Informatica

Informatica® PowerExchange for SAP NetWeaver

10.1.1 HotFix 1

User Guide for PowerCenter

Informatica PowerExchange for SAP NetWeaver User Guide for PowerCenter
10.1.1 HotFix 1
June 2017

© Copyright Informatica LLC 2009, 2018

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

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

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

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 © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © 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. Copyright © International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright © University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jQWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright© MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

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

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 various versions of 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 1996-2013, 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.

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 and at http://www.eclipse.org/org/documents/edl-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://asm.ow2.org/license.html, http://www.cryptix.org/LICENSE.TXT, http://hsqldb.org/web/hsqlLicense.html, http:// httpunit.sourceforge.net/doc/ license.html, http://jung.sourceforge.net/license.txt , http://www.gzip.org/zlib/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://www.w3.org/

Consortium/Legal/2002/copyright-software-20021231; http://www.slf4j.org/license.html; http://nanoxml.sourceforge.net/orig/copyright.html; http://www.json.org/license.html; http://forge.ow2.org/projects/javaservice/, http://www.postgresql.org/about/licence.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, http://www.slf4j.org/license.html; http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License; http://www.keplerproject.org/md5/license.html; http://www.toedter.com/en/jcalendar/license.html; http://www.edankert.com/bounce/index.html; http://www.net-snmp.org/about/license.html; http://www.openmdx.org/#FAQ; http://www.php.net/license/3_01.txt; http://srp.stanford.edu/license.txt; http://www.schneier.com/blowfish.html; http://www.jmock.org/license.html; http://xsom.java.net; http://benalman.com/about/license/; https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js; http://www.h2database.com/html/license.html#summary; http://jsoncpp.sourceforge.net/LICENSE; http://jdbc.postgresql.org/license.html; http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto; https://github.com/rantav/hector/blob/master/LICENSE; http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html; http://jibx.sourceforge.net/jibx-license.html; https://github.com/lyokato/libgeohash/blob/master/LICENSE; https://github.com/hjiang/jsonxx/blob/master/LICENSE; https://code.google.com/p/lz4/; https://github.com/jedisct1/libsodium/blob/master/LICENSE; http://one-jar.sourceforge.net/index.php?page=documents&file=license; https://github.com/EsotericSoftware/kryo/blob/master/license.txt; http://www.scala-lang.org/license.html; https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt; http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html; https://aws.amazon.com/asl/; https://github.com/twbs/bootstrap/blob/master/LICENSE; https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt; https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE, and https://github.com/apache/hbase/blob/master/LICENSE.txt.

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 new BSD License (http://opensource.org/licenses/BSD-3-Clause), the MIT License (http://www.opensource.org/licenses/mit-license.php), the Artistic License (http://www.opensource.org/licenses/artistic-license-1.0) and the Initial Developer's Public License Version 1.0 (http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/).

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

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

See patents at https://www.informatica.com/legal/patents.html.

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

NOTICES

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

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

The information in this documentation is subject to change without notice. If you find any problems in this documentation, please report them to us in writing at Informatica LLC 2100 Seaport Blvd. Redwood City, CA 94063.

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

Publication Date: 2018-11-22

Table of Contents

Preface 15

Informatica Resources. 15

Informatica Network. 15

Informatica Knowledge Base. 15

Informatica Documentation. 16

Informatica Product Availability Matrixes. 16

Informatica Velocity. 16

Informatica Marketplace. 16

Informatica Global Customer Support. 16

Part I: Getting Started with PowerExchange for SAP NetWeaver. 17

Chapter 1: Understanding PowerExchange for SAP NetWeaver. 18

Understanding PowerExchange for SAP NetWeaver Overview. 18

PowerCenter and SAP NetWeaver Integration Methods. 19

Data Integration Using the ABAP Program. 19

IDoc Integration Using ALE. 20

Data Integration Using BAPI/RFC Functions. 20

Data Migration Integration. 21

Business Content Integration. 21

PowerCenter and SAP NetWeaver BW Integration Methods. 21

Communication Interfaces. 21

Remote Function Call. 22

Common Program Interface-Communications. 22

Transport System. 22

Operating System Profiles for SAP Sessions. 22

Chapter 2: Installing and Configuring PowerExchange for SAP NetWeaver. . . 24

Installing and Configuring PowerExchange for SAP NetWeaver Overview. 24

Before You Begin. 24

Installing PowerExchange for SAP NetWeaver. 25

Upgrading PowerExchange for SAP NetWeaver. 25

Upgrading to PowerExchange for SAP NetWeaver in Unicode Mode. 25

Registering the Plug-in. 26

Configuration Checklist. 26

Configuration Tasks and Integration Methods. 26

Integrating with SAP Using ABAP. 27

Integrating with SAP Using ALE. 27

Integrating with SAP Using BAPI/RFC Functions. 28

Migrating Data to SAP. 28

Integrating with SAP Business Content. . . . 28
Installing the SAP NetWeaver RFC SDK Version 7.20 Libraries. . . . 28
Registering the PowerExchange for SAP NetWeaver Client. . . . 30
Defining PowerCenter as a Logical System in SAP. . . . 30
Creating a Logical System for IDoc ALE Integration. . . . 30
Creating a Logical System for Business Content Integration. . . . 34
Configuring sapnwrfc.ini. . . . 36
sapnwrfc.ini Connection Types. . . . 36
Sample sapnwrfc.ini File. . . . 37
Configuring a Connection in sapnwrfc.ini. . . . 37
Creating Entries in the Services File for Stream Mode Sessions. . . . 39
Uninstalling PowerExchange for SAP NetWeaver. . . . 39
Cleaning Up the SAP System. . . . 39

Chapter 3: Installing and Configuring PowerExchange for SAP NetWeaver BW. . . . 40

Installing and Configuring PowerExchange for SAP NetWeaver BW Overview. . . . 40
Before You Begin. . . . 40
Installing and Configuring PowerExchange for SAP NetWeaver BW. . . . 41
Upgrading PowerExchange for SAP NetWeaver BW. . . . 41
Installing the SAP NetWeaver RFC SDK Version 7.20 Libraries. . . . 41
Registering the PowerExchange for SAP NetWeaver Client. . . . 43
Creating Profiles for Production and Development Users. . . . 43
Profiles for Extracting Data from SAP BW 3.5. . . . 43
Profiles for Extracting Data from SAP BW. . . . 44
Profiles for Loading data to SAP NetWeaver BW or SAP BW 3.5. . . . 45
Defining PowerCenter as a Logical System in SAP BW. . . . 47
Configuring sapnwrfc.ini. . . . 47
sapnwrfc.ini Connection Types. . . . 48
Sample sapnwrfc.ini File. . . . 48
Configuring a Connection in sapnwrfc.ini. . . . 49
Creating the SAP BW Service. . . . 50
Load Balancing for the SAP BW System and the SAP BW Service. . . . 50
Steps to Create the SAP BW Service. . . . 51
Importing the ABAP Program into SAP BW. . . . 52
Troubleshooting PowerExchange for SAP NetWeaver BW Installation and Configuration. . . . 52

Chapter 4: Installing and Configuring Transports. . . . 53

Transports Installation and Configuration. . . . 53
Step 1. Delete Transport Programs. . . . 54
Step 2. Install Transport Objects. . . . 56
Step 3. Run Transport Programs. . . . 56
Step 4. Create Users. . . . 58

Step 5. Create Profiles. 58
Step 6. Create a Package. 60
Transports Installation. 61
Transports Installation Order. 61
Transports List based on SAP Functionality. 62
Secure Transports for SAP Unicode Versions 5.0 and Later. 63
Standard Transports for SAP Unicode Versions 5.0 and Later. 65
Secure Transports for SAP non-Unicode Versions 5.0 and Later. 67
Standard Transports for SAP non-Unicode Versions 5.0 and Later. 69

Part II: Data Integration Using ABAP. 71

Chapter 5: Importing SAP ECC Source Definitions. 72

Importing SAP ECC Source Definitions Overview. 72
Editing a Source Definition. 72
Table and View Definitions. 73
Importing Key Relationships. 73
Hierarchy Definitions. 74
Uniform Hierarchies. 74
Non-Uniform Hierarchies. 74
Importing Hierarchy Definitions. 75
Establishing Hierarchy Relationships. 76
IDoc Definitions 76
Importing IDoc Definitions. 76
Viewing IDoc Definitions. 76
Importing a Source Definition 77
Filtering Definitions in the Import Dialog Box. 77
Steps to Import an SAP ECC Source Definition. 77
Organizing Definitions in the Navigator. 78
Working with Business Components. 79
Troubleshooting SAP ECC Source Definitions. 79

Chapter 6: SAP ECC Target Definitions. 80

SAP ECC Target Definitions Overview. 80
Target Table Definitions. 80
Editing a Target Table Definition. 81
Target Update Override. 81
Target Table Name. 82
Keys and Key Relationships. 82
Importing an SAP ECC Target Definition. 82
Filtering Definitions in the Import Dialog Box. 83

Chapter 7: Working with ABAP Mappings. 84

Working with ABAP Mappings Overview. 84
Setting the Select Option. 84
Select Single. 85
Select Distinct. 85
Setting Order By Ports. 86
Transparent Tables. 86
Pool and Cluster Tables. 87
Viewing the Hierarchy Properties. 87
Viewing IDoc Properties. 87
Working with the ABAP/4 Program. 88
Selecting the Program Mode. 88
Naming the ABAP Program. 89
Adding Authority Checks. 89
Working with ABAP Programs and Versioned Mappings. 89
Generating and Installing the ABAP Program. 90
Deploying ABAP Mappings with ABAP Programs. 93
Viewing Program Information. 93
Uninstalling the ABAP Program. 94
Cleaning ABAP Program Information. 94
Copying Program Information 94
Troubleshooting ABAP Mappings. 95

Chapter 8: Working with SAP Functions in ABAP Mappings. 96

Working with SAP Functions in ABAP Mappings Overview. 96
Using SAP Functions in the ABAP Program Flow. 96
SAP Function Parameters. 97
Using SAP Functions in the ABAP Program Flow. 97
Importing SAP Functions. 98
Viewing SAP Functions. 99
Inserting SAP Functions in the ABAP Program Flow. 99
Configuring SAP Function Parameters in the ABAP Program Flow. 99
Steps for Inserting an SAP Function in the ABAP Program Flow. 100
Validating SAP Functions in the ABAP Program Flow. 101

Chapter 9: Application Source Qualifier for SAP ECC Sources. 103

Application Source Qualifier for SAP ECC Sources Overview. 103
Generating an ABAP Program. 104
Available ABAP Generation Mode. 104
Generating Open SQL. 105
Generating Exec SQL. 105
Generating ABAP Join Syntax. 105

Working with ABAP Program Flow. . . . 106
Validating the ABAP Program Flow. . . . 106
Joining Source Data. . . . 106
Joining Sources Using Open SQL. . . . 107
Joining Sources Using Exec SQL. . . . 108
Joining Sources Using ABAP Join Syntax. . . . 108
Selecting a Join Type. . . . 109
Using Multiple Outer Joins. . . . 109
Joining Tables and Hierarchies. . . . 109
Joining Tables and IDoc. . . . 110
Specifying Join Conditions. . . . 110
Creating an ABAP Code Block. . . . 111
Rules for Inserting the ABAP Code Block. . . . 111
Creating ABAP Program Variables. . . . 112
Naming Convention. . . . 113
Creating Structure and Structure Field Variables. . . . 113
Creating a Structure Variable. . . . 113
Creating a Structure Field Variable. . . . 114
Creating ABAP Type Variables. . . . 114
Viewing ABAP Program Variables. . . . 115
Using SAP System Variables. . . . 115
Entering a Source Filter. . . . 115
Using Dynamic Filters. . . . 116
Using Static Filters. . . . 117
Using Mapping Variables and Parameters. . . . 118
Using Mapping Variables in the ABAP Program Flow. . . . 119
Working with SAP Date Formats. . . . 119
Working with IDoc Sources. . . . 119
Working with IDoc Sources in the ABAP Program Flow. . . . 119
Entering an IDoc Filter. . . . 120
Validating the IDoc Filter Condition. . . . 120
Configuring an Application Source Qualifier. . . . 121
Configuring an Application Source Qualifier. . . . 121
Troubleshooting an Application Source Qualifier for SAP ECC Sources. . . . 121

Chapter 10: Configuring Sessions with SAP ECC Sources and Targets. 123

Configuring Sessions with SAP ECC Sources and Targets Overview. . . . 123
Running Stream Mode Sessions. . . . 124
Running File Mode Sessions. . . . 125
Enabling Compressed Data Transfer. . . . 125
Reusing Staging Files . . . 126
Overriding Filter Conditions. . . . 126
Accessing Staging Files for ABAP Mappings. . . . 127

Modes of Access. 127
Enabling Access to Staging Files on UNIX. 128
Configuring File Mode Session Properties. 128
Pipeline Partitioning for SAP ECC Sources. 129
Session Properties for SAP ECC Targets. 130
Reject Files. 131
Session Load Mode. 131
Configuring an SAP ECC Session 132
Troubleshooting Sessions. 132

Part III: IDoc Integration Using ALE. 134

Chapter 11: Creating Outbound IDoc Mappings. 135

Creating Outbound IDoc Mappings Overview. 135
Defining PowerCenter as a Logical System for Outbound IDocs. 136
Creating an Outbound IDoc Mapping. 136
Processing Invalid Outbound IDocs. 136
Creating an SAPALEIDoc Source Definition. 136
Using SAPALEIDoc Source Definitions in an Outbound IDoc Mapping. 137
Working with the SAP/ALE IDoc Interpreter Transformation 137
Segments and Groups. 138
Creating an SAP/ALE IDoc Transformation. 140
Editing an SAP/ALE IDoc Interpreter Transformation. 143
Processing Invalid Outbound IDocs. 144

Chapter 12: Creating Inbound IDoc Mappings. 145

Creating Inbound IDoc Mappings Overview. 145
Defining PowerCenter as a Logical System for Inbound IDocs. 145
Creating an Inbound IDoc Mapping. 145
Validating 6.x Inbound IDoc mappings. 146
Processing Invalid Inbound IDocs. 146
Working with the SAP/ALE IDoc Prepare Transformation. 146
IDoc Primary and Foreign Keys. 146
Creating an SAP/ALE IDoc Prepare Transformation. 148
Editing an SAP/ALE IDoc Prepare Transformation. 148
Creating an SAPALEIDoc Target Definition. 150
Configuring Inbound IDoc Mappings. 150
Passing the Document Number to SAP. 150
Passing the Sender Partner Number to SAP. 150
Processing Invalid Inbound IDocs. 151

Chapter 13: Configuring IDoc Workflows Using ALE. 152

Configuring Sessions for Outbound IDoc Mappings. 152

Terminating Conditions. . . . 152
Real-time Processing. . . . 153
Message Recovery. . . . 153
Pipeline Partitioning. . . . 154
Outbound IDoc Validation. . . . 154
Row-Level Processing. . . . 154
Continuous Workflows. . . . 154
Configuring Sessions for Inbound IDoc Mappings. . . . 155
Pipeline Partitioning. . . . 155
Sending IDocs to SAP. . . . 155
Inbound IDoc Validation. . . . 156
Caching Inbound IDoc and DMI Data. . . . 156
Steps to Configure Sessions for IDoc Mappings Using ALE. . . . 156
Error Handling for IDoc Sessions Using ALE. . . . 159
Running IDoc Workflows Using ALE 159
Troubleshooting IDoc Workflows Using ALE. . . . 159

Part IV: Data Integration Using BAPI/RFC Functions. . . . 160

Chapter 14: BAPI/RFC Transformation. . . . 161

BAPI/RFC Transformation Overview. . . . 161
BAPIs with Nested Structures. . . . 162
BAPI/RFC Transformation Components. . . . 162
Configuring BAPI/RFC Transformation Properties. . . . 162
Customizing BAPI/RFC Properties. . . . 163
Creating a BAPI/RFC Transformation. . . . 165
Using a BAPI/RFC Transformation in Mappings. . . . 167
Adding XML Transformations for Table Type Parameters. . . . 168
Working with Function Input Data for BAPI/RFC Functions. . . . 168
Configuring a Target Definition for BAPI/RFC Error Output. . . . 169
System Variables. . . . 169
Real-time BAPI/RFC Mappings. . . . 169
Troubleshooting a BAPI/RFC Transformation. . . . 170

Chapter 15: BAPI/RFC Sessions. . . . 171

BAPI/RFC Sessions Overview. . . . 171
Configuring a Session with a BAPI/RFC Transformation. . . . 171
Committing Data to SAP. . . . 172
Caching BAPI Data. . . . 172
Configuring Error Handling. . . . 173
Configuring a Session for BAPIs with Table Type Parameters. . . . 174
Configuring a Real-time BAPI/RFC Session. . . . 174

Part V: Data Migration. 175

Chapter 16: Creating Data Migration Mappings. 176

Creating Data Migration Mappings Overview. 176

Creating a DMI Mapping. 176

Working with the SAP DMI Prepare Transformation. 176

DMI Primary and Foreign Keys. 177

Creating an SAP DMI Prepare Transformation. 178

Editing an SAP DMI Prepare Transformation. 180

Error Handling with DMI Mappings. 180

Creating a Flat File Target for DMI Data. 181

Configuring Sessions for DMI Mappings. 181

Chapter 17: Migrating Legacy Data into SAP IS-U/CCS. 182

Migrating Legacy Data into SAP ISU/CCS Overview. 182

Migration Objects. 182

Configuring Migration Objects in SAP. 183

SAP IS-U/CCS Transformation Components. 183

SAP IS-U Transformation Properties. 184

Auto Structures and Fields. 185

Creating an SAP IS-U/CCS Transformation. 185

SAP IS-U/CCS Sessions. 186

Caching SAP IS-U/CCS Data. 186

SAP IS-U/CCS Migration Data to Import File. 187

Error Handling. 187

Part VI: Business Content Integration. 189

Chapter 18: Business Content Integration. 190

Business Content Integration Overview. 190

DataSources. 191

Logical System in SAP. 191

Mappings for Business Content Integration. 191

Workflows for Business Content Integration. 193

PowerCenter Integration Service Processing. 194

Before You Begin. 195

Steps to Integrate with SAP Business Content. 195

Step 1. Prepare DataSources in SAP. 195

Activating a DataSource in SAP. 195

Customizing Fields in a DataSource. 196

Step 2. Import and Configure PowerCenter Objects. 196

Importing PowerCenter Objects from BCI_Mappings.xml. 196

Creating Database Tables for PowerCenter Objects. 197
Configuring an LMAPITarget Application Connection. 198
Identifying and Verifying the Basic IDoc Type in the Listener Mapping. 199
Step 3. Create a Table to Process Empty DataSources. 200
Step 4. Configure and Start the Listener Workflow. 200
Step 5. Create the Processing Mappings. 202
Update Modes. 202
Request Files. 203
Non-Hierarchy and Hierarchy DataSource Processing Mappings. 204
Steps to Create a Processing Mapping. 205
Generating and Executing SQL for the Relational Targets. 210
Step 6. Deploy the Request Files. 211
Step 7. Create the Send Request Workflows. 211
Step 8. Create the Processing Workflows. 212
Creating a Processing Session. 212
Creating a Cleanup Session. 212
Configuring the Processing Workflow. 213
Step 9. Schedule the Processing and Send Request Workflows. 213
Example. 214
Steps to Schedule the Processing and Send Request Workflows. 215
Troubleshooting Business Content Integration. 216

Part VII: SAP BW Data Extraction. 218

Chapter 19: SAP BW Extraction Through Open Hub Destination. 219

SAP BW Extraction Through Open Hub Destination Overview. 219
Extracting Data from SAP BW. 220
Step 1. Create an Open Hub Destination. 221
Create a Data Transfer Process. 221
Step 2. Create an SAP BW OHS Source Definition. 221
Step 3. Start the SAP BW Service. 224
Step 4. Configure and Start a PowerCenter Workflow. 224
Configuring Raw Data Transfer. 224
Step 5. Configure and Start a Process Chain. 225
Configuring and Starting a Process Chain from SAP BW. 225
Inserting a Data Transfer Process. 227
Inserting an InfoSpoke Process. 227
Configuring and Starting the Process Chain from PowerCenter. 227
Changing the Data Extraction Status. 229
Viewing Data Extraction Status. 230
Viewing Data Extraction Status from PowerCenter. 230
Viewing Data Extraction Status from SAP. 230
Troubleshooting Data Extraction from SAP BW through OHD. 231

Part VIII: Loading Data to SAP BI. 232

Chapter 20: Building Components to Load Data into SAP BI. 233

Building Components to Load Data into SAP BI Overview. 233

InfoSources for Loading Data. 233

DataSources for Loading Data. 234

SAP BI Hierarchy. 234

Transfer Methods for Writing Data to SAP BI. 235

Steps to Build Components to Load Data into SAP BI. 236

Step 1. Create an InfoSource or DataSource. 237

Creating a 7.x InfoSource in SAP BI 7.3. 237

Creating an InfoSource in SAP BI 7.0. 237

Creating an InfoSource in SAP BW 3.5 or Earlier. 239

Creating a 7.x DataSource in SAP BI 7.3. 239

Configuring Hierarchy Structure. 241

Step 2. Assign an External Logical System. 242

Step 3. Activate the InfoSource or DataSource. 242

Chapter 21: Building PowerCenter Objects to Load Data into SAP BI. 243

Building PowerCenter Objects to Load Data into SAP BI Overview. 243

Step 1. Import an InfoSource or a DataSource. 243

Step 2. Create a Mapping. 245

Filtering Data to Load into SAP BI. 245

Filtering Data from a Relational Source. 245

Filtering Data from a Flat File Source. 246

Filtering Data from an SAP ECC Source. 247

Configuring Mapping Parameters for Data Selection. 248

Chapter 22: Loading Data into SAP BI. 250

Loading Data into SAP BI Overview. 250

Step 1. Configure a Workflow to Load Data into SAP BI. 251

Partitions for SAP BI sessions. 251

Creating an SAP BI Session. 251

Creating a PowerCenter Workflow for an SAP BI Session. 252

Step 2. Configure an InfoPackage. 253

Creating and Scheduling an InfoPackage. 253

Setting a Data Selection Entry to Filter Data. 254

Step 3. Configure a Data Transfer Process. 254

Step 4. Configure a Process Chain to Load Data. 254

Creating the Process Chain and Inserting the Start Process. 255

Inserting an InfoPackage Process. 256

Inserting the ZPMSENDSTATUS ABAP Program. 256

Log Events. 257
Viewing SAP BW Service Log Events in the SAP BI Monitor. 258
InfoPackage Status. 258
Recovering a PowerCenter Workflow. 259
Troubleshooting Data Loading into SAP BI. 259

Appendix A: Data Type Reference. 262

SAP Data Types. 262
PowerExchange for SAP NetWeaver and SAP Data Types. 263
Overriding Data Types in the Application Source Qualifier. 265
Binary Data Types. 266
CHAR, CUKY, and UNIT Data Types. 266
PowerExchange for SAP BW and SAP Data Types. 266
Date/Time Data Types. 268
Binary Data Types. 268
Numeric Data Types. 268
Writing to SAP BW Date Columns. 268

Appendix B: Code Pages and Unicode Support. 271

Language Code Selection. 271
Code Page Selection. 272
Supported Code Pages. 273
Processing Unicode Data. 274
Processing Unicode Data with a Single Session. 275
Processing Unicode Data with Multiple Sessions. 275
Processing Unicode Data in ABAP. 275
Processing Unicode Data with Different Code Pages. 275

Appendix C: Glossary. 276

Index. 282

Preface

The *Informatica PowerExchange® for SAP NetWeaver User Guide for PowerCenter®* provides information to build mappings and run sessions to extract data from SAP NetWeaver into a data warehouse and write data into SAP NetWeaver. It is written for the data warehouse developers and software engineers who are responsible for extracting data from SAP NetWeaver into a data warehouse or loading data into SAP NetWeaver.

This guide assumes you have knowledge of relational database concepts and the database engines, PowerCenter, and SAP NetWeaver. You should also be familiar with the interface requirements for other supporting applications.

Informatica Resources

Informatica Network

Informatica Network hosts Informatica Global Customer Support, the Informatica Knowledge Base, and other product resources. To access Informatica Network, visit https://network.informatica.com.

As a member, you can:

- Access all of your Informatica resources in one place.
- Search the Knowledge Base for product resources, including documentation, FAQs, and best practices.
- View product availability information.
- Review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to search Informatica Network for product resources such as documentation, how-to articles, best practices, and PAMs.

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

Informatica Documentation

To get the latest documentation for your product, browse the Informatica Knowledge Base at https://kb.informatica.com/_layouts/ProductDocumentation/Page/ProductDocumentSearch.aspx.

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

Informatica Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. If you are an Informatica Network member, you can access PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

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

If you are an Informatica Network member, you can access Informatica Velocity resources at http://velocity.informatica.com.

If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

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

Informatica Global Customer Support

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

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link: http://www.informatica.com/us/services-and-training/support-services/global-support-centers.

If you are an Informatica Network member, you can use Online Support at http://network.informatica.com.

Part I: Getting Started with PowerExchange for SAP NetWeaver

This part contains the following chapters:

- Understanding PowerExchange for SAP NetWeaver, 18
- Installing and Configuring PowerExchange for SAP NetWeaver, 24
- Installing and Configuring PowerExchange for SAP NetWeaver BW, 40
- Installing and Configuring Transports, 53

CHAPTER 1

Understanding PowerExchange for SAP NetWeaver

This chapter includes the following topics:

- Understanding PowerExchange for SAP NetWeaver Overview, 18
- PowerCenter and SAP NetWeaver Integration Methods, 19
- PowerCenter and SAP NetWeaver BW Integration Methods, 21
- Communication Interfaces, 21
- Transport System, 22
- Operating System Profiles for SAP Sessions, 22

Understanding PowerExchange for SAP NetWeaver Overview

SAP NetWeaver is an application platform that integrates multiple business applications and solutions, such as Customer Relationship Management (CRM), Advanced Planner and Optimizer (APO), and Bank Analyzer. Developers can add business logic within SAP NetWeaver using Java 2 Enterprise Edition (J2EE) or Advanced Business Application Programming-Fourth Generation (ABAP/4 or ABAP), a language proprietary to SAP.

You can use the following integration methods to extract data from or load data into SAP systems:

- You can use the ABAP, Application Link Enabling (ALE), BAPI/RFC functions, data migration, or business content integration methods.
- You can also extract data from or load data to the SAP Business Information Warehouse.

Note: PowerExchange for SAP NetWeaver and PowerExchange for SAP NetWeaver BW require separate licenses.

SAP NetWeaver is the basis for SAP solutions. Because PowerCenter works with the SAP NetWeaver application platform, you can integrate PowerCenter with any SAP industry solution or mySAP application that provides BAPI/RFC or ALE integration methods.

The following figure shows how PowerCenter integrates with SAP NetWeaver:

PowerCenter and SAP NetWeaver Integration Methods

PowerExchange for SAP NetWeaver integrates with mySAP applications in the following ways:

- Data integration using the ABAP program
- IDoc integration using ALE
- Data integration using BAPI/RFC functions
- Data migration integration
- Business content integration

Data Integration Using the ABAP Program

You can extract data from mySAP applications using the ABAP program. Create a mapping in the Designer that uses the ABAP program. Generate and install the ABAP program on the SAP server that extracts the source data. When you configure a session, you can access the source data through streaming or a staged file. The PowerCenter Integration Service accesses streamed data through RFC and the deprecated CPI-C mode. It accesses staged files through FTP, SFTP, or standard file I/O, typically using network file sharing, such as NFS.

Perform the following steps to extract data from mySAP applications using the ABAP program:

1. Import an SAP ECC source definition.
2. Create a mapping.
3. Generate and install the ABAP program.

4. Create a session and run a workflow.

IDoc Integration Using ALE

You can integrate PowerCenter with mySAP applications using Application Link Enabling (ALE) to send and receive Intermediate Documents (IDocs). IDocs are messages that exchange electronic data between SAP applications or between SAP applications and external programs.

The message-based architecture of ALE comprises three layers:

- **Application layer.** Provides ALE an interface to R/3 to send or receive messages from external systems.
- **Distribution layer.** Filters and converts messages to ensure that they are compatible between different releases of R/3 and R/2.
- **Communications layer.** Enables ALE to support synchronous and asynchronous communication. You use IDocs for asynchronous communication.

The architecture of ALE provides a way to send IDocs as text files without connecting to a central database. This allows applications to communicate with each other without converting between formats to accommodate hardware or platform differences.

ALE has the following components:

- **Logical component.** Determines how messages flow between the various applications or systems.
- **Physical component.** Transport layer that routes IDoc messages using the tRFC (Transactional RFC) protocol.
- **Message types.** Application messages that classify categories of data. For example, ORDERS and MATMAS (Material Master).
- **IDoc types.** Data structure associated with the message type. For example, MATMAS01, MATMAS02 for MATMAS. IDocs contain the data belonging to the message type.

IDocs contain three record types:

- **Control record.** Identifies the message type.
- **Data records.** Contain the IDoc data in segments.
- **Status records.** Describe the status of the IDoc. Status record names are the same for each IDoc type.

Data Integration Using BAPI/RFC Functions

Business Application Programming Interfaces (BAPI) provide a way for third-party applications to integrate data with SAP. You use BAPIs to create, change, delete, list, and detail objects in SAP.

The Business Application Programming Interfaces allow object-oriented access to the SAP system through methods for the business object types. Together with the business object types, BAPIs define and document the interface standard at the business level.

BAPIs also provide an object-oriented view of business components in SAP. You define BAPIs in the SAP Business Objects Repository. You implement and store them as Remote Function Call (RFC) enabled function modules in the Function Builder of the ABAP Workbench. You can call BAPIs as an ABAP program within SAP. You use RFCs to call BAPIs outside of SAP.

Use a BAPI/RFC transformation to create, change, or delete data in mySAP applications. When you run a session with a BAPI/RFC transformation, the PowerCenter Integration Service makes the RFC function calls to SAP to process SAP data.

You can use a BAPI/RFC transformation for one of the following reasons:

- **Migrate data to SAP.** For example, your organization uses PeopleSoft applications for enterprise purchasing management. You want to migrate to mySAP applications to manage enterprise purchasing. Use PowerExchange for for PeopleSoft to extract data from PeopleSoft and a BAPI/RFC transformation to write purchase management data to mySAP applications.
- **Synchronize data in SAP.** For example, a mySAP application contains customer orders. You want to add line items to some of the orders. You can use a BAPI/RFC transformation to make a BAPI/RFC call to add the line items to the orders in SAP.

Data Migration Integration

You can migrate data from legacy applications, from other ERP systems, or data from any number of other sources and prepare it for input into mySAP applications. The PowerCenter Integration Service extracts the data from the data source and prepares the data in an SAP format flat file that you can load into mySAP applications.

Business Content Integration

You can integrate PowerCenter with mySAP applications to provide an efficient, high-volume data warehousing solution. SAP business content is a collection of metadata objects that can be integrated with other applications and used for analysis and reporting. SAP produces the business content data, and PowerCenter consumes it. PowerCenter can consume all or changed business content data from mySAP applications and write it to a target data warehouse. You can then use the data warehouse to meet analysis and reporting needs.

PowerCenter and SAP NetWeaver BW Integration Methods

PowerExchange for SAP NetWeaver BW can perform the following tasks:

- Extracting data from SAP BW
- Loading data into SAP BW

PowerExchange for SAP NetWeaver BW interacts with InfoCubes and InfoSources. An InfoCube is a self-contained dataset created with data from one or more InfoSources. An InfoSource is a collection of data that logically belongs together, summarized into a single unit.

Communication Interfaces

TCP/IP is the native communication interface between PowerCenter and SAP NetWeaver. PowerCenter and SAP NetWeaver also use the following interfaces:

- Remote Function Call (RFC)
- CPI-C (deprecated)

Remote Function Call

RFC is the remote communications protocol used by SAP NetWeaver. It is based on Remote Procedure Call (RPC). PowerCenter makes remote function calls to communicate with SAP NetWeaver.

To execute remote calls from PowerCenter, SAP NetWeaver requires connection information, and the service name and gateway on the application server. This information is stored in a configuration file named `sapnwrfc.ini` on the node hosting the PowerCenter Client and the node where the PowerCenter Integration Service and SAP BW Service processes run.

Common Program Interface-Communications

CPI-C is the communication protocol that enables online data exchange and data conversion between SAP and PowerCenter. PowerExchange for SAP NetWeaver uses CPI-C to communicate with SAP NetWeaver only when you run ABAP stream mode sessions. Effective in version 9.6.0, the CPI-C protocol is deprecated.

To initialize CPI-C communication with PowerCenter, SAP NetWeaver requires information, such as the host name of the application server and SAP gateway. This information is stored in a configuration file named `sapnwrfc.ini` on the node where the PowerCenter Integration Service process runs. The PowerCenter Integration Service uses parameters in the `sapnwrfc.ini` file to connect to SAP NetWeaver when you run ABAP stream mode sessions.

Transport System

The transport system is a set of ABAP programs installed on the SAP system. The ABAP programs import SAP metadata into the repository. The ABAP programs also enable run-time functionalities, such as passing mapping variables and filters.

You use the transport system in the following situations:

Configuring PowerExchange for SAP NetWeaver.

You need to transport some customized objects that were provided by Informatica to the SAP system. These objects include tables, structures, programs, and functions. The PowerCenter Integration Service calls custom objects when it makes a request to the SAP system.

Deploying run-time transports and ABAP programs from development to production.

To use ABAP to integrate with mySAP applications, deploy the run-time transports provided by Informatica and the ABAP programs installed by the PowerCenter Integration Service to extract data when you move from development to production.

The SAP system administrator must install and configure transports on the SAP system.

Operating System Profiles for SAP Sessions

If the PowerCenter Integration Service uses operating system profiles, the operating system user specified in the operating system profile must have access to run time files.

Use the following rules and guidelines if the PowerCenter Integration Service uses operating system profiles for an SAP session:

- The operating system profile must have permissions to access the \server\bin directory to read the `sapnwrfc.ini` file and create the RFC trace file.
- If the session writes to SAP NetWeaver BW and there is a filter condition in the InfoPackage, the operating system profile must have permissions to access the BWParam directory.

CHAPTER 2

Installing and Configuring PowerExchange for SAP NetWeaver

This chapter includes the following topics:

- Installing and Configuring PowerExchange for SAP NetWeaver Overview, 24
- Configuration Checklist, 26
- Installing the SAP NetWeaver RFC SDK Version 7.20 Libraries, 28
- Registering the PowerExchange for SAP NetWeaver Client, 30
- Defining PowerCenter as a Logical System in SAP, 30
- Configuring sapnwrfc.ini, 36
- Uninstalling PowerExchange for SAP NetWeaver, 39

Installing and Configuring PowerExchange for SAP NetWeaver Overview

PowerExchange for SAP NetWeaver requires configuration on both the PowerCenter and SAP systems. The administrators for each of these systems must perform the configuration tasks for their respective systems.

Before You Begin

Before you can install and configure PowerExchange for SAP NetWeaver, install and configure SAP. To use business content integration, verify that you have installed the SAP plug-in version 2003_1 or later. Install Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package on the Windows machine that hosts PowerCenter Integration Service and the PowerCenter Client.

Tip: Use separate development, test, and production SAP systems. Perform all development, testing, and troubleshooting on the development and test systems before deploying to the production systems.

Installing PowerExchange for SAP NetWeaver

To install PowerExchange for SAP NetWeaver, perform the following steps:

1. Install PowerCenter.
2. Register the PowerExchange for SAP NetWeaver client.
3. Install the SAP NetWeaver RFC SDK libraries.
4. Register the PowerExchange for SAP NetWeaver plug-in with the PowerCenter repository.

After you install PowerExchange for SAP NetWeaver, perform the configuration tasks.

Upgrading PowerExchange for SAP NetWeaver

The PowerCenter installer zip file contains transports for new installations and upgrades.

If you are upgrading from a previous version of PowerExchange for SAP NetWeaver, perform the following steps:

1. Delete existing transports.
2. Install the latest transports.
3. Make a copy of the `sapnwrfc.ini` file.
4. Install the current version of PowerCenter. When you install the current version of PowerCenter, you also upgrade to the latest version of PowerExchange for SAP NetWeaver.
5. Install the SAP NetWeaver RFC SDK libraries.
6. Update the PowerExchange for SAP NetWeaver client registration.

Upgrading to PowerExchange for SAP NetWeaver in Unicode Mode

To upgrade to PowerExchange for SAP NetWeaver in Unicode mode, perform the following steps:

1. Delete existing transports.
2. Install the current version of the PowerExchange for SAP NetWeaver transports with the following overwrite options:
   - Import Transport Request Again
   - Overwrite Originals
3. Install the SAP NetWeaver RFC SDK libraries.
4. Update the PowerExchange for SAP NetWeaver client registration.
5. Regenerate and reinstall ABAP programs on the SAP system after upgrading the system to Unicode.
6. Configure the PowerCenter Integration Service to run in Unicode mode.
7. Set ValidateDataCodePages to No in the Informatica Administrator.
8. Set the code page to UTF-8 in the SAP NetWeaver application connection object.

Registering the Plug-in

After you install PowerExchange for SAP NetWeaver, register the plug-in with the repository. When you upgrade from any previous PowerCenter version to the PowerCenter 9.1 version or above, you must manually register the PowerExchange for SAP plug-in file in the PowerCenter repository.

A plug-in is an XML file that defines the functionality of PowerExchange for SAP NetWeaver. To register the plug-in, the repository must be running in exclusive mode. Use the pmrep RegisterPlugin command to register the plug-in.

The plug-in file for PowerExchange for SAP NetWeaver is pmsapplg.xml. Register the plug-in with the update option enabled. The RegisterPlugin command uses the following syntax:

```
Registerplugin -i <Informatica installation Directory>/server/bin/native/pmsapplg.xml -e
-N
```

For more information about the pmrep RegisterPlugin command, see the *Informatica Command Reference*.

Note: If you do not have the correct privileges to register the plug-in, contact the user who manages the PowerCenter Repository Service.

Configuration Checklist

After you install and configure SAP and PowerCenter, you must perform additional configuration tasks to use one or more of the following PowerExchange for SAP NetWeaver integration methods:

- Data integration using ABAP
- IDoc integration using ALE
- Data integration using BAPI/RFC
- Data migration
- Business content integration

Configuration Tasks and Integration Methods

Some configuration tasks apply to multiple integration methods. When you perform a configuration task for one integration method, you may be able to skip that task for another integration method. For example, if you configure a connection to a specific SAP application server in the `sapnwrfc.ini` file for ABAP, you can skip that task when you configure ALE integration.

Other configuration tasks apply to only one integration method.

If you have multiple SAP systems, you must repeat certain configuration tasks. For example, you must configure one SAP_ALE_IDoc_Reader application connection for each SAP system from which you want to receive outbound IDocs.

The following table lists the configuration tasks and the integration methods that apply to them:

Configuration Task	ABAP	ALE	RFC/ BAPI	DMI	Business Content
Install SAP NetWeaver RFC SDK libraries.	Yes	Yes	Yes	Yes	Yes
Define PowerCenter as a logical system in SAP.	No	Yes	No	No	Yes. Use the ZINFABCI ABAP program provided by Informatica.
Configure the `sapnwrfc.ini` file.	Yes	Yes	Yes	Yes	Yes
Configure an SAP_ALE_IDoc_Reader application connection.	No	Yes, if receiving outbound IDocs	No	No	Yes
Configure an SAP_ALE_IDoc_Writer application connection.	No	Yes, if sending inbound IDocs	No	No	Yes
Configure an SAP R/3 application connection.	Yes	No	No	No	No
Configure an FTP or SFTP connection.	Yes, if running file mode sessions	No	No	No	No
Configure an SAP BAPI/RFC application connection.	No	No	Yes	No	No
Prepare DataSources in SAP.	No	No	No	No	Yes

Integrating with SAP Using ABAP

To integrate with SAP using ABAP programs, perform the following steps:

1. Verify that the SAP user has the appropriate authorization.
2. Configure the `sapnwrfc.ini` file to run sessions in file mode or stream mode through RFC communication. You can also configure the `sapnwrfc.ini` file to run sessions in stream mode through the deprecated Common Program Interfaces-Communication (CPI-C) protocol. Refer to operating system support restrictions for sessions in stream mode when you install SAP NetWeaver RFC SDK version 7.20 libraries.
3. Configure an SAP R/3 application connection or an FTP or SFTP connection.

Integrating with SAP Using ALE

To integrate with SAP using ALE, perform the following steps:

1. Define PowerCenter as a logical system in SAP.

2. Configure the `sapnwrfc.ini` file for RFC communication with SAP. Create a connection for an RFC server program registered at an SAP gateway. Configure the connection to listen for outbound IDocs. The PROGRAM_ID parameter must match the Program ID you configured for the logical system in SAP.
3. Configure SAP_ALE_IDoc_Reader application connections to receive outbound IDocs from SAP.
4. Configure an SAP_ALE_IDoc_Writer application connection to send inbound IDocs to SAP.

Integrating with SAP Using BAPI/RFC Functions

To integrate with SAP using BAPI/RFC functions, perform the following steps:

1. Configure the `sapnwrfc.ini` file for RFC communication with SAP.
2. Configure an SAP R/3 application connection.

Migrating Data to SAP

To migrate data to SAP, configure an connection in the `sapnwrfc.ini` file for RFC communication with SAP.

Integrating with SAP Business Content

To integrate with SAP business content, perform the following steps:

1. Define PowerCenter as a logical system in SAP.
2. Configure the `sapnwrfc.ini` file for RFC communication with SAP. Configure a connection for an RFC server program registered at an SAP gateway to consume business content data. Set PROGRAM_ID to INFACONTNT.
3. Configure SAP_ALE_IDoc_Reader application connections to consume business content data from SAP.
4. Configure an SAP_ALE_IDoc_Writer application connection to send requests for business content data to SAP.
5. Activate and configure each DataSource in SAP before you create a processing mapping for the DataSource.

Installing the SAP NetWeaver RFC SDK Version 7.20 Libraries

Download and install the SAP NetWeaver RFC SDK 7.20 libraries for the PowerCenter Client and the operating system that hosts the PowerCenter Integration Service process when you configure PowerExchange for SAP NetWeaver. Contact SAP customer support if you have problems downloading the SAP NetWeaver RFC SDK libraries.

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:
https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

To download and install the SAP NetWeaver RFC SDK Version 7.20 libraries, perform the following steps:

1. Go to the SAP Service Marketplace: http://service.sap.com.

2. Download Unicode SAP NetWeaver RFC SDK 7.20 libraries that are specific to the operating system that hosts the Integration Service process and the Windows 32-bit SAP NetWeaver RFC SDK libraries.

   The following table lists the libraries corresponding to the different operating systems:

Operating System	Unicode SAP NetWeaver RFC SDK libraries
AIX 64-bit	- libsapucum.so - libsapnwrfc.so - libicuuc34.a - libicui18n34.a - libicudecnumber.so - libicudata34.a
Linux.64	- libicuuc.so.34 - libsapucum.so - libicudata.so.34 - libicui18n.so.34 - libsapnwrfc.so - libicudecnumber.so
Solaris 64-bit	- libicuuc.so.34 - libsapucum.so - libicudata.so.34 - libicui18n.so.34 - libsapnwrfc.so - libicudecnumber.so
Windows EM64T	- libsapucum.dll - libicudecnumber.dll - sapnwrfc.dll
Windows 32-bit	- libsapucum.dll - libicudecnumber.dll - sapnwrfc.dll
Windows 32-bit (client only)	- libsapucum.dll - libicudecnumber.dll - sapnwrfc.dll - icuin34.dll - icuuc34.dll - icudt34.dll

3. Copy the SAP NetWeaver RFC SDK libraries to the following directory: `<Informatica Installation Directory>\server\bin`

   Copy the Windows 32-bit (client only) files to the following directory: `<Informatica Installation Directory>\client\bin`

4. Set the following permissions for each SAP NetWeaver RFC SDK library:
   - Read, write, and execute permissions for the current user
   - Read and execute permissions for all other users

5. Download the libsapu16_mt.so library from the SAPEXE.SAR or UCLIB.SAR package that is available as part of the SAP kernel. You need to download the libsapu16_mt.so library only if you use UNIX or Linux operating systems.

6. Copy the libsapu16_mt.so library to the following directory: `<Informatica Installation Directory>\server\bin`

Registering the PowerExchange for SAP NetWeaver Client

When you install or upgrade PowerExchange for SAP NetWeaver, you need to register the PowerExchange for SAP NetWeaver Client.

To register the PowerExchange for SAP NetWeaver Client:

1. Go to <PowerCenter Installation Directory>\Client\bin.
2. Double-click PWX_SAP.reg. If the client machine is Windows 7 64-bit, double-click on PWX_SAP_64.reg.
3. Click Yes, and then click OK.

   The PowerExchange for SAP NetWeaver Client is registered.

Defining PowerCenter as a Logical System in SAP

To use IDoc ALE integration or business content integration, you must define PowerCenter as an external logical system in SAP. You perform different steps to create a logical system for each integration method.

Creating a Logical System for IDoc ALE Integration

Before PowerCenter can send and receive IDocs from SAP, define PowerCenter as a logical system in SAP. When you define PowerCenter as a logical system, SAP acknowledges PowerCenter as an external system that can receive outbound IDocs from SAP and send inbound IDocs to SAP.

Create a single logical system in SAP for IDoc ALE integration with PowerCenter. If you have multiple PowerCenter Integration Services in the PowerCenter installation, each PowerCenter Integration Service must use the same connection for the RFC server program in the `sapnwrfc.ini` file.

Perform the following steps to define PowerCenter as a logical system:

1. Create a logical system in SAP for PowerCenter.
2. Create an RFC destination for PowerCenter.
3. Create a tRFC port for the RFC destination.
4. Create a partner profile for PowerCenter.
5. Create outbound and inbound parameters for the partner profile.

Note: These steps might differ based on the version of the SAP system you use. For complete instructions on creating a logical system in SAP, see the SAP documentation.

Step 1. Create a Logical System for PowerCenter

Create a logical system for PowerCenter in SAP. When you create a logical system for PowerCenter, SAP acknowledges PowerCenter as an external system that can receive outbound IDocs from SAP and send inbound IDocs to SAP.

To create a logical system in SAP:

1. Go to transaction SALE.

   The Display IMG window appears.

2. Expand the tree to locate the Application Link Enabling > Sending and Receiving Systems > Logical Systems > Define Logical System operation.
3. Click the IMG - Activity icon to run the Define Logical System operation.

   An informational dialog box appears.
4. Click Enter.

   The Change View Logical Systems window appears.
5. Click New Entries.

   The New Entries window appears.
6. Enter a name and description for the new logical system entry for PowerCenter. For example, you can enter LSPowerCenterALE as the name.

Step 2. Create an RFC Destination

Create an RFC destination and program ID for PowerCenter. The program ID value you enter must match the PROGRAM_ID parameter that you specified in the connection for the RFC server program in the `sapnwrfc.ini` file defined for the PowerCenter Integration Service.

If the SAP system is a Unicode system and the PowerCenter Integration Service runs on AIX (64-bit), Linux (32-bit), Solaris (64-bit), or Windows, you must configure the logical system to communicate in Unicode mode. SAP provides Unicode RFC libraries for these operating systems. When the PowerCenter Integration Service runs on one of these operating systems, it uses the Unicode RFC libraries to process Unicode data.

To create an RFC destination in SAP:

1. Go to transaction SM59.

   The Display and Maintain RFC Destinations window appears.
2. Click Create.

   The RFC Destination window appears.
3. For RFC Destination, enter the name of the logical system you created in "Step 1. Create a Logical System for PowerCenter" on page 30.

   For example, LSPowerCenterALE.
4. Enter T for Connection Type to create a TCP/IP connection.
5. Enter a description for the RFC destination.
6. Click Save.

   The window refreshes.
7. For Activation Type, click Registration.
8. For Program ID, enter the same name that you entered for RFC Destination.

