  
GetSeqWithinLevel function 229  
GetSeqWithinParent function 230  
GetVRowKey function 221  
importing COBOL copybooks into data maps 185  
importing COBOL copybooks with redefines 189  
importing PL/I copybooks into data maps 192  
IMS data maps 87  
LTrim function 231  
PowerExchange Navigator 176  
RTrim function 231  
single-record data map with an array 180  
Split function 233  
Strip function 234  
user access method programs 247  
user-defined programs 237  
user-defined programs (Assembler language) 240  
user-defined programs (C language) 238  
user-defined programs (CL language) 240  
user-defined programs (COBOL language) 239  
user-defined programs (PL/I language) 241  
user-defined programs (RPG language) 240  
exported data maps  
deleting 64  
importing 59  
overview 33  
exporting  
metadata for extraction maps 145  
Expr(n) tab  
Record window 47  
expressions  
adding to data map records (IDMS) 80  
adding to data map records (IMS) 84  
PowerExchange functions for 216  
Extensions tab  
Extract Definition window 148  
extra properties  
for fields in data map records 283  
Extract Definition window  
Extensions tab 148  
Maps tab 148  
overview 3  
extraction groups  
adding 135  
deleting 147  
deleting all extraction maps in 147  
editing properties for 143  
overview 132  
viewing properties for 143

extraction map columns, PowerExchange-generated  
     DTL\_\_BI\_columnname 149  
     DTL\_\_CAPXACTION 149  
     DTL\_\_CAPXCASDELIND 149  
     DTL\_\_CAPXRESTART1 149  
     DTL\_\_CAPXRESTART2 149  
     DTL\_\_CAPXRRN 149  
     DTL\_\_CAPXTIMESTAMP 149  
     DTL\_\_CAPXUOW 149  
     DTL\_\_CAPXUSER 149  
     DTL\_\_CI\_columnname 149  
 extraction maps  
     adding 137  
     adding before image (BI) columns to data columns 139  
     adding change indicator (CI) columns to data columns 139  
     change data capture 132  
     changing capture registrations for 141  
     creating data maps from 145  
     Database Row Test dialog box 160  
     Database Row Test Output window 160  
     deleting 146  
     deleting all in an extraction group 147  
     exporting metadata for 145  
     finding columns in 290  
     merging with data maps 142  
     names for 133  
     overview 132  
     PowerExchange-generated columns 149  
     searching for columns in 14  
     testing 160  
     viewing 139  
     viewing BI columns for 148  
     viewing CI columns for 148  
     viewing properties for 144  
 extraction processing  
     adding columns to 140  
     removing columns from 140  
 extraction start point  
     resetting for applications 250

## F

field definition table (FDT) copybooks  
     data map types supported in 96  
 field definition tables (FDTs)  
     displaying for data maps (Adabas) 101  
     importing into data maps (Adabas) 26, 100  
     refreshing for data maps (Adabas) 101  
 fields in data map records  
     adding 37, 178  
     applying date mask to 191  
     applying date masks to 187  
     code pages for 285  
     converting characters to lowercase 234  
     converting characters to uppercase 234  
     copying 56  
     data checking properties for 284  
     defining as arrays 282  
     deleting 65  
     editing properties 276, 282, 283  
     editing properties for 52  
     generating multiple table rows for 221  
     getting contents of 233  
     getting length of 230  
     getting sequence number in hierarchy 229  
     overview 26  
     printing 63  
     searching for 13

stripping characters from 233  
 trimming characters from 231  
 variable length properties for 285  
 viewing 47  
 viewing properties for 52  
 fields in data map records (Adabas)  
     editing properties 283  
 fields in data map records (IDMS)  
     getting sequence number in hierarchy 229  
 fields in data map records (IMS)  
     getting sequence number in hierarchy 229  
 fields in data map segments  
     deleting 65  
     overview 26  
 fields in data map segments (IMS)  
     getting RBA values for 226  
 Fields tab  
     Record window 47  
 file-list files  
     associating with data maps 25, 60  
     creating 60  
     example (Linux) 61  
     example (UNIX) 61  
     example (Windows) 61  
     example (z/OS) 61  
     ODBC 61  
     processing for 60  
     syntax rules for 60  
 filtering  
     columns in data map tables 288  
     data in data map records 181  
     tables in data maps (DB2) 256, 257  
 flat files  
     data maps for 24, 263  
 Fragment function  
     example 221  
 functions  
     CallProg 198