   For example, LSPowerCenterALE. Use this Program ID as the value for the PROGRAM_ID parameter in the `sapnwrfc.ini` file.
9. If the SAP system is a Unicode system and the PowerCenter Integration Service runs on AIX (64-bit), Linux (32-bit), Solaris (64-bit), or Windows, click the Special Options tab, and then select the Unicode option under Character Width in Target System.

Step 3. Create a tRFC Port for the RFC Destination

Create a tRFC port for the RFC destination you defined in SAP. SAP uses this port to communicate with PowerCenter.

To create a tRFC port for the RFC destination:

1. Go to transaction WE21.
2. Click Ports > Transactional RFC.
3. Click Create.

   The Ports in IDoc Processing dialog box appears.
4. Click Generate Port Name, or click Own Port Name and enter a name.
5. Click Enter.
6. Enter a description for the port.
7. Select the IDoc record version type.
8. Enter the name of the RFC Destination you created in "Step 2. Create an RFC Destination" on page 31.

   For example, LSPowerCenterALE.

Step 4. Create a Partner Profile for PowerCenter

Create a partner profile for the logical system you defined for PowerCenter. When SAP communicates with an external system, it uses the partner profile to identify the external system.

To create a partner profile for PowerCenter:

1. Go to transaction WE20.
2. Click Create.
3. Enter the following properties:

Partner Profile Property	Description
Partner number	Name of the logical system you created for PowerCenter, for example, LSPowerCenterALE.
Partner type	Enter LS for logical system.

4. Click the Post-processing tab and enter the following properties:

Partner Profile Property	Description
Type	Enter US for user.
Agent	Enter the SAP user login name.
Lang	Enter EN for English.

5. Click the Classification tab and enter the following properties:

Partner Profile Property	Description
Partner class	Enter ALE.
Partner status	Enter A for active.

Step 5. Create Outbound and Inbound Parameters for the Partner Profile

Create outbound and inbound parameters for the partner profile defined for PowerCenter. Outbound parameters define the IDoc message type, IDoc basic type, and port number for outbound IDocs. SAP uses these parameters when it sends IDocs to PowerCenter. Create an outbound parameter for each IDoc message type that SAP sends to PowerCenter.

Inbound parameters define the IDoc message type for inbound IDocs. SAP uses these parameters when it receives IDocs from PowerCenter. Create an inbound parameter for each IDoc message type that SAP receives from PowerCenter.

To create outbound and inbound parameters for the partner profile:

1. From the partner profiles window, click Create Outbound Parameter.

   The Partner Profiles: Outbound Parameters window appears.

2. Enter the following properties:

Outbound Parameter Property	Description
Message Type	Select the IDoc message type the SAP system sends to PowerCenter.
Receiver Port	Select the tRFC port number you defined in "Step 3. Create a tRFC Port for the RFC Destination" on page 31.
IDoc Type	Select the IDoc basic type of the IDocs the SAP system sends to PowerCenter.

3. Click Save.

   The Packet Size property appears.

4. For Packet Size, enter a value between 10 and 200 IDocs.

   The packet size determines the number of IDocs that SAP sends in one packet to PowerCenter.

5. Click Enter.
6. Repeat steps from 1 to 5 to create an outbound parameter for each IDoc message type that SAP sends to PowerCenter.
7. Click Create Inbound Parameter.

   The Partner Profiles: Inbound Parameters window appears.

8. Enter the following properties for each inbound parameter:

Inbound Parameter Property	Description
Message Type	Select the IDoc message type the SAP system receives from PowerCenter.
Process Code	Select the process code. The SAP system uses the process code to call the appropriate function module to process the IDocs it receives.

9. Click Enter.
10. Repeat steps 7 through 9 to create an inbound parameter for each IDoc message type that SAP receives from PowerCenter.

Creating a Logical System for Business Content Integration

Before PowerCenter can consume business content data, define PowerCenter as a logical system in SAP.

You can create a single logical system in SAP for business content integration with PowerCenter. If you use multiple PowerCenter Integration Services, each PowerCenter Integration Service must use the same connection for the RFC server program in the `sapnwrfc.ini` file.

Informatica provides an ABAP program named /INFATRAN/ZINFABCI that creates or deletes the logical system in SAP for business content integration. The ZINFABCI ABAP program belongs to the /INFATRAN/ namespace reserved by Informatica in the SAP system. The ZINFABCI program performs the following steps in SAP when it creates the logical system:

1. Finds the logical system name of the SAP system.
2. Creates a unique IDoc type for business content integration.
3. Creates a logical system for PowerCenter.
4. Creates an RFC destination for PowerCenter.
5. Creates a tRFC port for the RFC destination.
6. Creates a partner profile with outbound and inbound parameters required for business content integration.
7. Registers the logical system with the business content integration framework.

A logical system uses an SAP user account to perform background tasks within SAP, such as activating DataSources, preparing data for transfer to PowerCenter, and extracting data. The PowerCenter Integration Service initiates these tasks in a session.

When you create a logical system, you need to choose a background user. Choose the default background user or a different background user. The default background user is ALEREMOTE. It is the default user name for ALE partner profile and the system level background user used in IDoc communication.

When you choose a different background user, use one of the following options:

- **Select an existing SAP user account.** When you select an existing SAP user account, the ABAP program updates the user account with the “S_BI-WX_RFC - Business Information Warehouse, RFC User Extraction” authorization profile. This authorization profile is required to perform background business content tasks within SAP.
- **Enter a new SAP user name and password.** The ABAP program creates the user account profile with the “S_BI-WX_RFC - Business Information Warehouse, RFC User Extraction” authorization profile only.

When you connect to SAP to create a business content integration processing mapping, use the logical system user name.

After you create a logical system for business content integration in SAP, you must configure a connection for an RFC server program registered at an SAP gateway in the `sapnwrfc.ini` file. Set the DEST and PROGRAM_ID parameters equal to the logical system name as configured in SAP for business content. The default name is INFACONTNT. You will use the value you set for the DEST parameter when you configure application connections.

For example:

```
DEST=INFACONTNT
PROGRAM_ID=INFACONTNT
GWHOST=salesSAP
GWSERV=sapgw00
```

After you create a logical system for business content integration in SAP, you must configure the gateway host and gateway services for the RFC destination that you created for business content integration. To configure the gateway host and gateway services, on the SAP server machine, go to transaction SM59. Click the **Technical Settings** tab and enter the gateway host name and gateway service name in the **Gateway Options** section.

Creating the Logical System

Use the /INFATRAN/ZINFABCI ABAP program to create the logical system.

To create the SAP logical system:

1. In SAP, enter /n/INFATRAN/ZINFABCI in the command field.

   The Administration: Communication Between SAP and Informatica window appears.
2. Select Create Communication Settings.
3. Select ALEREMOTE for Background User in SAP System. Or, enter a different user name.
4. If you enter a different user name, enter a password for Background User Password and enter it again to confirm.
5. Enter the logical system name for PowerCenter.

   The default name is INFACONTNT.
6. Enter an X for Update if User Exists.
7. Click Execute to save the logical system.

Configuring Unicode Mode

If the SAP system is a Unicode system, and the PowerCenter Integration Service runs on AIX (64-bit), Linux (32-bit), Solaris (64-bit), or Windows, you must configure the logical system to communicate in Unicode mode.

SAP provides Unicode RFC libraries for these operating systems. When the PowerCenter Integration Service runs on one of these operating systems, it uses the Unicode RFC libraries to process Unicode data.

To configure Unicode mode:

1. Go to the sm59 transaction.
2. Select the RFC Destination created for PowerCenter.

   The RFC Destination page appears.
3. Select the Special Options tab.
4. In the Character Width in Target System, select the Unicode option.

Deleting the Logical System Created for Business Content Integration

When you delete the logical system created for business content integration, you also delete all activated DataSources associated with the logical system. Delete a logical system only when you want to discontinue integrating PowerExchange for SAP NetWeaver with SAP business content.

Deleting the logical system also deletes the “S_BI-WX_RFC - Business Information Warehouse, RFC User Extraction” authorization profile from the user account profile. However, deleting a logical system does not delete the SAP user account.

Before you delete a logical system, make sure the business content integration listener workflow is running.

Important: You cannot recover deleted DataSources.

To delete the logical system created for business content integration:

1. In SAP, enter /n/INFATRAN/ZINFABCI in the command field.

   The Administration: Communication Between SAP and Informatica window appears.
2. Select a Background User in SAP System.
3. Select Delete Communication Settings, and then click Execute.

Configuring sapnwrfc.ini

SAP uses the communications protocol, Remote Function Call (RFC), to communicate with other systems. SAP stores RFC-specific parameters and connection information in a file named `sapnwrfc.ini`.

To use RFC, configure `sapnwrfc.ini` on the machines hosting the PowerCenter Client and PowerCenter Integration Service with parameters that enable communication with SAP. This file enables PowerCenter to connect to the SAP system as an RFC client.

You can also configure `sapnwrfc.ini` if you are integrating with mySAP applications using ABAP. Configure the file on the node where the PowerCenter Integration Service process runs to include parameters specific to SAP. This file enables PowerCenter to initiate CPI-C with the SAP system.

If SAP GUI is not installed on the machine that uses the `sapnwrfc.ini` file, you must make entries in the Services file to run stream mode sessions.

When you install PowerCenter, the installer copies a sample `sapnwrfc.ini` file to the following directories:

```
<PowerCenter Installation Directory>\clients\PowerCenter\client\bin
```

```
<PowerCenter Installation Directory>\server\bin
```

You can customize the sample file based on your requirements.

sapnwrfc.ini Connection Types

Use the `sapnwrfc.ini` file to configure the connections that you want to use.

You can configure the following types of connections in the `sapnwrfc.ini` file:

Connection to a specific SAP application server

Create this connection to enable communication between an RFC client and an SAP system. Each connection entry specifies one application server and one SAP system. Use the same connection for multiple integration methods.

Use this connection for data integration using ABAP, ALE, BAPI/RFC functions, data migration, and business content.

Connection to use SAP load balancing

Create this connection to enable SAP to create an RFC connection to the application server with the least load at run time. Use this connection when you want to use SAP load balancing.

Use this connection for data integration using ABAP, ALE, BAPI/RFC functions, data migration, and business content.

Connection to an RFC server program registered at an SAP gateway

Create this connection to connect to an SAP system from which you want to receive outbound IDocs or consume business content data using ALE.

To consume business content data using ALE, set the PROGRAM_ID to INFACONTNT.

Sample sapnwrfc.ini File

The following sample shows a connection entry for a specific SAP application server in the `sapnwrfc.ini` file:

```
DEST=sapr3
ASHOST=sapr3
SYSNR=00
```

The following sample shows a connection entry for SAP load balancing in the `sapnwrfc.ini` file:

```
DEST=sapr3
R3NAME=ABV
MSHOST=infamessageserver.informatica.com
GROUP=INFADEV
```

The following sample shows a connection entry for an RFC server program registered at an SAP gateway in the `sapnwrfc.ini` file:

```
DEST=sapr346CLSQA
PROGRAM_ID=PID_LSRECEIVE
GWHOST=sapr346c
GWSERV=sapgw00
```

Configuring a Connection in sapnwrfc.ini

The PowerCenter Client and the PowerCenter Integration Service use the connection entries defined in the `sapnwrfc.ini` file to connect to SAP. The connection entries that you define in the `sapnwrfc.ini` file will depend on the type of connection that you want to create.

The Designer connects to SAP to import metadata into the repository. The PowerCenter Integration Service connects to SAP to read and write data as an RFC client using the database connection that you create in the Workflow Manager.

Important: Use a DOS editor or Wordpad to configure the `sapnwrfc.ini` file. Notepad can corrupt the file.

To configure a connection entry in the `sapnwrfc.ini` file:

1. Open the `sapnwrfc.ini` file.

2. Enter the following connection parameters based on the type of connection that you want to create:

sapnwrfc.ini Parameter	Description
DEST	Logical name of the SAP system for the connection. Use this parameter for the following types of connections: - Connection to a specific SAP application server - Connection to use load balancing - Connection to an RFC server program registered at an SAP gateway All DEST entries must be unique. You must have only one DEST entry for each SAP system. You can use up to 32 characters.
ASHOST	Host name or IP address of the SAP application. PowerCenter uses this entry to attach to the application server. Use this parameter to create a connection to a specific SAP application server.
SYSNR	SAP system number. Use this parameter to create a connection to a specific SAP application server.
R3NAME	Name of the SAP system. Use this parameter to create a connection to use SAP load balancing.
MSHOST	Host name of the SAP Message Server. Use this parameter to create a connection to use SAP load balancing.
GROUP	Group name of the SAP application server. Use this parameter to create a connection to use SAP load balancing.
PROGRAM_ID	Program ID. The Program ID must be the same as the Program ID for the logical system that you define in the SAP system to send or receive IDocs or to consume business content data. For business content integration, set to INFACONTNT. Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
GWHOST	Host name of the SAP gateway. Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
GWSERV	Server name of the SAP gateway. Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
TRACE	Debugs RFC connection-related problems. Use this parameter for the following types of connections: - Connection to a specific SAP application server - Connection to use load balancing - Connection to an RFC server program registered at an SAP gateway Set one of the following values based on the level of detail that you want in the trace: - 0. Off. - 1. Brief. - 2. Verbose. - 3. Full.

3. If you are connecting to multiple SAP systems, define the connection parameters for each system in the `sapnwrfc.ini` file with unique DEST parameters.

Creating Entries in the Services File for Stream Mode Sessions

When you install SAP GUI, the installer makes entries in the Services file on the local system. If you have not installed SAP GUI on the node where the PowerCenter Integration Service process runs, you must manually create these entries in the Services file to run stream mode sessions.

To create entries in the Services file:

1. Open the Services file on the PowerCenter Integration Service system.

   On Windows 2000 or 2003, look in the \WINNT\system32\drivers\etc directory.

   On Windows XP, look in the \WINDOWS\system32\drivers\etc directory.

   On UNIX, look in the /etc directory.

2. Create the following entries:

   ```
   sapdp<system number>   <port number of dispatcher service>/tcp
   sapgw<system number>   <port number of gateway service>/tcp
   ```

   Check with the SAP administrator for the port numbers of the gateway service and the dispatcher service.

Uninstalling PowerExchange for SAP NetWeaver

You can uninstall PowerExchange for SAP NetWeaver when you uninstall PowerCenter. The uninstaller does not remove the `sapnwrfc.ini` file.

When you uninstall PowerExchange for SAP NetWeaver, you might need to clean up the SAP system.

Cleaning Up the SAP System

Perform the following tasks to clean up the SAP system:

- **Delete transport objects from SAP.** Use SE10 and SE80 transactions to delete the transport objects that you installed to run PowerExchange for SAP NetWeaver.
- **Uninstall ABAP programs.** If you used ABAP to extract data from SAP, uninstall ABAP programs from SAP.
- **Manually remove ALE configurations.** If you used Application Link Enabling (ALE) to send or receive outbound IDocs, manually remove the ALE configurations from SAP.
- **Delete communication settings using the ZINFABCI transaction.** If you used business content integration, use the ZINFABCI transaction to delete communication settings.

CHAPTER 3

Installing and Configuring PowerExchange for SAP NetWeaver BW

This chapter includes the following topics:

- Installing and Configuring PowerExchange for SAP NetWeaver BW Overview, 40
- Installing the SAP NetWeaver RFC SDK Version 7.20 Libraries, 41
- Registering the PowerExchange for SAP NetWeaver Client, 43
- Creating Profiles for Production and Development Users, 43
- Defining PowerCenter as a Logical System in SAP BW, 47
- Configuring sapnwrfc.ini, 47
- Creating the SAP BW Service, 50
- Importing the ABAP Program into SAP BW, 52
- Troubleshooting PowerExchange for SAP NetWeaver BW Installation and Configuration, 52

Installing and Configuring PowerExchange for SAP NetWeaver BW Overview

PowerExchange for SAP NetWeaver BW requires configuration on both the PowerCenter and SAP BW systems. The administrators for each of these systems must perform the configuration tasks for their respective systems.

Before You Begin

Perform the following steps before you configure PowerExchange for SAP NetWeaver BW:

1. Configure PowerExchange for SAP NetWeaver to extract data from SAP BW using Open Hub Services (OHS) (optional).
2. Configure SAP NetWeaver BW.
3. Transport objects to SAP BW if you are extracting data from SAP BW using OHS and if the SAP BW system is separate from the SAP system.

4. Install Microsoft Visual C++ 2005 Service Pack 1 Redistributable Package on the Windows machine that hosts PowerCenter Integration Service and the PowerCenter Client.

Installing and Configuring PowerExchange for SAP NetWeaver BW

To install PowerExchange for SAP NetWeaver BW, perform the following steps:

1. Install PowerCenter.
2. Install the SAP NetWeaver RFC SDK libraries.
3. Register the PowerExchange for SAP NetWeaver BW client.
4. Create profiles for development and production users.
5. Define PowerCenter as a logical system in SAP BW.
6. Configure the `sapnwrfc.ini` file to connect to SAP as an RFC client.
7. Create and enable the SAP BW Service.
8. Import the ABAP program.
9. Configure an application connection to connect to SAP BW.

Upgrading PowerExchange for SAP NetWeaver BW

Perform the following steps if you are upgrading from a previous version of PowerExchange for SAP NetWeaver BW:

1. Make a copy of the sapnwrfc.ini file.
2. Uninstall the previous version.
3. Install the current version of PowerCenter. When you install the current version of PowerCenter, you also upgrade to the latest version of PowerExchange for SAP NetWeaver BW.
4. Install the SAP NetWeaver RFC SDK libraries.
5. Update the PowerExchange for SAP NetWeaver BW Client registration.

Upgrading the SAP BW Service

If you used a linked domain repository in the previous installation, specify a repository that exists in the domain for the associated repository of the SAP BW Service. You cannot associate the SAP BW Service in the local PowerCenter domain with a PowerCenter Repository Service in another PowerCenter domain. When you upgrade a domain, the SAP BW Service loses the linked domain repository association.

Installing the SAP NetWeaver RFC SDK Version 7.20 Libraries

Download and install the SAP NetWeaver RFC SDK 7.20 libraries for the PowerCenter Client and the operating system that hosts the PowerCenter Integration Service process when you configure PowerExchange for SAP NetWeaver. Contact SAP customer support if you have problems downloading the SAP NetWeaver RFC SDK libraries.

For more information about product requirements and supported platforms, see the Product Availability Matrix on Informatica Network:
https://network.informatica.com/community/informatica-network/product-availability-matrices/overview

To download and install the SAP NetWeaver RFC SDK Version 7.20 libraries, perform the following steps:

1. Go to the SAP Service Marketplace: http://service.sap.com.
2. Download Unicode SAP NetWeaver RFC SDK 7.20 libraries that are specific to the operating system that hosts the Integration Service process and the Windows 32-bit SAP NetWeaver RFC SDK libraries.

   The following table lists the libraries corresponding to the different operating systems:

Operating System	Unicode SAP NetWeaver RFC SDK libraries
AIX 64-bit	- libsapucum.so - libsapnwrfc.so - libicuuc34.a - libicui18n34.a - libicudecnumber.so - libicudata34.a
Linux.64	- libicuuc.so.34 - libsapucum.so - libicudata.so.34 - libicui18n.so.34 - libsapnwrfc.so - libicudecnumber.so
Solaris 64-bit	- libicuuc.so.34 - libsapucum.so - libicudata.so.34 - libicui18n.so.34 - libsapnwrfc.so - libicudecnumber.so
Windows EM64T	- libsapucum.dll - libicudecnumber.dll - sapnwrfc.dll
Windows 32-bit	- libsapucum.dll - libicudecnumber.dll - sapnwrfc.dll
Windows 32-bit (client only)	- libsapucum.dll - libicudecnumber.dll - sapnwrfc.dll - icuin34.dll - icuuc34.dll - icudt34.dll

3. Copy the SAP NetWeaver RFC SDK libraries to the following directory: `<Informatica Installation Directory>\server\bin`

   Copy the Windows 32-bit (client only) files to the following directory: `<Informatica Installation Directory>\client\bin`
4. Set the following permissions for each SAP NetWeaver RFC SDK library:
   - Read, write, and execute permissions for the current user

- Read and execute permissions for all other users

5. Download the libsapu16_mt.so library from the SAPEXE.SAR or UCLIB.SAR package that is available as part of the SAP kernel. You need to download the libsapu16_mt.so library only if you use UNIX or Linux operating systems.
6. Copy the libsapu16_mt.so library to the following directory: `<Informatica Installation Directory>\server\bin`

Registering the PowerExchange for SAP NetWeaver Client

When you install or upgrade PowerExchange for SAP NetWeaver BW, you need to register the PowerExchange for SAP NetWeaver BW Client.

Note: If you completed this task when you configured PowerExchange for SAP NetWeaver, you can skip this task.

To register the PowerExchange for SAP NetWeaver BW Client:

1. Go to <PowerCenter Installation Directory>\Client\bin.
2. Double-click PWX_SAP.reg. If the client machine is Windows 7 64-bit, double-click on PWX_SAP_64.reg.
3. Click Yes, and then click OK.

   The PowerExchange for SAP NetWeaver BW Client is registered.

Creating Profiles for Production and Development Users

The SAP administrator needs to create product and development user authorization profiles when you extract data from or load data to SAP NetWeaver BW or SAP BW 3.5.

Profiles for Extracting Data from SAP BW 3.5

Use the following tables to determine the authorization profiles for extracting data from SAP BW 3.5.

Profiles for Development Users

The following table shows the authorization profile configuration for development users:

Integration Feature	Description	Class	Field Values
S_RFC	Authorization Check for RFC Objects	Cross Application Authorization Objects	Function groups: - RSB3RD - SYST - RFC1

Profiles for Production Users

The following table shows the authorization profile configuration for production users:

Integration Feature	Description	Class	Field Values
S_RFC	Authorization check for RFC objects	Cross Application Authorization Objects	Function groups: - RSB3RD - RSPC_API - SYST - RFC1
S_RS_ODSO	Administrator Workbench - ODS Object	Business Information Warehouse	Activity: Display, Maintain Sub Object for ODS Object: Data
S_RS_IOMAD	Administrator Workbench - Maintain Master data	Business Information Warehouse	Activity: Display, Maintain
S_RS_ICUBE	Administrator Workbench - InfoCube Object	Business Information Warehouse	Activity: Display, Maintain InfoCube subobject: Data
RSPC (TRANSACTION)	Transaction Code Check at Transaction start	Cross Application Authorization Objects	

When you add the RSPC transaction is added, SAP NetWeaver BW adds the following authorization objects:

- Background Processing: Background Administrator
- Background Processing: Operations on Background Jobs
- Background Processing: Background User Name

Make sure that these authorization objects are active.

Profiles for Extracting Data from SAP BW

Use the following tables to determine the authorization profiles for SAP BW development and production users.

Profiles for Development Users

The following table shows the authorization profile configuration for development users:

Integration Feature	Description	Class	Field Values
S_RFC	Authorization Check for RFC Objects	Cross Application Authorization Objects	Function groups: - RSB3RD - SYST - RFC1
S_RS_OHDST	Data Warehousing Workbench- Open Hub Destination	Business Information Warehouse	Activity: Display, Maintain Sub object for Open Hub Destination: Definition

Profiles for Production Users

The following table shows the authorization profile configuration for production users:

Integration Feature	Description	Class	Field Values
S_RFC	Authorization check for RFC objects	Cross Application Authorization Objects	Function groups: - RSB3RD - RSPC_API - SYST - RFC1
S_RS_ODSO	Administrator Workbench - ODS Object	Business Information Warehouse	Activity: Display, Maintain Sub Object for ODS Object: Data
S_RS_IOMAD	Administrator Workbench - Maintain Master data	Business Information Warehouse	Activity: Display, Maintain
S_RS_ICUBE	Administrator Workbench - InfoCube Object	Business Information Warehouse	Activity: Display, Maintain InfoCube subobject: Data
S_RS_OHDST	Data Warehousing Workbench- Open Hub Destination	Business information warehouse	Activity: Display, Maintain Sub object for Open Hub Destination: Data
S_RS_PC	Data Warehousing Workbench- Process Chains	Business information warehouse	Activity: Display, Execute, Maintain Sub object for Process Chain: Definition, protocols and Runtime

Profiles for Loading data to SAP NetWeaver BW or SAP BW 3.5

You need to determine the authorization profiles for loading data to SAP NetWeaver BW or SAP BW 3.5.

Profiles for Development Users

The following table shows the authorization profile configuration for development users:

Integration Feature	Description	Class	Field Values
S_RFC	Authorization Check for RFC Objects	Cross Application Authorization Objects	Activity: View, Maintain
S_RS_ADMWB	Administrative Workbench Objects	Business Information Warehouse	Activity: View, Maintain
S_DEVELOP	ABAP Workbench	Basis - Development environment	Activity: View, Maintain
S_RS_HIER	Administrative Workbench Hierarchy	Business Information Warehouse	Activity: View, Maintain

Integration Feature	Description	Class	Field Values
S_RS_ICUBE	Administrator Workbench - InfoCube Object	Business Information Warehouse	Activity: View, Maintain
S_RS_IOBC	Administrator Workbench - InfoObject Catalog	Business Information Warehouse	Activity: View, Maintain
S_RS_IOBJ	Administrator Workbench - InfoObject	Business Information Warehouse	Activity: View, Maintain
S_RS_IOMAD	Administrator Workbench - Maintain Master data	Business Information Warehouse	Activity: View, Maintain
S_RS_ISOUR	Administrator Workbench - InfoSource (Flexible update)	Business Information Warehouse	Activity: View, Maintain
S_RS_ISRCM	Administrator Workbench - InfoSource (Direct update)	Business Information Warehouse	Activity: View, Maintain
S_RS_OSDO	Administrator Workbench - ODS Object	Business Information Warehouse	Activity: View, Maintain
RSPC (TRANSACTION)	Transaction for maintain, execute process chain	n/a	Activity: Maintain, Execute

Profiles for Production Users

The following table shows the authorization profile configuration for development users:

Integration Feature	Description	Class	Field Values
S_RS_ADMWB	Administrative Workbench Objects	Business Information Warehouse	Activity: View, Maintain
S_RS_IOBC	Administrator Workbench - InfoObject Catalog	Business Information Warehouse	Activity: View, Maintain
S_RS_IOBJ	Administrator Workbench - InfoObject	Business Information Warehouse	Activity: View, Maintain
S_RS_IOMAD	Administrator Workbench - Maintain Master data	Business Information Warehouse	Activity: View, Maintain
S_RS_ISRCM	Administrator Workbench - InfoSource (Direct update)	Business Information Warehouse	Activity: View, Maintain
RSPC (TRANSACTION)	Transaction for maintain, execute process chain	n/a	Activity: Maintain, Execute

Defining PowerCenter as a Logical System in SAP BW

To import InfoSources and load data into SAP BW or to extract data from SAP BW, define PowerCenter as an external logical system in the SAP BW system.

Create a logical system in SAP BW for PowerCenter. If you have multiple SAP BW Services in the PowerCenter installation, each SAP BW Service must use the same connection for the RFC server program in the `sapnwrfc.ini` file.

If the SAP BW system is a Unicode system and the PowerCenter Integration Service and SAP BW Service run on AIX (64-bit), Linux (32-bit), Solaris (64-bit), or Windows, you must configure the logical system to communicate in Unicode mode. SAP provides Unicode RFC libraries for these operating systems. When the PowerCenter Integration Service and SAP BW Service run on these operating systems, they use the Unicode RFC libraries to process Unicode data.

To create a logical system:

1. Log on to the SAP BW system using SAP Logon.
2. Go to transaction RSA1 to open the Administrator Workbench.
3. Right-click the Source Systems folder and select Create.
4. From the Select Source System Type dialog box, select the following option:

   ```
   External System (Data and Metadata Transfer Using Staging BAPIs)
   ```

5. Click Enter.
6. In the Create Source System dialog box, enter the following information and click Enter:

Parameter	Description
Logical System Name	Name of the logical system. For example, you can enter LSPowerCenterBW.
Source System Name	Description of the source system.

7. On the RFC Destination screen, click the Technical settings tab.
8. Under Registered Server Program, enter the Program ID.

   SAP BW uses the Program ID to communicate with the SAP BW Service. The Program ID that you enter here must match the PROGRAM_ID in the connection for the RFC server program of the `sapnwrfc.ini` file defined for the SAP BW Service.
9. If the SAP BW system is a Unicode system and the PowerCenter Integration Service and SAP BW Service run on AIX (64-bit), Linux (32-bit), Solaris (64-bit), or Windows, click the Special Options tab, and then select the Unicode option under Character Width in Target System.
10. Click Save and return to the Administrator Workbench.

Configuring sapnwrfc.ini

SAP uses the communications protocol, Remote Function Call (RFC), to communicate with other systems. SAP stores RFC-specific parameters and connection information in a file named `sapnwrfc.ini`.

Configure the `sapnwrfc.ini` file on the PowerCenter Client and SAP BW Service with parameters that enable communication with the SAP BW system. This file enables PowerCenter to connect to the SAP BW system as an RFC client.

Note: The PowerCenter Integration Service does not use the `sapnwrfc.ini` file. When the SAP BW server requests data from PowerCenter, the SAP BW Service records the host name and system number of the SAP BW server. The SAP BW Service passes this information to the PowerCenter Integration Service. The PowerCenter Integration Service uses this information to connect and load data to the same SAP BW server.

sapnwrfc.ini Connection Types

Use the `sapnwrfc.ini` file to configure the connections that you want to use.

You can configure the following types of connections in the `sapnwrfc.ini` file for PowerExchange for SAP NetWeaver BW to connect to SAP BW:

Connection to a specific SAP application server

Create this connection for the PowerCenter Client to connect to the SAP BW server.

The PowerCenter Client uses this connection to import transfer structures from SAP BW. Use the DEST entry in the Import BW Transfer Structure dialog box.

Connection to use SAP load balancing

Create this connection to enable SAP to create an RFC connection to the application server with the least load at run time. Use this connection when you want to use SAP load balancing.

The PowerCenter Client uses this connection to import transfer structures from SAP BW. Use the DEST entry in the Import BW Transfer Structure dialog box.

Connection to an RFC server program registered at an SAP gateway

Create this connection to specify the logical system created for PowerCenter in the SAP BW system.

The SAP BW Service uses this connection to register as the RFC server and receive requests to run sessions. Use the DEST entry when you create the SAP BW Service.

Sample sapnwrfc.ini File

The following sample shows a connection entry for a specific SAP application server in the `sapnwrfc.ini` file:

```
DEST=sapr3
ASHOST=sapr3
SYSNR=00
```

The following sample shows a connection entry for SAP load balancing in the `sapnwrfc.ini` file:

```
DEST=sapr3
R3NAME=ABV
MSHOST=infamessageserver.informatica.com
GROUP=INFADEV
```

The following sample shows a connection entry for an RFC server program registered at an SAP gateway in the `sapnwrfc.ini` file:

```
DEST=sapr346CLSQA
PROGRAM_ID=PID_LSRECEIVE
GWHOST=sapr346c
GWSERV=sapgw00
```

Configuring a Connection in sapnwrfc.ini

The SAP BW Service uses the connection entries defined in the `sapnwrfc.ini` file to connect to an RFC server program registered at an SAP gateway. You register the SAP BW Service with SAP BW at the SAP gateway. The SAP BW Service is an RFC server and acts as a listener to receive requests from SAP BW to run a PowerCenter workflow.

The PowerCenter Client uses the connection entries defined in the `sapnwrfc.ini` file to connect to a specific SAP application server or to use SAP load balancing. The Designer connects to SAP BW to import metadata into the repository.

Important: Use a DOS editor or Wordpad to configure the `sapnwrfc.ini` file. Notepad can corrupt the file.

To configure an entry in the `sapnwrfc.ini` file:

1. Open the `sapnwrfc.ini` file.
2. Enter the following connection parameters based on the type of connection that you want to create:

sapnwrfc.ini Parameter	Description
DEST	Destination in RFCAccept. Use this parameter for the following types of connections: - Connection to a specific SAP application server - Connection to use load balancing - Connection to an RFC server program registered at an SAP gateway Use this parameter as the connect string when you import an InfoSource in the Target Designer and when you configure database connections in the Workflow Manager. You can use up to 32 characters.
ASHOST	Host name or IP address of the SAP BW application. Use this parameter to create a connection to a specific SAP application server.
SYSNR	SAP system number. Use this parameter to create a connection to a specific SAP application server.
R3NAME	Name of the SAP system. Use this parameter to create a connection to use SAP load balancing.
MSHOST	Host name of the SAP Message Server. Use this parameter to create a connection to use SAP load balancing.
GROUP	Group name of the SAP application server. Use this parameter to create a connection to use SAP load balancing.
PROGRAM_ID	Program ID for the logical system you create in SAP BW for the SAP BW Service. The Program ID in SAP BW must match this parameter, including case. For more information about the Program ID parameter, see "Defining PowerCenter as a Logical System in SAP BW" on page 47. Use this parameter to create a connection to an RFC server program registered at an SAP gateway.

sapnwrfc.ini Parameter	Description
GWHOST	Host name of the SAP gateway. Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
GWSERV	Server name of the SAP gateway. Use this parameter to create a connection to an RFC server program registered at an SAP gateway.
TRACE	Debugs RFC connection-related problems. Use this parameter for the following types of connections: - Connection to a specific SAP application server - Connection to use load balancing - Connection to an RFC server program registered at an SAP gateway Set one of the following values based on the level of detail that you want in the trace: - 0. Off. - 1. Brief. - 2. Verbose. - 3. Full.

3. If you are connecting to multiple SAP BW systems, configure multiple connections in the `sapnwrfc.ini` file with unique DEST parameters.

Creating the SAP BW Service

The SAP BW Service is an application service that performs the following tasks:

- Listens for RFC requests from SAP BW
- Initiates workflows to extract from or load to SAP BW
- Sends log events to the PowerCenter Log Manager

Use the Administrator tool to manage the SAP BW Service.

Load Balancing for the SAP BW System and the SAP BW Service

You can configure the SAP BW system to use load balancing. To support a SAP BW system configured for load balancing, the SAP BW Service records the host name and system number of the SAP BW server requesting data from PowerCenter. It passes this information to the PowerCenter Integration Service. The PowerCenter Integration Service uses this information to load data to the same SAP BW server that made the request.

You can also configure the SAP BW Service in PowerCenter to use load balancing. If the load on the SAP BW Service becomes too high, you can create multiple instances of the SAP BW Service to balance the load. To run multiple SAP BW Services configured for load balancing, create each service with a unique name but use the same values for all other parameters. The services can run on the same node or on different nodes. The SAP BW server distributes data to the multiple SAP BW Services in a round-robin fashion.

Steps to Create the SAP BW Service

Perform the following steps to create the SAP BW Service.

To create the SAP BW Service:

1. In the Informatica Administrator, click Create > SAP BW Service.

   The Create New SAP BW Service window appears.

2. Configure the SAP BW Service options.

   The following table describes the information to enter in the Create New SAP BW Service window:

Property	Description
Name	Name of the SAP BW Service. The name is not case sensitive and must be unique within the domain. The characters must be compatible with the code page of the associated repository. The name cannot have leading or trailing spaces, include carriage returns or tabs, exceed 79 characters, or contain the following characters: \ / * ? < > " |
Description	Description of the SAP BW Service.
Location	Name of the domain and folder in which the SAP BW Service is created. The Administrator tool creates the SAP BW Service in the domain where you are connected. Click Select Folder to select a new folder in the domain.
License	License Key for SAP BW.
Node	Node on which this service runs.
SAP Destination R Type	DEST entry defined in the `sapnwrfc.ini` file to connect to the SAP BW Service.
Associated PowerCenter Integration Service	PowerCenter Integration Service associated with the SAP BW Service.
Repository User Name	Account used to access the repository.
Repository Password	Password for the repository user. **Note:** If secure communication is enabled for the domain, you need not specify the repository password.
Security Domain	Security domain for the user. Appears when the Informatica domain contains an LDAP security domain.

3. Click OK.

   A message informs you that the SAP BW Service was successfully created.

4. Click Close.

   The SAP BW Service properties window appears.

5. Click Enable to start the SAP BW Service.

   Note: Before you enable the SAP BW Service, you must define PowerCenter as a logical system in SAP BW.

Importing the ABAP Program into SAP BW

The Informatica installer file includes an ABAP program that you must import into the SAP BW system. The ABAP program sends status information to the SAP BW Service. The SAP BW Service sends these as log events to the PowerCenter Log Manager.

To import the ABAP program:

1. From the SAP BW window, enter the SE38 transaction.
2. Enter ZPMSENDSTATUS for the program name and select create.
3. Enter a Title.
4. Select Executable Program as the Type.
5. Select Basis as the Application.
6. Select Start Using Variant.
7. Click Save.
8. Click Local Object.

   The ABAP Editor window appears.
9. Click Utilities > More Utilities > Upload/Download > Upload.
10. Unzip the Informatica installer `.zip` file and navigate to the zpmsendstatus.ab4 file located in the following directory:

    ```
    <Informatica Installer File>\saptrans\BW
    ```
11. Save the program.
12. Activate the program.

Troubleshooting PowerExchange for SAP NetWeaver BW Installation and Configuration

No metadata appears when I try to import from SAP BW.

If the profile for a user name does not have sufficient privileges, you cannot import SAP BW metadata into the PowerCenter repository. Instead, the **Transaction Transfer List** and **Master Transfer List** folders in the **Import from SAP BW** dialog box will be empty.

Import metadata from SAP BW using an SAP BW user name with a profile that lets you access metadata.

CHAPTER 4

Installing and Configuring Transports

This chapter includes the following topics:

- Transports Installation and Configuration, 53
- Transports Installation, 61
- Transports Installation Order, 61
- Transports List based on SAP Functionality, 62
- Secure Transports for SAP Unicode Versions 5.0 and Later, 63
- Standard Transports for SAP Unicode Versions 5.0 and Later, 65
- Secure Transports for SAP non-Unicode Versions 5.0 and Later, 67
- Standard Transports for SAP non-Unicode Versions 5.0 and Later, 69

Transports Installation and Configuration

The transport system is a set of ABAP programs installed on the SAP system. The ABAP programs import SAP metadata into the repository. The ABAP programs also enable run-time functionalities, such as passing mapping parameters and filters.

You use the transport system in the following situations:

Configuring PowerExchange for SAP NetWeaver

You need to transport customized objects that were provided by Informatica to the SAP system. These objects include tables, structures, programs, and functions. The PowerCenter Integration Service calls custom objects when it makes a request to the SAP system.

Deploying run-time transports and ABAP programs from development to production

When you use the deprecated SAP data object, you use ABAP to integrate with mySAP applications. You must deploy the run-time transports provided by Informatica and the ABAP programs installed by the PowerCenter Integration Service to extract data when you move from development to production.

The SAP system administrator must perform the following steps to integrate PowerCenter on the development, test, and production SAP systems:

1. Delete transport programs from previous versions.
2. Transport objects to the SAP system.

3. Run transport programs that generate unique IDs.
4. Create users in the SAP system for PowerCenter users.
5. Create profiles in the SAP system for PowerCenter users.
6. Create a package for the ABAP programs that the PowerCenter Integration Service installs on the SAP system. Create the package in the development environment only.

Step 1. Delete Transport Programs

When you integrate Informatica on the SAP system for the first time, you do not have to delete the transport programs. When you integrate Informatica on the SAP system that was configured for a previous version of Informatica, you need to delete the old transport programs from the SAP system. Note the current configuration in the /INFATRAN/ZPRGSQ SAP ECC custom table before you delete the transport objects.

Modifying /INFATRAN/

To delete a transport object, register the namespace /INFATRAN/ and enter the repair license. Also, change the status of /INFATRAN/ in the SAP system to Modifiable.

1. Go to transaction SE03 and double-click **Display/Change Namespaces**.

   The SAP system displays the list of namespaces.
2. Right-click /INFATRAN/ and click Display.
3. Make the following changes to the namespace:

Field	Description
Namespace	Unique name to identify the transport programs.
Namespace Role	Represents a namespace that you import into the SAP system. You cannot develop this namespace. However, you can repair the namespace if you have a valid Repair License.
Repair License	Unique license key required to modify or repair transport objects. Enter the license key 10357544012122787918 to delete or modify a namespace.
Short Text	Description of the namespace.

4. Click Save.
5. Go to transaction SE03 and double-click **Set System Change Option**.

   The System Change Option screen appears.
6. Change the Global Setting to Modifiable and click Save.

Deleting Transport Objects

PowerExchange for SAP NetWeaver contains the following packages:

- /INFATRAN/ZINFA_DESIGNTIME
- INFATRAN/ZINFA_RUNTIME
- ZINFABC_RUNTIME

Delete all packages. Before you delete a package, you must delete almost all the objects under it.

Important: If you use the deprecated SAP data object and upgrade from Informatica version 9.0.1 and later, do not delete the TBL_DESIGN_PROGINFO transport object that contains the table /INFATRAN/ZPRGIN. The table holds the details of all the ABAP programs installed from PowerExchange for SAP NetWeaver.

1. Go to transaction SE10 and verify whether there is any lock on the objects under the package that you want to delete.

   An object is locked when another user is modifying or transporting the object. You must check the list of modifiable requests for all users in transaction SE10 and verify if there is any request associated with an Informatica object.
2. Release all the modifiable requests associated with the Informatica object.
3. Go to transaction SE10 and create a workbench for deleting all objects.
4. Go to transaction SE80, select the package that you want to delete and click display.

   When you select a package, it displays all objects under it, such as function groups, programs, transactions, and dictionary objects. Dictionary objects include tables and structures.

   For example, select the package ZINFA_DESIGNTIME. When you select a package, it displays all objects under it, such as function groups, programs, transactions, and dictionary objects. Dictionary objects include tables and structures.

   Note: The ZINFA_DESIGNTIME package is needed only when you use the deprecated SAP data object.
5. Select a function group to view its function modules.
6. Right-click each function module and click Delete.
7. Right-click the function group and click Delete.

   When you delete a function group, you delete includes and other SAP standard dictionary objects.
8. Right-click each program and click Delete.
9. When prompted, select Includes and click OK.

   You must delete each program and includes.
10. Right-click each table and click Delete.

    If the tables contain data, delete the data before you delete the tables.
11. Click OK if prompted with the message that the table is used in the program.
12. Right-click each structure and click Delete.
13. Click OK if prompted with the message that structure is used in the program.
14. Go to transaction SE10 and select the transport request created for deleting objects.
15. Expand the request node and verify the list of objects.

    The list of objects in the request node and the list of objects that you delete from the package must match.
16. Go to transaction SE10, right-click the transport request for deleting the objects, and select Release Directly.

    Wait until the export of the change request is complete. Complete the export before you delete the package.
17. Go to transaction SE80, right-click the package, and click Delete.
18. When prompted, create a new local transport request to delete the package.
19. Go to transaction SE10 and delete the package.
20. Release the transport request that you created for the deleting package.

Step 2. Install Transport Objects

Informatica provides a group of design-time and run-time transports. Transports are customized objects necessary for SAP integration. These objects include tables, programs, structures, and functions that Informatica exports to data files. Import these transports on the SAP system. This process creates a package for each group of transports.

The transports you install depend on the version of the SAP system and whether or not the system is Unicode. The Informatica installer zip file contains the following transport directories:

saptrans/mySAP/UC

Contains transports for a Unicode SAP system. These transports are created from SAP ECC version 5.0.

saptrans/mySAP/NUC

Contains transports for an SAP system that is not Unicode.

Both of these directories contain separate directories for the data files and cofiles that you need to import on the SAP system. The data files contain the transport objects. The cofiles contain the transport conditions. Each set of program files represents a function group that has a specific purpose.

The SAP system administrator can import the transports using Transport Management System (STMS).

The Informatica installer zip file has separate directories for the data files and cofiles that you need to import on the SAP system. The data files contain the transport objects. The cofiles contain the transport conditions. Each set of program files represents a function group that has a specific purpose.

Import transports on the SAP system in the following order:

1. Import the ZINFABC run-time transport on the development system.
2. Import the run-time transports on the development system.
3. Import the design-time transports on the development system. The design-time transports you import on the development system depend on the PowerExchange for SAP NetWeaver features you want to use.
4. After you import the transports on the development system, import the run-time transports to the test and production systems.
5. To import the transports on SAP using the Transport Management System, go to transaction STMS.
6. Click Overview > Imports.
7. Open the target system queue.
8. Click Extras > Other Requests > Add.

   The Add Transport Request to Import Queue dialog box appears.
9. Add a transport request number.

   When you add a transport request number, delete the prefix. For example, when you add ZINFABC_RUN_R900101.R46, delete ZINFABC_RUN. Import the ZINFABC run-time transport first.
10. Click Enter.
11. From Request, select the transport request number you added, and click Import.
12. Repeat steps from 6 through 8 for each transport you want to add.

Step 3. Run Transport Programs

After you transport the integration objects, run the following programs:

/INFATRAN/YPMPARSQ

Part of package /INFATRAN/ZINFA_RUNTIME. This program generates unique parameter IDs. Run this program on the development, test, and production systems.

/INFATRAN/YPMPRGSQ

Part of package /INFATRAN/ZINFA_DESIGNTIME. Run this program on the development system only. Run this program to specify an ABAP program name prefix of up to 10 characters, provide a namespace that you have registered with SAP, and determine the starting sequence number. When you upgrade from a previous version, you run this program to use the same starting sequence number.

The ABAP program name prefix must start with the letter “Y” or “Z.” Use a unique prefix for each SAP system that you transport these objects to. For example, use YPC000001 as the prefix and current sequence for one SAP system and ZPM000001 for another SAP system.

When you run the /INFATRAN/YMPRGSQ program, you must select the option to use long names because both file mode sessions and stream mode sessions use long names. When you use long names, you generate a program name that is 30 characters in length, including the customer namespace.

When you run the /INFATRAN/YMPRGSQ program, you can also select the option to override the existing information in the /INFATRAN/ZPRGSQ custom table. For example, if you want to add a custom namespace, provide the related information and select the **Override existing information** option.

Run /INFATRAN/YPMPRGSQ again. When you run the program, set the initial sequence number to <last number before upgrade> + 1.

Note: PowerCenter version 9 and later transports are fully backward compatible with PowerCenter 8.0 and earlier, 7.0 and earlier, and 6.2.2 versions. However, note that effective in version 9.6.0, Informatica dropped support for deprecated BAPI mappings created in versions earlier than 8.5 and deprecated IDoc mappings created in versions earlier than 7.1. If you upgrade the deprecated mappings to version 9.6.0, the sessions will fail. The transports for the deprecated BAPI and IDoc mappings are not backward compatible.

Deploying Run-time Packages to the Test and Production Systems

After you install the transports on the SAP development system, deploy the run-time packages to the test and production systems. Before deploying the run-time packages, use the SAP transaction SE10 to verify that no existing transport requests include the run-time packages.

1. In the SAP development system, go to transaction SE80.

   The **Object Navigator** window appears.
2. Display the ZINFABC_RUNTIME package.
3. Right-click the package name and select **Write Transport Request**.

   The **Transport Package** dialog box appears.
4. Click **All Objects**.

   The **Enter Transport Request** dialog box appears.
5. Click **Create Request**.

   The **Select Request Type** dialog box appears.
6. Click **Transport of Copies** and then click **Enter**.

   The **Create Request** dialog box appears.
7. Enter a short description and click **Save**.
8. Go to transaction SE10.

   The **Transport Organizer** window appears.

9. For Request Type, select **Transport** of Copies.
10. For Request Status, select **Modifiable**.
11. Click **Display**.

    The **Transport Organizer: Requests** window appears.
12. Double-click the transport request you created.

    The **Display Request** dialog box appears.
13. On the **Properties tab**, select the target SAP system to which you want to deploy the package and click **Enter**.
14. Select the transport request you created and click **Release Directly**.

    SAP deploys the package to the target system.
15. Repeat steps 1 through 14 to deploy the /INFATRAN/ZINFA_RUNTIME package.

Step 4. Create Users

Create an appropriate user for development, test, and production environments in SAP. The user you create enables dialog-free communication between SAP and Informatica.

Depending on the version of the SAP installation, create a Common Program Interface-Communications (CPI-C) user, System user, or a communication user with the appropriate authorization profile.

Tip: Ensure that the Informatica user that you create in SAP and the user that completes the task in SAP have the same permissions.

Step 5. Create Profiles

The SAP administrator needs to create a profile in the development, test, and production SAP system so that you can use the integration features. This profile name must include authorization for the objects and related activities. The profile on the test system should be the same as the profile on the production system.

The following table shows the authorization necessary for integration:

Integration Feature	Production/ Development	Authorization Object	Activity
Install and uninstall programs	Development	S_DEVELOP	All activities. Also need to set Development ObjectID to PROG.
Install and uninstall programs	Development	S_TRANSPRT **Note:** You need authorization on the S_TRANSPRT object only when you use secure transports.	CREATE and CHANGE. Required for workbench requests TTYPE = DTRA.
Extract and load data	Production	S_TABU_DIS	READ and WRITE.
Run file mode sessions	Production	S_DATASET	WRITE. **Note:** You also need the READ activity authorization when you use secure transports.

Integration Feature	Production/ Development	Authorization Object	Activity
Submit background job	Production	S_PROGRAM	BTCSUBMIT, SUBMIT.
Release background job	Production	S_BTCH_JOB	DELE, LIST, PLAN, SHOW. Set Job Operation to RELE.
Run stream mode sessions	Production	S_CPIC	All activities.
Authorize RFC privileges	Production Development	S_RFC	All activities. Authorize RFC privileges for function group RFC objects of the following function groups: - ZPMV - ZERP - ZPMH - ZPMR - ZPMP - ZPMD - ZPMI - ZPMF - SYST - RFC1 - SDTX - /INFATRAN/* - SYSU Include the following function groups if you use secure transports: - SDIFRUNTIME - GSAC - SKEY - STR9 - SEU_COMPONENT - STRD - BTCH - SABC - KXUT - /1BCDWBEN/SEN4 - EDI1 Include the following function groups for BAPI sessions: - Function group for the BAPI - Function group containing ABAP4_COMMIT_WORK, BAPI_TRANSACTION_COMMIT, and ABAP4_ROLLBACK_WORK Include the following function groups for IDoc write sessions: - ARFC, ERFC, EDIN
IDoc authorization	Production Development	S_IDOCDEFT	READ. Transaction code: WE30.

Integration Feature	Production/ Development	Authorization Object	Activity
ALE authorization	Production	B_ALE_RECV	Message type of the IDoc that needs to be written to the SAP system.
Load data into cross client tables	Production	S_TABU_CLI	WRITE data into cross client tables.

Step 6. Create a Package

Create a package to integrate PowerExchange for SAP NetWeaver with mySAP applications using ABAP. When you create a mapping with an SAP source definition in the development system, you generate and install an ABAP program. By default, the ABAP programs that you generate from the mapping are installed in the $TMP package. For easy transport into a test or production system, the SAP administrator needs to create a package for the ABAP programs. You cannot transport items from the $TMP package.

1. Go to transaction SE80.

   The Object Navigator window appears.
2. From the list, select Package.
3. Enter a name for the new package and press Enter.

   SAP prompts you to create a new package.
4. Click OK.
5. Enter the following information and click the Save button.

Field	Description
Package	Name of the package.
Short text	Description of the package.
Software component	Name of the software component.
Appl. component	Name of the application component.

   SAP prompts you to select a workbench request.
6. Select a workbench request. Or, create a new workbench request.
7. Click Save.
8. Go to transaction SE10.

   You can view a list of workbench requests in transaction SE10.
9. Right-click the new workbench request and select Release.

Transports Installation

The transports that you need to install depend on the SAP functionality that you want to use. The transports also change depending on whether the SAP system is a Unicode system or a non-Unicode system, and also depending on whether you want to use standard transports or secure transports.

Secure Transports for Data Integration with ABAP

You can install secure transports to enforce security authorizations when you use ABAP to read data from SAP.

When you use secure transports to perform data integration tasks with ABAP, the PowerCenter Integration Service verifies that the following conditions are met:

- The SAP user who generates and installs the ABAP program has the developer license.
- The user who generates and installs the ABAP program is authorized for the following function groups:
  - SKEY
  - STR9
- Users cannot generate and install ABAP programs from the Designer tool if the SAP client is locked down for changes.
- A new workbench request is created to track object changes in SAP when users install or delete ABAP programs.
- The ABAP program can be installed only on development systems and not on production systems.

The following transports contain security features that enforce security authorizations when you use ABAP to read data from SAP:

- TBL run-time transport
- HIER_R3 run-time transport
- TBL design-time transport
- HIER_R3 design-time transport

Note: To enable security features, install these secure transports. Do not install the corresponding standard transports for TBL run time, HIER_R3 run time, TBL design time and HIER_R3 design time.

Transports Installation Order

If you use multiple methods to integrate with SAP, install the transports on the SAP development system in the following order:

1. ZINFABC run-time transport.
2. TBL run-time transport.

   Note: When you install the TBL run-time transport, you can choose to install standard transports or secure transports.
3. HIER_R3 run-time transport, if you use the ABAP integration method to extract hierarchies from the SAP system.

   Note: When you install the HIER_R3 run-time transport, you can choose to install standard transports or secure transports.

4. TRANS_VER run-time transport.
5. BCI run-time transport.
6. TBL_READ_RUN_CMP transport, if you want to enable compressed data transfer while reading data from SAP through ABAP.

   Note: Install the TBL_READ_RUN_CMP transport only for SAP ECC versions 5.0 or later.
7. TBL_DESIGN_PROGINFO design time transport.

   Note: Install the TBL_DESIGN_PROGINFO transport only for Unicode SAP systems. This transport contains the package /INFATRAN/ZINFA_DESIGNPROGINFO and the table /INFATRAN/ZPRGIN. Ensure that when you when you upgrade from Informatica version 9.0.1 and later, you do not delete or override the package /INFATRAN/ZINFA_DESIGNPROGINFO. The table /INFATRAN/ZPRGIN holds the details of all the ABAP programs that are installed from PowerExchange for SAP.
8. TBL design-time transport.

   Note: When you install the TBL design-time transport, you can choose to install standard transports or secure transports.
9. Remaining design-time transports in any order, depending on the PowerExchange for SAP NetWeaver features that you want to use.

   Note: When you install the HIER_R3 design-time transport, you can choose to install standard transports or secure transports.

When you install the transports on the development system, the transport process creates the following packages for each group of transports:

- **ZINFABC_RUNTIME.** Contains the ZINFABC run-time transport.
- **/INFATRAN/ZINFA_RUNTIME.** Contains the remaining run-time transports that you install.
- **/INFATRAN/ZINFA_DESIGNPROGINFO.** Contains the design-time transports required for storing the installed ABAP program information.
- **/INFATRAN/ZINFA_DESIGNTIME.** Contains the design-time transports that you install.

After you install the transports on the development system, deploy the run-time packages to the test and production systems.

Note: Some SAP systems refer to “package” as “development class.”

Before performing the transport, complete the following tasks:

1. Examine the transports for the SAP version table that corresponds to your SAP version.
2. Delete the prefix for each data file and cofile. For example, remove ZINFABC_RUN_ from the ZINFABC_RUN_R900183.R46 cofile.
3. Change the permission of the cofiles from read-only to write.

Transports List based on SAP Functionality

Apart from few generic transports, the transports that you need to install depend on the SAP functionality that you want to use.

The generic transports that you need to install for all SAP functionality are as follows:

- ZINFABC_RUN

- TRANS_VER_RUN

The following table lists the transports that you need to install based on the SAP functionality that you want to use:

SAP Functionality	Design-time Transports	Run-time Transports
Table extraction by using ABAP	- TBL_DESIGN	- TBL_READ_RUN
Hierarchy extraction by using ABAP	- HIER_R3_DESIGN	- HIER_R3_RUN
Compressed data transfer by using ABAP	Not applicable	- TBL_READ_RUN_CMP
Business content integration	Not applicable	- BCI_RUN
Data integration by using BAPI/RFC functions	- BAPI_DESIGN	Not applicable
Data migration integration	- DMI_DESIGN	Not applicable
IDoc integration by using ALE	- IDOC_DESIGN	Not applicable

Secure Transports for SAP Unicode Versions 5.0 and Later

If you want to install secure transports on SAP Unicode versions 5.0 and later, install the transports on the SAP development system in the same order as listed in the table.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - ZINFABC_RUN_R900245.U47 - ZINFABC_RUN_K900245.U47	Required	**Standard transports** - U47K900245	n/a	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Support for a namespace prefix in ABAP program names and for ABAP programs generated in a previous version.
Secure transports - TBL_READ_RUN_R900328.DV2 - TBL_READ_RUN_K900328.DV2	Required	**Secure transports** - DV2K900328	ZPMD ZPMP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - TBL_WRITE_RUN_R900036.DV2 - TBL_WRITE_RUN_K900036.DV2	Optional	**Standard transports** - DV2K900036	ZPMW	For SAP ECC and SAP add-on components, including CRM and SAP BW: Supports table load for data integration.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Secure transports - HIER_R3_RUN_R900236.DV2 - HIER_R3_RUN_K900236.DV2	Optional	**Secure transports** - DV2K900236	ZPMR	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - TRANS_VER_RUN_R900352.DV2 - TRANS_VER_RUN_K900352.DV2	Required	**Standard transports** - DV2K900352	ZPMV	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Checks versions for PowerExchange objects.
Standard transports - BCI_RUN_R900291.DV2 - BCI_RUN_K900291.DV2	Optional	**Standard transports** - DV2K900291	ZBCI	For SAP ECC and SAP BW: Supports business content integration.
Standard transports - TBL_READ_RUN_CMP_R900448.EC5 - TBL_READ_RUN_CMP_K900448.EC5	Optional	**Standard transports** - EC5K900448	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Enables compressed data transfer when you read data from SAP by using the ABAP program. **Note:** Install the TBL_READ_RUN_CMP transport only for SAP ECC versions 5.0 or later.
Standard transports - TBL_DESIGN_PROGINFO_R900051.UC5 - TBL_DESIGN_PROGINFO_K900051.UC5	Required	**Standard transports** - UC5K900051	n/a	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Holds the information about the installed ABAP programs.
Secure transports - TBL_DESIGN_R900289.DV2 - TBL_DESIGN_K900289.DV2	Required	**Secure transports** - DV2K900289	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - BAPI_DESIGN_R900246.U47 - BAPI_DESIGN_K900246.U47	Optional	**Standard transports** - U47K900246	ZPMF	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports data integration by using BAPI/RFC functions.
Standard transports - DMI_DESIGN_R900237.U47 - DMI_DESIGN_K900237.U47	Optional	**Standard transports** - U47K900237	ZDMI	For SAP ECC only: Supports data migration integration.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Secure transports - HIER_R3_DESIGN_R900264.DV2 - HIER_R3_DESIGN_K900264.DV2	Optional	**Secure transports** - DV2K900264	ZPMH	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - IDOC_DESIGN_R900239.U47 - IDOC_DESIGN_K900239.U47	Optional	**Standard transports** - U47K900239	ZPMI	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports IDoc integration by using ALE.
Standard transports - ISU_DESIGN_R900041.CCS - ISU_DESIGN_K900041.CCS	Optional	**Standard transports** - CCSK900041	ZISU	For SAP IS-U migration.

Standard Transports for SAP Unicode Versions 5.0 and Later

If you want to install standard transports on SAP Unicode versions 5.0 and later, install the transports on the SAP development system in the same order as listed in the table.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - ZINFABC_RUN_R900245.U47 - ZINFABC_RUN_K900245.U47	Required	**Standard transports** - U47K900245	n/a	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Support for a namespace prefix in ABAP program names and for ABAP programs generated in a previous version.
Standard transports - TBL_READ_RUN_R900318.DV2 - TBL_READ_RUN_K900318.DV2	Required	**Standard transports** - DV2K900318	ZPMD ZPMP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - TBL_WRITE_RUN_R900036.DV2 - TBL_WRITE_RUN_K900036.DV2	Optional	**Standard transports** - DV2K900036	ZPMW	For SAP ECC and SAP add-on components, including CRM and SAP BW: Supports table load for data integration.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - HIER_R3_RUN_R900241.U47 - HIER_R3_RUN_K900241.U47	Optional	**Standard transports** - U47K900241	ZPMR	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - TRANS_VER_RUN_R900352.DV2 - TRANS_VER_RUN_K900352.DV2	Required	**Standard transports** - DV2K900352	ZPMV	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Checks versions for PowerExchange objects.
Standard transports - BCI_RUN_R900291.DV2 - BCI_RUN_K900291.DV2	Optional	**Standard transports** - DV2K900291	ZBCI	For SAP ECC and SAP BW: Supports business content integration.
Standard transports - TBL_READ_RUN_CMP_R900448.EC5 - TBL_READ_RUN_CMP_K900448.EC5	Optional	**Standard transports** - EC5K900448	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Enables compressed data transfer when you read data from SAP by using the ABAP program. **Note:** Install the TBL_READ_RUN_CMP transport only for SAP ECC versions 5.0 or later.
Standard transports - TBL_DESIGN_PROGINFO_R900051.UC5 - TBL_DESIGN_PROGINFO_K900051.UC5	Required	**Standard transports** - UC5K900051	n/a	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Holds the information about the installed ABAP programs.
Standard transports - TBL_DESIGN_R900314.DV2 - TBL_DESIGN_K900314.DV2	Required	**Standard transports** - DV2K900314	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - BAPI_DESIGN_R900246.U47 - BAPI_DESIGN_K900246.U47	Optional	**Standard transports** - U47K900246	ZPMF	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports data integration by using BAPI/RFC functions.
Standard transports - DMI_DESIGN_R900237.U47 - DMI_DESIGN_K900237.U47	Optional	**Standard transports** - U47K900237	ZDMI	For SAP ECC only: Supports data migration integration.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - HIER_R3_DESIGN_R900238.U47 - HIER_R3_DESIGN_K900238.U47	Optional	**Standard transports** - U47K900238	ZPMH	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - IDOC_DESIGN_R900239.U47 - IDOC_DESIGN_K900239.U47	Optional	**Standard transports** - U47K900239	ZPMI	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports IDoc integration by using ALE.
Standard transports - ISU_DESIGN_R900041.CCS - ISU_DESIGN_K900041.CCS	Optional	**Standard transports** - CCSK900041	ZISU	For SAP IS-U migration.

Secure Transports for SAP non-Unicode Versions 5.0 and Later

If you want to install secure transports on SAP non-Unicode versions 5.0 and later, install the transports on the SAP development system in the same order as listed in the table.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - ZINFABC_RUN_R900183.R46 - ZINFABC_RUN_K900183.R46	Required	**Standard transports** - R46K900183	n/a	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Support for a namespace prefix in ABAP program names and for ABAP programs generated in a previous version.
Secure transports - TBL_READ_RUN_R900022.D47 - TBL_READ_RUN_K900022.D47	Required	**Secure transports** - D47K900022	ZPMD ZPMP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Secure transports - HIER_R3_RUN_R900307.R46 - HIER_R3_RUN_K900307.R46	Optional	**Secure transports** - R46K900307	ZPMR	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - TRANS_VER_RUN_R900037.D47 - TRANS_VER_RUN_K900037.D47	Required	**Standard transports** - D47K900037	ZPMV	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Checks versions for PowerExchange objects.
Standard transports - BCI_RUN_R900321.R46 - BCI_RUN_K900321.R46	Optional	**Standard transports** - R46K900321	ZBCI	For SAP ECC and SAP BW: Supports business content integration.
Standard transports - TBL_READ_RUN_CMP_R904055.NE5 - TBL_READ_RUN_CMP_K904055.NE5	Optional	**Standard transports** - NE5K904055	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Enables compressed data transfer when you read data from SAP by using the ABAP program. **Note:** Install the TBL_READ_RUN_CMP transport only for SAP ECC versions 5.0 or later.
Secure transports - TBL_DESIGN_R900315.R46 - TBL_DESIGN_K900315.R46	Required	**Secure transports** - R46K900315	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - BAPI_DESIGN_R900173.R46 - BAPI_DESIGN_K900173.R46	Optional	**Standard transports** - R46K900173	ZPMF	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports data integration by using BAPI/RFC functions.
Standard transports - DMI_DESIGN_R900178.R46 - DMI_DESIGN_K900178.R46	Optional	**Standard transports** - R46K900178	ZDMI	For SAP ECC only: Supports data migration integration.
Secure transports - HIER_R3_DESIGN_R900034.D47 - HIER_R3_DESIGN_K900034.D47	Optional	**Secure transports** - D47K900034	ZPMH	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - IDOC_DESIGN_R900175.R46 - IDOC_DESIGN_K900175.R46	Optional	**Standard transports** - R46K900175	ZPMI	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports IDoc integration by using ALE.

Standard Transports for SAP non-Unicode Versions 5.0 and Later

If you want to install standard transports on SAP non-Unicode versions 5.0 and later, install the transports on the SAP development system in the same order as listed in the table.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - ZINFABC_RUN_R900183.R46 - ZINFABC_RUN_K900183.R46	Required	**Standard transports** - R46K900183	n/a	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Support for a namespace prefix in ABAP program names and for ABAP programs generated in a previous version.
Standard transports - TBL_READ_RUN_R900011.D47 - TBL_READ_RUN_K900011.D47	Required	**Standard transports** - D47K900011	ZPMD ZPMP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - HIER_R3_RUN_R900180.R46 - HIER_R3_RUN_K900180.R46	Optional	**Standard transports** - R46K900180	ZPMR	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - TRANS_VER_RUN_R900037.D47 - TRANS_VER_RUN_K900037.D47	Required	**Standard transports** - D47K900037	ZPMV	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Checks versions for PowerExchange objects.
Standard transports - BCI_RUN_R900321.R46 - BCI_RUN_K900321.R46	Optional	**Standard transports** - R46K900321	ZBCI	For SAP ECC and SAP BW: Supports business content integration.
Standard transports - TBL_READ_RUN_CMP_R904055.NE5 - TBL_READ_RUN_CMP_K904055.NE5	Optional	**Standard transports** - NE5K904055	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Enables compressed data transfer when you read data from SAP by using the ABAP program. **Note:** Install the TBL_READ_RUN_CMP transport only for SAP ECC versions 5.0 or later.

Data and Cofile Names	Required/ Optional	Transport Request	Function Group	Functionality
Standard transports - TBL_DESIGN_R900010.D47 - TBL_DESIGN_K900010.D47	Required	**Standard transports** - D47K900010	ZERP	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports table extraction for data integration by using the ABAP program.
Standard transports - BAPI_DESIGN_R900173.R46 - BAPI_DESIGN_K900173.R46	Optional	**Standard transports** - R46K900173	ZPMF	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports data integration by using BAPI/RFC functions.
Standard transports - DMI_DESIGN_R900178.R46 - DMI_DESIGN_K900178.R46	Optional	**Standard transports** - R46K900178	ZDMI	For SAP ECC only: Supports data migration integration.
Standard transports - HIER_R3_DESIGN_R900177.R46 - HIER_R3_DESIGN_K900177.R46	Optional	**Standard transports** - R46K900177	ZPMH	For SAP ECC only: Supports hierarchy extraction for data integration by using the ABAP program.
Standard transports - IDOC_DESIGN_R900175.R46 - IDOC_DESIGN_K900175.R46	Optional	**Standard transports** - R46K900175	ZPMI	For SAP ECC and mySAP add-on components, including CRM, and SAP BI: Supports IDoc integration by using ALE.

Part II: Data Integration Using ABAP

This part contains the following chapters:

- Importing SAP ECC Source Definitions, 72
- SAP ECC Target Definitions, 80
- Working with ABAP Mappings, 84
- Working with SAP Functions in ABAP Mappings, 96
- Application Source Qualifier for SAP ECC Sources, 103
- Configuring Sessions with SAP ECC Sources and Targets, 123

CHAPTER 5

Importing SAP ECC Source Definitions

This chapter includes the following topics:

- Importing SAP ECC Source Definitions Overview, 72
- Table and View Definitions, 73
- Hierarchy Definitions, 74
- IDoc Definitions , 76
- Importing a Source Definition , 77
- Organizing Definitions in the Navigator, 78
- Troubleshooting SAP ECC Source Definitions, 79

Importing SAP ECC Source Definitions Overview

When you import source definitions from SAP, the Designer uses RFC to connect to the SAP application server. The Designer calls functions in the SAP system to import source definitions. SAP returns a list of definitions from the SAP dictionary. You can select multiple definitions to import into the PowerCenter repository. The Designer imports the definition as an SAP ECC source definition. After you import a definition, use it in a mapping to define the extract query.

You can import the following definitions into the PowerCenter repository:

- **SAP tables and views.** SAP tables include transparent, pool, and cluster tables. In addition, you can extract data from database views in SAP.
- **SAP hierarchies.** A hierarchy is a tree-like structure that defines classes of information.
- **SAP IDocs.** An IDoc is a generated text file that contains a hierarchical structure consisting of segments.

If the source changes after you import the definition, reimport the definition as a new SAP ECC source definition.

Editing a Source Definition

After you import a table or hierarchy definition from SAP, you can edit the definition properties. For example, you might create logical primary-foreign key relationships between two sources. You can also change the database name of the hierarchy if you want to display the definitions in separate nodes in the Navigator.

Table and View Definitions

You can import transparent, pool, and cluster table definitions as SAP ECC source definitions. You can also import database view definitions. Database views are based on views of transparent tables. PowerCenter does not differentiate between tables and views. You import definitions and extract data from views the same way you import and extract from tables.

You do not need to know the physical table structure on the underlying database server when you import the definition from the logical table on the application server.

When you import table definitions, the Designer displays the table name followed by the business name in the Import SAP Metadata dialog box. You can filter by table name or business name when you connect to the SAP system. Add source names to an import list before you import them. For table definitions, you can select to import all keys or a subset of all keys.

Note: Do not use table definitions as SAP sources if the sources have circular primary key-foreign key relationships.

The Designer imports the following SAP table information:

- Source name
- Column names
- Business descriptions
- Datatypes, length, precision, and scale
- Key relationships

Importing Key Relationships

After you connect to the SAP system through the Import SAP Metadata dialog box, you can designate which keys to import. You can select to import all keys or only primary and primary-foreign keys. The selection applies to all tables in the import list.

Importing All Keys

When you import all keys, the Designer imports relationships that are defined in the database and in the data dictionary including primary, primary-foreign keys, and foreign keys. This feature is useful when you build mappings between related tables and you need the foreign keys to generate the join.

Note: SAP does not always maintain referential integrity between primary key and foreign key relationships. If you use SAP ECC source definitions to create target definitions, you might encounter key constraint errors when you load the data warehouse. To avoid these errors, edit the keys in the target definition before you build the physical targets.

Importing a Subset of All Keys

When you import a subset of keys, the Designer imports relationships that are defined in the data dictionary. This includes primary and primary-foreign keys. However, foreign keys are not imported. SAP maintains referential integrity with the primary and primary-foreign key relationships.

Hierarchy Definitions

A hierarchy is a tree-like structure that defines classes of information. Each class is represented by a different level of the hierarchy. A hierarchy is related to one or more tables, called detail tables, that contain detail data associated with the hierarchy. The detail table is keyed to the root level of the hierarchy. The structure at each level of the hierarchy is called a node. A hierarchy has the following types of nodes:

- **Root node.** The highest node in the structure and the origin of all other nodes. The root node represents the hierarchy.
- **Leaf nodes.** The lowest level in the structure. The leaf nodes are keyed to the detail table through a range of values. This range of values is defined by a beginning and ending value, called From_Value and To_Value, respectively.
- **Higher nodes.** The nodes between the root and leaf nodes. Higher nodes represent the logical path from the root node to the leaf nodes. There may be multiple levels of higher nodes.

SAP has the following types of hierarchies:

- **Uniform.** All branches have the same number of levels.
- **Non-uniform.** Branches have different numbers of levels.

Uniform Hierarchies

A hierarchy is uniform if each level in the hierarchy represents the same type of information.

Non-Uniform Hierarchies

A hierarchy is non-uniform if one or more of the branches do not have the same number of levels.

The following figure shows a sample non-uniform hierarchy:

In this example, Tech Co is the root node. There is a level between Division and Employee ID that represents Department. The Sales Division branches directly to Employee ID.

When the PowerCenter Integration Service extracts data for this hierarchy, it inserts NULLs in the Department level and SetID columns associated with Sales. Some nodes do not have descriptions, but all nodes have SetIDs. If a node does not have a description, the PowerCenter Integration Service inserts NULLs in that column, and it extracts the SetID into the associated SetID column.

Importing Hierarchy Definitions

You can import single-dimension CO hierarchies as SAP ECC source definitions. Single-dimension hierarchies have one associated detail table. After you import a hierarchy definition, you can also import the detail table definition and establish a key relationship between the two definitions.

When you import hierarchy definitions, the Designer displays all available business names followed by the SetID in the Import SAP Metadata dialog box. If a business name does not exist, the Designer displays the SetID. You can enter filter criterion for business names only.

After you import a hierarchy definition, the Designer creates the following columns:

- **Root node and SetID.** The Designer creates two columns for the hierarchy root, one column for the root node, and one for the SetID of the root node.
- **Each node level and SetID.** The Designer creates two columns for each level representing higher nodes in the hierarchy, one column for each node level and the SetID of the node level.
- **Detail range for the leaf nodes.** The Designer creates two columns to represent the range of values for the leaf nodes in the hierarchy. These columns are named FROM_VALUE and TO_VALUE.

For example, you import a uniform hierarchy with a root node, a level 1 node, level 2 node, and leaf nodes. The Designer creates the following columns from the hierarchy structure:

	Column Name	Datatype	Prec	Scale	Not Null	Key Type
1	R300R3000_FLD	CHAR	50	0	☑	NOT A KEY
2	R300R3000_FLD_SetId	CHAR	30	0	☑	NOT A KEY
3	Level1	CHAR	50	0	☑	NOT A KEY
4	Level1_SetId	CHAR	30	0	☑	NOT A KEY
5	Level2	CHAR	50	0	☑	NOT A KEY
6	Level2_SetId	CHAR	30	0	☑	NOT A KEY
7	FROM_VALUE	CHAR	30	0	☑	FOREIGN KEY
8	TO_VALUE	CHAR	30	0	☑	NOT A KEY

In this example, R300R3000_FLD is the root node. R300R3000_FLD_SetId is the root node SetID. Level 1 and Level 2 are higher nodes. FROM_VALUE and TO_VALUE are leaf nodes.

When you import a hierarchy, the Designer creates all columns with a CHAR datatype. The Designer imports all hierarchy definitions with a precision of 50 for the root and higher nodes, and a precision of 30 for the detail range and the SetIDs.

The Designer also imports the following metadata:

- **Source name.** Hierarchy name.
- **Hierarchy SetID.** Unique identifier of the hierarchy.
- **Set table name.** Table name associated with the hierarchy.
- **Business descriptions.** Business description of the hierarchy.
- **Detail table name.** Table that contains the detail information for the hierarchy.
- **Related field name.** Field that joins the hierarchy with the detail table.

Establishing Hierarchy Relationships

To join the hierarchy with the detail table, import the detail table into the Designer. Use the Source Analyzer to establish the key relationship. After you import the definition, you can find the detail table information on the Properties tab of the definition in the Mapping Designer. You can then import the detail table definition and create the relationship.

However, when you import a hierarchy and its detail table, you create a logical relationship. The detail table contains the detail information for the leaf nodes. The hierarchy table contains the range of values for the details.

To establish hierarchy relationships:

1. Double-click the hierarchy table definition.
2. Change the key value of either the FROM_VALUE column or the TO_VALUE column to Foreign Key.
3. In the Primary table list, select the detail table.
4. In the Primary column list, select the primary key of the detail table. Click OK.

IDoc Definitions

SAP uses IDocs to integrate with Electronic Data Interchange (EDI) systems. You can import IDoc definitions as SAP ECC source definitions. An IDoc is a generated text file that contains a hierarchical structure consisting of segments. Each segment is an SAP structure defined in the SAP system.

An IDoc has the following components:

- **Header.** The header contains control information, such as creation date and status. The control information is in an SAP structure called EDIDC.
- **Data records.** The data records are in an SAP structure called EDIDD.

Import IDoc definitions when you want to extract data from the EDIDC and EDIDD structures.

Note: If you want to use IDocs to receive data from mySAP applications and send data to mySAP applications using ALE, do not import IDoc definitions. To use ALE to send and receive IDocs, use SAP/ALE IDoc Interpreter and SAP/ALE IDoc Prepare transformations in a mapping.

Importing IDoc Definitions

When you import an IDoc definition, the Designer connects to the SAP system and imports the metadata for IDocs from the EDIDC or EDIDD structures in the SAP system. During import, the Designer displays a list of basic IDocs. You can expand each IDoc to view a list of segments.

You can import an entire IDoc or an individual segment of an IDoc. When you import an entire IDoc, the Designer imports every segment in the IDoc. After you import an entire IDoc, each segment in the IDoc is independent of the other segments in the IDoc.

Viewing IDoc Definitions

After you import an IDoc definition, the Designer displays data records and some control information of the IDoc. Use the control information to perform lookups on the SAP system.

The Designer adds the following columns from the control information of the IDoc:

- **DOCNUM.** Document number. The SAP system assigns a unique documentation number to each IDoc.
- **STATUS.** The status of the IDoc.
- **CREDAT.** Creation date.
- **CRETIM.** Creation time.
- **SNDSAD.** Sender address.
- **DIRECT.** Direction of the IDoc. The direction can be inbound or outbound.

The Columns tab of the IDoc definition displays the added columns of control information.

The Designer also displays the following IDoc type properties on the Properties tab of the IDoc definition:

- **IDoc Type.** The name of the IDoc definition.
- **Basic IDoc Type.** The name of the basic IDoc type.
- **Extension IDoc Type.** The name of the user-defined extension of a basic IDoc type.

Importing a Source Definition

When you import source definitions, you connect to the SAP system through the Import SAP Metadata dialog box. The Designer provides the following tabs in the Import SAP Metadata dialog box:

- **Tables.** Import table and view definitions.
- **Hierarchies.** Import hierarchy definitions.
- **IDocs.** Import IDoc definitions.

You can also enter filter criterion to reduce the number of definitions the Designer displays in the selection list. If the first character of an SAP source name is an asterisk (*) or a number, the Designer converts the first character to an underscore (_) when you import the source definition.

Filtering Definitions in the Import Dialog Box

When you enter a business name filter, the Designer applies the filter to both tables and hierarchies. When you enter a table name filter, the Designer applies the filter to tables only, and returns all hierarchy definitions under the hierarchy tab.

The following rules apply to filter syntax:

- Use the percent sign (%) as a wildcard search for multiple characters.
- Use an underscore (_) as a wildcard search for single characters.
- Separate multiple table or business names with commas.

For example, if you select the Table tab and enter EKKO, BSE%, the SAP system returns the table named EKKO and all tables that begin with BSE.

Steps to Import an SAP ECC Source Definition

Complete the following steps to import an SAP ECC source definition.

To import an SAP ECC source definition:

1. In the Source Analyzer, click Sources > Import from SAP.

   The Import SAP Metadata dialog box appears.

2. To connect to the SAP system, enter the following information:

Field	Description
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Password for the user name.
Client	SAP client number.
Language	Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

3. Optionally, enter a filter.
4. Select Table or Business names filter button to apply the filter criterion.
5. Click one of the following tabs for the source you want to import: Tables, Hierarchies, or IDocs.
6. Click Connect.

   The Designer displays the table, hierarchy, or IDoc definitions.

7. If you are importing table definitions, clear All Keys if you want to import a subset of all key relationships.
8. Select the object or objects you want to import.
   - Hold down the Shift key to select blocks of sources.
   - Hold down the Ctrl key to make non-contiguous selections within a folder.
   - Use the Select All button to select all tables.
   - Use the Select None button to clear all highlighted selection.
9. Click Add To Import List.
10. To view the list, click View Import List.
11. To remove items from the list that you do not want to import, select the item and click Delete.
12. Click Close to close the Import List dialog box.
13. When the Import List is complete, click OK.

Organizing Definitions in the Navigator

By default, the Navigator displays both table definitions and hierarchy definitions under a single node in the Sources folder. The Designer provides the following ways to organize these definitions in the Navigator:

- Create separate repository nodes for the hierarchies.

- Create business components for related sources.

To display hierarchies separately in the Navigator, set the Designer options to group sources by database name. Then, edit the hierarchy definition and rename the database name to Hierarchy.

To edit the database name for hierarchies:

1. In the Source Analyzer, double-click the title bar of the hierarchy definition.

   The Edit Tables dialog box appears.
2. Click Rename.

   The Rename Source Table dialog box appears.
3. Change the Database Name to Hierarchy.
4. Click OK twice.

   Note: You also need to change the Designer options to display definitions in the Navigator by database name.

Working with Business Components

Business components provide a way to organize related sources. You can create business components in the Navigator to organize related SAP ECC sources. After you create a business component, drag the table and hierarchy definitions into the business component. The Navigator maintains the definition in the Sources folder and also in the Business Components folder. You can edit the definition in either folder. You can delete the definition in the business component without affecting the definition in the Sources folder.

Troubleshooting SAP ECC Source Definitions

When I tried to import an SAP ECC source definition, I received the following error:

```
SAP System Exception Failed
Key = RFC_ERROR_SYSTEM_FAILURE
Message = Function Module "<function module name>" not found
```

You connected to a production system. Connect to a development system.

When I view the properties of an imported SAP ECC table definition, I see the number sign (#) for several characters in the table description.

The Designer displays the number sign (#) for each character that is not converted while importing metadata from SAP. If the PowerCenter Integration Service is running in Unicode mode, the conversion error probably occurred because the imported table does not have a description in the connection language you selected in the Import SAP Metadata dialog box. Log in to the SAP system and enter a table description for this language.

CHAPTER 6

SAP ECC Target Definitions

This chapter includes the following topics:

- SAP ECC Target Definitions Overview, 80
- Editing a Target Table Definition, 81
- Importing an SAP ECC Target Definition, 82

SAP ECC Target Definitions Overview

When you import target definitions from SAP, the Designer uses RFC to connect to the SAP application server. The Designer calls functions in the SAP system to import target tables. SAP returns a list of tables from the SAP dictionary. You can select multiple tables to import into the PowerCenter repository. The Designer imports the tables as an SAP ECC target definition.

You can drag an SAP source definition into the Target Designer workspace. The Designer creates an SAP target definition that matches the source definition. You can edit the definition to change information, such as the description, columns, data types, and target type.

Target Table Definitions

You can import transparent, pool, and cluster table definitions as SAP ECC target definitions. After you import definitions you can load data into the SAP tables.

The Import SAP Metadata dialog displays the table name followed by the business name. You can filter by table name or business name when you connect to the SAP system. Add target names to an import list before you import them. For table definitions, you can select to import all keys or a subset of all keys.

The Designer imports the following SAP table information:

- Table name
- Column names
- Business descriptions
- Datatypes, length, precision, and scale
- Key relationships

Editing a Target Table Definition

After you import a target, you can edit the table definition properties on the **Properties** tab. You can override the default update statement and the target table name in the Properties tab of the target instance:

Note: You can edit the Update Override and the Target Table name properties at the session level. The properties set at the session level override the properties set at the target definition.

Target Update Override

By default, the PowerCenter Integration Service updates target tables based on key values. You can override the default UPDATE statement for each target in a target definition. You might want to update the target based on non-key columns.

For a mapping without an Update Strategy transformation or a Custom transformation with the update strategy property enabled, configure the session to mark source rows as update. The Target Update option only affects source rows marked as update. The PowerCenter Integration Service processes all rows marked as insert, delete, or reject normally. The Target Update Override only affects source rows marked as update by the Update Strategy or Custom transformation.

For example, a mapping passes the total sales for each salesperson to the ZT_SALES table. In the Designer enter the following UPDATE statement for the target ZT_SALES:

```
UPDATE ZT_SALES SET EMP_NAME = :TU.EMP_NAME DATE_SHIPPED = :TU.DATE_SHIPPED TOTAL_SALES
= :TU.TOTAL_SALES WHERE EMP_ID = :TU.EMP_ID
```

Because the target ports must match the target column names, the update statement includes the keyword:TU to specify the ports in the target transformation. If you modify the UPDATE portion of the statement, use :TU to specify ports.

Overriding the WHERE Clause

You can override the WHERE clause to include non-key columns. For example, you might want to update records for employees named Mike Smith only. You can edit the WHERE clause as follows:

```
UPDATE ZT_SALES SET DATE_SHIPPED = :TU.DATE_SHIPPED

TOTAL_SALES = :TU.TOTAL_SALES WHERE

EMP_NAME = 'MIKE SMITH'
```

Rules and Guidelines for Configuring the Target Update Override

Use the following rules and guidelines when you enter target update queries:

- You can use parameters and variables in the target update query. Use any parameter or variable type that you can define in the parameter file. You can enter a parameter or variable within the UPDATE statement, or you can use a parameter or variable as the update query. For example, you can enter a session parameter, $ParamMyOverride, as the update query, and set $ParamMyOverride to the UPDATE statement in a parameter file.
- If you update an individual row in the target table more than once, the SAP table only has data from the last update. If the mapping does not define an order for the result data, running different PowerCenter sessions on identical input data may result in different data in the target table.
- If the WHERE clause contains no port references, the mapping updates the same set of rows for each row of the mapping. For example, the following query updates all employees with EMP_ID > 100 to have the EMP_NAME from the last row in the mapping:

  ```
  UPDATE ZT_SALES set EMP_NAME = :TU.EMP_NAME WHERE EMP_ID > 100
  ```

- If the mapping includes an Update Strategy or Custom transformation, the target update statement only affects records marked for update.
- If you use the target update option, configure the session to mark all source records as update.

Target Table Name

You can override the target table name in the target definition. Override the target table name when you use a single mapping to load data to different target tables. Enter a table name in the target table name field. You can also enter a parameter or variable.

You can override the target name at the session level. The target table name you set at the session level overrides the target table name you set in the target definition.

Keys and Key Relationships

When you import all keys, the Designer imports relationships that are defined in the SAP data dictionary including primary, primary-foreign keys, and foreign keys.

After you connect to the SAP system through the Import SAP Metadata dialog box, you can designate which keys to import. You can select to import all keys or only primary and primary-foreign keys. The selection applies to all tables in the import list.

Importing an SAP ECC Target Definition

When you import a target definition, you connect to the SAP system through the Import SAP Metadata dialog box. The Designer provides the Tables tab in the Import SAP Metadata dialog box. You can enter the table name that you want to import.

1. In the Target Designer, click **Targets** > **Import from SAP**.

   The **Import SAP Metadata** dialog box appears.
2. To connect to the SAP system, enter the following information:

Field	Description
Connect String	DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Password for the user name.
Client	SAP client number.
Language	Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

3. Optionally, enter a filter.

4. Select Table or Business names filter button to apply the filter criterion.
5. Click Connect.

   The Designer displays the table definitions.
6. Clear **All Keys** if you want to import a subset of all key relationships.
7. Select the object or objects that you want to import.
   - Hold down the Shift key to select blocks of sources.
   - Hold down the Ctrl key to make non-contiguous selections within a folder.
   - Use the Select All button to select all tables.
   - Use the Select None button to clear all highlighted selection.
8. Click **Add To Import List**.
9. To view the list, click View Import List.
10. To remove items from the list that you do not want to import, select the item and click Delete.
11. Click Close to close the Import List dialog box.
12. When the Import List is complete, click OK.

Filtering Definitions in the Import Dialog Box

When you enter the business name or the table name, the Designer applies the filter condition to the SAP and returns the tables matching the filter condition.

The following rules apply to filter syntax:

- Use the percent sign (%) as a wildcard search for multiple characters.
- Use an underscore (_) as a wildcard search for single characters.
- Separate multiple table or business names with commas.

For example, if you select the Table tab and enter EKKO, BSE%, the SAP system returns the table named EKKO and all tables that begin with BSE.

CHAPTER 7

Working with ABAP Mappings

This chapter includes the following topics:

- Working with ABAP Mappings Overview, 84
- Setting the Select Option, 84
- Setting Order By Ports, 86
- Viewing the Hierarchy Properties, 87
- Viewing IDoc Properties, 87
- Working with the ABAP/4 Program, 88
- Troubleshooting ABAP Mappings, 95

Working with ABAP Mappings Overview

Complete the following steps when you create a mapping with an SAP ECC source definition:

1. **Configure the source definition.** The source definition has the following configuration properties that allow you to optimize performance when you extract from SAP:
   - **Select option.** Restricts the number of rows returned from the SAP ECC source.
   - **Order By.** Orders by primary keys or by a specific number of ports.
2. **Create and configure the Application Source Qualifier.**
3. **Install the ABAP program.** The ABAP program extracts source data from the SAP system.

Use the following guidelines when you create a mapping with an SAP ECC source definition:

- The mapping name cannot exceed 56 characters.
- The mapping name or description and the folder or repository name in which you save the mapping cannot contain the word “REPORT.” When the word “REPORT” exists in the mapping name or description and the folder or repository name, the ABAP program fails.

Setting the Select Option

You can restrict the number of rows returned from the SAP ECC source table. Configure the Select Option property in the source definition to select either a single row or a distinct set of rows from the source. By default, the Select Option property selects all rows from the source.

The property settings are:

- **Select All.** Default setting. Select all rows from a source.
- **Select Single.** Select a single row from a table using open SQL.
- **Select Distinct.** Select unique values from a table using open SQL or exec SQL.

The following table summarizes select options and the conditions for using them:

ABAP Generation	Select All	Select Single	Select Distinct
Open SQL	Yes	Yes	Yes
Exec SQL	Yes	No	Configure all sources to Select Distinct.
ABAP join syntax	Yes	No	Configure all sources to Select Distinct.
Hierarchy and IDoc definitions	n/a	n/a	n/a

Select Single

Select Single is an open SQL command that returns one row from the SAP ECC source. When you use open SQL, the Designer generates a Select Single statement for each source definition you configure with the Select Single option.

You might use this option with a nested loop join when you want to join tables based on key values in one table. The inner loop with Select Single matches one record for each join condition. Select Single improves performance in the select loop by selecting a single row of data rather than the entire table.

Select Single is not available with the following options:

- **Exec SQL and ABAP join syntax.** Exec SQL and ABAP join syntax do not recognize Select Single. Therefore, the Designer does not generate a Select Single statement in the ABAP program if you configure the Application Source Qualifier to generate exec SQL or ABAP join syntax.
- **Order By.** If you configure the source definition to use Select Single and Order By, the Designer generates a Select Single statement in the ABAP program, but it does not generate an Order By statement.
- **Hierarchy and IDoc definitions.** The Select Single option is not available for hierarchy and IDoc definitions.

Select Distinct

Select Distinct is a command that returns rows with a distinct set of key values. Use Select Distinct with open SQL, exec SQL, and ABAP join syntax. When you use open SQL, the Designer generates a select distinct statement for each source definition you configure with the Select Distinct option.

You might use this option when you only want to return records associated with particular key values. Select Distinct improves performance by filtering out unnecessary data early in the data flow.

If you join multiple sources in a single Application Source Qualifier configured for exec SQL and ABAP join syntax, and you want to use select distinct, choose Select Distinct for each source connected to the Application Source Qualifier. The Designer generates a Select Distinct statement with exec SQL only when you configure all of the source definitions with the Select Distinct option.

Note: If you use the Select Distinct option for LCHR when the length is greater than 2,000 characters and the underlying source database is Oracle, the session fails.

The Select Distinct option is not available for hierarchy and IDoc definitions.

Setting Order By Ports

You can improve session performance by sorting source data by primary key or a specified number of ports. Configure the Order By Ports property in the source definition. When the Designer generates ABAP using nested loop joins, it determines the number of columns for each source.

The Designer generates an Order By statement in the ABAP program when you specify a positive number of ports. Order by support differs for transparent tables and pool and cluster tables.

Note: If you include LRAW ports in the Order By statement, the session fails.

Transparent Tables

When you specify a sort order for a source definition in a mapping, the Designer generates an Order By statement beginning with the first column in the definition. The Designer generates the Order By statement using the following guidelines:

- If you specify a number of ports greater than the number of ports in the source definition, the ABAP program generates an Order By statement with all ports in the source definition.
- SAP requires all columns in the Order By statement to be part of the select statement. If you include a column in the Order By selection, but you do not project it into the Application Source Qualifier, the ABAP program adds that column to the select statement. However, the ABAP program does not extract the data from the column you excluded from the Application Source Qualifier.

The Order By statement differs for exec SQL, open SQL, and ABAP join syntax. The following samples are based on the same mapping that joins KONH and KONP in one Application Source Qualifier. Each source definition is configured to order by three ports.

Exec SQL

The Order By statement is similar to standard relational statements:

```
exec sql [...]
SELECT KONH.MANDT, KONH.KNUMH, KONH.ERNAM, KONH.ERDAT, KONH.KVEWE,
[...], KONP.MANDT, KONP.KNUMH, KONP.KOPOS, KONP.KAPPL,[...]
INTO [...]
FROM KONH, KONP
where [...] and [...]
order by KONH.MANDT, KONH.KNUMH, KONH.ERNAM, KONP.MANDT, KONP.KNUMH, KONP.KOPOS
endexec.
```

Open SQL

The Order By statement is generated within the nested loop for each source in the Application Source Qualifier:

```
select MANDT KNUMH ERNAM [...]
into [...]
from KONH where [...]
order by MANDT KNUMH ERNAM .
  select MANDT KNUMH KOPOS KAPPL [...]
  into [...]
  from KONP
  where [...]
  order by MANDT KNUMH KOPOS .
```

Note: The Designer does not generate an Order By clause if you use select single in the source properties.

ABAP Join Syntax

The Designer generates the Order By statement after the WHERE clause:

```
SELECT KONP~MANDT KONP~KNUMH[...] KONH~ERDAT KONH~KVEWE KONH~KOTABNR KONH~KAPPL [...]
INTO [...]
FROM KONH
INNER JOIN KONP
ON
KONP~KNUMH = KONH~KNUMH
WHERE
[...] and [...]
ORDER BY KONH~MANDT KONH~KNUMH KONH~ERNAM
KONP~MANDT KONP~KNUMH KONP~KOPOS .
```

Pool and Cluster Tables

You can order by primary keys for pool and cluster tables. When you specify any positive number to order by in the source definition, the Designer generates an Order By statement to order by primary key:

```
select MANDT KAPPL KSCHL LIFNR MATKL INFNR DATBI DATAB KNUMH
into [...]
from A015 where [...]
order by primary key .
```

Viewing the Hierarchy Properties

If you want to join the hierarchy with the detail table, you can view detail table and key field names related to the hierarchy on the Properties tab of the hierarchy definition in the Mapping Designer. The Select and ORDER BY options are not available for hierarchy definitions.

Viewing IDoc Properties

The Properties tab of the IDoc source definition displays the following information:

- **IDoc type**. Name of the IDoc definition.
- **Basic IDoc type.** Name of the basic IDoc type.
- **Extension IDoc type.** Name of the user-defined extension of a basic IDoc.

The Select and Order By options are not available for IDoc definitions.

In the Source Analyzer, you can import an entire IDoc or an individual segment of an IDoc. If two different IDocs have segments with the same name, you can edit the IDoc Type to indicate which segment you want to use in the mapping.

For example, the IDocs E1BPACAR01 and E1BPACAR02 both have a segment named E1MVKEM. You import E1MVKEM from E1BPACAR01 in the Source Analyzer. You cannot import E1MVKEM twice in the Source Analyzer. To use the E1MVKEM segment from E1BPACAR02, change the IDoc Type in the Mapping Designer to E1BPACAR02.

Working with the ABAP/4 Program

SAP uses its proprietary language, ABAP/4, to extract data. You need to generate and install an ABAP program from the Designer after you configure a mapping with an SAP ECC source definition. The Designer generates a unique program name and stores it in the repository. When you generate the ABAP program, you install it on the SAP system that contains the source data. The PowerCenter Integration Service uses the ABAP program to extract data from the SAP system.

Use the following options to maintain ABAP programs for mappings with SAP ECC sources:

- Generate ABAP programs to extract source data.
- Install ABAP programs directly or from a local copy.
- Uninstall ABAP programs you no longer want to use.
- Clean ABAP programs if you delete a folder from the repository.
- Copy ABAP programs when you copy a folder or mapping to another repository.