## G

generating  
     encrypted passwords 21  
     multiple table rows for fields in data map records 221  
     restart tokens for change data capture 163  
 GenVRowKey function 221  
 GetCurrentFileName function  
     examples 222  
 GetDatabaseKey function  
     alias for 225  
     to get the RBA 94  
     to get the RRN 94  
 GetDataFlowType function 224  
 GetDbKey function 225  
 GetDbKeyOfOwner function 225  
 GetFullDbKey function 225  
 GetFullDbKeyOfOwner function 226  
 GetIMSRBAByLevel function 226  
 GetPageGroup function 227  
 GetPageGroupOfOwner function 227  
 GetPgGrpAndRdx function 228  
 GetPgGrpAndRdxOfOwner function 228  
 GetPgGrpOfOwner function 228  
 GetRadix function 229  
 GetRadixOfOwner function 229  
 GetSeqWithinLevel function  
     example 229

GetSeqWithinParent function  
example 230  
getting  
length of fields in data map records 230  
RBA values for fields in data map segments (IMS) 226  
sequence number in hierarchy for fields in data map records (IDMS) 229  
sequence number in hierarchy for fields in data map records (IMS) 229  
GetVRowKey function  
example 221

## H

hiding  
Database Row Test Output window 159  
Resource Explorer 4  
status bar 12  
toolbar 10  
hierarchy  
defining for multiple-record data maps 184  
hierarchy created by DBD source import  
viewing in data maps (IMS) 81

## I

IBM MQSeries message queues  
data maps for 24  
IDMS access method  
data map properties 259  
IDMS capture registrations  
viewing parent hierarchy for base records in 129  
IDMS copybooks  
advanced properties for 109  
data map types supported in 96  
IDMS data map records  
editing properties for 79  
IDMS data maps  
adding 78  
adding CALC elements to records in 41, 42  
adding owner record and sets to records in 41  
adding owner records and sets to records in 42  
CALC elements in 33  
defining preferences for 20  
deleting CALC elements from records in 67  
deleting owner records and sets from records in 67  
editing properties for records in 274  
overview 78  
owner records and sets in 33  
viewing properties for records in 274  
IDMS databases  
data maps for 24, 259  
import details  
for data map records 276  
importing  
COBOL copybooks with REDEFINES statements into data maps 189  
copybooks into data map records 99  
copybooks into data map segments 99  
copybooks into data maps 97  
copybooks into IMS data maps 89  
exported data maps 59  
FDTs into data maps (Adabas) 26  
field definition tables (FDTs) into data maps (Adabas) 100  
keys into data maps (Adabas) 26  
PL/I copybook into data maps 192  
PL/I copybooks into data maps 192

improving  
bulk read performance for VSAM data sets 91  
IMS data maps  
adding 83, 87, 89  
editing properties for 85  
examples 87  
importing copybooks into 89  
overview 80  
prerequisites for adding 82  
sending to remote systems 85  
testing 87  
viewing records in 81  
IMS databases  
data maps for 24, 257  
maintaining hierarchical metadata in data maps 99  
using lookup transformations to look up data in 85  
writing data to 85  
IMS DL1 batch data map records  
CCK fields in 34  
search fields in 34  
IMS hierarchy created by DBD source import  
viewing in data maps (IMS) 81  
IMS ODBA access method  
data map properties 257  
IMS ODBA data map records  
CCK fields in 34  
search fields in 34  
IMS ODBA data maps  
configuring PowerExchange to use 82  
IMS options  
editing for data map tables 86  
IMS RBA  
getting in data map records (IMS) 84  
index buffers  
for VSAM data sets 92  
introduction  
PowerExchange Navigator 1  
ISAM access method  
data map properties 260  
items in data maps  
printing 61