The PowerCenter Integration Service extracts hierarchy data through a remote function call, not through an ABAP program. If you configure a mapping with a hierarchy definition and no detail table definition, the Designer does not generate an ABAP program for the mapping.

Note: You cannot generate and install ABAP programs from mapping shortcuts.

If a mapping becomes invalid after you install the ABAP program, validate the mapping, save the repository, and re-install the ABAP program. If you save the repository with the mapping open after you install the ABAP program, the session fails, and the session log instructs you to regenerate and install the program.

You can use the pmrep command line programs to generate and install ABAP programs for mappings with SAP ECC sources:

- GenerateAbapProgramToFile. Generates the ABAP program for a mapping and saves the program as a file.
- InstallAbapProgram. Generates and installs the ABAP program from a mapping in the PowerCenter repository, or installs the ABAP program from a file.
- UninstallAbapProgram. Uninstalls the ABAP program.

Selecting the Program Mode

The program mode is the mode in which the application server extracts data. When you generate an ABAP program, choose one of the following program modes:

- **File.** Extracts data to a staging file. The PowerCenter Integration Service accesses the file through FTP, SFTP, or NFS mount.
- **Stream.** Extracts data to buffers. The PowerCenter Integration Service accesses the buffers through RFC and the deprecated CPI-C mode.

Note: If a mapping contains both hierarchies and tables, generate the ABAP program using file mode. The Designer does not generate ABAP if you choose stream mode for a mapping that contains hierarchies and tables.

Related Topics:

- "Running Stream Mode Sessions" on page 124
- "Running File Mode Sessions" on page 125

Naming the ABAP Program

The Designer generates a unique ABAP program name when you generate or install the ABAP program for the first time. You can override the generated program name if you are installing or generating the ABAP program for the first time.

If you have registered a namespace with SAP, you can also choose to add a namespace prefix to the ABAP program name. The Designer adds the namespace to the ABAP program name if you are installing or generating the ABAP program for the first time.

You can use long names and namespace when you use the RFC file mode and RFC stream mode.

If you want to override the program name or add a namespace after installing or generating, you need to uninstall the ABAP program from all SAP systems. Then, you can install the ABAP program again with a namespace or program name override. You can enter an ABAP program name up to 30 characters.

Adding Authority Checks

When you generate the ABAP program, you can add authority checks. The Designer adds an authority check to the ABAP program for each Application Source Qualifier in the mapping.

The application server verifies that the user running the workflow has read permission on the sources. If the user does not have authorization to access one of the sources, the session fails with a NO AUTHORITY error.

If you enabled the Authority Check option when you generated the ABAP program, the SAP application server verifies that the user running the workflow has authorization to read the sources. SAP verifies the authorization before it reads the first source. If the user in the SAP ECC application connection does not have read authorization on any one of the sources, the session fails with a NO AUTHORITY error.

Working with ABAP Programs and Versioned Mappings

You can install and uninstall an ABAP program for versioned mappings. The repository can contain versioned mappings when you configure it to store multiple copies of the same object as you make changes and save it.

Generating and Installing an ABAP Program for Versioned Mappings

You can install an ABAP program for any or all versions of a mapping. When you create a new version of a mapping, install a new ABAP program. An ABAP program from a previous version of the mapping does not carry over with the new version.

When you generate an ABAP program, the Designer displays all versions of mappings in the Generate and Install dialog box.

Uninstalling an ABAP Program from Versioned Mappings

Uninstall an ABAP program when you no longer want to associate the program with a mapping. The Designer uninstalls the programs from the repository and the SAP system. You can uninstall an ABAP program for any or all versions of a mapping. You can also uninstall an ABAP program from older or deleted versions of a mapping.

When you uninstall an ABAP program, the Designer displays all versions of mappings in the Installed Programs dialog box.

Undoing a Checkout and Purging a Mapping with an ABAP Program

If you undo a checkout of a mapping or purge a mapping, the Designer marks that version of the mapping as deleted. Any valid versions of the mapping no longer use the ABAP program you installed for the deleted version of the mapping. Install an ABAP program for the valid version of the mapping.

The PowerCenter Repository Service does not delete the ABAP program. You need to clean the ABAP program information if you want to delete the program.

Generating and Installing the ABAP Program

The Designer installs the ABAP program in the development class shown in the Development Class field. The default development class is $TMP. The $TMP development class is temporary. You cannot transport ABAP programs from this class to another system. If you want to transport the ABAP program to a production system, create a development class within SAP for the ABAP programs. Install ABAP programs that use a namespace in a development class that is in the same namespace.

You must regenerate and install the ABAP program if there is a change in the source, Application Source Qualifier transformation, or their links.

You can install the ABAP program directly on the SAP system, or you can generate the ABAP program locally and install it using the local copy.

If the SAP system is a Unicode system, the Designer is connected to a Unicode repository, and the ABAP program contains a source filter with ISO 8859-1 or multibyte characters, generate a local copy of the ABAP program and upload the generated file into the SAP system.

Secure Transports

You can install secure transports to enforce security authorizations when you use ABAP to read data from SAP.

When you use secure transports to perform data integration tasks with ABAP, the PowerCenter Integration Service verifies that the following conditions are met:

- The SAP user who generates and installs the ABAP program has the developer license.
- The user who generates and installs the ABAP program is authorized for the following function groups:
  - SKEY
  - STR9
- Users cannot generate and install ABAP programs from the Designer tool if the SAP client is locked down for changes.
- A new workbench request is created to track object changes in SAP when users install or delete ABAP programs.
- The ABAP program can be installed only on development systems and not on production systems.

Note: If you use secure transports and any of these conditions are not met, the ABAP program installation fails.

Compressed Data Transfer

When you run a file mode session to read data from SAP through ABAP, you can configure the FileCompressEnable custom property to enable compressed data transfer. When you compress data, you can increase the session performance and decrease the disk storage that the staging file needs. To enable compressed data transfer, add the FileCompressEnable custom property at the session level or at the PowerCenter Integration Service level and set its value to 1.

Note: ABAP programs that you generate from version 9.6.1 contain code for compressed data transfer and uncompressed data transfer. To enable compressed data transfer, you must configure the FileCompressEnable custom property.

Generating the ABAP Program and Installing Directly onto the SAP System

You can install the ABAP program directly onto the SAP system. When you install directly onto the SAP system for the first time, the Designer generates a program name.

If you are generating the ABAP program for the first time, you can override the generated program name by selecting Enable Override. You cannot override the local file name.

To generate the ABAP program and install it directly onto the SAP system:

1. Click Mappings > Generate and Install R/3 Code.

   The Generate and Install dialog box appears.

2. Enter the following information to connect to the SAP system:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

3. Click Connect.

   A list of mappings in the folder appears. If the mappings are versioned, the version number appears next to each version of the mapping.

4. Select the ABAP mapping(s) for which you want to install ABAP.

   Use the following guidelines when you select the mappings for which you want to install ABAP:

   - You can install ABAP for all versions of a mapping.
   - You can install ABAP for different versions of the same mapping.
   - You can install ABAP for all versions of multiple mappings.

5. Select the Program Mode: File or Stream.
6. Optionally, select Enable Override to override the default ABAP program name.
7. Optionally, select Use Namespace to prefix a namespace you registered with SAP to the ABAP program name.
8. In the Development Class box, enter the name of the development class where you want to install the program.

   The default development class is $TMP.

Note: The $TMP development class is temporary. You cannot transport ABAP programs from this class to another system.

Install ABAP programs that use a namespace in a development class that is in the same namespace.

9. Click Direct Installation.

   Note: If you reinstall an existing ABAP program that uses the CPI-C protocol, you are prompted to overwrite the program to use the RFC protocol. Informatica recommends overwriting the program to use the RFC protocol.

10. Enter the ABAP program name if you selected Enable Override.

    This step also generates the ABAP program.

Generating and Installing from a Local Copy

Use the Designer to generate an ABAP program file in a specified local directory so you have a local copy. You can then install from the local copy. To view the file, click View File in the Generate and Install dialog box, or open the file with a text editor.

The naming convention for the local file is *mapping_name*_file.ab4 or *mapping_name*_stream.ab4, depending on the program mode you select. The Designer also creates a program name for generating the ABAP program on an SAP system.

After you use the Designer to generate the ABAP program file in a specified local directory, you can install from the local copy. When you install from a local copy, you need to be connected to the repository from which the ABAP program was generated.

To generate and install the ABAP program from a local copy:

1. Complete steps from 1 to 8 as described in "Generating the ABAP Program and Installing Directly onto the SAP System" on page 91.
2. In the Generate and Install dialog box, click Generate Files to generate the ABAP program.

   Or, if you want to override the generated program name, select Enable Override and click Generate Files.
3. If you selected to override the generated program name, enter the ABAP program name and click OK.

   ABAP program names must start with "Y" or "Z" when you select to override the program name.

   The output window displays a message indicating successful generation:

   ```
   Program generated successfully for mapping EKKO in file c:\temp\EKKO_File.ab4.
   ```
4. To view the local ABAP copy, click View File and double-click the file name.
5. Click Install From Files.

   Note: If you reinstall an existing ABAP program that uses the CPI-C protocol, you are prompted to overwrite the program to use the RFC protocol. Informatica recommends overwriting the program to use the RFC protocol. Note that when you overwrite a program, any customization that you made to the program is lost.
6. Double-click the ABAP file from the Open ABAP File dialog box.

   The output window displays a message indicating successful installation:

   ```
   Program YEKKO_99 installed successfully at destination sophie, from file c:\TEMP
   \EKKO_File.ab4.
   ```

Generating and Uploading into the SAP System

Use the Designer to generate an ABAP program file in a specified local directory. Use the SAP system to upload the generated file into the SAP system.

To generate and upload the ABAP program into the SAP system:

1. Generate and install the ABAP program from a local copy as described in "Generating and Installing from a Local Copy" on page 92.
2. Log in to the SAP system and upload the generated ABAP program file.

   When prompted, override the incorrect program you installed in step 1 with the correct program that you are uploading.

Deploying ABAP Mappings with ABAP Programs

When you add a versioned ABAP mapping with an ABAP program to a deployment group, you can choose to deploy the ABAP program along with the mapping.

Viewing Program Information

You can view information for ABAP programs on the SAP server to which you are connected. You can also view information for all ABAP programs installed for mappings in the repository folder.

To view ABAP programs:

1. Click Mappings > Generate and Install R/3 code.
2. Connect to the SAP server.
3. Click the View Prog Info button.

   The Installed Programs dialog box displays the following ABAP program information:

Attribute	Description
Program Name	Name of the ABAP program.
Mapping Name	Name of the SAP mapping.
Version Number	Version number of the SAP mapping.
Type	ABAP program mode. Displays one of the following values: - File: Indicates that the ABAP program runs in file mode. - File (C): Indicates that the ABAP program runs in file mode and can compress the source data before loading the data to the staging file. - Stream (CPI-C): Indicates that the ABAP program runs in the deprecated CPI-C stream mode. - Stream (RFC): Indicates that the ABAP program runs in RFC stream mode.
Install Time	Time at which the ABAP program is installed.
HostName	Host name of the SAP server.
User	Name of the user who created the ABAP program.
Client	Client number of the SAP server.

4. To view information for ABAP programs on the SAP server to which you are connected, select Show Programs Installed on the Current SAP Server. To view information for all ABAP programs installed for mappings in the repository folder, clear this option.

Uninstalling the ABAP Program

Uninstall an ABAP program when you no longer want to associate the program with a mapping. The Repository Manager uninstalls the programs from the repository and the SAP system.

You can only uninstall ABAP programs from the SAP system to which you are connected.

Note: If you remove a folder from a repository, you cannot uninstall any ABAP programs associated with mappings in the folder. To remove the programs, clean the ABAP program information.

To uninstall ABAP programs:

1. Click Mappings > Generate and Install R/3 code.
2. Connect to the SAP server.
3. Click the View Prog Info button.
4. Select the program you want to uninstall.

   Use the Ctrl key or Shift key to select multiple programs.

   To select all older or deleted mapping versions can, click Select Old/Deleted Mapping Versions.
5. Click Uninstall.

   The output window displays a message indicating the program was successfully uninstalled:

   ```
   Program YEKKO_99 un-installed successfully from destination sophie.
   ```

Cleaning ABAP Program Information

When you delete a folder from the repository, the PowerCenter Repository Service does not uninstall the ABAP program. The ABAP program remains on the SAP system. The repository entry of the ABAP program also remains in the repository.

You can clean the ABAP program information from the SAP system and the repository. You cannot use the Generate and Install dialog box to uninstall ABAP programs associated with deleted folders.

You may see the following types of ABAP programs when you clean ABAP program information:

- **Shared programs.** One or more mappings in other existing folders associate with a shared program. If you clean a shared program, the Designer only cleans the repository entry corresponding to the selected program in the deleted folder.
- **Orphan programs.** Only mappings in the deleted folder associate with an orphan program. If you clean an orphan program, the Designer uninstalls the program from the SAP system and removes all repository entries corresponding to the selected program in the deleted folder.

To clean ABAP program information in deleted folders:

1. Click Mappings > Clean ABAP Program Information.
2. Enter the connection information to connect to the application server.
3. Expand the SAP system node to select the orphan or shared ABAP program to clean.
4. Click Uninstall. Then, click Close.

Copying Program Information

When you copy a folder or a mapping to another repository, you can copy the ABAP program information along with the mapping. You can then run a session against the copied mapping without regenerating the ABAP program.

You might want to copy program information when you move into a test or production environment. For example, you develop a mapping, install the ABAP program, and run a successful session.

Complete the following steps to move to a test environment:

1. Transport the ABAP program to the SAP test system. Typically, SAP development systems have the PowerCenter transports that allow program installation, while the test and production systems do not allow program installation from PowerCenter. Therefore, when you move to test or production, you need to transport the ABAP program.
2. Copy the mapping and program information to a test repository.
3. Create and run a session against the mapping using the copied program ID and the transported ABAP program.

When you use the Mapping Copy command or the Folder Copy command, you can choose the mappings for which you want to copy the program information. The Designer copies the program ID and the timestamp of the ABAP program so the program is associated with both mappings. The Designer does not copy the ABAP program.

Use the following guidelines when you copy mappings:

- If you are still developing the mapping, do not copy the program information. Instead, generate a new ABAP program from within the copied mapping. If you modify one mapping, the ABAP program might not be valid for the other mapping.
- You cannot copy program information to a repository if another program with the same name exists. For example, you can copy program information once from Repository A to Repository B. After you copy from Repository A to Repository B, you cannot copy it again from Repository A to Repository B. You also cannot copy it back from Repository B to Repository A.
- You cannot copy program information within the same repository.
- When you copy a mapping from one repository to another, or from one folder to another, save the mapping at the target repository or folder before you modify the mapping in the target.
- If you modify a mapping, save the modified mapping to the repository before you copy the mapping to another repository.

Troubleshooting ABAP Mappings

When I tried to install an ABAP program, I received the following error:

```
SAP System Exception Failed
Key = RFC_ERROR_SYSTEM_FAILURE
Message = Function Module "<function module name>" not found
```

You connected to a production system. Connect to a development system.

CHAPTER 8

Working with SAP Functions in ABAP Mappings

This chapter includes the following topics:

- Working with SAP Functions in ABAP Mappings Overview, 96
- Using SAP Functions in the ABAP Program Flow, 96
- Importing SAP Functions, 98
- Viewing SAP Functions, 99
- Inserting SAP Functions in the ABAP Program Flow, 99

Working with SAP Functions in ABAP Mappings Overview

SAP functions are generic modules in the SAP system. The SAP system has a set of standard functions and user-defined functions. SAP functions can accomplish simple tasks such as getting the name of a field, or complex tasks such as calculating depreciation.

If the mapping requires an ABAP program to extract data from SAP, you can insert SAP functions in the ABAP program flow dialog box in the Application Source Qualifier to customize how the ABAP program extracts data. When you run a workflow, the ABAP program calls the SAP function to perform tasks.

To use SAP functions in the ABAP program, you first import the functions in the Source Analyzer and then insert them in the ABAP program flow. In the ABAP program flow, you assign values to function parameters so the function can perform calculations. You then assign variables to hold the result of the function output.

After you customize the ABAP program, you generate and install the ABAP program. The Designer generates a CALL FUNCTION statement in the ABAP program to use the SAP function.

Using SAP Functions in the ABAP Program Flow

When you use an SAP function in the ABAP program flow, you customize how the ABAP program uses selected rows. Use the result of an SAP function in the ABAP program flow. Or, use the result of an SAP function later in the mapping by creating an output port.

PowerExchange for SAP NetWeaver does not provide versioning for SAP functions. If you remove a function from a mapping, the Designer also removes the function from earlier versions of the mapping.

Note: You cannot use SAP functions if the ABAP program flow contains a hierarchy and no other sources.

SAP Function Parameters

Each SAP function has scalar input parameters (input values to the function) and scalar output parameters (output values from the function). The SAP function performs calculations with the values you assign to the scalar input parameters. The SAP function outputs the result of the calculations in the variables you assign to the scalar output parameters.

Some SAP functions may also have changing and table parameters. The Source Analyzer displays information such as the parameter name and parameter type for each parameter. You can assign source fields, constants, and variables to function parameters.

Using SAP Functions in the ABAP Program Flow

Complete the following steps to use an SAP function in the ABAP program flow:

1. Import the SAP function in the Source Analyzer.
2. Insert the SAP function in the ABAP program flow.
3. Assign values and variables to function parameters.
4. Generate the ABAP program.

   For example, you have a source table that contains company codes and information about each company. You want to get details about each company based on the company code. Use the SAP function BAPI_COMPANYCODE_GETDETAIL to get the information.

   Complete the following steps to use the SAP function BAPI_COMPANYCODE_GETDETAIL:

5. **Import the SAP function in the Source Analyzer.** The Source Analyzer displays the following parameters for BAPI_COMPANYCODE_GETDETAIL:
   - Scalar input parameter: CompanyCodeID
   - Scalar output parameters: CompanyCode_Detail, CompanyCode_Address, and Return
6. **Insert the SAP function into the ABAP program flow.** The ABAP program flow in the Application Source Qualifier shows objects in the ABAP program. Insert BAPI_COMPANYCODE_GETDETAIL below the source table in the ABAP program flow.
7. **Assign values to function parameters.** You want to get details about each company based on the company code. Assign the source field that contains company code to the scalar input parameter CompanyCodeID.

   Based on the company code, the SAP function gets details about each company and company address. The results of the function are the scalar output parameters CompanyCode_Detail and CompanyCode_Address. You need variables to hold the results of the function. You create a variable called VAR1 and assign it to CompanyCode_Detail. Create another variable and assign it CompanyCode_Address.
8. **Generate the ABAP program.** When you generate the ABAP program, the Designer generates a CALL FUNCTION statement to call the SAP function.

Importing SAP Functions

Import the SAP functions in the Source Analyzer before you insert the functions in the ABAP program flow. After you import an SAP function, use the function in any mapping in the open folder. If you want to use an SAP function in another folder, open that folder and import the function again. If the SAP system is a Unicode system, you can specify characters across several code pages in a single filter in the ABAP program flow dialog box.

To import an SAP function:

1. In the Source Analyzer, click Sources > SAP Functions.
2. Click Import.

   The Import SAP Metadata dialog box appears.
3. Enter the following information to connect to the SAP system:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

4. Enter filter criterion for the function import.

   Use the following filter syntax rules to define filter criterion:

   - Use the percent sign (%) as a wildcard search for multiple characters. For example, to find a function with “DATE” as part of the function name, enter `%DATE%` as the filter criterion.
   - Use an underscore (_) as a wildcard search for single characters.
5. Separate multiple function names or comments with commas.
6. Select one of the following options to apply to the filter criterion:

   - **Name.** Filters functions by function name.
   - **Comment.** Filters functions by function comment.

   Note: When you use the Comments filter option and specify EN as the language, the SAP Metadata dialog box displays the comments next to the function name for all functions that contain comments in English. If the functions contain comments in another language, the SAP Metadata dialog box does not display comments.
7. Click Connect.

   The Designer displays SAP functions according to the filter criterion.
8. Select the function you want to import.
9. Click Add To Import List.
10. To view the list, click View Import List.

11. To remove items from the list, select the item in the Import List dialog box and click Delete.
12. Click Close to close the Import List dialog box.
13. When the Import List is complete, click OK.

    The function and its parameters appear in the SAP Functions dialog box.

Viewing SAP Functions

After you import SAP functions, you can view the function parameters in the SAP Functions dialog box in the Source Analyzer. The parameters of the SAP function are read only. When you insert an SAP function into the ABAP program flow, you assign values or variables to the parameters.

Each SAP function has the following types of parameters:

- **Scalar Input parameters.** Input values to the SAP function. The ABAP program generates code to pass scalar input values to the SAP function.
- **Scalar Output parameters.** Output values from the SAP function. The SAP function returns function outputs in the scalar output parameters. The ABAP program generates code to receive scalar output values from the SAP function.
- **Changing parameters.** SAP function parameters that can require both input and output values. For example, an SAP function may use a scalar input parameter, modify it, and return the modified value as a scalar output parameter.
- **Table parameters.** SAP function parameters that are SAP structures. Table parameters have more than one row.

Inserting SAP Functions in the ABAP Program Flow

After you import an SAP function, use it in the Application Source Qualifier to help you extract source data. You insert SAP functions in the ABAP program flow and assign values to function parameters.

You assign values and variables to SAP function parameters in the ABAP Program Flow dialog box. You first assign values to the scalar input parameters so the SAP function can use them to perform calculations. You then assign variables to scalar output parameters to hold the return values of the function. If the SAP function contains table or changing parameters, you also assign variables to those parameters.

You can configure the Designer to create an output port in the Application Source Qualifier so you can use the values of function parameters later in the mapping.

Note: Error handling options you set in the session properties for the ABAP mappings do not apply to errors returned by SAP functions you use in the ABAP program flow of the mappings.

Configuring SAP Function Parameters in the ABAP Program Flow

You can view read-only fields and assign values to SAP functions in the ABAP Program Flow dialog box. In the ABAP Program Flow dialog box, each SAP function parameter has the following read-only fields:

- **Parameter.** Name of the parameter.

- **Type.** Type of the parameter. The parameter type can be a standard SAP datatype, a user-defined datatype, or a reference to a structure or structure field.
- **Optional.** When selected, the parameter is optional.

Configuring Scalar Input Parameters in the ABAP Program Flow

The SAP function performs calculations with the values or variables you assign to the scalar input parameters. In the ABAP Program Flow dialog box, configure the following fields for scalar input parameters:

- **Value Type.** Type of value for the parameter. Value type can be an ABAP program variable, a constant, or a field from a source table. If the parameter is optional, the value type can also be None.
- **Value.** Value of the parameter. Depending on the value type, the value of the parameter can be an ABAP program variable, a constant, or a field from a source table.

If the scalar input parameter is a variable, you can choose from a list of ABAP program variables you define in the program flow. You can also enter the name of a new variable. The Designer creates the variable if you enter the name of a new variable in the Value field. When the scalar input parameter is a variable, the Designer generates ABAP statements to assign values to the variables before the CALL FUNCTION statement.

Some scalar input parameters might also be structures. The Designer detects the Value Type for each field of the structure so you do not need to enter the value types.

Configuring Scalar Output, Changing, and Table Parameters in the ABAP Program Flow

The SAP function stores the result of the calculations in the variables you assign to the scalar output parameters. If you assigned a variable to a scalar output, changing, or table parameter, you cannot assign the same variable to another function parameter, except for scalar input parameters. You configure the following fields for scalar output, changing, and table parameters:

- **Variable.** ABAP program variable that holds the value of the parameter.
- **SQ Port.** Select to create an output port in the Application Source Qualifier from the function parameter.

You can create output ports from table parameters only if the SAP function is the last object in the ABAP program flow. The Designer generates a loop to create output ports from table parameters in the ABAP program. An SAP function might have several table parameters. However, you can only create output ports from fields in the same table parameter.

Note: If you choose to create output ports from a table parameter, but you later move the SAP function so that it is no longer the last object in the ABAP program flow, the Designer does not create output ports from the table parameter.

Steps for Inserting an SAP Function in the ABAP Program Flow

Use the following procedure to insert an SAP function.

To insert an SAP function:

1. In the ABAP Program Flow dialog box, click Insert Function.
2. Choose the SAP function you want to insert and click OK.
3. From the Scalar Input tab, assign a value type and value for scalar input parameters.

You can choose from variables already defined in the program flow. The Designer shows a list of variables that has matching datatype, precision, and scale with the parameter. You can also click in the value field and enter the name of a new variable. The Designer creates the new variable when you enter the new variable name.

4. From the Scalar Output tab, assign variables for scalar output parameters.
5. Select SQ Port if you want the Designer to create an output port for the import parameter.
6. From the Changing tab, assign variables for changing parameters.

   You do not need to assign variables for optional changing parameters.
7. Select SQ Port if you want the Designer to create an output port for the changing parameter.
8. Expand tables in the Table tab and assign values to the table parameters.
9. Click Validate.

   The Designer verifies that you assigned a variable or value for all required parameters.
10. Click Validate All.

    The Designer checks the placement of the SAP function in the ABAP program flow.
11. Click OK.

Validating SAP Functions in the ABAP Program Flow

When you click Validate, the Designer verifies that you assigned a variable or a value for each required parameter. When you click Validate All, the Designer checks the placement of the SAP function in the ABAP program flow. The rules for inserting SAP functions depends on the SQL type you generate.

Rules for Inserting SAP Functions in the ABAP Program Flow

You can insert an SAP function before the first source table or after the last source table in the program flow. If you use a nested loop to join tables, you can also insert an SAP function between source tables. Be aware of where the Designer inserts the SAP function in the ABAP program to generate the ABAP program successfully.

The following table shows the rules for inserting an SAP function:

SQL Type	Rules
Exec SQL	- You can insert SAP functions before the first source or after the last source. You cannot insert SAP functions between sources in the program flow. - If you insert an SAP function before the first source, the Designer calls the function before the exec statement. - If you insert an SAP function after the last source, the Designer inserts the function after the FORM WRITE_DSQNAME_TO_FILE statement.
ABAP join syntax	- You can insert SAP functions before the first source or after the last source. You cannot insert functions between sources in the program flow. - If you insert an SAP function before the first source, the Designer calls the function before the select statement. - If you insert an SAP function after the last source, the Designer inserts the function after the WHERE clause.
Open SQL (nested loop)	- You can insert SAP functions between sources in the program flow. The Designer inserts SAP functions between the select statements. You can insert SAP functions before the first source or after the last source. - If you insert an SAP function before the first source, the Designer calls the function before the first select statement. - If you insert an SAP function after the last source, the Designer inserts the function after the last WHERE clause.

CHAPTER 9

Application Source Qualifier for SAP ECC Sources

This chapter includes the following topics:

- Application Source Qualifier for SAP ECC Sources Overview, 103
- Generating an ABAP Program, 104
- Working with ABAP Program Flow, 106
- Joining Source Data, 106
- Creating an ABAP Code Block, 111
- Creating ABAP Program Variables, 112
- Entering a Source Filter, 115
- Using Mapping Variables and Parameters, 118
- Working with IDoc Sources, 119
- Configuring an Application Source Qualifier, 121
- Troubleshooting an Application Source Qualifier for SAP ECC Sources, 121

Application Source Qualifier for SAP ECC Sources Overview

When you add an SAP ECC source definition to a mapping, connect it to an Application Source Qualifier transformation. The Application Source Qualifier represents the record set queried from the SAP ECC source when you run a session.

When you complete the mapping, generate and install the ABAP program that the SAP application server uses to extract source data.

The Designer generates an ABAP program based on the properties in the source definition and the Application Source Qualifier. The Designer can generate open SQL, exec SQL, or ABAP join syntax. You can also set tracing level for session processing.

In the Application Source Qualifier, you can customize the ABAP program in the ABAP Program Flow dialog box. The ABAP Program Flow dialog box shows the order that the ABAP program processes objects. You can configure properties such as filter condition and join condition in the ABAP Program Flow dialog box.

In the ABAP Program Flow dialog box, use static or dynamic filters to specify how the ABAP program selects rows. You can also add more functionality to the ABAP program by adding ABAP code blocks. You can create variables and import SAP functions to use in the ABAP code or filter condition.

If you join multiple sources with one Application Source Qualifier, you can specify how the ABAP program joins the source tables. You can also specify the order that the ABAP program selects the source tables.

Generating an ABAP Program

After you design a mapping, generate an ABAP program to extract the data from the SAP system. The Designer uses the following generation modes to generate the ABAP program:

- Open SQL
- Exec SQL
- ABAP join syntax

You select the ABAP generation mode on the Properties tab of the Application Source Qualifier. To generate the ABAP program using ABAP join syntax, clear the Exec SQL and Force Nested Loop options.

Available ABAP Generation Mode

The available ABAP generation mode depends on the condition of the mapping and the SAP system. The following table shows the available ABAP generation modes for mappings:

Condition	Available SQL Generation Mode
Mapping contains pool or cluster tables.	Open SQL
Mapping contains hierarchies *and* related detail tables.	Open SQL
Mapping contains only transparent tables.	ABAP join syntax Open SQL Exec SQL
Mapping contains IDocs.	ABAP join syntax Open SQL
Mapping contains only hierarchies.	None

In the Application Source Qualifier, the Designer does not check whether you selected the correct ABAP generation mode for the mapping. When you connect to an SAP system and generate the ABAP program, the Designer validates that the selected ABAP generation mode matches the condition of the mapping and the SAP system version.

A hierarchy is a structure of metadata and is not accessible through SQL. The Designer does not generate an ABAP program to extract data from hierarchies. The PowerCenter Integration Service makes a remote function call to the application server to extract the hierarchy metadata.

Generating Open SQL

Open SQL is proprietary to SAP and is sometimes referred to as SAP SQL. Open SQL extracts data from buffers on the application server. When you generate the ABAP program with open SQL, the Designer uses a SELECT statement to select data. You can generate the ABAP program using open SQL for all mappings.

The following sample shows an open SQL statement that the Designer generated:

```
select MANDT KVEWE KAPPL KSCHL KOZGF DATVO DTVOB
into (T685-MANDT,T685-KVEWE,T685-KAPPL,T685-KSCHL,T685-KOZGF,
T685-DATVO,T685-DTVOB)
from T685 where [...].
endselect.
```

When you join several sources in one Application Source Qualifier, open SQL uses a nested loop to select data. The Designer issues multiple SELECT statements and then generates WHERE clauses for the join conditions within the nested loop.

To select Open SQL:

1. In the Application Source Qualifier transformation, click the Properties tab.
2. Select Force Nested Loop.

Generating Exec SQL

Exec SQL, or native SQL, is similar to standard SQL. Use the exec SQL option if the mapping contains only transparent tables or database views. The application server passes exec SQL requests directly to the database to execute. Exec SQL extracts data directly from the tables on the database server.

Although exec SQL can improve PowerCenter session performance, it may decrease performance of the SAP system. Extracting directly from the database may result in data inconsistencies due to application server buffering. Consult the SAP administrator before using the Exec SQL option.

The following sample shows an exec SQL statement:

```
exec sql [...]
  SELECT T685.MANDT, T685.KVEWE, T685.KAPPL, T685.KSCHL, T685.KOZGF,
  T685.DATVO, T685.DTVOB
  INTO
  :T685-MANDTT685-KVEWE, :T685-KAPPL, :T685-KSCHL, :T685-KOZGF,
  :T685-DATVO, :T685-DTVOB
  FROM
  T685
  where [...]
endexec.
```

Note: Exec SQL is not available for pool or cluster tables, hierarchies, or IDocs.

To select Exec SQL:

1. In the Application Source Qualifier, click the Properties tab.
2. Select Exec SQL.

Generating ABAP Join Syntax

ABAP join syntax is available for transparent tables and IDocs on SAP 4.x systems. When you have multiple sources connected to the same Application Source Qualifier, the ABAP program can select the tables using ABAP join syntax. ABAP join syntax uses an INNER JOIN or an OUTER JOIN statement to select multiple source tables. ABAP join syntax extracts data directly from the tables on the database server.

To use ABAP join syntax to generate the ABAP program, clear both the Force Nested Loop option and the Exec SQL option in the Application Source Qualifier transformation.

Working with ABAP Program Flow

The ABAP Program Flow dialog box in the Application Source Qualifier shows the order of objects in the ABAP program and how you can customize the program.

The ABAP program selects tables and objects according to the order in the program flow. Select a table in the program flow to configure the filter condition, join type, and join condition for the selected table. You can complete the following tasks in the ABAP Program Flow dialog box:

- **Change the order of the program flow.** Use the up and down arrows to arrange the order of objects in the program flow.
- **Insert SAP functions.** You can insert SAP functions in the program flow after you have imported them in the Source Analyzer.
- **Create and insert ABAP code blocks in the program flow.** You can add new functionality to the program flow by inserting more blocks of ABAP code. Use ABAP program variables in the code block.
- **Create ABAP program variables.** You can create ABAP program variables to represent values in the ABAP program. The ABAP program variables can also represent SAP system variables.
- **Filter data using dynamic or static filters.** Use dynamic or static filters to reduce the rows that the ABAP program selects. Use ABAP program variables with static filters.
- **Override the default join type and join condition.** If you use multiple sources in the mapping, you can join them with one Application Source Qualifier. You can choose how the ABAP program joins the sources.

In the Mapping Designer, you can also configure the source definition to override the default query with a Select Single or Select Distinct statement.

Validating the ABAP Program Flow

You can validate the ABAP program flow by clicking the Validate button or the Validate All button. When you click the Validate button, the Designer validates the syntax for the selected tab. You can validate each program object separately by using the Validate button.

When you click the Validate All button, the Designer validates all the objects in the program flow. The Designer also validates the placement of all the objects in the program flow.

If you add a new SAP ECC table source to the Application Source Qualifier or remove an existing SAP ECC table source from the Application Source Qualifier, the ABAP program flow rearranges the order of the objects in alphabetical order. You need to update the order of the objects in the ABAP program flow manually.

Joining Source Data

If a mapping uses multiple SAP ECC sources, use one Application Source Qualifier to join the sources. The sources you join in an Application Source Qualifier must be accessible from the same application server, and they must have primary-foreign key relationships. To join tables, link the columns to one Application Source Qualifier.

When you join sources, the ABAP program can perform an inner join or an outer join. The Designer generates a left outer join in the ABAP program if you select outer join in the ABAP Program Flow dialog box.

If you generate the ABAP program using exec SQL, the ABAP program can only perform an inner join. If you generate the ABAP program using open SQL or ABAP join syntax, you can select the join type in the ABAP Program Flow dialog box.

When you join sources, the Designer uses primary-foreign key relationships to determine the default join condition and, with open SQL, the default join order.

You can enter the join condition in the ABAP Program Flow dialog box.

Note: The ABAP Program Flow dialog box displays the join order in the mapping. You can change the join order by moving the program flow objects with the up and down arrows. Do not specify the join order by using the $Source_Join_Order attribute in the join condition.

Joining Sources Using Open SQL

When the Designer generates the ABAP program using open SQL, the ABAP program can perform an inner join or an outer join.

When you use open SQL, the Designer issues multiple SELECT statements. The Designer generates the WHERE clause for the join condition within the nested loop. For example, after the Designer issues the second SELECT statement, it generates the WHERE clause to join the second table with the first table.

The following sample ABAP program is generated using open SQL to join two transparent tables with an inner join. The join order is KONH, KONP. After the Designer generates the SELECT statement for KONH, it generates the WHERE clause to join KONH with KONP:

```
select KNUMH MANDT [...] LICDT
into (KONH-KNUMH,KONH-MANDT,[...] KONH-LICDT)
from KONH
where
(KONH_clause)
order by MANDT KNUMH ERNAM .
  select MANDT KNUMH KOPOS [...] VBEWA
  into (KONP-MANDT,KONP-KNUMH,KONP-KOPOS,[...] KONP-VBEWA)
  from KONP
  where
  KNUMH = KONH-KNUMH and
  (KONP_clause)
  order by MANDT KNUMH KOPOS .
  endselect. [...]
```

The following sample ABAP program is generated using open SQL to join KONH and KONP with an outer join:

```
select KNUMH MANDT [...] LICDT
into (KONH-KNUMH,KONH-MANDT, [...], KONH-LICDT)
from KONH
where
(KONH_clause)
order by MANDT KNUMH ERNAM .
  select MANDT KNUMH KOPOS [...] VBEWA
  into (KONP-MANDT,KONP-KNUMH,KONP-KOPOS,[...] KONP-VBEWA)
  from KONP
  where
  KNUMH = KONH-KNUMH and
  (KONP_clause)
  order by MANDT KNUMH KOPOS . [...]
  endselect.
  if sy-subrc <> 0.
    perform move_columns_to_output changing output.
    perform terminate_output changing output.
  endif.
endselect. [...]
```

Joining Sources Using Exec SQL

Exec SQL joins tables much like standard SQL. Exec SQL selects all tables with one SELECT statement. The Designer generates the WHERE clause for the join condition after the SELECT statement. When you join tables using exec SQL, the ABAP program performs an inner join.

The following sample exec SQL is generated to join the same transparent tables (KOHN, KOHM, and KONP) as above:

```
exec sql [...]
  SELECT KONH.MANDT, [...], KONM.KOPOS, [...], KONP.MANDT, [...]
  INTO [...]
    FROM
  KONP, KONH, KONM
  where
  KONP.MANDT = :client_var and
  KONH.MANDT = :client_var and
  KONH.MANDT = KONP.MANDT and
  KONM.MANDT = :client_var and
  KONM.KNUMH = KONP.KNUMH
endexec.
```

Joining Sources Using ABAP Join Syntax

ABAP join syntax is available for transparent tables and IDocs on SAP 4.x systems. When you generate the ABAP program using ABAP join syntax, the ABAP program can perform either an inner join or an outer join. You choose the join type in the ABAP Program Flow dialog box. Each table in the ABAP program flow must have a key relationship with at least one table above it. If a relationship does not exist between the tables you want to join, you can specify a join condition in the ABAP Program Flow dialog box.

For example, you have two tables, KONP and KONH. You generate the ABAP program with ABAP join syntax and choose inner join for the join type. The following sample ABAP program is generated to join KONP and KONM using an inner join:

```
SELECT KONH~MANDT [...]
INTO (KONH-MANDT, [...]
FROM KONP
INNER JOIN KONH
ON KONH~KNUMH = KONP~KNUMH
WHERE
(KONP_clause) and
(KONH_clause)
ORDER BY KONP~MANDT KONP~KNUMH KONP~KOPOS
KONH~MANDT KONH~KNUMH KONH~ERNAM .
endselect.
```

You can also choose outer join for the join type. The following sample of the ABAP program is generated to join KONP and KONH using an outer join:

```
SELECT KONH~MANDT [...]
INTO (KONH-MANDT, [...]
FROM KONP
LEFT OUTER JOIN KONH
ON KONH~KNUMH = KONP~KNUMH
WHERE
(KONP_clause) and
(KONH_clause)
ORDER BY KONP~MANDT KONP~KNUMH KONP~KOPOS
KONH~MANDT KONH~KNUMH KONH~ERNAM .
endselect.
```

To use ABAP join syntax, clear both the Force Nested Loop option and the Exec SQL option in the Application Source Qualifier transformation. You cannot use ABAP join syntax to join hierarchies.

When you have a mapping with inner/outer joins between multiple SAP ECC tables and you use a dynamic filter on an outer join table, the session fails. This occurs due to ABAP join restrictions on certain combinations of inner and outer joins between SAP tables. SAP produces the following error message:

```
[CMRCV: 18 Illegal access to the right-hand table in a LEFT OUTER JOIN].
```

Selecting a Join Type

Use the following procedure to select a join type.

To select a join type:

1. Edit an Application Source Qualifier transformation and select the Properties tab.
2. Select how you want to generate the ABAP program: Exec SQL, open SQL (Force Nested Loop), or ABAP join syntax. To select ABAP join syntax, clear both Exec SQL and Force Nested Loop.
3. Click the right corner of the Program Flow field to open the ABAP Program Flow dialog box.
4. Select the Join Type tab in the ABAP Program Flow dialog box.
5. Select a table from Program Flow and choose the join type (inner or outer).

   If you generate the ABAP program using exec SQL, only choose inner join in the Join Type tab.
6. Select the sources you want to join in Source(s) to Join to.
7. If necessary, change the order of the program flow by clicking the up and down arrows.
8. Click Validate.

   The Designer validates that a key relationship exists between the sources you want to join.
9. Click OK.

   Note: You cannot use Select Single in source tables when you use ABAP join syntax. When you generate the ABAP program, the Designer returns an error message if you use Select Single with ABAP join syntax.

Using Multiple Outer Joins

When you use an outer join, the Designer generates a LEFT OUTER JOIN statement in the ABAP program. You can use more than one outer join in the ABAP program flow. However, the tables you can join using outer join depends on how you join other tables in the program flow.

For example, you have three tables, KONH, KONP, and KONM in the ABAP program flow. You join KONP with KONH using an outer join. When you select KONM in the program flow, you cannot join KONM with KONP using an outer join. If you join KONM with KONP using an outer join, the Designer returns an error message when you generate the ABAP program.

In this example, you can only choose KONH because you already joined KONP with KONH using an outer join. When you join KONP and KONH with an outer join, the ABAP program selects rows that exist in both KONP and KONH, and it discards rows that only exist in KONP.

The same outer join concept applies to an ABAP program with any number of objects.

Joining Tables and Hierarchies

You can join a hierarchy and the detail table in an Application Source Qualifier to extract table data along with hierarchy metadata. A hierarchy definition can only appear as the first object in the ABAP program flow.

When you join a hierarchy and a detail table in an Application Source Qualifier, the Designer generates open SQL to extract table data. Because the hierarchy is a structure of metadata, you cannot access it through SQL.

When you run a workflow, the ABAP program uses SQL to extract data from the detail table. The PowerCenter Integration Service calls an RFC function on the application server to extract the hierarchy metadata. The PowerCenter Integration Service then joins the hierarchy and detail data.

Joining Tables and IDoc

You can join an IDoc with one or more tables in an Application Source Qualifier. An IDoc can only appear as the first object in the program flow. When you join an IDoc with more than one table, each table must have a key relationship with at least one other table above it. If a table has a key relationship with an IDoc source but has no relationship with any other table in the program flow, the Designer returns an error message when you generate the ABAP program.

When you generate SQL using ABAP join syntax, you cannot override the default join conditions between an IDoc and a table in the Join Condition tab. To override the default join condition, specify the join condition as a static filter condition for the IDoc. Enter the condition in the static filter and clear the IDoc source check box in the Join Type tab.

Note: You cannot override the default join condition between an IDoc and a table if the table is the second object in the ABAP program flow.

Specifying Join Conditions

You can specify the join condition for each table you join in the Application Source Qualifier. If you do not specify a join condition, the Designer uses one of the following conditions as a default join condition:

- Default join condition imported from the SAP system.
- Key relationship you entered in the Source Analyzer. PowerCenter converts the key relationship to a join condition when it runs the session.

The ABAP Program Flow dialog box does not display the default join condition. When you enter a join condition in the ABAP Program Flow dialog box, you override the default join condition.

You can also create a relationship in the join condition for tables that do not have a relationship. Creating a relationship in the join condition is similar to specifying a key relationship in the Source Analyzer. However, the key relationship you specify in the Source Analyzer applies to all the mappings in the folder. The relationship you specify in the join condition applies only to the mapping. You can then join the tables using an inner join or an outer join.

The join condition must follow ABAP syntax. The Designer validates the syntax of the join condition in the ABAP Program Flow dialog box.

Rules for Specifying Join Conditions

Use the following rules to specify join conditions:

- Join conditions must follow ABAP syntax. Specify the join condition using the following syntax:

  ```
  TABLE_NAME2-FIELD_NAME = TABLE_NAME1-FIELD_NAME
  ```

- If you want to enter two statements in the join condition, separate the statements with a semicolon (;) or AND.
- The syntax for join conditions is the same regardless of whether you configure the Application Source Qualifier to use open SQL or exec SQL.
- If you do not include the table name with the field name in a join condition, the session might fail due to ambiguous references.
- If you create a join condition that joins fields in the same table, the session might result in incorrect data.

To specify a join condition:

1. In the ABAP Program Flow dialog box, select the Join Condition tab.
2. Select the table in Program Flow to edit the join condition.

   Double-click a table name in Source Level Attributes to view a list of field names.
3. Double-click the field name to enter it in the join condition.
4. Enter the join condition.
5. Click Validate to validate the syntax of the join condition.
6. Click OK.

Creating an ABAP Code Block

You can add more functionality in the ABAP program by adding more ABAP code in the program flow. An ABAP code block is additional ABAP code you can add to the ABAP program. In the ABAP code block, use source fields and ABAP program variables defined for the Application Source Qualifier to customize the ABAP program.

The ABAP code block should follow ABAP syntax. Comments in the ABAP code block must start with an asterisk (*). The ABAP program flow represents the order that the ABAP program selects tables and objects. Use source fields or values that are placed above the code block.

Note: You cannot use an ABAP code block if the ABAP program flow contains a hierarchy and no other sources.

To create an ABAP code block:

1. In the ABAP Program Flow dialog box, click New ABAP Block.
2. Enter the name for the new ABAP code block and click OK.

   The name of the ABAP code block cannot exceed 28 characters.
3. Expand the source table name and the Variables folder to view the source fields and variable names.
4. Double-click the source field or the variable name to enter it in the ABAP code block.
5. Write the code block.
6. Click OK to save the ABAP code block.

Rules for Inserting the ABAP Code Block

Generally, you can create a new ABAP code block and insert it before the first source table or after the last source table in the program flow. If you use a nested loop to join tables, you can also insert an ABAP code block between source tables. The SQL generation mode you select in the Application Source Qualifier determines the position you can insert the ABAP code block in the program flow.

The following table lists the rules for inserting an ABAP code block:

Generation Mode	Rules
Exec SQL	- You can insert code blocks before the first source or after the last source. You cannot insert code blocks between sources in the program flow. - If you insert a code block before the first source, the Designer inserts the code block before the exec statement. - If you insert a code block after the last source, the Designer inserts the code block after the FORM WRITE_DSQNAME_TO_FILE statement.
ABAP join syntax	- You can insert code blocks before the first source or after the last source. You cannot insert code blocks between sources in the program flow. - If you insert a code block before the first source, the Designer inserts the code block before the select statement. - If you insert a code block after the last source, the Designer inserts the code block after the where clause.
Open SQL (nested loop)	- You can insert code blocks between sources in the program flow. The Designer inserts code blocks between the select statements. You can insert code blocks before the first source or after the last source. - If you insert a code block before the first source, the Designer inserts the code block before the first select statement. - If you insert a code block after the last source, the Designer inserts the code block after the last where clause.

The following rules apply when you create a code block to specify the initial value of ABAP program variables:

- If you create a code block to initialize variables used in a filter condition, insert the code block before the first source.
- If you create a code block to initialize variables used in data movement, insert the code block after the last source.

Validating the ABAP Code Block

To validate the placement of the code block, click the Validate All button in the ABAP Program Flow dialog box. You cannot generate the ABAP program if the placement of the ABAP code block does not follow the rules for inserting an ABAP code block. The SAP system validates the syntax of the code block when you install the ABAP program.

Creating ABAP Program Variables

When you write ABAP code blocks or static filter conditions, use variables to represent SAP structures, fields in SAP structures, or values in the ABAP program. You can create the following types of variables in the ABAP Program Flow dialog box:

- **Structure and structure field variables.** Represents structures and fields in structures defined in the SAP system.
- **ABAP type variable.** Represents values in the ABAP program.

After you create the variable, use the variable as many times as you need in the ABAP program flow. The Designer generates a data statement to declare the variable when you generate the ABAP program. The SAP system validates the syntax used with variables in the ABAP code blocks or filter conditions when you install the ABAP program.

When you assign an ABAP program variable to another variable in an ABAP code block, be sure that the variables have the same precision and datatype.

Note: You cannot use an ABAP program variable if the ABAP program flow contains a hierarchy and no other sources.

Naming Convention

When you create a variable in the ABAP program flow, be aware of the following rules:

- Variable names cannot contain reserved words and special characters such as the pound sign (#).
- Variable names are not case sensitive.
- The maximum length for variable names is 25 characters.
- The maximum length for the definition of structure or structure field variables is 30 characters.

When you generate and install the ABAP program, the SAP system validates the ABAP program variable according to the following rules:

- The maximum length for the initial value of a variable is 40 characters.
- Variable names cannot contain any SAP datatype names, table, structure, and structure field names, or any ABAP keyword.

Creating Structure and Structure Field Variables

Structures are virtual tables defined in the SAP dictionary. You can create a structure variable and a structure field variable.

When you create a structure variable, the Designer generates a data statement in the ABAP program to declare the variable. For example, you create a structure variable named struc1 to represent an SAP structure called AENVS. The Designer generates the following statement in the ABAP program to declare struc1:

```
data: struc1 like AENVS occurs 5 with header line.
```

The structure AENVS has a field called EXIST. You can create a structure field variable called field1 to represent this field. The Designer generates the following statement in the ABAP program to declare field1:

```
data: FIELD1 like AENVS-EXIST.
```

After you create the structure field variable, you specify its initial value in an ABAP code block.

Creating a Structure Variable

Create a structure variable to represent any structure in the SAP system. A structure can contain many fields.

To create a structure variable:

1. In the ABAP Program Flow dialog box, click Variables to open the ABAP Program Variables dialog box.
2. Click Add.
3. Enter a name for the new ABAP program variable.
4. Choose Structure for the Variable Category and click OK.
5. Enter a definition for a structure variable.

The variable definition must be the name of an existing structure in the SAP system. You cannot specify an initial value for a structure variable.

6. Click OK.

Creating a Structure Field Variable

Create a structure field variable to represent a field in a structure.

To create a structure field variable:

1. In the ABAP Program Flow dialog box, click Variables to open the ABAP Program Variables dialog box.
2. Click Add.
3. Enter a name for the new structure field variable.
4. Choose Structure Field for the Variable Category and click OK.
5. Enter a definition for a structure field variable.

   Enter a variable definition that is an existing field in a structure defined in the SAP system. Use the following format:

   ```
   STRUCTURE_NAME-FIELD_NAME
   ```

6. Click OK.

Creating ABAP Type Variables

An ABAP type variable can represent a value in the ABAP program. After you create an ABAP type variable, you can specify the datatype, precision, scale, and initial value. You can specify any SAP datatype for the ABAP type variable. Some datatypes, such as floating point, have fixed precision and scale so you cannot specify these fields.

If you want to use the value of the variable in other transformations or in the target definition, make the ABAP type variable an output port in the Application Source Qualifier.

For each ABAP type variable, the Designer generates the following data statement in the ABAP program to declare the variable:

```
data: Variable_Name(precision) type ABAP_Type Decimals Scale.
Variable_Name = 'Initial_Value'.
```

For example, you create an ABAP type variable named var1 to represent a currency value. If you specify zero for the initial value and one for precision and scale, the Designer generates the following statement in the ABAP program:

```
data: var1(1) type P decimals 1.
var1 = '0'.
```

The ABAP statement does not include precision and scale if you choose a datatype that has fixed precision and scale.

To create an ABAP type variable:

1. In the ABAP Program Flow dialog box, click Variables to open the ABAP Program Variables dialog box.
2. Click Add.
3. Enter a name for the new ABAP program variable.
4. Choose ABAP Type for the Variable Category and click OK.
5. Choose a datatype for Type from the list.
6. Enter the precision and scale for the variable.

You cannot edit these fields if you choose a datatype that has fixed precision and scale.

If you select DEC, the precision must be 14 or less.

7. Enter the initial value for the variable.
8. Select SQ Port if you want the Designer to create an output port for this variable.
9. Click OK.

Viewing ABAP Program Variables

You can view existing ABAP program variables by clicking Variables in the ABAP Program Flow dialog box. You can change the type, precision, and other fields of each existing ABAP program variable. You can also delete existing ABAP program variables in the ABAP Program Variables dialog box. However, you cannot delete variables still being used in the ABAP program.

Using SAP System Variables

When you write ABAP code blocks or static filter conditions, use current system information such as system date and system user name. SAP system variables such as SY-DATUM and SY-UNAME can provide current system information. You need to first create a structure field variable to represent a system variable.

For example, you want to create a structure field variable named sysvar1 to represent SY-DATUM. The Designer generates the following statement in the ABAP program:

```
data: SYSVAR1 like sy-datum.
```

You can then use sysvar1 to represent SY-DATUM in the ABAP program flow. You cannot enter an initial value for system variables in the ABAP Program Variables dialog box. You can assign initial values for system variables in the ABAP code block.

Entering a Source Filter

For each source table connected to the Application Source Qualifier, use a dynamic or a static filter to reduce the number of rows the ABAP program returns. Use constants in a dynamic filter to select rows. Use constants and variables in a static filter to select rows.

You specify the type of filter and the filter condition for each source table in the ABAP Program Flow dialog box. The type of filter you use is independent of how the Designer generates the ABAP program. You cannot use hierarchy columns in a filter condition.

The PowerCenter Integration Service processes filter conditions differently for static and dynamic filters. The following table describes the differences between static and dynamic filter handling:

Filter Handling	Dynamic Filter	Static Filter
Filter condition	Use the following on the right side of the filter condition: - Constants - User-defined mapping variables and parameters - Built-in mapping variables	Use the following on the right side of the filter condition: - Constants - User-defined mapping variables and parameters - ABAP program variables
Filter stored	Designer stores the filter condition in repository.	Designer writes the filter condition to the ABAP program.
Filter processing	When you run the workflow, the PowerCenter Integration Service moves the filter from the repository to the SAP system. The ABAP program calls a function to process the filter.	When you run the workflow, the SAP server processes the filter directly from the ABAP program.
Session properties	You can override the filter condition in session properties.	You cannot override the filter condition in session properties.
Session log file or Log Events window in the Workflow Monitor	PowerCenter Integration Service includes the filter syntax in log events.	PowerCenter Integration Service does not include the filter syntax in log events.

When you specify a port in a dynamic filter condition, static filter condition, or join override, link that port from the Application Source Qualifier to the target or next transformation in the mapping. Do the same if you specify a dynamic filter condition at the session level.

If the SAP system is a Unicode system, the Designer is connected to a Unicode repository, and you enter a source filter that contains ISO 8859-1 or multibyte characters, generate a local copy of the ABAP program and upload the generated file into the SAP system.

Note: You cannot use a source filter if the ABAP program flow contains a hierarchy and no other sources.

Using Dynamic Filters

A dynamic filter uses the constant in the filter condition to select rows. When you use a dynamic filter, the Designer stores filter information in the repository. The dynamic filter condition is not part of the ABAP program.

When you run a workflow, the PowerCenter Integration Service moves the dynamic filter condition from the repository to the SAP system and the ABAP program applies it to the rows read from the source tables. You can override dynamic filter conditions in the session properties.

The PowerCenter Integration Service writes the dynamic filter syntax to the session log.

Note: You cannot use dynamic filters on IDoc source definitions in the ABAP program flow. You cannot override a dynamic filter if you use Exec SQL to generate the ABAP program. When you use Exec SQL, the Designer applies the dynamic filter as a static condition in the select statement during ABAP code generation.

To enter a dynamic filter:

1. In the ABAP Program Flow dialog box, select the table you want to filter.

2. Select the Dynamic Filter tab.

   Double-click the table name in Source Level Attributes to view a list of field names.

3. Double-click the field name to enter it in the filter condition.

   Source Level Attributes do not show ABAP program variables or other tables in the program flow because you cannot use them in a dynamic filter condition.

4. Finish entering the filter condition.
5. Click Validate to validate the syntax of the filter condition. Then, click OK.

Using Static Filters

A static filter uses constants and variables in the filter condition to select rows. When you use a static filter, the Designer writes the filter information in the SQL portion of the ABAP program as a WHERE clause. The ABAP program executes the filter condition with the SELECT statement. Because the static filter is part of the ABAP program, you cannot override it in the session properties. The PowerCenter Integration Service does not include the static filter condition in log events.

Syntax Rules for Entering Filters

When you validate filter conditions, the Designer performs validation according to the following rules:

- When you enter filter conditions, use the following syntax:

  ```
  table_name-field_name [=, >=, <=, <, >, < >] 'value'
  ```

  - If you use variables in a static filter condition, the filter condition must be in the following format:

    ```
    table_name-field_name [=, >=, <=, <, >, < >] :variable_name
    ```

  - If you use fields from another table in a static filter condition, the filter condition must be in the following format:

    ```
    table_name1-field_name [=, >=, <=, <, >, < >] table_name2-field_name
    ```

  Note: The left side of the filter condition must be a field from the table you selected in the ABAP program flow.

- Enclose constants on the right side of the condition in single quotes.
- If you use an ABAP program variable in a static filter condition, a colon (:) must precede the variable name.
- Filter conditions for character strings must match the full precision of the column if the condition is numeric. For example, if you want to filter records from CSKS where KOSTL is greater than 4,000, enter the following condition:

  ```
  KOSTL > '0000004000'
  ```

- All valid SAP operators can be used in filter conditions.
- Use a semicolon (;) or boolean operators (such as AND) to separate multiple conditions.
- Always leave a space after every token except commas.

Specifying Filter Conditions for NUMC Columns

The following issues apply when you specify filter conditions for NUMC columns:

- The PowerCenter Integration Service ignores negative comparisons for NUMC columns in a filter condition. The PowerCenter Integration Service treats all comparisons as positive.

- Because SAP does not store signs with NUMC data, do not use negative filter conditions in the Application Source Qualifier for NUMC columns. SAP does not recognize the negative conditions and treats all comparisons for NUMC columns as positive.

To enter a static filter:

1. In the ABAP Program Flow dialog box, select the table you want to filter, and select the Static Filter tab.
2. In Source Level Attributes, double-click the table name or the Variables folder to view a list of fields or ABAP program variables.
3. Double-click the field names or variables to enter it into the filter condition. If you use a variable in a static filter condition, enter a colon (:) before the variable name.

   You can also use fields from other tables in the ABAP program flow in the right side of the filter condition.
4. Finish entering the filter condition.
5. Click Validate to validate the syntax of the filter condition and click OK.

Using Mapping Variables and Parameters

In the ABAP program flow, use mapping variables and parameters in the following ABAP program objects: filter conditions, join conditions, and ABAP code blocks. The type of mapping variables and parameters you use depends on how the SAP server processes the ABAP program object.

Use both user-defined and built-in mapping variables and parameters in dynamic filter conditions. The Designer stores dynamic filter conditions in the repository. When you run a workflow, the PowerCenter Integration Service evaluates the variable or parameter and passes the filter from the repository to the SAP system. Then the ABAP program calls a function to process the dynamic filter.

You cannot use built-in mapping variables in static filter conditions because the SAP server processes the static filter directly from the ABAP program. The PowerCenter Integration Service does not pass values from the repository to the SAP system. Similarly, you cannot use built-in mapping variables in join conditions and ABAP code blocks.

The following table shows the types of mapping variables you can use with each object in the ABAP program flow:

ABAP Program Object	Mapping Variable Type
Dynamic filter condition	User-defined and built-in mapping variables.
Static filter condition	User-defined mapping variables only.
Join condition	User-defined mapping variables only.
ABAP code block	User-defined mapping variables only.

Use mapping parameters in all ABAP program objects.

Using Mapping Variables in the ABAP Program Flow

In the ABAP Program Flow dialog box, you use mapping variables in the filter condition, join condition, or ABAP code blocks. To update the value of the mapping variables, use a variable function in the Expression transformation in the mapping.

For example, you want to select source data from a time period that ends at the session start time. You create a mapping variable named $$FROMTIME to represent the beginning of the time period. You enter the following statement in the dynamic filter condition:

```
TABLE_NAME-FIELD_NAME >= $$FROMTIME
```

To update the beginning of the time period for the next session, you set the $$FROMTIME variable to the session start time of the current session. The built-in variable $$$SESSSTARTTIME returns the session start time. In the mapping, you update the $$FROMTIME variable by entering the following statement:

```
SETVARIABLE($$FROMTIME, TO_DATE(SESSSTARTTIME))
```

In an ABAP code block, use a mapping variable as a constant in the right-hand side of comparisons. You cannot modify the mapping variable by assigning a value to it. For example, you cannot assign a value to a mapping variable in the ABAP code block.

Working with SAP Date Formats

Mapping variables that return date time values have the PowerCenter default date format of MM/DD/YYYY HH:MI:SS.US. The SAP date format is YYYYMMDD. When you run a session, the PowerCenter Integration Service converts the date time format if necessary. You do not need to perform any conversion when you specify a date time variable.

Working with IDoc Sources

An IDoc is a hierarchical structure that contains segments. Each IDoc segment has a header and data records. The header contains control information such as creation date and status. The control information is in an SAP structure called EDIDC.

When you add an IDoc source definition to a mapping, connect the IDoc source definition to an Application Source Qualifier. You can connect one IDoc source definition and other SAP tables to an Application Source Qualifier. You cannot connect more than one IDoc source definition to an Application Source Qualifier.

Working with IDoc Sources in the ABAP Program Flow

You can connect one or more tables to an IDoc source definition in the Application Source Qualifier. You can then customize the ABAP program in the ABAP Program Flow dialog box. You cannot have more than one IDoc source definition in the ABAP program flow.

Use the following guidelines when you have an IDoc in the ABAP program flow:

- You cannot generate the ABAP program using Exec SQL.
- The IDoc source definition can only be in the first position in the ABAP program flow.
- You cannot have both an IDoc source definition and a hierarchy in the ABAP program flow.
- You cannot use the dynamic filter in the ABAP Program Flow dialog box to select rows from the IDoc source.

IDoc Control Information

IDoc sources contain control information, such as creation date (CREDAT). When you import the IDoc source definition, the Designer adds the following columns from the control information to the source definition: DOCNUM, STATUS, CREDAT, CRETIM, SNDSAD, and DIRECT. You cannot use any of these columns in any ABAP program flow objects, including static filters, code blocks, and join conditions.

Entering an IDoc Filter

In the ABAP Program Flow dialog box, use a static filter to select rows based on the data records of the IDoc. You cannot use IDoc control information in static filter conditions. To use IDoc header information in a filter condition, use the IDoc filter in the Application Source Qualifier.

In the IDoc filter condition, you select rows based on any field in EDIDC, the table that contains IDoc control information. Use constants, built-in and user-defined mapping variables, and mapping parameters on the right side of the IDoc filter condition.

The Designer processes IDoc filters similar to the way it processes dynamic filters. The Designer stores IDoc filter information in the repository. The IDoc filter condition is not a part of the ABAP program. When you run a workflow, the PowerCenter Integration Service moves the filter condition from the repository to the SAP system and the ABAP program applies it to rows read from the IDoc definition. You can override the IDoc filter condition when you run a workflow.

To enter an IDoc filter:

1. In the Application Source Qualifier, click the Properties tab.
2. Click the right corner of the IDoc Filter field to open the Source Editor dialog box.
3. Expand the IDoc type to view all the header information in the IDoc definition.

   Header information of an IDoc definition is in the format of the SAP structure EDIDC.
4. Double-click a field in EDIDC to enter it in the IDoc filter condition.
5. If you want to use mapping variables and parameters in the IDoc filter condition, click the Variables tab to view a list of mapping variables and parameters.
6. Double-click the mapping variable or parameter name to enter it in the IDoc filter condition.
7. Click Validate to validate the syntax of the IDoc filter.

Validating the IDoc Filter Condition

When you validate IDoc filter conditions, the Designer performs validation according to the following guidelines:

- Use the following syntax:

  ```
  EDIDC-field_name [=, >=, <=, <, >, < >] 'value'
  ```
- Enclose on the right side of the filter condition in single quotes.
- Filter conditions for character strings must match the full precision of the column if the condition is numeric.
- All valid SAP operators can be used in IDoc filter conditions.
- Use a semicolon (;) or boolean operators (such as AND) to separate multiple conditions.
- Always leave a space after every token except commas.

Configuring an Application Source Qualifier

You can use one Application Source Qualifier to join SAP ECC sources. Join sources when to join data from multiple tables or application systems with identical versions.

When the ABAP program extracts from SAP, it stores all data, including dates and numbers, in character buffers. When you configure an Application Source Qualifier, you might want to change some of the dates and number datatypes to string to maintain accuracy during conversion.

Configuring an Application Source Qualifier

After you create an Application Source Qualifier, you can set several configuration options.

To configure an Application Source Qualifier:

1. Edit an Application Source Qualifier.

   The Edit Transformations dialog box appears.
2. On the Ports tab, optionally change numeric or date datatypes to string to maintain accuracy during conversion.
3. On the Properties tab, optionally edit the following properties:

Property	Description
Exec SQL	Generates native SQL to access transparent tables.
Tracing Level	Sets the amount of detail included in the session log when you run a session containing this transformation.
Force Nested Loop	Generates open SQL to access SAP tables.
Program Flow	Customizes the ABAP program with SAP functions, ABAP code blocks, variables, filters, and join conditions.
IDoc Filter	Specifies the filter condition for selecting IDoc source definitions.

4. On the Sources tab, select additional source definitions you want to associate with the Application Source Qualifier.
5. Click OK.

   Note: Do not pass SAP PREC data to the pipeline. You can connect PREC columns to an Application Source Qualifier, but you cannot connect them to other transformations.

Troubleshooting an Application Source Qualifier for SAP ECC Sources

I copied an Application Source Qualifier from an imported mapping and pasted it into another mapping. When I try to install the ABAP program, the program installation fails.

When you create an ABAP mapping, do not use a copy of an Application Source Qualifier from an imported mapping.

CHAPTER 10

Configuring Sessions with SAP ECC Sources and Targets

This chapter includes the following topics:

- Configuring Sessions with SAP ECC Sources and Targets Overview, 123
- Running Stream Mode Sessions, 124
- Running File Mode Sessions, 125
- Accessing Staging Files for ABAP Mappings, 127
- Pipeline Partitioning for SAP ECC Sources, 129
- Session Properties for SAP ECC Targets, 130
- Reject Files, 131
- Session Load Mode, 131
- Configuring an SAP ECC Session , 132
- Troubleshooting Sessions, 132

Configuring Sessions with SAP ECC Sources and Targets Overview

If the mapping requires an ABAP program, select the appropriate reader to run the session. The reader for an SAP session determines the work process under which the SAP application server extracts data. You can run the session in one of the following modes:

- **Stream mode.** Extracts SAP data to buffers. Data from the buffers streams to the PowerCenter Integration Service. Use stream mode if the dataset is small and the execution of data is fast.
- **File mode.** Extracts SAP data to a staging file. Use file mode if the dataset is large.

For a mapping with an ABAP program, you select stream or file mode when you generate and install the ABAP program.

If the mapping contains a hierarchy only, you can run the session in stream mode or file mode. If the mapping contains a hierarchy or IDoc definition, the PowerCenter Integration Service makes a remote function call to extract the hierarchy data.

If the mapping contains a hierarchy definition and the detail table definition in one Application Source Qualifier, the ABAP program that extracts the detail table data also joins the detail data to the hierarchy data

extracted through the remote function call. Select the SAP Staging Reader if the mapping contains both a hierarchy and detail table definition.

Although you can choose SAP Streaming Reader or SAP Staging Reader as the reader type in the session properties, the PowerCenter Integration Service always extracts hierarchy data through RFC.

The following table shows properties for file and stream mode sessions:

Reader Type	Connection File	Protocol	Data Access	Work Process	Sources in Mapping
SAP Staging Reader	`sapnwrfc.ini`	RFC (File Mode)	FTP, SFTP, NFS, Direct	Dialog, background	Table, hierarchy, hierarchy and table
SAP Streaming Reader	`sapnwrfc.ini`	RFC (Stream Mode)	Data stream	Dialog	Table, hierarchy
SAP Streaming Reader	`sapnwrfc.ini`	CPI-C (Deprecated Stream Mode)	Data stream	Dialog	Table, hierarchy

You can configure the session to load data to the SAP tables. The PowerCenter Integration Service sends data to the SAP system as packets. The SAP transport module loads data to the SAP tables either as bulk mode or normal mode based on the sessions configuration. SAP tables include transparent, pool, and cluster tables.

Running Stream Mode Sessions

You can run PowerExchange for SAP NetWeaver sessions in stream mode when the dataset is small and the execution of data is fast. PowerExchange for SAP NetWeaver uses the RFC protocol to run stream mode sessions.

Note: To run ABAP programs that were created in versions earlier than 9.6.0 in stream mode, you can choose to use the CPI-C protocol.

To run a session in stream mode, select SAP Streaming Reader as the source reader type in the session properties. Select SAP Streaming Reader if you installed an ABAP program for the mapping, and you selected stream mode when you generated the ABAP program.

If you have separate application connections for file and stream modes, select the application connection that you configured for stream mode.

When you run a session in stream mode, the installed ABAP program creates buffers on the application server. The program extracts source data and loads it into the buffers. When a buffer fills, the program streams the data to the PowerCenter Integration Service. With this method, the PowerCenter Integration Service can process data when it is received.

Tip: Stream mode sessions require an online connection between PowerCenter and SAP to transfer data in buffers. For this reason, all stream mode sessions use a dialog process. You might want to use streaming when the window of time on the SAP system is large enough to perform both the extraction and the load.

Running File Mode Sessions

You can run PowerExchange for SAP NetWeaver sessions in file mode when the dataset is large.

To run a session in file mode, select the SAP Staging Reader as the source reader type in the session properties. Select SAP Staging Reader if you installed an ABAP program in the Mapping Designer, and you selected file mode when you generated the ABAP program.

If you have separate application connections for file and stream modes, select the application connection that is configured to connect to the SAP system using RFC.

When you run a file mode session, the installed ABAP program creates a staging file on the application server. The program extracts source data and loads it into the file. The PowerCenter Integration Service process is idle while the program extracts data and loads the file. When the file is complete, the PowerCenter Integration Service accesses the file and continues processing the session. The ABAP program then deletes the staging file unless you configure the session to reuse the file.

Tip: File mode sessions do not require an online connection between the PowerCenter Integration Service and SAP while the generated ABAP program is extracting data. For this reason, you can run a file mode session offline using a background process. Choose background processing when the data volume is high and the extraction time exceeds the limit for dialog processes.

Enabling Compressed Data Transfer

In file mode, the ABAP program extracts data from SAP and loads it to the staging file record by record. Similarly, the PowerCenter Integration Service extracts the data from the staging file record by record and writes it to the target.

If you have an SAP source that contains a large volume of data, you need a high number of read and write operations to process the data. As the number of read and write operations increase, the session performance decreases.

To improve the session performance and more efficiently process the data, you can enable compressed data transfer while reading data from SAP through ABAP. You can enable compressed data transfer when you read data from SAP ECC version 5.0 or later. To enable compressed data transfer, add the FileCompressEnable custom property at the session level or at the PowerCenter Integration Service level and set its value to 1.

When you enable compressed data transfer, the ABAP program compresses the source data into blocks and then loads the data to the staging file. The PowerCenter Integration Service then reads the compressed data from the staging file and writes it to the target. By compressing the data, you reduce the number of read and write operations that are required to process the data. You also reduce the amount of disk storage that the staging file needs. Therefore, the session performance improves.

For example, you want to read data from an SAP table that contains 5 million records. You need 5 million write operations for the ABAP program to load the data to the staging file. Similarly, you need 5 million read operations for the PowerCenter Integration Service to extract the data from the staging file and load it to the target. When you enable compressed data transfer, assume that 100 records are compressed and stored in one block. Then, you would need only 50000 write operations and 50000 read operations to process the data. Similarly, you also significantly decrease the disk storage that the staging file needs because the records are compressed.

Note: To enable compression for an ABAP program that was generated earlier than version 9.6.1, you must first upgrade to version 9.6.1, generate and reinstall the ABAP program, and then configure the FileCompressEnable custom property.

Reusing Staging Files

When you run a file mode session, the SAP application server creates a staging file for each Application Source Qualifier in the mapping. By default, the PowerCenter Integration Service deletes the file after reading it. If you run multiple sessions that use identically configured Application Source Qualifiers, you can save the staging file and reuse it in another session. You can also reinitialize the file if the source data has changed.

Use the following session properties when you want to reuse staging files:

- **Persist the Stage File.** The PowerCenter Integration Service checks for the existence and validity of a staging file in the specified staging file directory.
  - If the file exists and is valid, the ABAP program does not run. The PowerCenter Integration Service reads the existing file.
  - If the file does not exist or is invalid, the ABAP program creates a staging file.

  Note: If you use FTP or SFTP, the PowerCenter Integration Service might record a message in the session log indicating that it cannot find the specified file. The PowerCenter Integration Service logs an informational message when it checks for the existence of the file.

  The PowerCenter Integration Service does not delete the staging file after reading the file.

  When the PowerCenter Integration Service checks the validity of a staging file, it verifies that the total length of all projected columns equals the total length of the file. The PowerCenter Integration Service does not verify individual columns, or verify the expected size of the entire file.
- **Reinitialize the Stage File.** The ABAP program extracts and replaces the existing staging file. Use this option when source data has changed and you want to refresh the file. Use this option only in conjunction with the File Persistence option.

The following table describes the PowerCenter Integration Service actions for the File Persistence and File Reinitialize options:

Persist the Stage File	Reinitialize the Stage File	Action
On	Off	ABAP program creates the staging file if it does not exist. If the staging file exists, the ABAP program validates and reuses the file. If validation fails, the ABAP program recreates the file. After the PowerCenter Integration Service reads the file, it remains on the system for reuse.
On	On	ABAP program creates the staging file even if it exists. The file remains on the system for reuse.
Off	Off	PowerCenter Integration Service reads the staging file and deletes the file.

Note: File persistence is not available for hierarchies.

Overriding Filter Conditions

When you persist the staging file, session-level overrides do not apply. For example, if a staging file exists, and you want to apply a one-time session-level override for a filter condition, clear the Persist the Stage File option before you run the session. The SAP application server generates a staging file based on the session-level override. PowerCenter reads and deletes the staging file. This ensures the following actions:

- A new file is generated using the filter condition.
- The next session run based on the same or identical Application Source Qualifier generates a new staging file without the temporary filter condition.

If you do not clear the Persist the Stage File option, the PowerCenter Integration Service validates and uses the existing file. PowerCenter does not pass the filter condition to the SAP application server. If you persist the staging file and reinitialize the staging file, the ABAP program generates and saves a new staging file. The next session uses the filtered file when you want to use the full extract.

Accessing Staging Files for ABAP Mappings

When you run a file mode session, SAP creates the file in the directory specified in the session properties. The session settings determine how PowerCenter accesses the staging files. To run a file mode session, establish access to the staging file as follows:

- **Mode of access.** Determine the mode of access and establish connection to the files.
- **Enable access to the files.** Configure access to staging files on UNIX.
- **Configure a file mode session.** Configure specific session properties for file mode sessions.

Modes of Access

You can access staging files for an SAP session in the following ways:

- File Direct
- NFS Mount
- FTP or SFTP

File Direct

Use File Direct when the file system is shared between the two machines. There are two file direct situations:

- The SAP host and the PowerCenter Integration Service host are on the same machine.
- The SAP host and the PowerCenter Integration Service host are on different machines, but have a common view of the shared file system. Map a drive from the PowerCenter Integration Service to the machine where the staging files reside.

The user accessing the file must be the user that runs the PowerCenter Integration Service. If the SAP system is on Windows, that user must have standard read permissions on the directory where you stage the file.

NFS Mount

Use NFS Mount when the file path and name are different for the SAP system and the PowerCenter Integration Service. Use NFS Mount in the following situations:

- **One host is Windows and the other is UNIX.** Map a drive from the PowerCenter Integration Service to the machine where the staging files reside. The path names map differently between the two platforms.
- **The file system shared between the two hosts are mounted differently.** Map a drive from the PowerCenter Integration Service to the machine where the staging files reside.

The user accessing the file must be the user that runs the PowerCenter Integration Service. If the SAP system is on Windows, that user must have standard read permissions on the directory where you stage the file.

FTP or SFTP

Use FTP or SFTP when the PowerCenter Integration Service accesses the file system through an FTP or SFTP connection in the following situations:

- **The FTP or SFTP server is configured to view the entire file system.** When the PowerCenter Integration Service accesses SAP through FTP or SFTP, the path to the file is identical.
- **The FTP or SFTP server is restricted to a certain directory or directories.** The paths for the staging directory and source directory are different.

Configure an FTP or SFTP connection in the Workflow Manager.

The user who accesses the staging file must be the FTP or SFTP user. If the SAP system is on Windows, that user must have standard read permissions on the directory where you stage the file.

If the PowerCenter Integration Service fails to access the staging file through FTP or SFTP, it logs the error message returned by SAP in the session log. Use transaction ST22 from the SAP client to get more information about the SAP error message.

Enabling Access to Staging Files on UNIX

If the SAP system is on UNIX, by default SAP creates the staging file with read and write access to the owner and the users in the owner group. The owner of the file is generally the SAP administrator. Outside users have no access to the file.

There are several ways to ensure that the user accessing the staging file has proper permissions:

- **Access the file as the SAP administrator.** If you access the file through File Direct or NFS, use the SAP administrator as the user that runs the PowerCenter Integration Service. If you access the file through FTP, use the SAP administrator as the FTP user.
- **Place the user accessing the file in the SAP administrator group.** If you access the file through File Direct or NFS, place the user that runs the PowerCenter Integration Service in the SAP administrator group. If you access the file through FTP, place the FTP user in the SAP administrator group.
- **Prepare the staging directory.** If you access the file through File Direct, NFS, FTP, or SFTP, configure the directory so that SAP creates the staging file with the group ID of the directory, not the SAP user who creates the file.

To prepare the staging directory:

1. The user accessing the staging file must create the staging directory.
2. Run the following UNIX command from the directory where you want to generate the files:

   ```
   % chmod g+s .
   ```

   When you run this command, the staging files inherit the group ID of the directory, not the SAP user that creates the file. There are no permission issues because the user who accesses the file also owns the directory.

   Note: If the SAP system is on Windows, the user who accesses the file must have standard read permissions on the directory where you stage the file.

Configuring File Mode Session Properties

When you run file mode sessions, configure the following source session properties:

- **Stage File Directory**. The SAP path containing the staging file.
- **Source File Directory**. The PowerCenter Integration Service path containing the source file.

- **Stage File Name**. The name of the staging file. You can edit this file name. If you access the staging file through FTP or SFTP and the FTP or SFTP server is running on Windows, do not use a period (.) at the end of the file name.
- **Reinitialize the Stage File**. If enabled, the ABAP program extracts data and replaces the existing staging file. You can only enable this option if you also enable Persist the Stage File.
- **Persist the Stage File**. If enabled, the PowerCenter Integration Service reads an existing, valid staging file. If the staging file does not exist or is invalid, the ABAP program creates a new staging file.
- **Run Session in Background**. Use this option when the data volume is high and the extraction time is long.

The following table summarizes entries for the staging file and source file directories for different access methods:

Access Method	Staging Directory	Source Directory	Situation	Establish Connection
File Direct	/data/sap	/data/sap	Paths are the same.	Mapped drive if different machines.
NFS	/data/sap	e:\sapdir	Specify the path from each machine.	Mapped drive.
FTP, SFTP	/data/sap	/data/sap	Unrestricted FTP or SFTP.	FTP or SFTP connection.
FTP-Restricted	e:\ftp\sap	/sap	FTP server is restricted to e:\ftp. You want the file in e:\ftp\sap. Specify the full path in the staging directory (e:\ftp\sap). Specify the path from the restricted directory in the source directory (/sap).	FTP connection.

Pipeline Partitioning for SAP ECC Sources

You can increase the number of partitions in a pipeline to improve session performance. Increasing the number of partitions allows the PowerCenter Integration Service to create multiple connections to sources and targets and process partitions of sources and targets concurrently.

When you configure an ABAP mapping to use pipeline partitioning, the PowerCenter Integration Service processes the partition information similar to processing dynamic filters. The PowerCenter Integration Service saves the partition information in the repository.

The PowerCenter Integration Service moves the partition information to the SAP system. The ABAP program then calls a function to process the partition information.

When you create partitions, you can override the dynamic filter condition in the Application Source Qualifier.

The following partitioning restrictions apply to SAP ECC sources:

- You can only use pass-through and key range partition types.
- The pipeline cannot contain multiple partitions if the mapping contains hierarchies.
- The pipeline cannot contain multiple partitions if you generate the ABAP program using exec SQL.
- Use PowerCenter default date format (MM/DD/YYYY HH:MI:SS.US) to enter dates in key ranges for datatypes such as DATS and ACCP.
- You cannot use a RAW or LRAW column as a key for partitions.

Session Properties for SAP ECC Targets

You can configure session properties for SAP ECC targets in the Targets view on the Mapping tab. Define the properties for each target instance in the session. When you click the Targets view on the Mapping tab, you can view and configure the settings of a specific target. Select the target under the Targets node.

The following table describes the properties available in the Properties settings on the Mapping tab of the session properties:

Attribute Name	Description
Target load type	You can choose Normal or Bulk. - If you select Normal, the PowerCenter Integration Service loads SAP ECC targets normally. - If you select Bulk, the PowerCenter Integration Service loads all the rows in a packet to SAP ECC targets at the same time. Loading in bulk mode can improve session performance. Default is Normal.
Packet Size	Set the packet size that the PowerCenter Integration Service commits table data to SAP. Packet size determines the maximum number of rows the PowerCenter Integration Service stores in its buffer before it makes an RFC call to the SAP system. Default is 10000.
Insert	Integration Service inserts all rows flagged for insert. Default is enabled.
Update	Select one of the Update options. - None. PowerCenter Integration Service does not update any rows. - Update as Update. PowerCenter Integration Service updates all rows flagged for update. Default option. - Update as Insert. PowerCenter Integration Service inserts all rows flagged for update. - Update else Insert. PowerCenter Integration Service updates rows flagged for update if they exist in the target, else inserts the rows.
Delete	PowerCenter Integration Service deletes all rows flagged for delete. Default is enabled.
Truncate target table option	PowerCenter Integration Service truncates the target before loading. Default is disabled.
Reject file directory	Reject file directory name. By default, the Integration Service writes all reject files to the service process variable directory, $PMBadFileDir.
Reject file name	File name for the reject file.
Target Table name	You can override the default target table name.
Update Override	Overrides the UPDATE statement in the target definition Properties tab.

Reject Files

The PowerCenter Integration Service creates a reject file for each target instance in the mapping.

When loading data to the SAP target tables, the rows can be rejected either by the writer or by the SAP target tables. The writer rejects the rows if there is a data conversion error or if you try to inset, update, or delete rows without configuring the session properties. The PowerCenter Integration Service writes the rejected rows into the reject file.

When the SAP target table rejects a row during normal mode, the SAP transport module sends the data to PowerCenter Integration Service. The PowerCenter Integration Service writes the reject data to the reject file. If SAP rejects data during bulk mode, the PowerCenter Integration Service does not writes the bad data to the reject file. It updates the session log with the number of rows in the packet rejected by SAP.

Each row indicates if the row is set for update, insert, or delete by marking U, I, or D at the beginning of each row. Every row in the reject file has either INFA or SAP indicating where the row was rejected.

The PowerCenter Integration Service creates the reject file based on the target instance.

The following sample reject file shows the SAP and INFA indicators:

```
I,SAP,800,000000000000000968,20040813,MELEWSKY,,,KEBV,KEBV,
I,INFA,800,000000000000001009,20041001,SAHAD,,,KVEDPALSQBG
```

Session Load Mode

The PowerCenter Integration Service stores the rows in memory until the total row count reaches the packet size. It then loads the rows stored in the buffer into the SAP ECC target table as a packet. You can choose to load data into the SAP tables in normal or bulk mode.

Normal mode inserts the rows one by one into the target table. Bulk mode inserts rows into the SAP ECC target table at the same time.

Normal mode is the default load method. You must use the normal mode to load data into the SAP system for any or all of the following conditions:

- The mapping contains an Update Strategy transformation.
- The packet has mixed operations.
- You want to override the default UPDATE statement for each target in a target definition.
- You want to view the SAP rejected rows in the bad file.
- You manually edit the primary keys in the Designer and want to update or delete rows based on the modified key.
- You want to update select fields in the SAP target table.

When you load data using bulk mode, all the rows within a packet must have the same operation, such as insert, update, or delete. When the rows within a packet have mixed operations, then the PowerCenter Integration Service reverts to normal load for the packet.

When the PowerCenter Integration Service shifts from bulk loading to normal loading, it writes an informational message in the sessions log. When a bulk load fails, PowerCenter Integration Service updates the session log with load statistics. It updates the session log with the number of rows in the packet that were loaded into SAP and the number of rows that were rejected. The PowerCenter Integration Service does not write the reject rows rejected by SAP in the reject file that caused the bulk load to fail.

Configuring an SAP ECC Session

You can configure session properties for sessions for mappings with SAP ECC sources and targets.

1. In the Task Developer, edit an SAP session.

   The Edit Tasks dialog box appears.
2. On the Properties tab, select Fail Task and Continue Workflow or Restart Task for the Recovery Strategy property.
3. From the Connections settings on the Mapping tab (Sources node), select the connection values for the SAP ECC sources.

   Tip: For a file mode session, use an FTP or SFTP connection to access the staging file.
4. Click Readers and select the appropriate readers for the SAP ECC sources.
5. If you specify the SAP Staging Reader for an SAP ECC source, click Properties and edit the following properties for the file source:

Attribute Name	Description
Stage File directory	SAP path containing the stage file. You can parameterize the stage file directory.
Source File directory	PowerCenter Integration Service path containing the source file. You can parameterize the source file directory.
Stage File Name	Name of the stage file. You can parameterize the stage file name.
Reinitialize the Stage File	If enabled, the ABAP program extracts data and replaces the existing staging file. You can only enable this option if you also enable Persist the Stage File.
Persist the Stage File	If enabled, the PowerCenter Integration Service reads an existing, valid staging file. If the staging file does not exist or is invalid, the ABAP program creates a new staging file. If disabled, the PowerCenter Integration Service deletes the staging file after reading it. Default is disabled.
Run Session in Background	Use when the data volume is high and the extraction time is very long.

6. On the Targets node, enter the connection values for the targets in the mapping.
7. Use the Partitions view on the Mapping tab to specify multiple partitions or change the partitioning information for each pipeline in a mapping.

Troubleshooting Sessions

The source session failed with a NO AUTHORITY error.

The source session failed because you selected the Authority Check option when you generated the ABAP program for the mapping, but did not have the proper authorization.

If you enabled the Authority Check option when you generated the ABAP program, the SAP application server verifies that the user running the workflow has authorization to read the sources. SAP verifies the authorization before it reads the first source.

If the user in the SAP ECC application connection does not have read authorization on any one of the sources, the session fails with a NO AUTHORITY error.

Part III: IDoc Integration Using ALE

This part contains the following chapters:

- Creating Outbound IDoc Mappings, 135
- Creating Inbound IDoc Mappings, 145
- Configuring IDoc Workflows Using ALE, 152

CHAPTER 11

Creating Outbound IDoc Mappings

This chapter includes the following topics:

- Creating Outbound IDoc Mappings Overview, 135
- Creating an SAPALEIDoc Source Definition, 136
- Working with the SAP/ALE IDoc Interpreter Transformation , 137
- Processing Invalid Outbound IDocs, 144

Creating Outbound IDoc Mappings Overview

You can configure PowerExchange for SAP NetWeaver to receive outbound SAP IDocs in real time as they are generated by mySAP applications. To receive outbound IDocs, PowerExchange for SAP NetWeaver integrates with mySAP applications using Application Link Enabling (ALE). ALE is an SAP proprietary technology that enables data communication between SAP systems. ALE also enables data communication between SAP and external systems.

Note: Receiving outbound SAP IDocs is different from sourcing IDocs from static EDIDC and EDIDD structures.

You can capture changes to the master data or transactional data in the SAP application database in real time. When data in the application database changes, the SAP system creates IDocs to capture the changes and sends the IDocs to the PowerCenter Integration Service.

The PowerCenter Integration Service and SAP use transactional RFC (tRFC) communication to send and receive IDocs. tRFC is an SAP method that guarantees the RFCs are executed only once. As a result, the PowerCenter Integration Service receives each IDoc only once.

If the PowerCenter session is not running when the SAP system sends outbound IDocs, the PowerCenter Integration Service does not receive the IDocs. However, the SAP system stores the outbound IDocs in EDI tables, which are a staging area for guaranteed message delivery.

You can configure the SAP system to resend the IDocs by configuring the tRFC port used to communicate with the PowerCenter Integration Service. When you configure the port, you can enable background processes in SAP that try to resend the IDocs to the PowerCenter Integration Service a set number of times.

Defining PowerCenter as a Logical System for Outbound IDocs

Before you can receive IDocs from SAP using ALE, define PowerCenter as a logical system that receives IDocs from the base logical system in SAP.

Creating an Outbound IDoc Mapping

To receive outbound IDocs from mySAP applications, create an outbound IDoc mapping. The outbound IDoc mapping can contain the following components:

- **SAPALEIDoc source definition.** Source definition to read data from the SAP source system.
- **Application Multi-Group Source Qualifier transformation.** Determines how the PowerCenter Integration Service reads data from the SAP source.
- **SAP/ALE IDoc Interpreter transformation (optional).** Processes IDoc data according to the IDoc type you specify when you create the transformation.
- **Target definition.** Target definition for the target to which you want the PowerCenter Integration Service to write the IDoc data.

Processing Invalid Outbound IDocs

You can validate outbound IDocs. If you validate outbound IDocs, you can configure an outbound IDoc mapping to write invalid IDocs to a flat file or relational target.

Related Topics:

- "Processing Invalid Outbound IDocs" on page 144

Creating an SAPALEIDoc Source Definition

To receive outbound IDocs from SAP using ALE, create an SAPALEIDoc source definition in the Designer. An SAPALEIDoc source definition represents metadata for outbound IDocs. When you create an SAPALEIDoc source definition, the Designer displays a table with IDoc fields and SAP datatypes. When the PowerCenter Integration Service extracts data from the SAP source, it converts the data based on the datatypes in the Source Qualifier transformation associated with the source.

An SAPALEIDoc source definition contains predefined ports. You cannot edit the ports.

The following table describes the SAPALEIDoc source definition ports:

Port Name	Description
Basic IDoc Type	Basic IDoc type name.
Extended IDoc Type	Extended IDoc type name.
IDocRecord	IDoc message data.
DocumentNumber	Unique message number of the IDoc.

Tip: You only need to maintain one SAPALEIDoc source definition per repository folder. When you include an SAPALEIDoc source definition in a mapping, you can add an instance of the source definition to the mapping.

Using SAPALEIDoc Source Definitions in an Outbound IDoc Mapping

When you include an SAPALEIDoc source definition and its associated Application Multi-Group Source Qualifier transformation in an outbound IDoc mapping, you can only connect the Source Qualifier to one SAP/ALE IDoc Interpreter transformation.

If you want to include multiple SAP/ALE IDoc Interpreter transformations in the mapping, include an SAPALEIDoc source definition and Application Multi-Group Source Qualifier transformation for each SAP/ALE IDoc Interpreter transformation.

Working with the SAP/ALE IDoc Interpreter Transformation

Include an SAP/ALE IDoc Interpreter transformation in an outbound IDoc mapping when you want to process the outbound IDoc data you receive from the SAP system. The transformation receives data from upstream transformations in the mapping and interprets the segment data.

Each SAP/ALE IDoc Interpreter transformation can interpret data for one IDoc type. For information about reading multiple types of outbound IDocs, see Informatica Knowledge Base article 17410.

After you create an SAP/ALE IDoc Interpreter transformation, you can edit the transformation to change the data segments you want to include in the transformation.

When you edit the transformation, you can also view details about the IDoc type and segments. To view details, double-click the title bar of the transformation and select the IDoc Display tab.

The following table shows the information that displays when you view details about an IDoc on the IDoc Display tab of an SAP/ALE IDoc Interpreter or SAP/ALE IDoc Prepare transformation:

Property	Description
Message Type	IDoc message type.
Basic Type	Basic IDoc type, where applicable.
Extended Type	Extended IDoc type, where applicable.
Show Transformation Metadata	Select to display transformation metadata.
Show Group Status	Select to display the Group Status column.
Segment Name	Segment names of the IDoc type.
Description	Description of the segments, where applicable.
Select	Select the data segments to include in the transformation.
Segment Status	Required segments.
Group Status	Required groups. Displays only if you select Show Group Status.

Property	Description
Minimum Occurrence	Minimum number of occurrences of the segment in an IDoc.
Maximum Occurrence	Maximum number of occurrences of the segment in an IDoc.
Field Name	Field names of a segment.
Description	Description of the field.
SAP Datatype	SAP datatype of the field.
SAP Precision	SAP precision of the field.
SAP Scale	SAP scale of the field.
Transformation Datatype	Transformation datatype of the field. Displays only if you select Show Transformation Metadata.
Transformation Precision	Transformation precision of the field. Displays only if you select Show Transformation Metadata.
Transformation Scale	Transformation scale of the field. Displays only if you select Show Transformation Metadata.

Segments and Groups

An IDoc is a hierarchical structure that contains segments. A segment can be a parent or child. A child segment depends on another segment. A parent segment contains child segments. A parent segment can be a child of another segment.

IDoc segments are organized into groups. The following rules determine to which group a segment belongs:

- **A parent segment starts a new group.** For example, in the following figure, the E1MARCM segment contains a child and therefore starts a new group.
- **A child segment that is not a parent belongs to the group that is started by its immediate parent.** For example, in the following figure, the E1MARA1 segment does not contain a child and therefore belongs to the group of its parent E1MARAM.

A group can also be a parent or a child.

The following figure provides an example, where the E1MARAM group is a parent of the E1MARCM group:

Some segments and groups are required. In an SAP/ALE IDoc Prepare transformation, SAP/ALE IDoc Interpreter transformation, and SAP DMI Prepare transformation, a required segment must exist in the IDoc only if its group, its parent groups, and its parent segments are required or selected. For example, the E1MARAM group is required. Therefore, its required child segment E1MAKTM must exist in the IDoc. Its optional child segment E1MARA1 does not have to exist in the IDoc.

If a required segment belongs to an optional group that is not selected, then the segment does not have to exist in the IDoc. For example, the E1MARCM group is optional. Therefore, the required E1MARCM segment also becomes optional.

Note: These rules describe the hierarchy of a standard IDoc. The hierarchy of a custom IDoc may differ. However, the PowerCenter Integration Service processes the data in the same way.

Viewing Group and Segment Status

To help you understand when segments are required, use the read-only Segment Status and Group Status columns. To display the Group Status column, click Show Group Status. When a group is required, the Group Status column is selected. When a segment is required, the Segment Status column is selected.

The following table provides an example that shows how you can use the Segment Status and Group Status columns to understand which segments are required in the MATMAS04 IDoc:

Segment Name	Segment Group	Segment Status	Group Status	Required in IDoc
E1MARAM	E1MARAM	Required	Required	Required
E1MARA1	E1MARAM	Optional	Required	Optional

Segment Name	Segment Group	Segment Status	Group Status	Required in IDoc
E1MAKTM	E1MARAM	Required	Required	Required
E1MARCM	E1MARCM	Required	Optional	Optional

When you clear Show Group Status to hide the Group Status column, the Segment Status column uses different rules to determine which segments are selected, depending on the type of segment:

- **Child segment that is not a parent.** When the segment is required, the Segment Status column is selected. For example, in the following figure, the E1MAKTM segment is required, so its Segment Status column is selected.
- **Parent segment.** When both the segment and its group are required, the Segment Status column is selected. For example, in the following figure, the E1MARAM segment and group are required, so its Segment Status column is selected. The E1MARCM segment is required, however its group is optional. The E1MARCM Segment Status column is cleared.

The following figure shows how the Segment Status column changes for the E1MARCM segment when you clear Show Group Status:

Creating an SAP/ALE IDoc Transformation

Use the Generate SAP IDoc Interpreter Transformation Wizard to create an SAP/ALE IDoc Interpreter transformation to read outbound IDocs from the SAP system. You can import basic or extended IDoc type metadata.