## K

key segments in data map records (Datacom)  
adding 39, 41  
deleting 67  
editing properties for 53  
overview 33  
viewing properties for 53  
keys  
adding to data map tables (Adabas) 46, 72  
for VSAM data sets 92  
importing into data maps (Adabas) 26  
keys in data map records (Datacom)  
adding 39  
deleting 66  
editing properties for 53  
overview 33  
viewing properties for 53  
keys in data map tables (Adabas)  
editing properties for 289  
viewing properties for 289  
keys in data map tables (Datacom)  
editing properties for 289  
viewing properties for 289

keys in data maps (Adabas)  
  properties 271  
KSDS access method  
  data map properties 261

## L

layout of data  
  in data source or target 26  
LengthOf function 230  
LISTTASK command  
  issuing in a database row test 162  
local DB2 catalog details  
  for importing copybooks 105  
logon credentials  
  to access data 57  
  to access data maps 57  
lookup transformations  
  using to look up data in IMS databases 85  
LTrim function  
  example 231

## M

Maps tab  
  Extract Definition window 148  
menu bar  
  overview 5  
menu commands  
  overview 5  
  PowerExchange Navigator 5  
menus  
  on menu bar 5  
merging  
  data maps with extraction maps 142  
metadata  
  exporting for extraction maps 145  
  refreshing in personal metadata profiles 157  
MQSeries access method  
  data map properties 261  
MQSeries message queues  
  data maps for 24, 261  
multibyte data  
  in database row tests 159  
multiple-record data map  
  testing 184  
multiple-segment data maps  
  adding 89

## N

names  
  for extraction maps 133  
naming  
  data maps 23, 34  
Natural data definition module (DDM) copybooks  
  data map types supported in 96  
normalization of an array  
  testing 197

## O

OCCURS clauses  
  defining arrays with 29

OCCURS DEPENDING ON clause  
  displaying multiple output rows for 188  
ODBC  
  file-list files 61  
opening  
  PowerExchange resources 12  
overview  
  application groups 248  
  applications 248  
  arrays 29  
  capture registrations 112  
  code pages 31  
  columns in data map tables 27  
  copybooks 25, 95  
  data checking 32  
  data map tables 27  
  data maps 22, 23  
  data maps (Adabas) 70  
  data maps (Datacom) 72  
  data maps (DB2 catalog) 74  
  data maps (DB2 unload file) 75  
  data maps (IDMS) 78  
  data maps (IMS) 80  
  data maps for specific data sources 69  
  database row test 158  
  database row tests 158  
  exported data maps 33  
  extraction groups 132  
  extraction maps 132  
  menu commands 5  
  personal metadata profiles 152  
  PowerExchange Navigator 1  
  record-level exit programs 27  
  registration groups 112  
  shortcut keys 5  
  user access method programs 34, 243  
  user-defined fields 26  
owner record and sets in data map records (IDMS)  
  adding 41  
  sort element details for 275  
owner records and sets for data map records (IDMS)  
  getting page groups for 228  
  getting radix for 229  
owner records and sets in data map records (IDMS)  
  adding 42  
  deleting 67  
  editing properties for 54  
  getting database key for 225  
  getting fully qualified database keys for 226  
  getting page group and radix for 228  
  getting page groups for 227  
  overview 33  
  viewing properties for 54

## P

page groups  
  getting for data map records (IDMS) 227, 228  
  getting for owner records and sets in data map records (IDMS) 227, 228  
parameter list  
  passed to user-defined programs 236  
parameter structure  
  passed to user access method programs 244  
personal metadata profile tables  
  searching for columns in 14

personal metadata profiles  
     adding 153  
     Database Row Test dialog box 161  
     Database Row Test Output window 161  
     deleting 157  
     editing properties for 156  
     finding tables in 290  
     overview 152  
     refreshing metadata in 157  
     searching for a table in 14  
     testing 161  
     viewing metadata for tables in 155  
     viewing properties for 156  
         viewing properties for tables in 156  
 personal metadata tables  
     viewing properties for 155  
 PL/I copybooks  
     data map types supported in 96  
     importing into data maps 192  
 PL/I user-defined programs  
     operating system support 241  
 PowerExchange  
     code pages supported 31  
     resources 12  
 PowerExchange functions  
     for expressions in user-defined fields 216  
     for user-defined fields 216  
 PowerExchange Listener commands  
     issuing in a database row test 162  
 PowerExchange Navigator  
     Application window 3  
     arranging windows 5  
     Capture Registration window 3  
     examples 176  
     Extract Definition window 3  
     interface 2  
     introduction 1  
     menu bar 5  
     overview 1  
     Record window 3  
     Resource Explorer 3  
     Resource Inspector 3  
     Resources window 3  
     shortcut keys 11  
     shortcut menus 8  
     status bar 12  
     Table window 3  
     title bar 5  
     toolbar 10  
 PowerExchange resources  
     closing 15  
     deleting 15  
     opening 12  
     searching for 13  
 PowerExchange-generated extraction map columns  
     DTL\_\_BI\_columnname 149  
     DTL\_\_CAPXACTION 149  
     DTL\_\_CAPXCASDELIND 149  
     DTL\_\_CAPXRESTART1 149  
     DTL\_\_CAPXRESTART2 149  
     DTL\_\_CAPXRRN 149  
     DTL\_\_CAPXTIMESTAMP 149  
     DTL\_\_CAPXUOW 149  
     DTL\_\_CAPXUSER 149  
     DTL\_\_CI\_columnname 149  
 PREDICT copybooks  
     data map types supported in 96

preferences  
     defining for data maps 18  
 Preferences - IDMS Properties dialog box 20  
 prerequisites  
     for adding IMS data maps 82  
 printing  
     columns in data map tables 64  
     data map records 63  
     data map tables 63  
     data maps 62  
     fields in data map records 63  
 properties  
     application groups 250  
     CALC elements in data map records (IDMS) 54  
     capture registrations 125  
     columns in data map tables 56  
     data checking for data maps 269  
     data defaults for data maps 268  
     data map records 52, 272, 276  
     data map records (Datacom) 273  
     data map records (IDMS) 79, 274  
     data map tables 55  
     data map tables (IMS) 289  
     data maps 51  
     data maps (ADABAS access method) 252  
     data maps (DATACOM access method) 254  
     data maps (DB2 access method) 254  
     data maps (DB2 unload file) 255  
     data maps (DL/1 Batch access method) 257  
     data maps (ESDS access method) 258  
     data maps (flat files) 263  
     data maps (IDMS access method) 259  
     data maps (IMS ODBA access method) 257  
     data maps (ISAM access method) 260  
     data maps (KSDS access method) 261  
     data maps (MQSeries access method) 261  
     data maps (RRDS access method) 262  
     data maps (sequential data sets) 263  
     data maps (tape data set) 265  
     data maps (USER access method) 266  
     definitions for data map tables 286  
     elements in owner records and sets (IDMS) 275  
     extraction groups 143  
     extraction maps 144  
     fields in data map records 52, 276, 282, 283, 285  
     fields in data map records (Adabas) 283  
     filtering criteria for columns in data map tables 288  
     key segments in data map records (Datacom) 53  
     keys in data map records (Datacom) 53  
     keys in data map tables (Adabas) 289  
     keys in data maps (Adabas) 271  
     owner records and sets in data map records (IDMS) 54  
     personal metadata profile tables 156  
     personal metadata profiles 156  
     record-level exit programs defined for data maps 270  
     registration groups 124  
 pwxcmd -epassword option  
     generating encrypted password for 21