When you create an SAP/ALE IDoc Interpreter transformation, you can import the IDoc metadata in the following ways:

- **From file.** Use to import the metadata for the IDocs in the SAP/ALE IDoc Interpreter from a metadata file.

  When you import the IDoc metadata from file, SAP converts numeric datatypes in the IDoc metadata to CHAR. After you create the transformation, apply the appropriate transformation datatypes to the ports that have numeric datatypes in SAP.
- **Connect to SAP.** Use to import the IDoc metadata from the SAP system to use in the transformation. The PowerCenter Integration Service can validate the input data you pass into the transformation.

Generating IDoc Metadata to File for Import

If you want to import IDoc metadata for an SAP/ALE IDoc transformation from file, run the RSEIDoc3 program from the SAP client to generate the metadata. When you run the program, select the IDoc type and range for the IDoc metadata you want to generate. The program exports the metadata it generates into a metadata file. For example, you can export the metadata to a file with the *.idc* extension. You can then use the metadata file to import the metadata into the Designer for use in the SAP/ALE IDoc transformation.

To generate IDoc metadata using the RSEIDoc3 program:

1. Enter transaction se38 from the SAP client.
2. Execute the RSEIDoc3 program.
3. Select the basic IDoc type and range.
4. If you are using SAP version 5.0 or later, click Control Record.
5. Optionally, select extended IDoc type and range, if available.
6. Optionally, select extended grammar, if available.
7. Click Parser.
8. Click System > List > Save > Local File.
9. On the Save List in File dialog box, select Unconverted.
10. Enter the path and file name where you want to save the metadata file. Save the file with the *.idc* extension.

Steps to Create an SAP/ALE IDoc Transformation

Use the following procedure to create SAP/ALE IDoc Interpreter and SAP/ALE IDoc Prepare transformations.

To create an SAP/ALE IDoc Interpreter or SAP/ALE IDoc Prepare transformation:

1. In the Transformation Developer, click the SAP/ALE IDoc Interpreter transformation button or SAP/ALE IDoc Prepare transformation button.

   The pointer becomes a cross.
2. Click in the Transformation Developer workspace.

   If you clicked the SAP/ALE IDoc Interpreter button, the Generate SAP IDoc Interpreter Transformation Wizard appears. If you selected the SAP/ALE IDoc Prepare button, the Generate SAP IDoc Prepare Transformation Wizard appears.
3. To import IDoc metadata from a file, click Local File.

   To import IDoc metadata from the SAP system, go to step 7.
4. If you clicked Local File, enter the name and path of the file from which you want to import IDoc metadata. Or, click Browse to locate the file you want to use.
5. Click Import.

6. Go to step 13.
7. If you want to import IDoc metadata from the SAP system, enter the following information:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

8. Click Connect.

   After you connect to the SAP system, enter a filter to display specific IDoc types.
9. Select one of the following filter types:
   - **Message Type.** Select to display IDocs by message type. The Designer displays the basic and extended type for each IDoc that meets the filter condition.
   - **Basic IDoc Type.** Select to display IDocs by basic IDoc type. The Designer displays only the basic type for each IDoc that meets the filter condition.
   - **Extended IDoc Type.** Select to display IDocs by extended IDoc type. The Designer displays only the extended type for each IDoc that meets the filter condition.
10. Enter a filter condition.

    You can enter an IDoc name. Or, use an asterisk (*) or percent sign (%) as wildcard characters to display IDocs that meet the filter condition. Use the following syntax when you enter a wildcard character:
    - **Enter the filter condition as a prefix.** For example, enter MAT* or MAT% to display all IDocs that begin with MAT.
    - **Enter the filter condition as a suffix.** For example, enter *AT or %AT to display all IDocs that end with AT.
    - **Enter the filter condition as a substring.** For example, enter *MAT* or %MAT% to display all IDocs that contain MAT.
11. Click Show IDoc Types.

    All IDocs that meet the filter condition appear.
12. To refine the IDocs that display, you can select one or both of the following options:
    - **Show Only Unknown Message Type.** If IDocs display that are of an unknown message type, you can select this option to display only those IDocs.
    - **Show Release For Message Type.** Select to display IDocs by SAP release.
13. Expand an IDoc Type to see a list of basic and extended IDocs.
14. Select the basic or extended IDoc whose metadata you want to import and click Next.

    Step 2 of the wizard appears.
15. Click Show Group Status to show the required groups in the Group Status column.

16. Select the IDoc segments you want to include in the transformation.

    You can manually select the segments you want to include. Or, click Select All Segments to include all segments.

    Click Clear All Segments to remove all optional segments.

    When you select the segments, the transformation uses the following rules to select parent and child segments:

    - If you select a segment, its parent segments and required child segments are selected.
    - If you clear a segment, its child segments are cleared.
17. Click Next.

    Step 3 of the wizard appears. The wizard provides a name for the transformation.
18. Optionally, modify the name of the transformation.

    If you clicked Transformation > Create to create the transformation, you cannot modify the name of the transformation in Step 3 of the wizard. The Designer uses the transformation you entered in the Create Transformation dialog box.
19. Optionally, modify the description of the transformation.
20. Click Finish.

Editing an SAP/ALE IDoc Interpreter Transformation

You can edit an SAP/ALE IDoc Interpreter transformation to change the data segments you want to include in the transformation. You can also modify the name and description of the transformation.

To edit an SAP/ALE IDoc Interpreter transformation:

1. In the Transformation Developer or Mapping Designer, double-click the title bar of the SAP/ALE IDoc Interpreter transformation.

   The Edit Transformations window appears.
2. On the Properties tab, select Output is Deterministic if you want to enable recovery for an outbound IDoc session.
3. Click the IDoc Display tab.
4. Optionally, modify the segments you want to include in the SAP/ALE IDoc Interpreter transformation.

   You can manually select the segments you want to include in the transformation. Or, click one of the following options to select IDoc segments:

   - **Select All Segments.** Click to include all segments.
   - **Clear All Segments.** Click to remove all selected segments except required segments.

   When you select the segments to include in the transformation, the transformation uses the following rules to select parent and child segments:

   - If you select a segment, its parent segments all required child segments are selected.
   - If you clear a segment, all child segments are cleared.
5. Click OK.

Processing Invalid Outbound IDocs

You can configure an outbound IDoc mapping to write invalid IDocs to a relational or flat file target. To write invalid IDocs to a relational or flat file target, connect the Error Output port in the SAP/ALE IDoc Interpreter transformation to a relational or flat file target definition.

To write invalid outbound IDocs to a relational or flat file target, you must also configure the outbound IDoc session to check for invalid IDocs.

CHAPTER 12

Creating Inbound IDoc Mappings

This chapter includes the following topics:

- Creating Inbound IDoc Mappings Overview, 145
- Working with the SAP/ALE IDoc Prepare Transformation, 146
- Creating an SAPALEIDoc Target Definition, 150
- Configuring Inbound IDoc Mappings, 150

Creating Inbound IDoc Mappings Overview

You can configure PowerExchange for SAP NetWeaver to send inbound SAP IDocs to mySAP applications. To send inbound IDocs, PowerExchange for SAP NetWeaver integrates with mySAP applications using Application Link Enabling (ALE). ALE is an SAP proprietary technology that enables data communication between SAP systems. ALE also enables data communication between SAP and external systems.

For example, you have a legacy application that processes sales transactions. You want to synchronize the transactional data in the legacy application with the data in the SAP application database. Use an inbound SAP IDoc mapping to send the transactional data from the legacy application database to the SAP system. The PowerCenter Integration Service extracts the data from the legacy application data source, prepares the data in SAP IDoc format, and sends the data to the SAP system as inbound IDocs using ALE.

Defining PowerCenter as a Logical System for Inbound IDocs

Before you can send inbound IDocs to SAP using ALE, define PowerCenter as a logical system that sends IDocs to SAP.

Creating an Inbound IDoc Mapping

To send inbound IDocs to mySAP applications, create an inbound IDoc mapping. The inbound IDoc mapping needs to contain the following components:

- **Source definition.** Source definition to read data from the source system.
- **Source Qualifier transformation.** Determines how the PowerCenter Integration Service reads data from the source.
- **SAP/ALE IDoc Prepare transformation.** Processes IDoc data according to the IDoc type you specify when you create the transformation.
- **SAPALEIDoc target definition.** Target definition that writes IDocs to the SAP system.

Validating 6.x Inbound IDoc mappings

If you export an inbound IDoc mapping from PowerCenter 6.x and then import it into an 8.5.x repository, the mapping loses all links between transformations and becomes invalid. To use a 6.x inbound IDoc mapping, link the ports and then validate the mapping.

Processing Invalid Inbound IDocs

You can validate inbound IDocs before sending them to the SAP system. If you validate inbound IDocs, you can configure an inbound IDoc mapping to write invalid IDocs to a flat file or relational target instead of sending them to SAP.

Related Topics:

- "Processing Invalid Inbound IDocs" on page 151

Working with the SAP/ALE IDoc Prepare Transformation

You need to include an SAP/ALE IDoc Prepare transformation in an inbound IDoc mapping. The transformation receives data from upstream transformations in the mapping and interprets the segment data.

Each SAP/ALE IDoc Prepare transformation can only interpret data for a particular IDoc type. You can include multiple SAP/ALE IDoc Prepare transformations to represent multiple IDoc types in a single mapping.

After you create an SAP/ALE IDoc Prepare transformation, you can edit the transformation to set values for control record segments and to change the data segments you want to include in the transformation.

When you edit the transformation, you can also view details about the IDoc type and segments. To view details, double-click the title bar of the transformation and select the IDoc Display tab.

IDoc Primary and Foreign Keys

An IDoc message is organized hierarchically with one top-level parent segment and one or more second-level child segments. Second-level child segments can also have one or more third-level child segments.

To maintain the structure of the IDoc data, the IDoc Prepare transformation uses primary and foreign keys. The top-level parent segment has a primary key. Each child segment has a primary key and a foreign key. The foreign key of each child segment references the primary key of its parent segment. For example, the foreign key of a second-level child segment references the primary key of the top-level parent segment. Similarly, the foreign key of a third-level child segment references the primary key of the second-level child segment.

The IDoc Prepare transformation groups incoming IDoc data based on the values in the primary and foreign key fields. The Control Input group of the IDoc Prepare transformation represents the parent segment. All other groups of the IDoc Prepare transformation except the ErrorIDocData group represent second-level or third-level child segments.

Note: The ErrorIDocData group is used for processing invalid IDocs.

The following table shows the groups of the IDoc Prepare transformation and the fields used for the primary and foreign keys:

Group(s)	Field	Description
Control Input Group	GPK_DOCNUM	Primary key of the parent segment.
Child Segment 1	GPK_<Child1_name>	Primary key of Child Segment 1.
Child Segment 1	GFK_DOCNUM_<Child1_name>	Foreign key of Child Segment 1 references the primary key of the parent segment.
Child Segment A of Child Segment 1	GPK_<Child1A_name>	Primary key of Child Segment A of Child Segment 1.
Child Segment A of Child Segment 1	GFK_<Child1_name>_<Child1A_name>	Foreign key of Child Segment A of Child Segment 1 references the primary key of Child Segment 1.
Child Segment 2	GPK_<Child2_name>	Primary key of the IDoc child segment.
Child Segment 2	GFK_DOCNUM_<Child2_name>	Foreign key of Child Segment 2 references the primary key of the parent segment.
Child Segment B of Child Segment 2	GPK_<Child2B_name>	Primary key of Child Segment B of Child Segment 2.
Child Segment B of Child Segment 2	GFK_<Child2_name>_<Child2B_name>	Foreign key of Child Segment B of Child Segment 2 references the primary key of Child Segment 2.

Each value for the GPK_<name> field needs to be unique. Each GFK_<parent_name>_<group_name> field needs to reference the primary key of its parent segment.

For example, the following table shows the relationship of primary and foreign keys in an IDoc message named ABSEN1 with four child segments:

Group	Field	Primary/Foreign Keys
CONTROL_INPUT_ABSEN1	GPK_DOCNUM	P1
E2ABSE1	GPK_E2ABSE1	C1
-	GFK_DOCNUM_E2ABSE1	P1
E2ABSE2	GPK_E2ABSE2	C2
-	GFK_DOCNUM_E2ABSE2	P1
E2ABSE2A	GPK_E2ABSE2A	C2A
-	GFK_E2ABSE2_E2ABSE2A	C2
E2ABSE3	GPK_E2ABSE3	C3
-	GFK_DOCNUM_E2ABSE3	P1

Group	Field	Primary/Foreign Keys
E2ABSE3B	GPK_E2ABSE3B	C3B
-	GFK_E2ABSE2_E2ABSE2A	C3
E2ABSE4	GPK_E2ABSE4	C4
-	GFK_DOCNUM_E2ABSE4	P1

The IDoc Prepare transformation uses these primary and foreign key relationships to maintain the structure of the IDoc data. Any foreign key field that does not match the primary key of its parent segment results in an orphan row. Any primary key field that is not unique results in a duplicate row.

Verify that each IDoc message has a unique primary key for the top-level parent segment, each child segment, and that each foreign key matches the primary key of its parent.

Creating an SAP/ALE IDoc Prepare Transformation

Use the Generate SAP IDoc Prepare Transformation Wizard to create an SAP/ALE IDoc Prepare transformation. The wizard lets you import basic or extended IDoc type metadata.

When you create an SAP/ALE IDoc Prepare transformation, you can import the IDoc metadata in the following ways:

- **From file.** Use to import the metadata for the IDocs in the SAP/ALE IDoc Prepare transformation from a metadata file. For information about generating IDoc metadata, see "Generating IDoc Metadata to File for Import" on page 141.

  When you import the IDoc metadata from file, SAP converts numeric datatypes in the IDoc metadata to CHAR. When you create or edit the transformation, apply the appropriate transformation datatypes to the ports that have numeric datatypes in SAP.
- **Connect to SAP.** Use to import the IDoc metadata from the SAP system to use in the transformation. The PowerCenter Integration Service can validate the input data you pass into the transformation.

Editing an SAP/ALE IDoc Prepare Transformation

You can edit an SAP/ALE IDoc Prepare transformation to set values for control record segments and change the data segments you want to include in the transformation. You can also modify the name and description of the transformation.

To edit an SAP/ALE IDoc Prepare transformation:

1. In the Transformation Developer or Mapping Designer, double-click the title bar of the SAP/ALE IDoc Prepare transformation.

   The Edit Transformations window appears.
2. Click the IDoc Control Record tab.

   The IDoc Control Record tab displays the control record segments in the IDoc, their values, and precision. The Designer provides values for some of the segments when you create an SAP/ALE IDoc Prepare transformation. You can provide values for other segments. The PowerCenter Integration Service writes these values to the SAP system.

You can enter values in the following ways:

- Manually enter values for segments.
- Connect to the SAP system to get predefined values for required segments.

3. If you do not want to connect to the SAP system to get IDoc control record segment values, enter values for the segments you want.

   You can also enter a mapping variable for any segment.

4. To get predefined values for required control record segments, click Get Partner Profile to connect to the SAP system.

   Tip: If you want to connect to the SAP system to get values for required control record segments, and you imported the IDoc metadata for the transformation from file, you can enter a value for MESTYP before clicking Get Partner Profile. When you connect to the SAP system, it displays the control record segments for the message type you want. Otherwise, locate the message type.

   The Connect to SAP dialog box appears.

5. Enter the following connection information to connect to the SAP system:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

6. Enter a partner number.

   The Connect to SAP dialog box displays the segment names and values for the Designer to include in the SAP/ALE IDoc Prepare transformation.

7. If you imported IDoc metadata for the SAP/ALE IDoc Prepare transformation or entered a message type value for MESTYP on the IDoc Control Record tab, click Select. Go to step 9.

8. If there is no value for the message type on the IDoc Control Record tab, click Next until you find the appropriate message type.

   Find the segments for the message type for which you created the SAP/ALE IDoc Prepare transformation. For example, if you created an SAP/ALE IDoc Prepare transformation for the message type MATMAS, find the segments for the message type MATMAS.

9. Click Select.

   Click Cancel if you do not want to use these values.

   The Designer prompts you to update the control record.

10. Click Yes to update the control record. Or, click No to cancel.

11. Click the IDoc Display tab.

12. Select the IDoc segments you want to include in the transformation.

You can manually select the segments you want to include in the transformation. Or, click Select All Segments to include all segments.

Click Clear All Segments to remove all optional segments.

When you select the segments, the transformation uses the following rules to select parent and child segments:

- If you select a segment, its parent segments and all required child segments are selected.
- If you clear a segment, all child segments are cleared.

13. Click OK.

Creating an SAPALEIDoc Target Definition

To send inbound IDocs to SAP using ALE, create an SAPALEIDoc target definition in the Designer. An SAPALEIDoc target definition represents metadata for inbound IDocs. When you create an SAPALEIDoc target definition, the Designer displays a table with IDoc fields and SAP datatypes. When the PowerCenter Integration Service sends data to the SAP target, it converts the data based on the transformation datatypes in the mapping to the target datatypes.

An SAPALEIDoc target definition contains the predefined port IDocData. You cannot edit this port in the Designer.

Tip: You only need to maintain one SAPALEIDoc target definition per repository folder. When you include an SAPALEIDoc target definition in a mapping, you can add an instance of the target definition to the mapping.

Configuring Inbound IDoc Mappings

When you configure an inbound IDoc mapping, use the following guidelines:

- Pass a value to the DOCNUM port of the SAP/ALE IDoc Prepare transformation.
- Optionally, pass the sender partner number to SAP.
- Determine how you want to process invalid IDocs.

Passing the Document Number to SAP

An inbound IDoc mapping must link the DOCNUM port of the SAP/ALE IDoc Prepare transformation to an upstream transformation. The DOCNUM port represents a unique number for each IDoc. The SAP system does not accept inbound IDocs without a unique document number. If the PowerCenter Integration Service does not pass a value to the DOCNUM port, the session fails.

Passing the Sender Partner Number to SAP

You can pass the sender partner number to SAP. Pass a value to the CONTROL_SNDPRN port in the control input group of the SAP/ALE IDoc Prepare transformation. If you do not connect this port to an upstream transformation, the PowerCenter Integration Service use the partner number value of SNDPRN on the IDoc Control Record tab of the transformation, if any.

Processing Invalid Inbound IDocs

You can configure an inbound IDoc mapping to write invalid IDocs to a relational or flat file target instead of the SAP system.

To write invalid IDocs to a relational or flat file target, complete the following steps:

1. Connect the ErrorIDocData port in the SAP/ALE IDoc Prepare transformation to a relational or flat file target definition.
2. Configure the inbound IDoc session to check for invalid IDocs.

CHAPTER 13

Configuring IDoc Workflows Using ALE

This chapter includes the following topics:

- Configuring Sessions for Outbound IDoc Mappings, 152
- Configuring Sessions for Inbound IDoc Mappings, 155
- Steps to Configure Sessions for IDoc Mappings Using ALE, 156
- Error Handling for IDoc Sessions Using ALE, 159
- Running IDoc Workflows Using ALE , 159
- Troubleshooting IDoc Workflows Using ALE, 159

Configuring Sessions for Outbound IDoc Mappings

When you configure an outbound IDoc session to receive IDocs from SAP using ALE, configure the following properties:

- Terminating conditions
- Real-time processing
- Message recovery
- Pipeline partitioning
- IDoc validation
- Continuous workflows

The following properties can affect outbound IDoc session performance:

- Pipeline partitioning
- Outbound IDoc validation
- Row-level processing

Terminating Conditions

Terminating conditions determine when the PowerCenter Integration Service stops reading from the source and ends the session. You can define the following terminating conditions:

- Idle Time

- Packet Count
- Reader Time Limit

When you enter values for multiple terminating conditions, the PowerCenter Integration Service stops reading IDocs from the SAP source when the first terminating condition is met. For example, if you set idle time to 10 seconds and the packet count to 100 packets, the PowerCenter Integration Service stops reading IDocs from the SAP source after 10 seconds or after reading 100 packets, whichever comes first.

Idle Time

Configure idle time to indicate the number of seconds the PowerCenter Integration Service waits for IDocs to arrive before it stops reading from the SAP source. For example, if you enter 30 for idle time, the PowerCenter Integration Service waits 30 seconds after reading from the SAP source. If no new IDocs arrive within 30 seconds, the PowerCenter Integration Service stops reading from the SAP source.

Packet Count

Configure packet count to control the number of packets the PowerCenter Integration Service reads from SAP before stopping. For example, if you enter 10 for Packet Count, the PowerCenter Integration Service reads the first 10 packets from the SAP source and then stops. The packet Size property in the ALE configuration determines the number of IDocs the PowerCenter Integration Service receives in a packet.

If you enter a Packet Count value, and you configure the session to use pipeline partitioning, the outbound IDoc session can run on a single node only. The PowerCenter Integration Service that runs the session cannot run on a grid or on primary and backup nodes.

Reader Time Limit

Configure reader time limit to read IDocs from the SAP source for a set period of time in seconds. The PowerCenter Integration Service reads IDocs from SAP for the number of seconds you specify. For example, if you specify 10 for the reader time limit, the PowerCenter Integration Service stops reading from the SAP source after 10 seconds.

Real-time Processing

You can configure flush latency to process data in real time. A real-time session reads, processes, and writes data to targets continuously. Flush latency determines how often the PowerCenter Integration Service flushes data from the source.

When you configure real-time flush latency and configure the session to use source-based commits, the PowerCenter Integration Service commits IDocs to the target using the source-based commit interval and the real-time flush latency interval. The PowerCenter Integration Service uses the number of IDoc records or control segments for the commit interval.

Message Recovery

When you configure message recovery for a real-time session, the PowerCenter Integration Service can recover unprocessed IDoc messages from a failed session. When you enable message recovery for a real-time session, the PowerCenter Integration Service stores IDoc messages in a recovery file. If the session fails, run the session in recovery mode to recover the IDoc messages that the PowerCenter Integration Service did not process.

To enable message recovery, select Output is Deterministic in the SAP/ALE IDoc Interpreter transformation.

Important: If you configure a transformation as repeatable and deterministic, it is your responsibility to ensure that the data is repeatable and deterministic. If you try to recover a session with transformations that do not produce the same data between the session and the recovery, the recovery process can result in corrupted data.

Pipeline Partitioning

You can increase the number of partitions in a pipeline to increase session performance. Increasing the number of partitions allows the PowerCenter Integration Service to create multiple connections to sources and targets and process partitions of sources and targets concurrently.

You can specify pass-through partitioning for Application Multi-Group Source Qualifiers in an outbound IDoc mapping.

Specifying Partitions and a Recovery Cache Folder

When you specify partitions for an outbound IDoc mapping in a session, and you configure the Recovery Cache Folder attribute in the session properties, enter a cache folder on a different device for each source partition in the pipeline.

Outbound IDoc Validation

You can configure an IDoc session to check for IDocs which are not valid. The PowerCenter Integration Service validates the IDoc before passing the data to a downstream transformation. If the IDoc is not valid, the PowerCenter Integration Service writes the IDoc to a relational or flat file target through the SAP/ALE IDoc Interpreter transformation Error Output port.

To check for IDocs which are not valid, select Extended Syntax Check in the session properties. You must also connect the Error Output port in the SAP/ALE IDoc Interpreter transformation to a relational or flat file target definition.

Checking for IDocs that are not valid can decrease session performance. Validate IDocs in the development environment. After you verify that the IDocs are valid, you can remove IDoc validation in the production environment to increase session performance.

Row-Level Processing

The PowerCenter Integration Service can process each row of an outbound IDoc according to the IDoc metadata and pass it to a downstream transformation. Enable row-level processing in the session properties. Row-level processing increases session performance.

If you disable row-level processing, the PowerCenter Integration Service buffers each IDoc before passing it to a downstream transformation. Buffering IDocs can decrease session performance.

If you enable row-level processing, the PowerCenter Integration Service does not validate outbound IDocs, even if outbound IDoc validation is enabled.

Continuous Workflows

You can schedule a workflow to run continuously. A continuous workflow starts as soon as the PowerCenter Integration Service initializes. When the workflow stops, it restarts immediately. To schedule a continuous workflow, select Run Continuously from the Schedule tab in the scheduler properties when you schedule the workflow.

Configuring Sessions for Inbound IDoc Mappings

When you configure an inbound IDoc session, select an SAP_ALE_IDoc_Writer application for the SAPALEIDoc target definition. You can also configure the following session properties:

- Pipeline partitioning
- Sending IDocs to SAP
- IDoc validation
- Caching inbound IDoc and DMI data

The following properties can affect inbound IDoc session performance:

- Pipeline partitioning
- Inbound IDoc validation

Pipeline Partitioning

You can increase the number of partitions in a pipeline to improve session performance. Increasing the number of partitions enables the PowerCenter Integration Service to create multiple connections to sources and process partitioned data concurrently.

When you configure an inbound IDoc session to use pipeline partitioning, use key range partitioning to make sure that all data belonging to an IDoc message is processed in the same logical partition. Use the port connected to the GPK_DOCNUM port in the SAP/ALE IDoc Prepare transformation as the partition key.

The transformation where you define the partitioning depends on the type of source definition the mapping contains. If the mapping contains a relational source definition, define key range partitioning for the Source Qualifier transformation. If the mapping contains a flat file source definition, the Source Qualifier transformation does not support key range partitioning for a flat file source definition. Therefore, include an Expression transformation in the inbound IDoc mapping preceding the SAP/ALE IDoc Prepare transformation. Define key range partitioning for the Expression transformation.

Sending IDocs to SAP

The PowerCenter Integration Service sends IDoc messages to SAP as a packet. By default, SAP allows packets up to10 MB. The SAP administrator can change the packet size configuration. When you configure how the PowerCenter Integration Service sends IDocs, configure the packet size to be equal to or less than the packet size configured in SAP.

Select one of the following options for the Send IDocs Based On property in the session properties:

- **Packet Size.** The PowerCenter Integration Service sends IDoc messages based on the value in the Packet Size property.
- **Commit Call.** The PowerCenter Integration Service sends IDoc messages at every commit point.

When you send IDocs based on packet size, the PowerCenter Integration Service stores the IDoc messages in memory until the total count reaches the packet size. It then sends the messages as a packet to SAP. A larger packet size reduces the number of calls to SAP. However, if the session fails, the PowerCenter Integration Service must resend a larger amount of data in the next session.

Calculate the value for the Packet Size session property based on the packet size configuration in SAP and the maximum number of rows per IDoc message that you expect to send to SAP. For example, you configured SAP to handle a packet of 10 MB. One row in an IDoc message is equal to 1,000 bytes. You are sending IDoc messages that contain a maximum of 50 rows. Set the Packet Size property to 200.

When you send IDocs based on commit call, the PowerCenter Integration Service commits IDocs to SAP based on the commit properties of the session. To ensure that the commit occurs at the IDoc message boundary, use a user-defined commit. The PowerCenter Integration Service commits IDocs based on transactions you define in the mapping properties. If you use a source-based commit, the PowerCenter Integration Service may send partial IDocs to SAP.

Inbound IDoc Validation

You can configure an IDoc session to check for IDocs which are not valid. The PowerCenter Integration Service validates the IDoc before passing the data to a downstream transformation. If the IDoc is not valid, the PowerCenter Integration Service writes the IDoc to a relational or flat file target through the SAP/ALE IDoc Prepare transformation Error Output port.

To check for IDocs which are not valid, select Extended Syntax Check in the session properties. You must also connect the Error Output port in the SAP/ALE IDoc Prepare transformation to a relational or flat file target definition.

Checking for IDocs which are not valid can impact session performance. Validate IDocs in the development environment. After you verify that the IDocs are valid, you can remove IDoc validation in the production environment to improve session performance.

Caching Inbound IDoc and DMI Data

The PowerCenter Integration Service creates caches in memory for SAP/ALE IDoc Prepare transformations and SAP DMI Prepare transformations. The transformations receive inbound data from upstream transformations and prepare the segment data. The PowerCenter Integration Service stores this data in the cache.

Configure the cache size in the session properties. Default cache size is 10 MB. You can set the cache size equal to 20 percent of the available memory for the system. For optimum session performance, calculate the cache size based on factors such as processing overhead and the size of the source data.

If you configure a large cache size, the PowerCenter Integration Service may run out of memory. If you configure a cache size that is more than the available memory for the system, the PowerCenter Integration Service fails the session.

If the PowerCenter Integration Service requires more memory than the configured cache size, it stores overflow values in cache files. Since paging to disk can slow session performance, configure the cache size to store data in memory. You configure the directory where cache files are stored in the session properties. When the session completes, the PowerCenter Integration Service releases cache memory and deletes the cache files. You may find cache files in the cache directory if the session does not complete successfully.

Steps to Configure Sessions for IDoc Mappings Using ALE

You can configure session properties for IDoc sessions using ALE.

To configure an IDoc session:

1. In the Task Developer, double-click an SAP session to open the session properties.
2. If you are configuring an outbound IDoc session, select a recovery strategy from the General Options in the Properties tab.

To enable message recovery, select Resume from Last Checkpoint.

If you enable message recovery, you can configure a value for the recovery cache folder from the Properties settings of the Mapping tab (Sources node). Or, use the default cache folder $PMCacheDir\.

3. In the Config Object tab, configure advanced settings, log options, and error handling properties.
4. Click the Mapping tab.
5. From the Connections settings on the Mapping tab (Sources node), select the connection values for the sources.

   If you are configuring an outbound IDoc session, select an SAP_ALE_IDoc_Reader application connection for the Application Source Qualifier associated with the SAPALEIDoc source definition.
6. If you are configuring an outbound IDoc session, click Properties.
7. Optionally, edit the values for the Idle Time, Message Count, and Reader Time Limit terminating conditions.

   The Workflow Manager assigns the following default values to the terminating conditions:

Terminating Condition	Default Value	Description
Idle Time	-1	SAP can remain idle for an infinite period of time before the PowerCenter session ends.
Packet Count	-1	PowerCenter Integration Service can read an infinite number of packets before the session ends.
Reader Time Limit	0	PowerCenter Integration Service can read IDocs from SAP for an infinite period of time.

8. Optionally, configure real-time flush latency.

   By default, the PowerCenter Integration Service does not run the session in real time. Default value is 0.
9. On the Targets node, enter the connection values for the targets in the mapping.

   If you are configuring an inbound IDoc session, select an SAP_ALE_IDoc_Writer application connection for the SAPALEIDoc target definition.
10. If you are configuring an inbound IDoc session, click the Properties settings.
11. Edit the following properties:

Property	Description
Packet Size	Number of IDocs you want the PowerCenter Integration Service to send in a packet to SAP.
Number of Retries	Number of times you want the PowerCenter Integration Service to attempt to connect to the SAP system.

Property	Description
Delay Between Retries	Number of seconds you want the PowerCenter Integration Service to wait before attempting to connect to the SAP system if it could not connect on a previous attempt.
Send IDoc Based on	Select one of the following options: - Packet Size. The PowerCenter Integration Service commits IDocs to SAP based on the value you set for the Packet Size property. The PowerCenter Integration Service collects IDoc messages until the total count reaches the packet size. It then sends the messages as a packet to SAP. - Commit Call. The PowerCenter Integration Service commits IDocs to SAP based on the commit properties at every commit point.

Do not select Generate Request ID. Use this property only when you configure send request workflows for business content integration.

12. On the Transformations node, edit the following properties, depending on whether you are configuring an inbound or outbound IDoc session:

Property	Outbound/ Inbound	Description
Duplicate Parent Row Handling	Both	Determines how the PowerCenter Integration Service handles duplicate parent rows. Choose one of the following values: - First Row. The PowerCenter Integration Service passes the first duplicate row to the target. The PowerCenter Integration Service rejects rows with the same primary key that it processes after this row. - Last Row. The PowerCenter Integration Service passes the last duplicate row to the target. - Error. The PowerCenter Integration Service passes the first row to the target. Rows that follow with duplicate primary keys increment the error count. The session fails when the error count exceeds the error threshold.
Orphan Row Handling	Both	Determines how the PowerCenter Integration Service handles orphan rows. Choose one of the following values: - Ignore. The PowerCenter Integration Service ignores orphan rows. - Error. The session fails when the error count exceeds the error threshold.
Extended Syntax Check	Both	Checks for IDocs that are not valid. Validating IDocs can decrease session performance.
Row Level Processing	Outbound	PowerCenter Integration Service can process each row of an outbound IDoc according to the IDoc metadata and pass it to a downstream transformation. If you enable row-level processing, the PowerCenter Integration Service does not validate IDocs. Select to increase session performance.

Property	Outbound/ Inbound	Description
NULL Field Representation	Inbound	Determines how the PowerCenter Integration Service handles fields with a null value when preparing the data in IDoc format. Choose one of the following values: - Blank. The PowerCenter Integration Service inserts all blanks for the field. - Slash (/). The PowerCenter Integration Service inserts a single slash (/) for the field.
Cache Directory	Inbound	Default directory used to cache inbound IDoc or DMI data. By default, the cache files are created in a directory specified by the variable $PMCacheDir. If you override the directory, make sure the directory exists and contains enough disk space for the cache files. The directory can be a mapped or mounted drive.
Cache Size	Inbound	Total memory in bytes allocated to the PowerCenter Integration Service for the caching of data prepared by SAP/ALE IDoc Prepare or SAP DMI Prepare transformations. Default is 10 MB.

13. Click OK.

Error Handling for IDoc Sessions Using ALE

A session to read outbound IDocs or write inbound IDocs using ALE fails if the PowerCenter Integration Service encounters a row error. This is because the PowerCenter Integration Service validates IDocs for consistency by group before it writes the data to the target. Failing the session ensures data consistency.

Running IDoc Workflows Using ALE

When the PowerCenter Integration Service writes inbound IDocs to the SAP system, SAP does not report status details back to PowerCenter. Therefore, if SAP fails to post an IDoc or if an error occurs after PowerCenter calls SAP, the PowerCenter session log does not contain the reason for the error. However, you may be able to access detailed information from within SAP. If a PowerCenter call to SAP fails, PowerCenter writes the error to the session log.

Troubleshooting IDoc Workflows Using ALE

The session failed while writing IDocs to the SAP system. The session log shows that the session failed but provides no detailed information.

When PowerCenter writes inbound IDocs to the SAP system, SAP does not report status details back to PowerCenter. If SAP fails to post an IDoc, for example, the PowerCenter session log does not contain the reason for the error. However, you may be able to access detailed information from within SAP.

Part IV: Data Integration Using BAPI/RFC Functions

This part contains the following chapters:

- BAPI/RFC Transformation, 161
- BAPI/RFC Sessions, 171

CHAPTER 14

BAPI/RFC Transformation

This chapter includes the following topics:

- BAPI/RFC Transformation Overview, 161
- BAPI/RFC Transformation Components, 162
- Creating a BAPI/RFC Transformation, 165
- Using a BAPI/RFC Transformation in Mappings, 167
- Troubleshooting a BAPI/RFC Transformation, 170

BAPI/RFC Transformation Overview

The BAPI/RFC transformation is an active transformation. It can process data in mySAP applications. Use a BAPI/RFC transformation to make a BAPI/RFC call in SAP. BAPI/RFC calls include requests to the SAP system, such as creating, changing, or deleting data in mySAP applications. For example, if you want to update sales order data in SAP, create a BAPI/RFC transformation from the function BAPI_SALESORDER_CHANGE.

BAPI/RFC functions use function parameter values to perform tasks. They can have the following parameters:

- **Scalar input parameters.** Scalar input values. Some BAPI functions require scalar input values to perform tasks. For example, some BAPI functions require scalar inputs to change data.
- **Scalar output parameters.** Scalar output values that a BAPI function returns after performing a task.
- **Table parameters.** SAP structures with more than one row. Table parameters can be input, output, or both. Input table parameters pass table input values to a BAPI/RFC function. For example, some BAPI/RFC functions to change data require table inputs.

The BAPI/RFC transformation can contain the following groups:

- **Scalar input.** Input group for scalar parameters. It contains a port for each scalar input parameter. The group name is SCALAR_INPUT.
- **Table type scalar input.** Input group for table type parameters. It contains one port for table type parameters and one Integration ID port. The group name is TTYPE_<parameter name>.
- **Table input.** One group for each table parameter. The ports represent import structures. The group name is TABLE_INPUT_<structure name>. For example, TABLE_INPUT_EXTERNAL_DOCUMENTS.
- **Scalar output.** Output group for scalar parameters. It contains a port for each scalar output parameter. The group name is SCALAR_OUTPUT.

- **Table type scalar output.** Output group for table type parameters. It contains one port for table type parameters and one Integration ID port. The group name is TTYPE_<parameter name>.
- **Table output.** One group for each table parameter. The ports represent output structures. The group name is TABLE_OUTPUT_<structure name>.
- **Error output.** Passes data from data conversion errors and invalid BAPI/RFC calls.

All groups, except the error output group, contain an Integration ID port that requires an input value.

BAPIs with Nested Structures

You can create a BAPI/RFC transformation for a BAPI with nested structures. A BAPI/RFC transformation includes the following groups: scalar input, scalar output, table input, table output. When the BAPI contains a nested structure, ports for these groups in a BAPI/RFC transformation use the following naming convention:

```
<group_name>_<parameter_name>_<field name>
```

For example:

```
SCALAR_INPUT_PARAM1_FIELD1
```

If there are multiple input or output structures, the BAPI/RFC transformation includes each structure parameter name in the port names. For example, BAPI Z_TST2 has the parameter INPUT1, which is of the type ZTYPE1. ZTYPE1 has several components such as FIELD1 and FIELD2. FIELD2 is a component of the type structure. It contains field F1. The naming convention in the BAPI/RFC transformation for FIELD 1 is:

```
SCALAR_INPUT_INPUT1_FIELD1
```

The naming convention in the BAPI/RFC transformation for the field F1 is:

```
SCALAR_INPUT_INPUT1_FIELD2_F1
```

BAPI/RFC Transformation Components

A BAPI/RFC transformation contains the following tabs:

- **Transformation.** You can rename the transformation and add a description on the Transformation tab. BAPI transformations are reusable and non-reusable.
- **Ports.** View ports on the ports tab.
- **Properties.** Configure transformation properties, such as the runtime location.
- **Metadata Extensions.** Create a non-reusable metadata extension to extend the metadata of the BAPI/RFC transformation. Configure the extension name, datatype, precision, and value. You can also promote a metadata extension to be reusable if you want to make it available to all transformations.
- **Port Attribute Definitions.** Attributes that apply to all ports in the transformation. You cannot create port attribute definitions for BAPI/RFC transformations.
- **Customize BAPI.** You can view the BAPI/RFC parameters. You can view and edit the return structure properties that you selected during import. You can also generate XSD files for table type parameters.

Configuring BAPI/RFC Transformation Properties

Configure transformation properties on the Properties tab.

The following table describes the BAPI/RFC transformation properties:

Option	Description
Runtime Location	Leave this property blank.
Tracing Level	Amount of detail displayed in the session log for this transformation. Default is Normal.
Is Partitionable	Indicates if you can create multiple partitions in a pipeline that uses this transformation: - No. The transformation cannot be partitioned. The transformation and other transformations in the same pipeline are limited to one partition. - Locally. The transformation can be partitioned, but the PowerCenter Integration Service must run all partitions in the pipeline on the same node. Choose Local when different partitions of the BAPI/RFC transformation must share objects in memory. - Across Grid. The transformation can be partitioned, and the PowerCenter Integration Service can distribute each partition to different nodes. Default is No.
Inputs Must Block	Indicates if the procedure associated with the transformation must be able to block incoming data. Default is disabled.
Update Strategy Transformation	Indicates if this transformation defines the update strategy for output rows. Default is disabled.
Transformation Scope	Indicates how the PowerCenter Integration Service applies the transformation logic to incoming data: - Row - Transaction - All Input Set the transformation scope to Transaction when you configure the BAPI session for real time. Otherwise, set the transformation scope to All Input. Default is All Input.
Generate Transaction	Indicates if this transformation can generate transactions. When enabled, the transformation generates transactions for all output groups. Default is disabled.
Requires Single Thread Per Partition	Indicates if the PowerCenter Integration Service processes each partition at the procedure with one thread. When you enable this option, the procedure code can use thread-specific operations. Default is enabled if you import the BAPI/RFC transformation in the Transformation Developer. If you use another method to the create the transformation, default is disabled.
Output is Deterministic	Indicates whether the transformation generates consistent output data between session runs. You must enable this property to perform recovery on sessions that use this transformation.

Important: If you configure a transformation as repeatable and deterministic, it is your responsibility to ensure that the data is repeatable and deterministic. If you try to recover a session with transformations that do not produce the same data between the session and the recovery, the recovery process can result in corrupted data.

Customizing BAPI/RFC Properties

After you import a BAPI/RFC transformation, you can perform the following tasks on the Customize BAPI tab:

- View and edit parameter properties and return structures.
- Generate XSD files for table type parameters.

- Change the precision of the table type parameters defined in the transformation.

Viewing and Editing Parameter Properties and Return Structures

You can view parameter properties and return structures and change the direction of the BAPI/RFC table parameters. The direction of the table parameters determine which groups in the transformation are input groups and which are output groups.

If the BAPI return structure is custom, you can edit the return structure properties you selected during import for reusable BAPI/RFC transformations. Otherwise, you can view the BAPI/RFC return structure parameters.

Note: You can customize return structure properties when you import a BAPI/RFC transformation.

Click Launch Customize BAPI Interface to view parameter properties and return structures and change the direction of the BAPI/RFC table parameters.

The following table describes properties on the Parameters tab:

Property Name	Description
Name	Name of the export, import, and table parameters and columns.
Associated Type	Definition of the parameter in SAP.
Optional	Indicates if the PowerCenter Integration Service is required to pass a value to the parameter when it calls the BAPI.
Direction	Indicates if the parameter requires input or provides output. Values are: Input, Output, Both, or None.
Datatype	Object datatype.
Precision	Object precision.
Scale	Object scale.
Default	Default value of the parameter in SAP, if any. SAP uses this value when the PowerCenter Integration Service does not pass a value to SAP.

The following table describes the information that you can edit on the Return Structure tab:

Custom Return Structure Parameter	Description
Return Structure	Return parameter name to determine the status of function calls. Value is RETURN if the BAPI contains a default return structure. If the BAPI does not contain a default return structure, select any table output parameter or scalar output parameter of type STRUCTURE. Default is None.
Status Field	Required if you select a value for the return structure. Select a field from the structure for status. If you select None for the return structure, this parameter is blank.
Text Field	Required if you select a value for the return structure. Select a field from the structure for status messages. If you select None for the return structure, this parameter is blank.

Custom Return Structure Parameter	Description
Status Indicator For Warning	Enter an indicator message for warning. If you select None for the return structure, the value is W. Default is W.
Status Indicator for Error	Enter an indicator message for error. If you select None for the return structure, the value is E. Default is E.
Status Indicator for Abort	Enter an indicator message for abort. If you select None for the return structure, the value is A. Default is A.

To view and edit BAPI/RFC transformation parameters and return structure:

1. Edit a BAPI/RFC transformation.
2. Click the Customize BAPI tab.
3. Click Launch Customize BAPI Interface.
4. On the Parameters tab, optionally edit the direction of table parameters.
5. Click the Return Structure tab.

   The Customize BAPI dialog box displays the BAPI/RFC parameters.
6. Optionally, edit the return structure properties.

Generating XSD Files for Table Type Parameters

The PowerCenter Client generates XSD files for table type parameter groups in a BAPI/RFC transformation that are used in scalar input and scalar output parameters. If you export and then import a BAPI/RFC transformation, or if you use a BAPI/RFC transformation that you created on a different PowerCenter Client machine, you need to regenerate the XSD files.

Use the XSD files to create XML Generator or XML Parser transformations to pass data to the table type input data port or to pass data from the table type output data port.

To regenerate XSD files:

1. Edit a BAPI/RFC transformation.
2. Click the Customize BAPI tab.
3. Click Generate XSD Files.

Creating a BAPI/RFC Transformation

You can create a BAPI/RFC transformation in the Transformation Developer or the Mapping Designer. If you click Transformation > Import SAP BAPI/RFC in the Transformation Developer, you can select one or more BAPIs to import. The BAPI/RFC Wizard creates a transformation for each BAPI you import. If you click the BAPI icon in the Transformation Developer or import a BAPI/RFC transformation, you can select one BAPI to import.

Connect to SAP to import BAPI/RFC metadata. You can enter a DEST entry from the `sapnwrfc.ini` file or a host name to connect to SAP. When you start the wizard, click Connect String to enter a DEST entry. Click Host Name to enter a host name.

You can enter the following connection string parameters when you connect to SAP:

Field	Description
Connect String	Required. DEST entry for a connection to a specific SAP application server or DEST entry for a connection that uses SAP load balancing defined in the `sapnwrfc.ini` file.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

You can enter the following host name values when you connect to SAP:

Field	Description
Host Name	Required. DEST entry for a connection to a specific SAP application server or DEST entry for a connection that uses SAP load balancing defined in the `sapnwrfc.ini` file.
System No	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

You can filter BAPIs based on the name or the description of the BAPI. When you enter a filter condition, you can enter an exact name or description. Or, you can perform wildcard search. Use the following filter syntax rules and guidelines:

- To filter on a description, enter a filter condition of 20 characters or fewer.
- Use the percent sign (%) as a wildcard search for multiple characters. For example, to find a BAPI whose name begins with the letter “I”, enter `I%` as the filter criterion.
- Use an underscore (_) as a wildcard search for single characters.
- If you filter BAPIs by description, the filter condition is case sensitive. If you filter BAPIs by name, the filter condition is not case sensitive.

When you select a BAPI to import, you can view information about the BAPI parameters on the Parameters tab. You can also view and edit information about the return structure on the Return Structure tab.

When you create a BAPI/RFC transformation with table type parameters, the PowerCenter Client generates an XSD file for each table type group in the transformation. You cannot import BAPIs with a parameter whose

name includes a forward slash (/) and was defined using a table type parameter. The Designer does not generate the XSD file for the corresponding parameter.

The Designer uses the following naming convention for XSD files:

```
<parameter_name>-<associated_type_name>.xsd
```

If the parameter name or associated type name includes a name space, the Designer replaces the forward slash (/) with a pound sign (#).

You cannot create a transformation from a BAPI that contains a field with the same name as its parent structure. If you create a transformation from a BAPI with a parameter defined with a deep structure or table or a table type with a deep structure in the line type, the wizard shows the root and leaf nodes. It does not show intermediate nodes.

To create a BAPI/RFC transformation:

1. In the Transformation Developer, click Transformation > Create, select SAP BAPI/RFC transformation, enter a transformation name, and click Create. Or, in the Transformation Developer, click Transformation > Import SAP BAPI/RFC. Or, in the Mapping Designer, click Transformation > Create, select SAP BAPI/RFC transformation, enter a transformation name, and click Create.

   Step 1 of the wizard appears.
2. Enter connection string parameters to connect to SAP with the `sapnwrfc.ini` file. Or, click Host Name and enter host name parameters to connect to SAP. Click Next.
3. Optionally, enter a filter condition to filter BAPIs by name or description.
4. Click Get Objects.

   The wizards displays BAPIs that you can import.
5. Expand the list of objects and select the BAPIs you want to import.

   If you clicked Transformation > Import SAP BAPI/RFC to start the wizard, you can select multiple BAPIs to import. Otherwise, you can import one BAPI.
6. Optionally, select a BAPI name to view information about the BAPI parameters on the Parameters tab.
7. Optionally, click the Return Structure tab to edit information about the BAPI return structure.
8. Click Next.

   Step 3 of the wizard provides information about the BAPI you want to import.
9. Optionally, edit the name of the BAPI/RFC transformation.
10. Click Finish.

Using a BAPI/RFC Transformation in Mappings

When you configure a mapping with a BAPI/RFC transformation, use the following rules and guidelines:

- Pass a value to the Integration ID ports in the scalar input group and all mandatory table input groups of the BAPI/RFC transformation.
- Add an XML transformation when the BAPI/RFC contains table type parameters groups.
- Add a target definition if you want to receive BAPI/RFC function call errors from the BAPI error group.