## R

radix  
     getting for data map records (IDMS) 228, 229  
     getting for owner records and sets in data map records (IDMS) 228, 229  
 record IDs  
     in data map records 183

record layout  
     editing in data maps (DB2) 75  
 Record window  
     Expr(n) tab 47  
     fields in 47  
     Fields tab 47  
     overview 3  
     records in 47  
 record-level exit programs  
     overview 27  
     properties 270  
 records in data maps  
     adding 37  
     adding fields to 37  
     deleting 65  
     deleting fields from 65  
     editing properties for 52  
     editing properties for fields in 52  
     importing copybooks into 99  
     overview 26  
     printing 63  
     printing fields in 63  
     record-level exit programs for 27  
     sorting 51  
     user-defined fields in 38  
     viewing 47  
     viewing fields in 47  
     viewing properties for 52  
     viewing properties for fields in 52  
 records in data maps (Datacom)  
     adding elements to 39  
     adding items to 39  
     adding key segments to 41  
     adding keys to 39  
     deleting elements from 66  
     deleting key segments from 67  
     deleting keys from 66  
     editing properties for key segments in 53  
     editing properties for keys in 53  
     viewing properties for key segments in 53  
     viewing properties for keys in 53  
 records in data maps (IDMS)  
     adding CALC elements to 42  
     adding items to 41  
     deleting CALC elements from 67  
     editing properties for CALC elements in 54  
     viewing properties for CALC elements in 54  
 records in data maps (IMS)  
     viewing 81  
 REDEFINES statements  
     in COBOL copybooks 189  
 references affected by the current delete list 65  
 refreshing  
     Adabas field definition tables (FDTs) 101  
     data map tables with missing columns 38  
     metadata in personal metadata profiles 157  
 registration groups  
     adding 116  
     adding application groups with 115  
     adding extraction groups with 115  
     deleting 130  
     deleting all capture registrations in 130  
     overview 112  
     viewing properties 124  
 registration tag names 113  
 registrations groups  
     editing properties 124  
 relative byte address (RBA)  
     GetDatabaseKey function 94  
     getting for data set records (VSAM) 93  
 relative record number (RRN)  
     GetDatabaseKey function 94  
     getting for data set records (VSAM) 93  
 remote Datacom details  
     copybooks 107  
 remote DB2 catalog details  
     for importing copybooks 108  
 remote details  
     copybooks 106  
 remote systems  
     sending data maps to 58  
 removing  
     columns from capture registrations 129  
     columns from extraction processing 140  
 resetting  
     extraction start point for applications 250  
 resource configurations  
     adding 16  
     creating shared 17  
     defining 16  
     deleting 17  
     editing 16  
     selecting 17  
 Resource Explorer  
     hiding 4  
     overview 2, 3  
     showing 4  
 Resource Inspector  
     overview 2, 3  
 resources  
     PowerExchange 12  
 Resources window  
     changing view type for 4  
     overview 2, 3  
 restart tokens  
     generating for change data capture 163  
 return codes  
     for user-defined programs 237  
     identifying for programs called with CallProg function 218  
     user access method programs 246  
 RPG programs  
     operating system support 240  
 RRDS access method  
     data map properties 262  
 RTrim function  
     example 231  
 running  
     database row tests 158

## S

search criteria  
     to find columns in capture registrations 290  
     to find columns in data map tables 290  
     to find columns in extraction maps 290  
     to find columns in personal metadata profile tables 290  
     to find fields in data map records 290  
     to find tables in personal metadata profiles 290  
     wildcards in 13  
 search fields  
     adding to data map records (IMS) 44, 45  
     deleting from data map records (IMS) 68  
     editing in data map records (IMS) 44  
     editing properties for in data map records (IMS) 55  
     viewing properties for in data map records (IMS) 55

search fields in data map records (IMS DL/1 batch)  
   overview 34  
 search fields in data map records (IMS ODBA)  
   overview 34  
 searching  
   for columns in capture registrations 14  
   for columns in data map tables 13  
   for columns in extraction maps 14  
   for columns in personal metadata tables 14  
   for fields in data map records 13  
   for PowerExchange resources 13  
   for tables in personal metadata profiles 14  
 segments in data maps  
   deleting 65  
   deleting fields from 65  
   importing copybooks into 99  
   overview 26  
 selecting  
   columns in data map tables (DB2 catalog) 109  
   resource configurations 17  
 sending  
   data maps to remote systems 33, 58  
 SEQ access method  
   data map properties 263  
 sequence numbers  
   getting for fields in data map records (IDMS) 230  
   getting for fields in data map records (IMS) 230  
 sequential data sets  
   data maps for 24, 263  
 SetBitA function 232  
 SetBitN function 232  
 shared resource configurations  
   creating 17  
 shortcut keys  
   overview 5  
   PowerExchange Navigator 11  
 shortcut menus  
   PowerExchange Navigator 5, 8  
 showing  
   Database Row Test Output window 159  
   Resource Explorer 4  
   status bar 12  
   toolbar 10  
 simple tables  
   in data maps 27  
 single-record data map  
   testing 180  
 single-record data maps  
   adding 177  
   adding fields to records in 178  
   adding records to 178  
 single-segment data maps (IMS)  
   adding 87  
 sort element details  
   for owner records and sets (IDMS) 275  
 sorting  
   records in data maps 51  
   tables in data maps 51  
 source details  
   for importing copybooks 101  
 source files  
   getting name of for data map records 222  
 Split function  
   example 233  
 splitting  
   field data in data map records 220  
 SQL escape sequences  
   DTLDSN 61  
     DTLFILELIST 61  
 status bar  
   hiding 12  
   PowerExchange Navigator 12  
   showing 12  
 status options  
   for capture registrations 122  
 STOPTASK command  
   issuing in database row test 162  
 storing  
   data maps (IMS) 85  
 Strip function  
   example 234  
 syntax rules  
   for file-list files 60

## T

Table window  
   columns in 49  
   overview 3  
   tables in 49  
 tables  
   adding to data maps 179  
 tables in data maps  
   adding 46  
   adding Adabas keys to 46  
   adding columns to 47  
   deleting 66  
   deleting columns from 66  
   editing properties for 55, 286  
   editing properties for columns in 56  
   finding columns in 290  
   overview 27  
   printing 63  
   printing columns in 64  
   properties 197, 214  
   searching for columns in 13  
   sorting 51  
   viewing 49  
   viewing properties for 55, 286  
   viewing properties for columns in 56  
 tables in data maps (Adabas)  
   adding keys to 72  
 tables in data maps (DB2)  
   filtering 256  
 tables in data maps (IMS)  
   editing properties for 86, 289  
   viewing 81  
   viewing properties for 86, 289  
 tables in personal metadata profiles  
   finding columns in 290  
   searching for 14  
   searching for columns in 14  
   viewing metadata for columns in 155  
   viewing properties for 156  
 tag names  
   for capture registrations 113  
 TAPE access method  
   data map properties 265  
 tape data sets  
   data maps for 24, 265  
 testing  
   data maps 159  
   data maps (IMS) 87  
   extraction maps 160  
   multiple-record data map 184

personal metadata profiles 161  
 single-record data map 180  
 title bar  
 PowerExchange Navigator 5  
 ToLower function 234  
 toolbar  
     hiding 10  
     PowerExchange Navigator 10  
     showing 10  
 ToUpper function 234  
 trimming  
     characters from fields in data map records 231