You can also configure a mapping with a BAPI/RFC transformation to perform the following tasks:

- Pass input data to BAPI/RFC function input parameters.
- Use SAP system variables as default values of scalar non-structure parameters.

- Run a BAPI session in real time.

Adding XML Transformations for Table Type Parameters

Add an XML Generator transformation when the BAPI/RFC transformation contains a table type parameters input group. Add an XML Parser transformation when the BAPI/RFC transformation contains a table type parameters output group.

Create the XML transformation from the XSD file that the PowerCenter Client generates when you create the BAPI/RFC transformation. The XSD files are in the following location:

```
<Workspace file path\SAPFiles
```

Select **Tools** > **Options** > **Workspace File**, in the Designer to configure the workspace file path.

To pass the integration ID from an XML Generator transformation, add a pass-through port in the XML Generator transformation. You can use the reference port that you select when you add a pass-through port to propagate the integration ID.

If you know the length of the data from the SAP system, configure the precision of the ports in the XML Generator and XML Parser transformations to be equal to the precision of the corresponding port in the BAPI/RFC transformation. Also, configure the precision of the ports in the BAPI/RFC transformation on the Customize BAPI tab in the transformation properties.

If the precision of the output port in the XML Generator transformation is less than the corresponding port in the BAPI/RFC transformation and the precision of the input port in the XML Parser transformation is greater than the corresponding port in BAPI/RFC transformation, session performance might decrease. If the precision of the output port in the XML Generator is greater than the corresponding port in the BAPI/RFC transformation and the precision of the input port in the XML Parser transformation is less than the corresponding port in the BAPI/RFC transformation, data loss or data inconsistency might occur.

If you do not know the length of the data from the SAP system, enable streaming for the XML Parser transformation in the session properties.

Related Topics:

- "Configuring a Session for BAPIs with Table Type Parameters" on page 174

Working with Function Input Data for BAPI/RFC Functions

Use the following guidelines when passing data to BAPI/RFC function input parameters:

- When the function input parameter datatype is INT1 or NUMC, provide positive values for the function input.
- When the source input data for an RFC/BAPI function is of the integer datatype, do not use string data in the source definition. Otherwise, the session fails.
- If the input data for an RFC/BAPI function mapping has a higher scale than the SAP metadata specification, the PowerCenter Integration Service rounds the data to comply with the SAP metadata. When you run a session in high precision mode, the session can fail due to overflow if the round-off data cascades to the precision digits. For example, the datatype and precision for an RFC/BAPI function parameter is DEC (6,5). The input data you pass to the function parameter is 9.99999. When the PowerCenter Integration Service processes the input data, it rounds the input data to 10, which is not compatible with the SAP metadata. The PowerCenter Integration Service fails the session.

Configuring a Target Definition for BAPI/RFC Error Output

You can add a target definition to a BAPI/RFC mapping if you want to receive input data corresponding to a BAPI/RFC function call or data conversion errors from SAP. Create a target definition with a column of the String datatype and precision of 65535. Connect the column in the target definition to the Error Output Group in the BAPI/RFC transformation.

The PowerCenter Integration Service writes the error output data up to 65,535 characters to the target in comma-delimited format. If the error output data is longer than 65,535 characters, the PowerCenter Integration Service truncates the data.

The following example shows sample error output data:

```
SCALAR_INPUT_Integration_Id=1,SCALAR_INPUT_FKEY_IMP=1,SCALAR_INPUT_FCHAR_IMP=a,SCALAR_INP
UT_FCLNT_IMP=100,SCALAR_INPUT_FCUKY_IMP=USD,SCALAR_INPUT_FCURR_IMP=61.11,SCALAR_INPUT_FDE
C_IMP=1232.12300,
TABLE_INPUT_ALLDTYPES_IN_Integration_Id=1,TABLE_INPUT_ALLDTYPES_IN_FKEY=1,TABLE_INPUT_ALL
DTYPES_IN_FCHAR=a,TABLE_INPUT_ALLDTYPES_IN_FCLNT=100,TABLE_INPUT_ALLDTYPES_IN_FCUKY=USD,T
ABLE_INPUT_ALLDTYPES_IN_FCURR=61.11
```

System Variables

SAP uses system variables to set default values for some BAPI import parameters. They provide information, such as current date and time for the operating system on which SAP runs. System variables start with “SY-”. For example, SY-DATLO represents the local date of the SAP system.

PowerCenter provides values for some system variables when it needs them to define default input values for BAPI/RFC parameters. It uses the values as default input values for some ports of BAPI/RFC transformations. It uses these default values when there is no input for a port or when the port is not connected to an upstream transformation or source.

PowerCenter provides the following system variables:

System Variable Name	Description
SY-LANGU	Log in language from the SAP R/3 application connection properties.
SY-MODNO	RFC handle value.
SY-MANDT	Value taken from the SAP R/3 application connection properties.
SY-DATUM	Local date of the PowerCenter Integration Service machine processing the data.
SY-UZEIT	Local time of the PowerCenter Integration Service machine processing the data.
SY-UNAME	Logon user ID from the SAP R/3 application connection properties.
SY-HOST	SAP host name from the SAP R/3 application connection properties.

Real-time BAPI/RFC Mappings

To configure a BAPI mapping to run in real time, use the following rules and guidelines:

- Include a real-time source in the mapping.
- Set the transformation scope in the BAPI/RFC transformation to Transaction.
- You can only link input ports from one upstream transformation to a BAPI/RFC transformation.

Troubleshooting a BAPI/RFC Transformation

The BAPI/RFC Import Wizard or Customize BAPI dialog box fails to launch with the following error message:

```
The wizard could not be launched.
```

The system might not have enough memory. Close any unnecessary programs and restart the Designer.

The Customize BAPI dialog box shows the following error message:

```
Some special characters were found in the metadata. The customization dialog could not
be displayed.
```

If the repository is non-Unicode and the BAPI/RFC description is not in English, BAPI/RFC information does not appear in the Customize BAPI dialog box. Reimport the BAPI/RFC transformation. Edit the BAPI/RFC information on the Return Structure tab of the BAPI/RFC Import Wizard.

I imported a BAPI/RFC transformation with a table type parameter, but the Designer did not generate the XSD file.

You might have imported a BAPI with a parameter whose name includes a forward slash (/) and was defined using a table type parameter. The Designer does not generate the XSD file for the corresponding parameter.

CHAPTER 15

BAPI/RFC Sessions

This chapter includes the following topics:

- BAPI/RFC Sessions Overview, 171
- Configuring a Session with a BAPI/RFC Transformation, 171
- Configuring a Session for BAPIs with Table Type Parameters, 174
- Configuring a Real-time BAPI/RFC Session, 174

BAPI/RFC Sessions Overview

When you run a session with a BAPI/RFC transformation, the PowerCenter Integration Service makes a BAPI/RFC call to SAP to process the data. When it reaches the end of file, the PowerCenter Integration Service makes the BAPI/RFC call. Depending on the session configuration, the PowerCenter Integration Service can also issue a commit.

The BAPI/RFC call is based on the input data of the Integration ID ports. The PowerCenter Integration Service makes one call to SAP for each Integration ID.

If the BAPI/RFC call fails or if there is a data conversion error, SAP passes the data for the integration ID in comma-separated format to an error output group. If the mapping contains a target instance that is connected to the error output group, the PowerCenter Integration Service writes the data to the target.

Configuring a Session with a BAPI/RFC Transformation

When you configure a BAPI/RFC session, you can configure the following session behavior:

- Commit behavior
- Caching
- Error handling
- Partitioning. Each partition makes separate BAPI/RFC calls to SAP.
- Verbose logging. The session log contains the information about the BAPI/RFC call return code and status. It also includes detailed messages.

When you configure a session, select an SAP RFC BAPI Interface application connection for the BAPI/RFC transformation.

Committing Data to SAP

The PowerCenter Integration Service can make BAPI/RFC calls to SAP without issuing a commit. Or, it can commit data after a specified number of BAPI/RFC calls. It uses a BAPI/RFC commit function to commit the data. The PowerCenter Integration Service can wait until a commit is complete before continuing the session.

The PowerCenter Integration Service can commit data after a number of BAPI/RFC calls if the BAPI/RFC call writes data to the SAP database or changes data in the SAP database. When the PowerCenter Integration Service issues a commit BAPI/RFC call, SAP commits the data to the database.

If you run a session for a BAPI that does not change data in the SAP database, the PowerCenter Integration Service does not need to issue a commit BAPI/RFC call to commit data. For example, if the BAPI performs an SAP process, the PowerCenter Integration Service does not need to commit data during the session.

The following table describes the commit properties you can configure for a BAPI/RFC transformation in a session:

Property	Description
Perform Commit	Issues a commit with a BAPI commit function. If you disable this option, the PowerCenter Integration Service does not call a commit function. It also ignores the commit interval and does not wait for a commit to continue session processing. Default is enabled.
Perform Commit Wait	Waits until the commit is complete to continue session processing. SAP sends a return call to the PowerCenter Integration Service when the commit is complete. If you disable this option, the PowerCenter Integration Service continues to process data during the commit. Default is enabled.
Commit Interval	Number of BAPI/RFC calls after which the PowerCenter Integration Service issues a commit. To issue a commit, the PowerCenter Integration Service makes a BAPI/RFC commit call. If you disable the Perform on Commit option, the PowerCenter Integration Service ignores this value. Default is 1000.
Commit Function	Commit function the PowerCenter Integration Service calls when it issues a commit. Select one of the following BAPIs: - ABAP4_COMMIT_WORK - BAPI_TRANSACTION_COMMIT If you disable the Perform on Commit option, the PowerCenter Integration Service ignores this value. Default is BAPI_TRANSACTION_COMMIT.

Caching BAPI Data

The PowerCenter Integration Service creates caches in memory for BAPI/RFC transformations. A BAPI/RFC transformation receives data from upstream transformations in the mapping and prepares the data. It stores this data in the cache.

Configure the cache size in the session properties. Default cache size is 10 MB. You can set the cache size equal to 20 percent of the available system memory. For optimal session performance, calculate the cache size based on factors, such as processing overhead and the size of the source data. For example, if the mapping or session contains other transformations that use auto caching, do not increase the cache size beyond 20 percent. If there are more than five input groups in the BAPI/RFC transformation, increase the cache size to 30 to 40 percent of the available system memory.

If you configure a large cache size, the PowerCenter Integration Service might run out of disk space and fail the session.

The following table shows the data caching properties you can configure for a BAPI/RFC transformation in a session:

Property	Description
Cache Directory	Default directory used to cache BAPI data. By default, the cache files are created in a directory specified by the variable $PMCacheDir. If you override the directory, make sure the directory exists and contains enough disk space for the cache files. The directory can be a mapped or mounted drive.
Cache Size	Total memory in bytes allocated to the PowerCenter Integration Service to cache data prepared for BAPI/RFC transformations. Default is 10 MB.

Configuring Error Handling

The PowerCenter Integration Service increments the error count in the following cases:

- **When it processes an orphan row.** Any value of the table input Integration ID that does not match the value of the scalar input Integration ID results in an orphan row.
- **When it processes a duplicate row.** Any Integration ID value in the scalar input group that is not unique results in a duplicate row.
- **When it receives a BAPI/RFC error.** The PowerCenter Integration Service can receive BAPI/RFC errors with the error code W (warning), A (abort), or E (error).

The following table shows the error handling properties you can configure for a BAPI/RFC transformation in a session:

Property	Description
Duplicate Parent Row Handling	Determines how the PowerCenter Integration Service handles duplicate parent rows during a session. Choose one of the following values: - First Row. The PowerCenter Integration Service passes the first duplicate row to the target. The PowerCenter Integration Service rejects rows with the same primary key that it processes after this row. - Last Row. The PowerCenter Integration Service passes the last duplicate row to the target. - Error. The PowerCenter Integration Service passes the first row to the target. Rows that follow with duplicate primary keys increment the error count. The session fails when the error count exceeds the error threshold. Default is First Row.
Orphan Row Handling	Determines how the PowerCenter Integration Service handles orphan rows during a session. Choose one of the following values: - Ignore. The PowerCenter Integration Service ignores orphan rows. - Error. The session fails when the error count exceeds the error threshold. Default is Ignore.
Continue on Error	PowerCenter Integration Service continues the session when it receives a BAPI/RFC error. Default is enabled.

Configuring a Session for BAPIs with Table Type Parameters

If you do not know the length of the data from the SAP system, enable streaming for the XML Parser transformation in the session properties.

The PowerCenter Integration Service passes BAPI/RFC data to the XML Parser transformation in segments based on the precision of the table type scalar output port. The XML Parser transformation stores the data in a buffer until it reaches the end of file. The PowerCenter Integration Service then passes the data to downstream transformations.

Configuring a Real-time BAPI/RFC Session

You can run a session with a BAPI/RFC transformation in real time. Run a real-time BAPI/RFC session to call BAPIs that do not change data in the SAP database. For example, you receive a customer order from JMS. You want to get customer address data from SAP in real time and load the customer order data and address data to the target. Create a mapping with a JMS source and BAPI/RFC transformation and configure the session to run in real time.

Important: If you run a real-time session to call BAPIs to write data to SAP, data duplication or inconsistency can occur.

During a real-time BAPI/RFC session, the PowerCenter Integration Service makes a BAPI/RFC call to the SAP system when it reaches a commit interval or flush latency value. When the PowerCenter Integration Service issues a commit, it removes the data from the cache.

Because the PowerCenter Integration Service issues a commit before end of file in a real-time session, verify that the BAPI/RFC call conforms to a transaction boundary. Make sure that the rows intended for a single BAPI/RFC call are in the same transaction. Otherwise, the PowerCenter Integration Service might process data that you intended for a single BAPI/RFC call in different calls.

To configure a real-time BAPI/RFC session, configure the session properties for the source for real time.

Related Topics:

- "Real-time BAPI/RFC Mappings" on page 169

Part V: Data Migration

This part contains the following chapters:

- Creating Data Migration Mappings, 176
- Migrating Legacy Data into SAP IS-U/CCS, 182

CHAPTER 16

Creating Data Migration Mappings

This chapter includes the following topics:

- Creating Data Migration Mappings Overview, 176
- Working with the SAP DMI Prepare Transformation, 176
- Creating a Flat File Target for DMI Data, 181
- Configuring Sessions for DMI Mappings, 181

Creating Data Migration Mappings Overview

You can migrate data from legacy applications, other ERP systems, or any number of other sources into mySAP applications. Create an SAP Data Migration Interface (DMI) mapping to prepare data for migration into mySAP applications. After you create a DMI mapping, you can configure a session. The PowerCenter Integration Service extracts the data from the data source and prepares the data in an SAP format flat file that you can load into SAP.

Creating a DMI Mapping

To migrate data into SAP, create a DMI mapping with the following components:

- **Source definition.** Reads data from the source system.
- **Source Qualifier transformation.** Determines how the PowerCenter Integration Service reads data from the source.
- **SAP DMI Prepare transformation.** Processes the data to migrate into SAP.
- **Flat file target definition.** Loads data to the target.

Working with the SAP DMI Prepare Transformation

The SAP DMI Prepare transformation receives data from upstream transformations in the mapping and interprets the segment data. After you create an SAP DMI Prepare transformation, you can edit the transformation to change the data segments you want to include in the transformation. When you edit the

transformation, you can also view details about the segments. To view details, double-click the title bar of the transformation and select the DMI Display tab.

DMI Primary and Foreign Keys

A DMI message is organized hierarchically with one top-level parent segment and one or more second-level child segments. Second-level child segments may have one or more third-level child segments.

To maintain the structure of the DMI data, the SAP DMI Prepare transformation uses primary and foreign keys. The top-level parent segment has a primary key. Each child segment has a primary key and a foreign key. The foreign key of each child segment references the primary key of its parent segment. For example, the foreign key of a second-level child segment references the primary key of the top-level parent segment. Similarly, the foreign key of a third-level child segment references the primary key of the second-level child segment.

The SAP DMI Prepare transformation groups incoming data based on the values in the primary and foreign key fields. The Control Input group of the SAP DMI Prepare transformation represents the parent segment. All other groups of the SAP DMI Prepare transformation except the DMI_Prepare_Error_Output group represent second-level or third-level child segments.

Note: The DMI_Prepare_Error_Output group is used for processing invalid DMI documents. You can process invalid DMI documents the same way as invalid IDocs.

The following table shows the groups of the SAP DMI Prepare transformation and the fields used for the primary and foreign keys:

Group(s)	Field	Description
Control Input Group	GPK_DOCNUM	Primary key of the parent segment.
Child Segment 1	GPK_<Child1_name>	Primary key of Child Segment 1.
Child Segment 1	GFK_DOCNUM_<Child1_name>	Foreign key of Child Segment 1 references the primary key of the parent segment.
Child Segment A of Child Segment 1	GPK_<Child1A_name>	Primary key of Child Segment A of Child Segment 1.
Child Segment A of Child Segment 1	GFK_<Child1_name>_<Child1A_name>	Foreign key of Child Segment A of Child Segment 1 references the primary key of Child Segment 1.
Child Segment 2	GPK_<Child2_name>	Primary key of the child segment.
Child Segment 2	GFK_DOCNUM_<Child2_name>	Foreign key of Child Segment 2 references the primary key of the parent segment.
Child Segment B of Child Segment 2	GPK_<Child2B_name>	Primary key of Child Segment B of Child Segment 2.
Child Segment B of Child Segment 2	GFK_<Child2_name>_<Child2B_name>	Foreign key of Child Segment B of Child Segment 2 references the primary key of Child Segment 2.

Each value for the GPK_<name> field must be unique. Each GFK_<parent_name>_<group_name> field must reference the primary key of its parent segment.

The following example shows the relationship of primary and foreign keys in a DMI document named ABSEN1 with four child segments:

Group	Field	Primary/Foreign Keys
CONTROL_INPUT_ABSEN1	GPK_DOCNUM	P1
E2ABSE1	GPK_E2ABSE1	C1
-	GFK_DOCNUM_E2ABSE1	P1
E2ABSE2	GPK_E2ABSE2	C2
-	GFK_DOCNUM_E2ABSE2	P1
E2ABSE2A	GPK_E2ABSE2A	C2A
-	GFK_E2ABSE2_E2ABSE2A	C2
E2ABSE3	GPK_E2ABSE3	C3
-	GFK_DOCNUM_E2ABSE3	P1
E2ABSE3B	GPK_E2ABSE3B	C3B
-	GFK_E2ABSE2_E2ABSE2A	C3
E2ABSE4	GPK_E2ABSE4	C4
-	GFK_DOCNUM_E2ABSE4	P1

The SAP DMI Prepare transformation uses these primary and foreign key relationships to maintain the structure of the DMI data. Any foreign key field that does not match the primary key of its parent segment results in an orphan row. Any primary key field that is not unique results in a duplicate row.

Verify that each DMI document has a unique primary key for the top-level parent segment, each child segment, and that each foreign key matches the primary key of its parent.

Creating an SAP DMI Prepare Transformation

Use the Generate SAP DMI Prepare Transformation Wizard to create an SAP DMI Prepare transformation. The wizard lets you import DMI metadata.

When you create an SAP DMI Prepare transformation, you can import the DMI metadata in the following ways:

- **From file.** Use to import the DMI metadata in the SAP DMI Prepare transformation from a DMI file.
- **Connect to SAP.** Use to import the DMI metadata from the SAP system to use in the transformation.

Generating DMI Metadata to File for Import

If you want to import DMI metadata for an SAP DMI Prepare transformation from file, you can run the RSAPEXP program from the SAP client to generate the metadata. When you run the program, select the range for the DMI metadata you want to generate. The program exports the metadata it generates into a metadata file. For example, you can export the metadata to a file with the *.dmi* extension. You can then use the metadata file to import the metadata into the Designer for use in the SAP DMI Prepare transformation.

To generate DMI metadata using the RSAPEXP program:

1. Use transaction SXDA_TOOLS from the SAP client.
2. Run the RSAPEXP program.
3. Select the range.
4. Optionally, select extended grammar, if available.
5. Click Execute.
6. Click List > Download.
7. From the Save list in file dialog box, select Unconverted.
8. On the Transfer list to a local file dialog box, enter the path where you want to save the metadata file.
9. Click Transfer.

Steps to Create an SAP DMI Prepare Transformation

To create an SAP DMI Prepare transformation:

1. In the Transformation Developer, click the SAP DMI Prepare transformation button.

   The pointer becomes a cross.
2. Click in the Transformation Developer workspace.
3. To import DMI metadata from a file, click Local File.

   To import DMI metadata from the SAP system, go to step 6.
4. Enter the name and path of the file from which you want to import DMI metadata. Or, click Browse to locate the file you want to use.
5. Click Import. Go to step 8.
6. If you are importing DMI metadata from the SAP system, enter the following information:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

7. Click Connect.
8. Expand a Data Transfer Object, select an Activity, and click Next.

   Step 2 of the wizard appears.
9. Click Show Group Status to show the required groups in the Group Status column.
10. Select the DMI segments you want to include in the transformation.

You can manually select the segments you want to include in the transformation. Or, click Select All Segments to include all segments.

Click Clear All Segments to remove all optional segments.

When you select the segments, the transformation uses the following rules to select parent and child segments:

- If you select a segment, all its parent segments are selected, and all its required child segments are selected.
- If you clear a segment, all its child segments are cleared.

11. Click Next.

    Step 3 of the wizard appears. The wizard provides a name for the transformation.

12. Optionally, modify the name of the transformation.

    If you clicked Transformation > Create to create the transformation, you cannot modify the name of the transformation in Step 3 of the wizard. The Designer uses the transformation you entered in the Create Transformation dialog box.

13. Optionally, modify the description of the transformation.
14. Click Finish.

Editing an SAP DMI Prepare Transformation

You can edit an SAP DMI Prepare transformation to change the data segments you want to include in the transformation. You can also modify the name and description of the transformation.

To edit an SAP DMI Prepare transformation:

1. In the Transformation Developer or Mapping Designer, double-click the title bar of the SAP DMI Prepare transformation.

   The Edit Transformations dialog box appears.

2. Click the DMI Display tab.
3. Click Show Group Status to show which groups are required in the Group Status column.
4. Select the DMI segments you want to include in the transformation.

   You can manually select the segments you want to include in the transformation. Or, click Select All Segments to include all segments.

   Click Clear All Segments to remove all optional segments.

   When you select the segments, the transformation uses the following rules to select parent and child segments:

   - If you select a segment, all parent segments and all required child segments are selected.
   - If you clear a segment, all child segments are cleared.

5. Click OK.

Error Handling with DMI Mappings

The SAP DMI Prepare transformation includes an ErrorDMIData group. You can create a flat file target definition and connect output port of the ErrorDMIData group to an input port in the flat file target. PowerCenter writes data errors in the DMI Prepare transformation to this flat file target.

Creating a Flat File Target for DMI Data

To migrate DMI files to SAP, create a flat file target definition. The definition must have one port with a datatype of String that matches the precision of the DMIData field in the SAP DMI Prepare transformation.

Configuring Sessions for DMI Mappings

You use the same procedures to configure a DMI session as you use to configure a session for inbound IDoc mappings.

CHAPTER 17

Migrating Legacy Data into SAP IS-U/CCS

This chapter includes the following topics:

- Migrating Legacy Data into SAP ISU/CCS Overview, 182
- Configuring Migration Objects in SAP, 183
- SAP IS-U/CCS Transformation Components, 183
- SAP IS-U/CCS Sessions, 186

Migrating Legacy Data into SAP ISU/CCS Overview

Use the SAP IS-U/CCS transformation to migrate data from external data sources or legacy applications into Industry-Specific Utilities/Customer Care and Services (IS-U/CCS). IS-U/CCS is the SAP industry solution for the utilities industry. The SAP IS-U/CCS transformation is an active transformation.

The IS-U/CCS transformation prepares data in a file format that you can load into the IS-U/CCS. The file consists of migration objects. The PowerCenter Integration Service extracts the data from the data source and prepares the data in a file format that you can load into SAP.

Migration Objects

A migration object is a collection of auto structures. Each auto structure consists of fields. The migration objects, auto structures, and fields have a set of pre-determined properties. You can change these properties based on how you want to import the metadata from the SAP system into the IS-U/CCS transformation.

Configuring Migration Objects in SAP

You must use the data model of the SAP IS-U/CCS system while migrating data from a legacy system. Before you import the data model from SAP, you must configure the migration object, the auto-structure, and the fields.

The SAP administrator must install the SAP IS-U/CCS transports on the SAP system before configuring migration objects in SAP.

1. On the SAP Menu screen, enter EMIGALL in the Transaction field.

   The Choose Company dialog box appears.
2. Select the migration company that contains the migration object that you want to configure.

   All the Migration Objects listed under the Company are displayed.
3. Select a migration object that you want to configure.
4. Click **Migration object** > **Change**.
5. Select the auto-structure.
6. Click **Automatic Structure** > **Change**.

   The Structure Maintenance dialog box appears.
7. Select the Generation option if you want the data to be populated into this auto-structure.
8. Optionally, select the **Multiple** option if you want to pass multiple records into the auto-structure for each migration object.
9. Select the field.
10. Click **Field** > **Change**.

    The Field Maintenance dialog box appears.
11. Select the **Processing type** to set the automation data of the migration object.

    The following table lists the processing type options:

Option	Description
Transfer	Data is transferred from third party legacy system.
Via KSM	Data is transferred from the key and status management.

12. Click **Save**.

SAP IS-U/CCS Transformation Components

When you configure an SAP IS-U/CCS transformation, you define the run-time location and the metadata extension of the transformation.

An SAP IS-U/CCS transformation contains the following tabs:

Transformation

You can rename the transformation and add a description on the Transformation tab.

Ports

View ports on the ports tab.

Properties

Configure transformation properties, such as the run-time location.

Metadata Extensions

Create a non-reusable metadata extension to extend the metadata of the SAP IS-U/CCS transformation. Configure the extension name, datatype, precision, and value. You can also promote a metadata extension to be reusable if you want to make it available to all transformations.

Port Attribute Definitions

Attributes that apply to all ports in the transformation. You cannot create port attribute definitions for SAP IS-U/CCS transformations.

ISU/CCS Display

You can view the SAP IS-U/CCS parameters.

SAP IS-U Transformation Properties

Configure transformation properties that determine run-time processing such as partitioning and row processing.

The following table describes the SAP IS-U/CCS transformation properties:

Option	Description
Runtime Location	Leave this property blank.
Tracing Level	Amount of detail displayed in the session log for this transformation. Default is Normal.
Is Partitionable	Indicates if you can create multiple partitions in a pipeline that uses this transformation: - No. The transformation cannot be partitioned. The transformation and other transformations in the same pipeline are limited to one partition. - Locally. The transformation can be partitioned, but the PowerCenter Integration Service must run all partitions in the pipeline on the same node. Choose Local when different partitions of the SAP IS-U/CCS transformation must share objects in memory. - Across Grid. The transformation can be partitioned, and the PowerCenter Integration Service can distribute each partition to different nodes. Default is Locally.
Inputs Must Block	Indicates if the procedure associated with the transformation must be able to block incoming data. Default is disabled.
Update Strategy Transformation	Indicates if this transformation defines the update strategy for output rows. Default is disabled.
Transformation Scope	Indicates how the PowerCenter Integration Service applies the transformation logic to incoming data. All Input option applies the transformation logic on all incoming data.
Generate Transaction	Indicates if this transformation can generate transactions. When enabled, the transformation generates transactions for all output groups. Default is disabled.

Option	Description
Requires Single Thread Per Partition	Indicates if the PowerCenter Integration Service processes each partition at the procedure with one thread.
Output is Deterministic	Indicates whether the transformation generates consistent output data between session runs. You must enable this property to perform recovery on sessions that use this transformation.

Auto Structures and Fields

The **IS-U/CC Display** tab displays the auto structures of the migration object.

Select an auto structure to view the field level details. The following table describes the fields and its properties of an auto structure:

Property	Description
Field Name	Name of the field.
SAP Datatype	SAP datatype of the field.
Precision	SAP precision of the field.
Scale	SAP scale of the field.
Required	Field required or not.
Processing Type	Processing type specified for each field of the automation data of a migration object.
KSM Config	Names of the higher level migration objects.
Format	Internal or external data format.

Creating an SAP IS-U/CCS Transformation

Use the SAP IS-U/CCS Transformation Wizard to create an SAP IS-U/CCS transformation. The wizard connects to the SAP system to import the metadata of the migration objects.

1. In the Transformation Developer, click **Transformation** > **Create**.
2. Select **SAP IS-U/CSS Migration Prepare** transformation, enter a transformation name, and click Create.

   The Generate IS-U/CSS Transformation Wizard appears.
3. Enter connection string parameters to connect to SAP with the `sapnwrfc.ini` file.

   The following table explains the connection string parameters:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.

Field	Description
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

4. Click Connect.

   The wizard displays the companies that you can import.

5. Select the company from which you want to import the metadata.

   A list of migration objects and their descriptions appear.

6. Select the required migration object and click **Next**.

   The wizard displays the auto structure and the corresponding fields and relevant metadata for the selected migration object.

7. Select each auto structure that you want to write data to.
8. Select a code page that matches the code page of the migration object configured in SAP and click Next.
9. Click Finish.

The binary file generated after the transformation is complete is stored at the location you specified when you set up the session details. Copy the generated binary file from this location and place it under the working directory of the SAP system. Use the EMIGALL transaction to import the file into the SAP System.

SAP IS-U/CCS Sessions

When you configure an SAP IS-U/CCS session, you define the session properties that determine how the PowerCenter Integration Service reads data from the legacy data sources and writes to an SAP IS-U/CCS target. You can configure the SAP IS-U/CCS session to allocate the cache memory. You can set file name and the location to write the migration data. You can also configure the session to handle errors and exceptions during the data migration.

Caching SAP IS-U/CCS Data

The PowerCenter Integration Service creates caches in memory for SAP IS-U/CCS transformations. An SAP IS-U/CCS transformation receives data from upstream transformations in the mapping and prepares the data. It stores this data in the cache.

Configure the cache size in the session properties. Default cache size is 10 MB. You can set the cache size equal to 20 percent of the available system memory. For optimal session performance, calculate the cache size based on factors, such as processing overhead and the size of the source data. For example, if the mapping contains other transformations that use auto caching, do not increase the cache size beyond 20 percent. If there are more than five input groups in the SAP IS-U/CCS transformation, increase the cache size to 30 to 40 percent of the available system memory.

If you configure a large cache size, the PowerCenter Integration Service might run out of disk space and fail the session.

The following table describes the data caching properties you can configure for an SAP IS-U/CCS transformation in a session:

Property	Description
Cache Directory	Default directory used to cache SAP IS-U/CCS data. By default, the cache files are created in a directory specified by the variable $PMCacheDir. If you override the directory, make sure the directory exists and contains enough disk space for the cache files. The directory can be a mapped or mounted drive.
Cache Size	Total memory in bytes allocated to the PowerCenter Integration Service to cache data prepared for SAP IS-U/CCS transformations. Default is 10 MB.

SAP IS-U/CCS Migration Data to Import File

Import the legacy system data into an SAP IS-U/CCS transformation. You can generate SAP ISU/CCS migration data into the import file using SAP IS-U/CCS transformation.

The following table shows the location at which you can save the import file for an SAP IS-U/CCS transformation in a session:

Property	Description
Import File Directory	Enter the directory name in this field. By default, the Integration Service writes import files in the service process variable directory, $PMTargetFileDir. You can also use the $OutputFileName session parameter to specify the file directory.
Import File Name	Enter the file name, or file name. By default, the Workflow Manager names the import file based on the target definition used in the mapping: target_name.out. Enter the file name, or file name and path. Optionally use the $OutputFileName session parameter for the file name.
Code Page	Optional. Name of the code page compatible with the SAP server.

Error Handling

In the Error Handling section of the Config Object tab, you must configure the SAP IS-U/CCS session to handle any error that occurs during data migration. The PowerCenter Integration Service increments the error count when it processes the following row types:

- Duplicate row. Any legacy key that is not unique results in a duplicate row. Duplicate row handling is applicable to the parent group or the control input group.
- Orphan row. Any value of the field data that does not match the value of the LegacyKey results in an orphan row. If the legacy key in the auto structure group is not present in the control input group, it results in an orphan row.

The following table describes the error handling properties that you can configure for an SAP IS-U/CCS transformation in a session:

Property	Description
Duplicate Parent Row Handling	Determines how the PowerCenter Integration Service handles duplicate parent rows. Choose one of the following values: - First Row. The PowerCenter Integration Service passes the first duplicate row to the target. The PowerCenter Integration Service rejects rows with the same primary key that it processes after this row. - Last Row. The PowerCenter Integration Service passes the last duplicate row to the target. - Error. The PowerCenter Integration Service passes the first row to the target. Rows that follow with duplicate primary keys increment the error count. The session fails when the error count exceeds the error threshold. Default is First Row.
Orphan Row Handling	Determines how the PowerCenter Integration Service handles orphan rows during a session. Choose one of the following values: - Ignore. The PowerCenter Integration Service ignores orphan rows. - Error. The session fails when the error count exceeds the error threshold. Default is Ignore.
Validate Field MetaData	Determines whether the metadata imported from the SAP System is current during run time. You have to provide the connection details for the specific SAP application server to be able to validate field metadata.

Part VI: Business Content Integration

This part contains the following chapter:

- Business Content Integration, 190

CHAPTER 18

Business Content Integration

This chapter includes the following topics:

- Business Content Integration Overview, 190
- Step 1. Prepare DataSources in SAP, 195
- Step 2. Import and Configure PowerCenter Objects, 196
- Step 3. Create a Table to Process Empty DataSources, 200
- Step 4. Configure and Start the Listener Workflow, 200
- Step 5. Create the Processing Mappings, 202
- Step 6. Deploy the Request Files, 211
- Step 7. Create the Send Request Workflows, 211
- Step 8. Create the Processing Workflows, 212
- Step 9. Schedule the Processing and Send Request Workflows, 213
- Troubleshooting Business Content Integration, 216

Business Content Integration Overview

PowerExchange for SAP NetWeaver integrates with SAP business content to provide an efficient, high-volume data warehousing solution. SAP business content is a collection of metadata objects that can be integrated with other applications and used for analysis and reporting. mySAP applications produce the business content data and the PowerCenter Integration Service consumes it. The PowerCenter Integration Service can consume all or changed business content data from mySAP applications and write it to a target data warehouse.

Informatica provides an XML file that you can use to import mappings and workflows that integrate with SAP business content.

Related Topics:

- "Step 2. Import and Configure PowerCenter Objects" on page 196

DataSources

The PowerCenter Integration Service consumes business content data from SAP DataSources. A DataSource is a customized business object used to retrieve data from internal SAP sources, including SAP function modules like SAP Financials. It contains the following components:

- An extraction structure that describes the fields containing extracted data
- An extraction type
- An extraction method that transfers data to an internal table of the same type as the extraction structure

SAP provides the following types of DataSources:

- Master data attributes
- Master data text
- Hierarchy
- Transaction

Use standard SAP DataSources or custom DataSources. SAP predefines all standard DataSources. You need to create custom DataSources.

Related Topics:

- "Hierarchy Definitions" on page 74

Logical System in SAP

To consume data from SAP business content, you define PowerCenter as a logical system in SAP. The logical system enables the PowerCenter Integration Service to initiate tasks within SAP. For example, the PowerCenter Integration Service connects to SAP as a logical system when you create a mapping to process a DataSource. It also connects to SAP as a logical system during a PowerCenter session to request data from the SAP system. Create a logical system in SAP for business content integration before you create a mapping to process a DataSource.

Mappings for Business Content Integration

You need to import several PowerCenter mappings from BCI_Mappings.xml. Use these mappings and the processing mappings that you create to integrate with SAP business content.

Use the following mappings to integrate with SAP business content:

1. **Listener mapping.** Receives DataSource data from SAP, loads it to staging targets, and requests that the PowerCenter Integration Service start the appropriate processing and send request sessions for the DataSource. You configure and use the imported listener mapping.
2. **Send request mapping.** Sends requests for DataSource data to SAP. The PowerCenter Integration Service uses request files generated when you create the processing mapping to request data from SAP. When SAP receives the request, it sends data to the listener mapping. You configure and use the imported send request mapping.
3. **Processing mapping.** Processes DataSource data staged by the listener mapping and loads it to the target data warehouse. When you create processing mappings, you can specify data selection parameters and determine whether to consume all the DataSource data or only the data that changed

since the last time it was processed. You use the Generate Mapping for BCI Wizard in the Mapping Designer to create processing mappings. You create one processing mapping for each non-hierarchy DataSource you want to process. You create one processing mapping for all hierarchy DataSources.

4. **Cleanup mapping.** Cleans up data in the staging targets. You configure and use the imported cleanup mapping.

Note: You cannot customize the listener mapping or the send request mapping that you import from BCI_Mappings.xml. You also cannot customize the processing mapping that you generate from PowerCenter.

The following table shows the relationship of the business content integration mappings:

Mapping	Source(s)	Target(s)	Notes
Listener	SAP/ALEIDoc source	- BCI_Scheduling_Target LMAPI target - RSINFOStaging relational target - Indicator relational target - Source_For_BCI relational target	BCI_Scheduling_Target LMAPI target determines which processing and send request mappings are run for each DataSource. Source_For_BCI relational target is the source of the processing mapping.
Send Request	Flat file source	SAP/ALEIDoc target	Source is the request file created when you create the processing mapping.
Processing	Source_For_BCI relational source	Relational target data warehouse	Source_For_BCI relational source is the target of the listener mapping.
Cleanup	- Source_For_BCI relational source - DocumentNumber relational source	Source_For_BCI relational target	Removes processed data from Source_For_BCI relational source.

Listener Mapping for Partitioning

In addition to the `BCI_Mappings.xml` listener mapping, Informatica ships the `BCI_Listener_Partition.xml` listener mapping that you can use to partition the data that the Source_For_BCI relational target receives from SAP. When you extract large volumes of data from SAP, you can use partitioning to improve performance.

The Source_For_BCI table in the `BCI_Listener_Partition.xml` listener mapping contains a column called DataSourceName. You can use this field to partition the data that the Source_For_BCI relational target receives from SAP. Define the partition type based on the database that you use.

To use partitioning, you must import the `BCI_Listener_Partition.xml` listener mapping. If you have made any customizations in your previous listener mapping, you must manually restore the customizations in the imported `BCI_Listener_Partition.xml` listener mapping.

The Source_For_BCI table is used as a target in the listener mapping and as a source in the processing mapping. Therefore, you must update the Source_For_BCI table definitions in the listener mapping and processing mapping to include the DataSourceName field.

After you import the `BCI_Listener_Partition.xml` listener mapping, generate and execute SQL to create the Source_For_BCI relational table in the database with the DataSourceName field or edit the table definition to include the DataSourceName field. Similarly, before you generate the processing mapping, you must either reimport the Source_For_BCI table from the `BCI_Listener_Partition.xml` listener mapping or edit the table definition manually to include the DataSourceName field.

Workflows for Business Content Integration

You need to import several PowerCenter workflows from BCI_Mappings.xml. You use the imported listener workflow and the send request and processing workflows that you create to integrate with SAP business content.

Use the following workflows to integrate with SAP business content:

1. **Listener workflow.** Receives DataSource data from SAP, stages it for the processing workflow, and requests that the PowerCenter Integration Service start the appropriate processing and send request workflows for the DataSource. You configure and use the imported listener workflow.
2. **Send request workflow.** Sends requests for DataSource data to SAP. You create one send request workflow for each DataSource.
3. **Processing workflow.** Processes DataSource data staged by the listener workflow, writes data to the target, and cleans up the data staged by the listener workflow. You create one processing workflow for each processing mapping.

Running the Listener Workflow

Run the listener workflow before you create processing mappings. The PowerCenter Integration Service connects to SAP as a logical system when you create processing mappings and run processing workflows to perform tasks within SAP. For example, when you create a processing mapping, activate an extract program for the DataSource within the SAP system.

Scheduling the Send Request and Processing Workflows

After you create processing and send request workflows, stop the listener workflow so that you can configure the BCI_Scheduling_Target LMAPI target in the listener mapping. Use BCI_Scheduling_Target to schedule the processing and send request workflows that run for each DataSource. You then restart the listener workflow.

Running the Send Request Workflows

The send request workflows request DataSource data from SAP. When SAP receives the request, it sends the data to the PowerCenter Integration Service. The PowerCenter Integration Service uses the listener workflow to consume the data and load it to a relational target. The relational target is a staging area for the data, which the processing mappings use as a source.

Manually run the send request workflow that requests the first DataSource data you want to process. When BCI_Scheduling_Target in the listener workflow receives the DataSource data for this request, it requests that the PowerCenter Integration Service start the remaining workflows in the order you configured.

Running the Processing Workflows

When the listener workflow completes writing DataSource data to the staging area, BCI_Scheduling_Target requests that the PowerCenter Integration Service start the appropriate processing workflow for the DataSource. Processing workflows contain a processing session and a cleanup session. The processing session reads the data from the staging area, processes it, and loads it to the target. After the processing session loads the data to the target, it initiates the cleanup session to remove processed data from the staging area.

When the processing workflow completes, BCI_Scheduling_Target in the listener workflow requests that the PowerCenter Integration Service start another send request workflow to request the next DataSource data.

You can run one processing workflow at a time for a DataSource. When there are multiple requests sent at the same time for the same DataSource, you can configure the PowerCenter Integration Service to wait until the previous processing workflow request is complete before it processes the next request.

PowerCenter Integration Service Processing

To process business content data, you start the listener workflow and the send request workflow for the first DataSource data you want to process.

When you start these workflows, the send request workflow, listener workflow, processing workflow, and SAP interact in the following order:

1. The send request workflow sends a request for DataSource data to SAP.
2. SAP sends the requested data to the listener workflow.
3. The listener workflow writes data to the staging area.

   Note: Configure a relational target connection to handle SAP DataSources with no data. For more information about configuring the relational target connection, see "Step 3. Create a Table to Process Empty DataSources" on page 200.
4. When the listener workflow receives the complete DataSource data, BCI_Scheduling_Target in the listener mapping requests that the PowerCenter Integration Service start the appropriate processing workflow for the next DataSource.
5. The processing session processes the DataSource data from the staging area.
6. When the processing session finishes processing data, it initiates the cleanup session.
7. The cleanup session removes the processed data from the staging area.
8. When the processing workflow finishes, BCI_Scheduling_Target in the listener mapping requests that the PowerCenter Integration Service start the next send request workflow.
9. Repeat steps 1 through 8 until the last DataSource is processed.

The following figure shows how the send request and processing workflows interact with the listener workflow and SAP:

Before You Begin

Complete the following tasks before you can configure business content integration:

- Verify that you have installed the latest business content integration transport on the SAP system.
- Verify that you have installed the SAP plug-in version 2003_1 or later.

Steps to Integrate with SAP Business Content

Complete the following steps to integrate with SAP business content:

1. **Prepare DataSources in SAP.** Activate and configure each DataSource in SAP before you create a processing mapping for the DataSource.
2. **Import and configure PowerCenter objects.** Import and configure mappings and workflows to integrate with business content. Unzip the Informatica installer `.zip` file, navigate to the following directory, and import the mappings and workflows:

   ```
   <Informatica Installer File>\sapsolutions\mySAP\bci
   ```

3. **Create a relational table.** Create a relational table with the default table name BCILookupTable to handle SAP DataSources with no data.
4. **Configure and start the listener workflow.** Configure and start the imported listener workflow.
5. **Create the processing mappings.** Create processing mappings to select SAP DataSources, specify data extraction parameters, and create request files.
6. **Deploy request files.** Deploy request files from the PowerCenter Client to the PowerCenter Integration Service so that you can use them when you configure sessions.
7. **Create the send request workflows.** Create the send request workflows to request data from SAP.
8. **Create the processing workflows.** Create the processing workflows to consume DataSource data from SAP.
9. **Schedule the processing and send request workflows.** Stop the listener workflow and configure BCI_Scheduling_Target in the listener mapping to start the appropriate processing and send request workflows for each DataSource.

Step 1. Prepare DataSources in SAP

Before you create a processing mapping, activate DataSources in SAP. You can also customize whether DataSource fields are visible. Use the following SAP transactions to activate DataSources and customize DataSource fields in SAP:

- **RSA5 Transfer DataSources function.** Changes the status of DataSources from Delivered to Active.
- **RSA6 Postprocess DataSources and Hierarchy.** Customizes the visibility of fields in a DataSource.

Activating a DataSource in SAP

Use transaction RSA5 in SAP to change the status of DataSources from Delivered to Active.

To activate a DataSource in SAP:

1. In SAP, open transaction RSA5 for the DataSource for which you want to create a processing mapping.

2. Execute the Transfer DataSources function.

   The DataSource is now active.

Customizing Fields in a DataSource

Use transaction RSA6 in SAP to customize fields in a DataSource.

To customize fields in a DataSource:

1. In SAP, open transaction RSA6 for the DataSource that you want to customize.
2. Select a DataSource and click DataSource > Display DataSource.
3. Select Hide Field for fields you want to hide.

   When you create a processing mapping, you cannot see hidden fields or consume data from them.
4. Clear Hide Field for fields you want to make visible. Then, click the Save button.

Step 2. Import and Configure PowerCenter Objects

To integrate with business content, import the following PowerCenter objects from BCI_Mappings.xml:

- Listener mapping and workflow
- Send request mapping
- Cleanup mapping
- Sample processing mapping and workflow

The sample processing mapping and workflow are for reference purposes only. Because the sample processing mapping and workflow are not based on an SAP DataSource from the SAP system, you cannot process any data using them.

Note: Because the imported workflow sessions do not have valid connection information for the environment, they are invalid. You configure valid connections for the sessions when you configure the listener, send request, and processing workflows.

Complete the following tasks after you import PowerCenter objects from BCI_Mappings.xml:

- Generate and execute SQL for the relational targets.
- Configure an LMAPITarget application connection.
- Verify the basic IDoc type in the Router transformation of the listener mapping.

Importing PowerCenter Objects from BCI_Mappings.xml

Use the Repository Manager to import objects from BCI_Mappings.xml.

Tip: Create a development folder and a production folder in the Repository Manager. Import objects into the development folder. Imported relational source and target definitions use a default database type that may not match the database type. If you use the Designer to change the database type once in the development folder, you can copy the mappings with relational source and target definitions to multiple production folders without having to change the database type again.

To import PowerCenter objects from BCI_Mappings.xml:

1. In the Repository Manager, connect to a repository.

2. Click Repository > Import Objects.
3. In the Import Wizard, select BCI_Mappings.xml from the Informatica installer file directory, and click Next.

   Note: To import the BCI_Mappings.xml, unzip the Informatica installer .zip file, navigate to the following directory, and select the BCI_Mappings.xml: `<Informatica Installer File>\sapsolutions\mySAP\BCI`
4. Select Add All, and click Next.
5. Select a folder, and click Next.

   For example, use a development folder.
6. Complete the import from BCI_Mappings.xml using the Import Wizard.

Creating Database Tables for PowerCenter Objects

Create database tables for the following relational targets:

- Source_For_BCI
- RSINFOStaging
- Indicator
- DocumentNumber

Creating a Database Table for the Source_For_BCI Relational Target

Create a database table for the Source_For_BCI target. The Source_For_BCI target is used as a target in the listener mapping, a source in the processing mapping, and both a source and a target in the cleanup mapping.

To create a database table for the Source_For_BCI relational target:

1. In the Target Designer, add the Source_For_BCI target definition to the workspace.
2. Edit the Source_For_BCI target definition.
3. On the Table tab, verify that the database type matches the relational database, and click OK.
4. Select the Source_For_BCI target definition, and click Targets > Generate/Execute SQL.
5. Click Connect.
6. Select the ODBC connection, enter the user name and password, and click Connect.
7. Select Create Table, and clear Primary Key and Foreign Key.

   Note: The IDocRecord column must be the primary key in the Designer but not in the Source_For_BCI relational table in the database. A primary or foreign key in the Source_For_BCI table in the database causes the cleanup session to fail.
8. Click Generate and execute SQL.

   The Designer creates the database table using the default table name Source_For_BCI.