## U

UDB\_CAPI\_CONNECTION statement EPWD option  
     generating encrypted password for 21  
 USER access method  
     data map properties 266  
     data maps 243  
 user access method programs  
     data maps for 266  
     examples 247  
     overview 34, 243  
     parameter structure passed to 244  
     return codes 246  
 user exit program  
     testing 214  
 user exit programs  
     compiling 201  
     creating 201  
     creating and calling in data map records 198  
     sample 201  
 user-defined code pages 32  
 user-defined fields  
     adding to data map records 38  
     functions for expressions in 216  
     overview 26  
     PowerExchange functions for 216  
     testing in data maps 242  
     to demonstrate the normalization of an array 196  
     with expressions 211  
 user-defined functions  
     CallProg 26, 217  
     Check 218  
     Concat 219  
     CopyData 219  
     Fragment 220  
     GenVRowKey 221  
     GetCurrentFileName 222  
     GetDatabaseKey 223  
     GetDataFlowType 224  
     GetDbKey 225  
     GetDbKeyOfOwner 225  
     GetFullDbKey 225  
     GetFullDbKeyOfOwner 226  
     GetIMSRBAByLevel 226  
     GetPageGroup 227  
     GetPageGroupOfOwner 227  
     GetPgGrpAndRdx 228  
     GetPgGrpAndRdxOfOwner 228  
     GetPgGrpOfOwner 228  
     GetRadix 229  
     GetRadixOfOwner 229  
     GetSeqWithinLevel 229  
     GetSeqWithinParent 230  
     LengthOf 230  
     LTrim 231

RTrim 231  
 SetBitA 232  
 SetBitN 232  
 Split 233  
 Strip 233  
 ToLower 234  
 ToUpper 234  
 user-defined programs  
     calling from data map records 217  
     errors 235  
     exampe C programs 238  
     example (Assembler language) 239  
     example (COBOL language) 238  
     example (PL/I language) 241  
     example (RPG language) 240  
     parameter list passed to 236  
     return codes for 237

## V

variable length properties  
     for fields in data map records 285  
 viewing  
     capture registrations 124  
     CCK fields in data map records (IMS) 55  
     columns in data map tables 49  
     data files 49  
     data map records 47, 183  
     data map records (IMS) 81  
     data map tables 49  
     elements for capture registrations (Datacom) 128  
     extraction maps 139  
     fields in data map records 47  
     hierarchy created by DBD source import in data maps (IMS) 81  
     parent hierarchy for base records for capture registrations (IDMS) 129  
     properties for elements in data map records (Datacom) 53  
     properties for search fields in data map records (IMS) 55  
 viewing properties  
     application groups 250  
     CALC elements in data map records (IDMS) 54  
     capture registrations 125  
     columns in data map tables 56  
     data map records 52, 272  
     data map records (IDMS) 274  
     data map tables 55  
     data map tables (IMS) 86, 289  
     data maps 51  
     definitions for data map tables 286  
     extraction groups 143  
     extraction maps 144  
     fields in data map records 52  
     key segments in data map records (Datacom) 53  
     keys in data map records (Datacom) 53  
     keys in data map tables (Datacom) 289  
     owner records and sets in data map records (IDMS) 54  
     personal metadata profile tables 155, 156  
     personal metadata profiles 156  
     registration groups 124  
 VSAM data maps  
     adding 90  
     overview 90  
 VSAM data set records  
     getting the RBA in 93, 94  
     getting the relative byte address (RBA) in 93  
     getting the relative record number (RRN) in 93  
     getting the RRN in 93, 94

VSAM data sets  
control interval access 92  
data buffers for 91  
improving bulk read performance for 91  
index buffers for 92  
keys for 92  
VSAM entry-sequenced data sets (ESDSs)  
  data maps for 24, 258  
VSAM key-sequenced data sets (KSDSs)  
  data maps for 24  
VSAM keyed-sequenced data sets (KSDSs)  
  data maps for 261  
VSAM relative record data sets (RRDSs)  
  data maps for 24, 262

windows  
Application 3  
Capture Registration 3  
Database Row Test Output 159–161  
Extract Definition 3, 139, 148  
Record 3, 47  
Resource Explorer 2  
Resource Inspector 2  
Resources 2–4  
Table 3, 49  
writing data  
  to IMS databases 85

## W

wildcards  
  in search criteria for PowerExchange resources 13