Creating a Database Table for the RSINFOStaging, Indicator, and DocumentNumber Targets

Create a database table for the Source_For_BCI target. The RSINFOStaging target is used as a target in the listener mapping. The Indicator target is used as a target in the listener mapping. The DocumentNumber target is used as a target in the processing mapping and as a source in the cleanup mapping.

To create a database table for the RSINFOStaging, Indicator, and DocumentNumber targets:

1. In the Target Designer, add the RSINFOStaging, Indicator, and DocumentNumber target definitions to the workspace.
2. Edit each target definition.
3. On the Table tab, verify that the database type matches the relational database, and click OK.
4. Select the target definitions.
5. Click Targets > Generate/Execute SQL.
6. Click Connect.
7. Select the ODBC connection, enter the user name and password, and click Connect.
8. Select Create Table, Primary Key, and Foreign Key.
9. Click Generate and execute SQL.

   The Designer creates the database tables using the default table names RSINFOStaging, Indicator, and DocumentNumber.

Configuring an LMAPITarget Application Connection

Configure an LMAPITarget application connection before running the listener workflow.

To configure an LMAPITarget application connection:

1. In the Workflow Manager, connect to a repository.
2. Click Connections > Application.

   The Application Connections Browser appears.
3. Select LMAPITarget as the application connection type.
4. Click **New**.

   The Connection Object Definition dialog box appears.
5. Enter the connection options

   The following table describes the connection options:

Connection Option	Description
Name	Connection name used by the Workflow Manager.
Use Kerberos Authentication	Indicates that the LMAPITarget application runs on a network that uses Kerberos authentication. If this option is selected, you cannot set the user name and password in the connection object. The connection uses the credentials of the user account that runs the session that connects to the LMAPITarget application. The user account must have a user principal on the Kerberos network where the application runs.
User Name	Repository user name. Not available if the **Use Kerberos Authentication** option is selected.
Password	Password for the user name. Not available if the **Use Kerberos Authentication** option is selected.

Connection Option	Description
Use Parameter in Password	Indicates that the password for the repository user name is a session parameter, $Param*Name*. Define the password in the workflow or session parameter file, and encrypt it by using the *pmpasswd* CRYPT_DATA option. Default is disabled.
Code Page	Code page compatible with the SAP server. Must also correspond to the Language Code.
Domain Name	Name of the domain for the associated PowerCenter Integration Service.
PowerCenter Integration Service Name	Name of the associated PowerCenter Integration Service.
User NameSpace	Security domain for the user account.

6. Click OK.

Identifying and Verifying the Basic IDoc Type in the Listener Mapping

The listener mapping contains a Router transformation that tests the basic IDoc type. The basic IDoc type the PowerCenter Integration Service passes through the Router transformation must match the basic IDoc type in the SAP system. If the basic IDoc types do not match, the PowerCenter Integration Service writes the data to the RSINFOstaging target. However, it does not write data to the SOURCE_FOR_BCI target.

You can identify the basic IDoc type in the SAP system. You can also verify the basic IDoc type in the Router transformation to make sure it matches the basic IDoc type in the SAP system.

Identifying the Basic IDoc Type in SAP

When you configure PowerExchange for SAP NetWeaver for business content integration, you run the ZINFABCI program to create a logical system in SAP.

When you run ZINFABCI, the program creates a row in the RSBASIDOC table with the RLOGSYS field. RLOGSYS has the same value that you specified for the logical system in the ZINFABCI program. You need to use this name when you identify the basic IDoc type of the SAP system.

To identify the basic IDoc type in SAP:

1. Log on to the SAP system.
2. Go to transaction SE11 and view the contents of the RSBASIDOC table.
3. Query the RSBASIDOC table using the logical system name you provided when you ran the ZINFABCI transaction. For example, INFACONTNT.

   The row contains a field called BIDOCTYP. The value of this field is the basic IDoc type.

   Note: The basic IDoc type to verify it against the basic IDoc type in the Router transformation.

Verifying the Basic IDoc Type in the Router Transformation

After you identify the basic IDoc type, edit the Router transformation in the Listener mapping to verify that it matches the basic IDoc type in the SAP system.

To verify the basic IDoc type in the Router transformation of the listener mapping:

1. In the Mapping Designer, open the listener mapping.
2. Edit the Router transformation.
3. Click the Groups tab.

   The default group filter condition for the Source_For_BCI group includes a basic IDoc type. By default, the basic IDoc type is ZSIN1000. For example:

   ```
   BasicIDocType='ZSIN1000'
   ```
4. If the basic IDoc type in the SAP system is not ZSIN1000, modify the group filter condition to match the basic IDoc type of the SAP system.

Step 3. Create a Table to Process Empty DataSources

Create a table in any relational database to handle SAP DataSources with no data. Configure the listener workflow and the send request workflow sessions to write and read data from the relational table.

The default table name is BCILookupTable. Use the example syntax to create the table in the database:

```
CREATE TABLE BCILookupTable (Request_Id varchar(30), DataSourceName varchar(30))
```

If you create a table name with another name, you must use same table name that you created in the send request and the listener workflow sessions.

To change the table name in the send request source definition, open the Source Analyzer and click Sources:

1. Select the **SAPALEIDoc** and click **BCI_Listner**.
2. In the **Edit Tables** dialog box, click the **Metadata Extensions** tab.
3. In the **Edit Metadata Extension Value** dialog box, edit the table name.

To change the table name in the listener workflow, open the Target Designer and select Targets:

1. Select the target definition for the listener workflow.
2. In the **Edit Tables** dialog box, click the **Metadata Extensions** tab.
3. In the **Edit Metadata Extension Value** dialog box, edit the table name.

Step 4. Configure and Start the Listener Workflow

Before you create processing mappings, configure and start the listener workflow.

To configure the listener workflow, create a session from the listener mapping. The listener mapping includes the following target definitions:

BCI_Scheduling_Target

Determines which processing and send request workflows are run for each DataSource.

RSINFOStaging

Contains RSINFO IDoc messages sent by SAP. These messages include the status of the DataSource extraction from SAP. When the status is complete, BCI_Scheduling_Target requests that the PowerCenter Integration Service start the appropriate processing workflow.

Indicator

Contains the status of the processing and send request workflows scheduled for the DataSource. When the status of a processing workflow is complete, BCI_Scheduling_Target requests that the PowerCenter Integration Service start the next send request workflow.

Source_For_BCI

Contains the DataSource data received from SAP in IDoc message format. The processing mapping uses this as a source definition. This staging table must have sufficient space to store the data. When the processing mapping is complete, the cleanup mapping also uses this as a source definition. The cleanup mapping removes the processed data.

1. In the Workflow Designer, drag the listener workflow into the workspace.
2. Open the session properties for s_BCI_listener.
3. From the connections settings on the **Mapping** tab (Sources node), select the SAP_ALE_IDoc_Reader application connection that you configured for business content integration.
4. From the connections settings on the **Mapping** tab (Sources node), select the BCI Metadata Connection type and from the drop-down assign the SAP_ALE_IDoc_Writer application connection to access segments in SAP longer than 1000 characters.
5. Optionally, from the connections settings on the **Mapping** tab (Sources node), select the relational connection that you created.
6. Set the real-time flush latency attribute to 10.

   The real-time flush latency must be greater than 0 (zero). Otherwise, the session fails.
7. Set the delay BCI processing workflow trigger attribute to define the time in seconds by which you want to delay the trigger of the BCI processing workflow.

   Set this attribute to flush all the data to the Source_For_BCI staging target before the processing workflow starts.
8. Click the BCI_Scheduling_Target.
9. From the connections settings, select the LMAPITarget application connection that you configured in "Configuring an LMAPITarget Application Connection" on page 198.
10. Set the wait for commit attribute to configure how the PowerCenter Integration Service must treat multiple processing workflow requests sent at the same time for the same DataSource.

    The following table describes the options:

Option	Description
Yes	When a processing workflow is running and the PowerCenter Integration Service receives a request from the listener workflow for the same DataSource, the PowerCenter Integration Service saves the request in a queue. The PowerCenter Integration Service writes a message in the listener log stating that the request will be processed after the current running processing workflow completes. After the current processing workflow completes, the PowerCenter Integration Service processes the request.

Option	Description
No	When a processing workflow is running and the PowerCenter Integration Service receives a request from the listener workflow for the same DataSource, the PowerCenter Integration Service does not process the request. The PowerCenter Integration Service writes a message in the listener log stating that the request was not processed because another processing workflow for the same DataSource is running.

11. Click each relational target (RSINFOStaging, Indicator, and Source_For_BCI) and select the same relational connection for each target.

    If the targets use different relational connections, the session fails.
12. Click the Source_For_BCI target.
13. Set the target load type to normal.
14. Click **OK** to close the session properties.
15. Open the listener workflow.
16. On the **Scheduler** tab, click the **Edit Scheduler** button.

    The **Edit Scheduler** dialog box appears.
17. On the **Schedule** tab, select **Run Continuously** and click **OK**.

    A continuous workflow starts as soon as the PowerCenter Integration Service initializes. When the workflow stops, it restarts immediately.
18. Click **OK** to close the **Edit Workflow** dialog box.
19. Save the workflow.
20. Start the listener workflow.

Step 5. Create the Processing Mappings

Use the Mapping Designer to create a processing mapping. When you create the mapping, use the wizard to enter the connection information for the SAP system, select the DataSource, select a transport mode, activate the extraction program in SAP for the DataSource, select data extraction parameters, and select an update mode.

Create one processing mapping for each non-hierarchy DataSource. You create one processing mapping for all hierarchy DataSources.

Note: The PowerCenter objects you imported in "Step 2. Import and Configure PowerCenter Objects" on page 196 included a sample processing mapping for a non-hierarchy DataSource. This mapping is for reference purposes only. Because the sample processing mapping is not based on an SAP DataSource from the SAP system, you cannot process any data with it.

Update Modes

When you create a mapping, specify one of the following update modes:

- **Full.** Extracts all the data that meets the selection parameters you specify.
- **Delta.** Extracts only the data that has changed since the last data extraction.

A series of delta updates is called a delta queue. When you create a processing mapping and select delta update instead of full update, select one of the following delta update options:

- **Delta initialization with transfer.** Extracts all the data from a DataSource.
- **Delta update.** Extracts the changed data since the last data extraction.
- **Delta repeat.** Repeats a previous delta update if there were errors.

If you do not want to do an initial full transfer of data, but want to extract only changed data, you can configure for delta initialization without transfer. For more information about configuring delta initialization without transfer, see the Informatica How-To Library article "How to Configure SAP BCI Delta Initialization Without Transfer":
https://kb.informatica.com/h2l/HowTo%20Library/1/0647-Configuring-SAP-BCI-Delta-Initialization-Without-Transfer-H2L.pdf

When you create the processing mapping, based on your business needs, you can initialize more than one delta queue for each DataSource. Use multiple delta queues when you want to select data from a non-continuous range that cannot be defined in a single delta queue.

For example, to compare data from item numbers 1 through 4 to item numbers 11 through 14, you need to use two delta queues because the item numbers are not in one continuous range.

The following table shows two delta queues for two item ranges updated at the end of each quarter:

Delta Queue	Quarter 1	Quarter 2	Quarter 3	Quarter 4
Delta Queue 1 Items 1 through 4	Delta Initialization Full Update	Delta Update Changed Data	Delta Update Changed Data	Delta Update Changed Data
Delta Queue 2 Items 11 through 14	Delta Initialization Full Update	Delta Update Changed Data	Delta Update Changed Data	Delta Update Changed Data

At the end of the first quarter, you initialize both delta queues and perform full updates, because there are no previous delta updates. At the end of the second, third, and fourth quarters, both delta queues extract only the data that has changed since the previous update.

When you create a processing mapping, you can create a new delta queue or edit an existing delta queue. For each new delta queue, you need to specify selection criteria.

Request Files

A request file requests data from an SAP DataSource. Each DataSource has one request file. When you create a processing mapping, you select a local directory where you want to store the request file. When you save a request file, the processing mapping wizard saves it using the following syntax:

```
<DataSource_name>_<update_mode>
```

For example, if you save a request file for the 0ACCOUNT_ATTR DataSource in full update mode, the request file is named as follows:

```
0ACCOUNT_ATTR_Full_Update
```

If you have an existing request file for the DataSource and update mode, you can revert to the contents of the saved request file to overwrite the current settings. When you save the current settings, and a request file for the DataSource and update mode exists, the current settings overwrite the contents of the request file.

When you use hierarchy DataSources, the send request file uses the language that you specify in the **Language** field in step 1 of the **Generate Mapping for BCI** wizard.

After you create a processing mapping, deploy the request file so that you can use it when you configure a session to send requests to SAP.

Non-Hierarchy and Hierarchy DataSource Processing Mappings

When you create a processing mapping, the Designer creates a different mapping for non-hierarchy DataSources than for hierarchy DataSources. Hierarchy DataSources have parent-child segments. To maintain the hierarchical relationship in the target, the processing mapping for hierarchies contains additional target definitions that share a primary key.

Note: There can only be one processing mapping for all hierarchy DataSources. This mapping is named Mapping_For_Hierarchy and cannot be changed. If you create an additional processing mapping for a hierarchy, you overwrite the existing Mapping_For_Hierarchy mapping.

Processing mappings include the following target definitions:

- **Control record.** Contains the control record data for the DataSource document, including status and type.
- **Document number.** Contains the DataSource document number as a unique key. The cleanup mapping uses this as a source definition.
- **Parameter file.** Contains the folder name and the largest document number processed. The Source Qualifier transformation uses the largest document number processed to ensure that only higher document numbers are extracted from the Source_For_BCI source.
- **DataSource data targets.** Non-hierarchy processing mappings contain one target definition for all of the business content data extracted from the DataSource. The hierarchy processing mapping contains one target definition for the hierarchy parent IDoc segment and four target definitions for the hierarchy child IDoc segments.

The following figure shows a processing mapping for a non-hierarchy DataSource:

The following figure shows a processing mapping for a hierarchy DataSource:

Note: The name of the source definition in processing mappings is based on the name of the DataSource. However, the source for all processing mappings is the Source_For_BCI relational table.

Steps to Create a Processing Mapping

Before you create a processing mapping, start the listener workflow. When you create the processing mapping, you can send a test request file to SAP. For SAP to return data to the PowerCenter Integration Service, the listener workflow must be running.

Tip: Create processing mappings in the same development folder that you created in "Step 2. Import and Configure PowerCenter Objects" on page 196. The relational target definitions created when you create a processing mapping use a default database type that may not match the database type. If you use the Designer to change the database type once in the development folder, you can copy processing mappings with relational target definitions to multiple production folders without having to change the database type again.

Complete the following steps to create a processing mapping:

1. **Connect to SAP and select a DataSource.** Use Step 1 of the Generate Mapping for BCI Wizard to connect to SAP and select a DataSource.
2. **Select a transfer mode and activate the ABAP program in SAP.** Use Step 2 of the wizard to select a transfer mode and activate the ABAP extraction program in SAP.
3. **Configure the request file and data extraction parameters.** Use Step 3 of the wizard to create, revert, or test the request file and select data extraction parameters.

4. **Name and generate the processing mapping.** Use Step 4 of the wizard to name, describe, and generate the processing mapping.
5. **Override the SQL query in the Source Qualifier**. If you use Oracle or IBM DB2 and the DataSource is not a hierarchy, update the SQL query in the Source Qualifier.

After you create the processing mapping, create the relational tables for the relational target definitions.

Connecting to SAP and Selecting a DataSource

Use step 1 of the wizard to connect to SAP and select a DataSource.

To connect to SAP and select a DataSource:

1. In the Mapping Designer, click Mappings > Generate BCI Mapping.

   The Generate Mapping for BCI Wizard appears.
2. To connect to the SAP system, enter the login information.

   The following table describes the login information:

Field	Description
Connect String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name. Must be a user for which you have created a source system connection.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

3. Click Connect.
4. Optionally, enter the name or a partial name of an Application Component or a DataSource to search for and click the Find button. Or, go to step 7.

   An application component is an expandable container for DataSources.

   You can include an asterisk (*) or percent sign (%) in a name to use as a wildcard in the search.

   The Search Results dialog box appears.
5. Select a DataSource or application component.

   The DataSources listed are those you activated in "Step 1. Prepare DataSources in SAP" on page 195.
6. Click Select to close the Search Results dialog box, and click Next.
7. Expand the application component that contains the DataSource you want to use.

   The Designer displays a list of DataSources.
8. Select the DataSource from which you want to create the mapping.

   If you select a hierarchy DataSource, go to "Configuring the Request File and Data Extraction Parameters" on page 209.

Note: If you select a hierarchy DataSource and have already created a processing mapping for a hierarchy DataSource, you receive a warning. You can only have one processing mapping for all hierarchy DataSources in a folder. If you create another processing mapping for a hierarchy DataSource, you overwrite the existing mapping for hierarchy DataSources.

9. Click Next.

   Step 2 of the wizard appears.

Selecting the Transfer Mode and Activating the ABAP Extraction Program

Use step 2 of the Generate Mapping for BCI Wizard to select the transfer mode and activate the ABAP extraction program.

The following table describes the DataSource fields:

Field	Description
DataSource Name	SAP name of the DataSource used for the mapping.
Type	DataSource type.
Description	DataSource description.
Extractor	Extractor program name.
Extract Structure	Extract structure name.
Delta	Delta designation.

To select the transfer mode and activate the ABAP extraction program:

1. In step 2 of the Generate Mapping for BCI Wizard, select a transfer mode:
   - **tRFC**. Sends data faster and requires fewer resources than IDoc, but cannot be used for hierarchy DataSources.
   - **IDoc**. Stages all data to IDocs before sending them to the PowerCenter Integration Service. IDoc is the default for hierarchy DataSources.
2. Click Activate DataSource to activate the ABAP extraction program in SAP for the DataSource.

   If you activated this DataSource in another processing mapping, then you do not need to activate the DataSource again. However, you need to reactivate the DataSource when one of the following conditions is true:
   - This mapping uses a different transfer mode.
   - The DataSource metadata has changed.

   Otherwise, the session you create for this mapping fails.
3. Click Next.

   Step 3 of the wizard appears.

Activating the ABAP Extraction Program in SAP for Multiple DataSources

Informatica ships a `DataSourceMassActivation.xml` mapping that you can use to simultaneously activate the ABAP extraction program in SAP for multiple DataSources.

Before you use the DataSource mass activation mapping, create an SAP ECC connection to connect to the SAP source from which the listener session extracts data. You must also create an SAP RFC/BAPI connection to make BAPI/RFC function calls and activate the ABAP extraction program in SAP for multiple DataSources.

The DataSource mass activation mapping contains the following components:

Sources

The mapping contains a flat file and the SAP ROOSOURCE table as sources. You must create a flat file source in the PowerCenter source directory and list the names of the DataSources for which you want to activate the ABAP extraction program in SAP.

Transformations

The mapping contains a Joiner transformation, an Expression transformation, and a BAPI/RFC transformation.

The Joiner transformation joins the flat file source and the SAP ROOSOURCE table to fetch the DataSource details. It matches the DataSource name in the flat file with the SAP ROOSOURCE table to fetch DataSource details such as the DataSource type.

The Expression transformation contains details such as the language, transfer mode, and logical system that you use to activate the ABAP extraction program in SAP. It contains the following ports that you can edit:

SCALAR_INPUT_I_TFMETHODE

Defines the transfer mode. Default is T. To use the IDoc transfer mode, enter the value as I.

SCALAR_INPUT_I_LANGU

Defines the language. Default is E, which stands for English. To use a different language, enter the relevant language code.

SCALAR_INPUT_I_RLOGSYS

Holds the $$RLOGSYS mapping variable. You must create a parameter file and define the $ $RLOGSYS mapping variable in the file. Set the variable value to the name of the SAP source from which the listener session extracts data. To change the logical system value, update the value of the $$RLOGSYS mapping variable in the parameter file.

The BAPI/RFC transformation makes the BAPI/RFC function calls to activate the ABAP extraction program in SAP.

Target

The mapping contains a flat file target to record the errors that are encountered in the DataSource mass activation process.

When you import the DataSource mass activation mapping, an associated session also gets imported. After you import the mapping, you must generate and install the R/3 code in SAP. Then, edit the session to include the SAP ECC connection and RFC/BAPI connection. Run the session to simultaneously activate the ABAP extraction program in SAP for multiple DataSources.

1. Import the `DataSourceMassActivation.xml` mapping from the Repository Manager.
2. Navigate to the PowerCenter source directory and create a flat file source.

3. In the flat file source, enter the names of the DataSources for which you want to activate the ABAP extraction program in SAP.
4. Open the session that is associated with the DataSource mass activation mapping.
5. In the session properties, set the source filename property to the name of the flat file source.
6. Create a parameter file and define the $$RLOGSYS mapping variable in the file. Set the variable value to the name of the SAP source from which the listener session extracts data.
7. Click **Mappings** > **Generate and Install R/3 Code**.

   The **Generate and Install** dialog box appears.
8. Enter the connect string, user name, password, client, and language to connect to the SAP source system.
9. Click **Connect**.

   The DataSource mass activation mapping appears.
10. Select the DataSource mass activation mapping and click **Direct Installation**.
11. In the session properties, select the SAP ECC and RFC/BAPI connections.
12. Start the listener workflow.
13. Run the listener session.

    The ABAP extraction program is activated in SAP for the DataSources that you specified in the flat file source.

Configuring the Request File and Data Extraction Parameters

Use step 3 of the Generate Mapping for BCI Wizard to configure the request file and data extraction parameters. The procedure varies depending on the update mode and the DataSource type that you select.

When you perform a delta update, select a delta update mode and optionally enter a selection value range in the Initialize Delta Queue area. Enter a selection value range if you want to filter the data.

When you perform a full update, optionally enter a selection value range in the Selection Criteria area. Enter a selection value range if you want to filter the data. If you are creating a processing mapping for a hierarchy DataSource, perform the same steps to configure the request file and data extraction parameters. However, before you send the request file, you must select a particular hierarchy within the hierarchy DataSource.

To configure the request file and data extraction parameters:

1. In step 3 of the Generate Mapping for BCI Wizard, optionally click Revert to populate all settings from a previously saved request file for the DataSource.

   If you click Revert, go to "Naming and Generating the Processing Mapping" on page 210.
2. Enter a directory for the request file or click the Browse button and select a directory.
3. Select an existing Delta Initialization Request.

   - or -

   Select Create New.
4. If you are creating a new delta initialization request, enter a from and to value range.
5. If you are creating a processing mapping for a hierarchy DataSource, select the hierarchy for which you want to send the request.
6. Select an update mode:
   - **Full**. Extracts all the data. If you select Full, enter a from and to value range for the selection criteria.

- **Delta**. Extracts the data that has changed since the last data extraction.

7. Click the Initialize button.

   The delta queue is initialized.

8. Select a delta update mode:
   - **Delta Init with Transfer.** Extracts all the data from the DataSource.
   - **Delta.** Extracts the changed data since the last data extraction.
   - **Delta Repeat.** Repeats the previous delta update in case of errors.

9. Click Save to save all the settings to the request file.

   The PowerCenter Client saves the request file to the specified directory.

10. Click Send Request to send the request to SAP as a test.
11. Click Next.

    Step 4 of the wizard appears.

Naming and Generating the Processing Mapping

Use step 4 of the Generate Mapping for BCI Wizard to name and generate the processing mapping.

To name and generate the processing mapping:

1. In step 4 of the Generate Mapping for BCI Wizard, optionally modify the default name for the mapping if you selected a non-hierarchy DataSource.

   If you selected a hierarchy DataSource, you cannot modify the default name.

2. Optionally, enter a description for the mapping.
3. Click Generate Mapping.
4. Click Yes to close the wizard or No if you want to leave the wizard open to create more processing mappings.
5. Click Finish.

Overriding the SQL Query for Non-Hierarchy DataSources

If you use Oracle or IBM DB2 and the DataSource is not a hierarchy, update the SQL query in the Source Qualifier transformation.

To override the default query for a non-hierarchy DataSource on Oracle or IBM DB2:

1. In the Mapping Designer, edit the Source Qualifier transformation.
2. Click the Properties tab.
3. Open the SQL Editor for the SQL Query field.
4. Change the SUBSTRING function to SUBSTR.

Generating and Executing SQL for the Relational Targets

Generate and execute SQL to create relational tables in the database for the relational targets in the processing mapping.

To generate and execute SQL for the relational targets:

1. In the Target Designer, add the relational target definitions for the processing mapping to the workspace.

For non-hierarchy processing mappings, add the following relational target definitions to the workspace:

- ControlRecord
- BIC_CII<DataSource_name>
- DocumentNumber

For the hierarchy processing mapping, add the following relational target definitions to the workspace:

- ControlRecord
- E2RSHIE000
- E2RSHTX000
- E2RSHND000
- E2RSHIV000
- E2RSHFT000
- DocumentNumber

2. Edit each relational target definition and verify that the database type matches the database type.
3. Select all the relational target definitions.
4. Click Targets > Generate/Execute SQL.
5. Click Connect.
6. Select the ODBC connection, enter the user name and password, and click Connect.
7. Select Create Table, Primary Key, and Foreign Key.
8. Click Generate and Execute SQL.

Step 6. Deploy the Request Files

To use processing mappings in PowerCenter workflows, deploy all request files created with the processing mapping wizard. To deploy request files, copy or move them to the PowerCenter Integration Service source file directory. The $PMSourceFileDir service variable configured for the PowerCenter Integration Service process specifies the source file directory.

You access the request files when you create the send request workflows. When the send request workflows run, the PowerCenter Integration Service reads the appropriate request file from the PowerCenter Integration Service source directory.

Step 7. Create the Send Request Workflows

Create one send request workflow for each processing workflow. The send request workflow includes a send request session which sends requests for data to SAP. You can use the send request workflows to access data from segments in SAP.

Note: The processing and send request workflows must have unique names within the repository. If a workflow has the same name as another workflow in the same repository, then the PowerCenter Integration Service fails to start the workflow.

To create a send request workflow:

1. In the Workflow Manager, click Connections > Application.
2. Verify that the Destination Entry for the SAP_ALE_IDoc_Writer application connection is the same as the DEST parameter you configured for the logical system in the SAP system.
3. Create a session for the send request mapping.
4. Open the session properties.
5. On the Mapping tab, click the Targets node.
6. Optionally, from the Connections settings on the Mapping tab (Targets node), select the relational connection that you created.
7. Select the Generate Request ID attribute.
8. Click the Files and Directories node.
9. For the Send_request attribute, enter the source file directory where you deployed the request file.
10. Enter the source file name for the request file.
11. Click OK to close the session properties.
12. Link the start task to the send request session.

Step 8. Create the Processing Workflows

Create one workflow for each processing mapping. Create a processing workflow and include the following components in it:

- **Processing session.** Processes data from the listener staging area and loads it into the target.
- **Cleanup session.** Cleans up the staging area.

Note: The processing and send request workflows must have unique names within the repository. If a workflow has the same name as another workflow in the same repository, the PowerCenter Integration Service cannot start the workflow.

After you add each task to the workflow, link them in the correct order.

Creating a Processing Session

Create a processing session to process data staged by the listener mapping and load it into the target data warehouse. Configure the processing session like an outbound IDoc session.

To create the processing session:

1. In the Workflow Designer, create a session for the processing mapping.
2. From the Connections settings on the Mapping tab (Sources node), select the relational connection value for the source.
3. Click the Targets node.
4. Select the relational connection values for the targets.

Creating a Cleanup Session

Create a cleanup session to remove processed data from the staging area.

To create the cleanup session:

1. In the Workflow Designer, create a session for the cleanup mapping.
2. Click the Properties tab.
3. Choose to treat source rows as delete.
4. From the Connections settings on the Mapping tab (Sources node), select the relational connection values for the sources.

   The connection value must be the same value as the Source_For_BCI target definition of the listener mapping.
5. Click the Targets node.
6. Select the connection values for the targets.

   The connection value must be the same value as the Source_For_BCI target definition of the listener mapping.
7. From the Properties settings, verify that the Delete attribute is enabled.

Configuring the Processing Workflow

After you create the tasks in the workflow, link the tasks in the following order:

1. Start task
2. Processing session
3. Cleanup session

Configure each session or task to run when the previous session or task completes.

Step 9. Schedule the Processing and Send Request Workflows

You can determine the order in which you want to receive and process DataSource data. To do this, you edit the BCI_Scheduling_Target in the listener mapping. In the target, enter which processing and send request workflows run for each DataSource. You enter the following information:

- The name of each DataSource that you want to process
- The name of the processing workflow that processes the DataSource data
- The name of the send request workflow that requests data from SAP for the next DataSource

When you schedule the workflows, you enter the names of each DataSource and each processing workflow. You also enter the names of all but one of the send request workflows. You do not enter the name of the send request workflow that requests the first DataSource you want to process.

The following table shows how to schedule the workflows for each DataSource:

DataSource	Processing Workflow	Send Request Workflow
DS1	pr_DS1	sr_DS2
DS2	pr_DS2	sr_DS3
DS3	pr_DS3	

The PowerCenter Integration Service requests the DS1 DataSource first. It uses the sr_DS1 send request workflow. BCI_Scheduling_Target does not schedule the sr_DS1 send request workflow. Instead, start the sr_DS1 workflow. BCI_Scheduling_Target requests that the PowerCenter Integration Service start the remaining workflows in the following order:

1. When BCI_Scheduling_Target receives the complete data for DS1, it requests that the PowerCenter Integration Service start the pr_DS1 processing workflow.
2. When pr_DS1 completes, BCI_Scheduling_Target requests that the PowerCenter Integration Service start the sr_DS2 send request workflow. This workflow requests data from SAP for the DS2 DataSource.
3. When BCI_Scheduling_Target receives the complete data for the DS2 DataSource, it requests that the PowerCenter Integration Service start the pr_DS2 processing workflow.
4. When pr_DS2 completes, BCI_Scheduling_Target requests that the PowerCenter Integration Service start the sr_DS3 send request workflow. This workflow requests data from SAP for the DS3 DataSource.
5. When BCI_Scheduling_Target receives the complete data for the DS3 DataSource, it requests that the PowerCenter Integration Service start the pr_DS3 processing workflow.

Example

The following table lists the sample processing and send request workflows that you imported from BCI_Mappings.xml:

Workflow Name	Processing/ Send Request	Action
wf_send_request_article_attr	Send Request	Sends a request to SAP for the 0ARTICLE_ATTR DataSource.
wf_article_attr	Processing	Processes the 0ARTICLE_ATTR DataSource.
wf_send_request_article_text	Send Request	Sends a request to SAP for the 0ARTICLE_TEXT DataSource.
wf_article_text	Processing	Processes the 0ARTICLE_TEXT DataSource.

The following figure shows how BCI_Scheduling_Target schedules these sample processing and send request workflows:

The wf_send_request_article_attr workflow sends the request for the 0ARTICLE_ATTR DataSource which is the first DataSource to be processed. BCI_Scheduling_Target does not include this workflow. Instead, you manually start this first send request workflow. BCI_Scheduling_Target requests that the PowerCenter Integration Service start the remaining workflows in the following order:

1. wf_article_attr processing workflow
2. wf_send_request_article_text send request workflow
3. wf_article_text processing workflow

Steps to Schedule the Processing and Send Request Workflows

When you schedule the processing and send request workflows, you also enter the names of each session in the workflows. Before you start this procedure, you need to know the name of each workflow, the name of each session within the workflow, and the order that the sessions are scheduled to run.

To schedule the processing and send request workflows:

1. In the Workflow Manager, stop the listener workflow.

   If you are editing workflows you have already scheduled, make sure that no send request or processing workflow is running before you stop the listener workflow.
2. In the Designer, drag the BCI_Scheduling_Target LMAPI target into the Target Designer workspace.
3. Open BCI_Scheduling_Target.
4. Click the Scheduling Information tab.
5. In the Indicator Table field, enter the name of the database table you created for the Indicator relational target, as explained in "Creating Database Tables for PowerCenter Objects" on page 197.

   By default, the name is Indicator.

6. Delete the sample keys.
7. Add a new key and enter the key information.

   The following table describes the key information:

Property	Description
Key	Name of the DataSource that you want to process. You can enter each DataSource only once. **Note:** If you use a hierarchy DataSource, enter the key value as HIERARCHY. If you enter the key value as the DataSource name, the processing workflow for the hierarchy DataSource does not start.
Primary Workflow Name	Name of the processing workflow that processes this DataSource.
Secondary Workflow Name	Name of the send request workflow that sends a request to SAP for the next DataSource. Do not enter a send request workflow for the last DataSource that you process.

8. Select the primary workflow name and in Session Task Information, add a new task for each session in the workflow.
9. Enter the name of each session in the order the tasks are scheduled to run.
10. Select the secondary workflow name and add the name of each session.
11. Complete steps 7 to 10 to create a key for each DataSource that you want to process.

    You do not need to enter the keys in any particular order.
12. Click OK.
13. Click Repository > Save.
14. In the Workflow Manager, start the listener workflow.
15. Start the send request workflow that requests the first DataSource data you want to process.

Troubleshooting Business Content Integration

I cannot connect to SAP to create a processing mapping with my user name.

You may not have configured your user name as a logical system user name in SAP.

I cannot see the DataSource for which I want to create a processing mapping.

You may not have activated the DataSource within SAP. Activate the DataSource within SAP.

Not all the fields in the DataSource are visible when I create a processing mapping.

Some fields in the DataSource may be hidden. Use transaction RSA6 in SAP to clear the hidden fields.

How can I verify if the listener workflow received all the data that was sent by SAP?

The PowerCenter Integration Service prints a message in the session log that specifies the number of records that was sent by SAP and the number of records that was received by the listener workflow. Review the session log to see if the number of records match.

The update did not return the data I expected. How can I make sure everything is working properly?

There may be problems within SAP or the DataSource may have hidden fields. Use transaction RSA3 in SAP to test the extraction within SAP using the same data selection criteria as the delta update. You can then compare the results with the results from the delta update.

The BCI listener workflow fails when I configure the BCI listener mapping for delta extraction and the delta extraction does not have any row.

The connection parameters or user names that you defined in the relational connections for the send request workflow and the listener workflow may be different. Ensure that the connection parameters and user names in all the relational connections match.

The session failed with a timestamp error.

The request file is older than the latest processing mapping for that DataSource. Deploy the request file for the latest processing mapping created for the DataSource.

Part VII: SAP BW Data Extraction

This part contains the following chapter:

- SAP BW Extraction Through Open Hub Destination, 219

CHAPTER 19

SAP BW Extraction Through Open Hub Destination

This chapter includes the following topics:

- SAP BW Extraction Through Open Hub Destination Overview, 219
- Step 1. Create an Open Hub Destination, 221
- Step 2. Create an SAP BW OHS Source Definition, 221
- Step 3. Start the SAP BW Service, 224
- Step 4. Configure and Start a PowerCenter Workflow, 224
- Step 5. Configure and Start a Process Chain, 225
- Viewing Data Extraction Status, 230
- Troubleshooting Data Extraction from SAP BW through OHD, 231

SAP BW Extraction Through Open Hub Destination Overview

You can use the Open Hub Destination (OHD) to extract data from SAP BW. If you extract data through SAP BI version 7.0, you can extract through the open hub destination. The open hub destination is an independent object that you can integrate into the data flow. You can configure InfoCubes, DataStore Objects, InfoObjects, and InfoSets as open hub data sources.

The following figure shows how PowerCenter extracts data:

PowerCenter data extraction from SAP BW occurs in the following sequence:

1. A process chain directs an open hub destination to extract data. The process chain can also help you identify the point of failure in case of a system failure.
2. Use the ZPMSENDSTATUS ABAP program or pmsaputil to configure the third-party parameters. Use pmsaputil to start the process chain.
3. The open hub destination extracts data and loads it to a into a database table in the BW system.
4. The open hub destination notifies the RFC destination that is specified in the open hub destination that data extraction is complete.
5. The SAP BW Service initiates the PowerCenter workflow that the Integration Service uses to extract data from SAP BW.
6. The Integration Service extracts data from the SAP BW

When the data extraction begins, you can view log events to track the interactions between PowerCenter and SAP BW.

Extracting Data from SAP BW

Complete the following steps to configure the SAP BW system and PowerExchange for SAP NetWeaver BW to extract data from SAP BW:

1. Create an open hub destination and a data transfer process in the SAP BW system to extract the data from SAP BW and write it to an SAP transparent table.
2. Create a mapping with an SAP BW OHS source definition that you import from an open hub destination.
3. Start the SAP BW Service in PowerCenter.
4. Configure a PowerCenter workflow to extract data from SAP BW.
5. Create and start a process chain to extract data.

Step 1. Create an Open Hub Destination

Create an open hub destination to extract data from SAP BW. You can configure SAP BW objects such as InfoCubes, DataStore Objects, InfoObjects, and InfoSets as open hub destination data sources.

1. In the SAP Menu screen, enter RSA1 in the Transaction field.

   The **Administrator Workbench** opens.
2. Select **Open Hub Destination** in the left pane.
3. Right-click an **InfoArea** in the right pane and then select **Create Open Hub Destination**.

   The **Create Open Hub Destination** dialog box appears.
4. On the **General** tab in the **Create Open Hub Destination** dialog box, select a data source from which to extract data.
5. Enter an open hub destination name and description.
6. Select the **ObjectType** from which you want to update the data to the open hub destination.
7. Select the object listed for the selected object type and click enter. The open hub destination is added to the InfoArea.
8. Select **Third-Party Tool** as the destination in the **Destination** tab.
9. Enter an RFC destination.
10. Save and activate the open hub destination.

Create a Data Transfer Process

Create a data transfer process (DTP) to transfer data from source objects to target objects in SAP BW.

1. Right-click the open hub destination and select **Create Data Transfer Process**.

   The **Creation of Data Transfer Process** dialog box appears.
2. Enter a name for the DTP in the Data Transfer Proc field.
3. Select the object type and the name of the source for the DTP.

   Note: Use the same object type and the name that you selected for the open hub destination.
4. Save and activate the DTP.

Step 2. Create an SAP BW OHS Source Definition

Create a mapping with an SAP BW OHS source definition. When you create an SAP BW OHS source definition, use a connection string or a host name to connect to SAP. To connect to SAP using a connection string, use a connection string specified as a DEST entry in the `sapnwrfc.ini` file.

The following table lists the connection parameters you enter to connect to SAP:

Connection Parameter	Description
Connection String	Required. DEST entry defined in the `sapnwrfc.ini` file for a connection to a specific SAP application server or for a connection that uses SAP load balancing.
User Name	Required. SAP source system connection user name.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

To connect to SAP using a host name, use a host name and a system number that you specify in the Connection Properties window.

The following table lists the connection parameters you enter to connect to SAP:

Connection Parameter	Description
Host Name	Required. Host name of the machine where the SAP application is running.
System Number	Required. SAP system number.
User Name	Required. SAP source system connection user name.
Password	Required. Password for the user name.
Client	Required. SAP client number.
Language	Optional. Language you want for the mapping. Must be compatible with the PowerCenter Client code page. If you leave this option blank, PowerCenter uses the default language of the SAP system.

When you connect to SAP, the Import InfoSpokes from SAP BW wizard displays the InfoSpokes that you can import as SAP BW OHS source definitions. You can enter a filter condition to filter the InfoSpokes you want to appear.

The format of InfoSpokes depends on the DataSource they belong to. You can import an InfoSpoke from one of the following DataSources:

- DataStore
- InfoCube
- InfoObject(Texts)
- InfoObject(Attributes)

Note: Previous versions of SAP NetWeaver BW refer to DataStore as ODS Object.

When you import multiple InfoSpokes, the Designer creates separate SAP BW OHS source definitions for each InfoSpoke. You can import multiple InfoSpokes at a time. Each mapping can include only one SAP BW OHS source definition.

When you import an SAP BW OHS source definition, Designer client replaces the slash (/) in the business name of the Source Qualifier with an underscore (_).

You can filter InfoSpokes based on the name or the description of the InfoSpoke. When you enter a filter condition, you can enter an exact name or description. Or, you can perform wildcard search. Use the following filter syntax rules and guidelines:

- If you want to filter on a description, enter a filter condition of 20 characters or fewer.
- Use the percent sign (%) as a wildcard search for multiple characters. For example, to find an InfoSpoke whose name begins with the letter "I", enter `I%` as the filter criterion.
- Use an underscore (_) as a wildcard search for single characters.

The following table describes the information displayed when you select an InfoSpoke:

Field	Description
Name	Name of the record. A record can be user-defined or SAP defined. An SAP-defined record is uniquely identified by a Request ID, Data Packet ID, or Record ID.
Description	Description of the field.
Datatype	Datatype of the object.
Precision	Precision of the object.
Scale	Scale of the object.

To create an SAP BW OHS source definition:

1. In the Source Analyzer, click Sources > Import InfoSpoke from SAP BW.

   The Import InfoSpoke from SAP BW Wizard appears.
2. Enter the information in the Connection Properties page and click Next.

   The InfoSpoke Selection window appears.
3. Enter a filter condition to filter InfoSpokes based on the name or description.
4. Click Get Objects.

   The wizards displays a list of InfoSpokes based on the filter criteria you entered.
5. Select the InfoSpokes you want to import.
6. Optionally, select Show Selected Objects Only to view only the selected InfoSpokes in the Objects pane.
7. Click Next.

   The InfoSpoke Summary Page appears.
8. Click Finish.

Step 3. Start the SAP BW Service

Start the SAP BW Service from the Administrator tool. The SAP BW Service is an application service that listens for RFC requests from the SAP BW system, initiates workflows to extract from or load to the SAP BW system, and sends log events to the PowerCenter Log Manager.

Step 4. Configure and Start a PowerCenter Workflow

To configure a PowerCenter workflow to extract data from SAP BW, configure the workflow with an SAP BW OHS source definition. Assign and SAP_BWOHS_Reader application connection to the source. Schedule the workflow to run on demand.

Note: Use the exact name of the workflow when you insert the ZPMSENDSTATUS ABAP program in "Step 5. Configure and Start a Process Chain" on page 225.

The following table describes the session level properties:

Attribute	Description
Raw Data Transfer	Enables raw data transfer.
Log Lock Re-try Count	The number of attempts to acquire a lock to write to log. If you do not enable raw data transfer, set the value to 300. The maximum value is 2147483645. Default value is 10000.
Recovery Cache Folder	The file cache folder to enable recovery for the session. Default cache folder is $PMCacheDir.

Configuring Raw Data Transfer

You can configure the SAP BW OHS session to enable the PowerCenter Integration Service to read data from an SAP BW source through a data transfer API. The performance of the SAP BW OHS session increases when you read data from the SAP BW source using the RAW data transfer API.

To read data, you must install SAP BW version 7.3 Support Package 5 or later.

You can enable raw data transfer for SAP BW source. The PowerCenter Integration Service uses the UTF-16 code page if you extract data in Unicode mode and the ISO 8859-1 code page if you extract data in ASCII mode.

1. In the Task Developer of the **Workflow Manager**, double-click the SAP BW OHS session to open the session properties.
2. Click the **Mapping** tab.
3. From the Source settings on the **Mapping** tab, select **Raw Data Transfer**.
4. Optionally, change the **Log Lock Re-try Count**.

Step 5. Configure and Start a Process Chain

Configure and start a process chain to extract data in SAP BW or PowerCenter.

Configuring and Starting a Process Chain from SAP BW

Complete the following steps to configure and start a process chain in SAP BW:

1. Create a process chain and insert the start process.
2. Insert the ZPMSENDSTATUS ABAP program.
3. Insert an InfoSpoke process.
4. Start the process chain.

Creating the Process Chain and Inserting the Start Process

When you create the process chain and insert the start process, you also schedule the process chain.

To create the process chain and insert the start process:

1. From the Administrator Workbench in SAP BW, click SAP Menu > Administration > RSPC - Process Chains.

   The Process Chain Maintenance Planning View window appears.
2. Click Create.

   The New Process Chain dialog box appears.
3. Enter a unique name for the process chain and enter a description.
4. Click Enter.

   The Insert Start Process dialog box appears.
5. Click Create.

   The Start Process dialog box appears.
6. Enter a unique name for the start process variant and enter a description.
7. Click Enter.

   The Maintain Start Process window appears.
8. Click Change Selections to schedule the process chain.

   The Start Time window appears.
9. To schedule the process chain to run immediately after you execute it, click Immediate.
10. Click Save.
11. In the Maintain Start Process window, click Cancel.
12. In the Insert Start Process dialog box, click Enter.

    The start process appears in the Process Chain Maintenance Planning View window.

Inserting the ZPMSENDSTATUS ABAP Program

Before you can insert the ZPMSENDSTATUS ABAP program into a process chain, import the program into SAP BW.

To insert the ZPMSENDSTATUS ABAP program:

1. In the Process Chain Maintenance Planning View window in SAP BW, click Process Types.

2. From the Process Types menu, click General Services > ABAP Program.

   The Insert ABAP Program dialog box appears.
3. Click Create.

   The ABAP Program dialog box appears.
4. Enter a unique name for the ABAP program process variant and enter a description.
5. Click Enter.

   The Process Maintenance: ABAP Program window appears.
6. In the Program Name field, click the browse button to select the ZPMSENDSTATUS ABAP program.
7. Click Change next to the Program Variant field.

   The ABAP: Variants - Initial Screen window appears.
8. Click Create.
9. In the ABAP: Variants dialog box, enter a name for the ABAP variant and click Create.

   The Maintain Variant window appears.
10. In the Maintain Variant window, enter the Maintain Variant fields:

    The following table describes the Maintain Variant fields.

Field	Description
DEST	Select the name of the RFC destination.
INFPARAM	Enter the PowerCenter workflow name that you created to extract SAP BW data in one of the following formats: - <PowerCenter workflow name> - <PowerCenter folder name>: <PowerCenter workflow name> - <PowerCenter folder name>: <PowerCenter workflow name>: <PowerCenter session name>
CONTEXT	Enter OHS API.
INFOPAK	Leave this field blank.
OHDEST	Value you specified for the Destination field when you created the InfoSpoke.

11. Click Save and Exit in the Maintain Variant window.
12. Click Save and Exit in the ABAP Variants window.
13. Click Save and Exit in the Process Maintenance: ABAP Program window.
14. Click Enter in the Insert ABAP Program dialog box.

    The ABAP program appears in the Process Chain Maintenance Planning View window.
15. Link the Start process to ZPMSENDSTATUS ABAP program.
16. Enter selection criteria to specify which process chains you want to monitor and click Execute.

    The Job Overview window appears.
17. Select the BI_PROCESS_ABAP job and click Job Log.

    The Job Log Entries window appears. It includes an entry about the status of the PowerCenter workflow that the process chain was configured to start.

Inserting a Data Transfer Process

Insert a process for open hub destination that you created in SAP BW.

To insert an open hub destination process:

1. In the Process Chain Maintenance Planning View window, click Process Types.
2. From the Process Types menu, click Load Process and Post-Processing > Data Transfer Process.

   The Insert Data Transfer Process dialog box appears.
3. For the Data Transfer Process field, click the browse button to select the Data Transfer Process you created.
4. Click Enter.

   The data transfer process appears in the Process Chain Maintenance Planning View window.
5. Link ZPMSENDSTATUS ABAP program to the data transfer process.

   The process chain flow is: Start -> ABAP Program (zpmsendstatus) -> data transfer process.
6. Click Checking View and then click Activate.
7. Click Execute and assign the process chain to a specific SAP BW server.

   If you scheduled the process chain to run immediately, the process chain starts running on the assigned SAP BW server.
8. Optionally, to view the status of the process chain, click Job Overview.

Inserting an InfoSpoke Process

Insert a process for the InfoSpoke you created in SAP BW.

To insert an InfoSpoke process:

1. In the Process Chain Maintenance Planning View window, click Process Types.
2. From the Process Types menu, click Load Process and Post-Processing > Data Export Into External Systems.

   The Insert Data Export into External Systems dialog box appears.
3. For the Process Variants field, click the browse button to select the InfoSpoke you created.
4. Click Enter.

   The InfoSpoke process appears in the Process Chain Maintenance Planning View window.
5. Link ZPMSENDSTATUS ABAP program to the InfoSpoke process.

   The process chain flow is: Start -> ABAP Program (zpmsendstatus) -> InfoSpoke.
6. Click Checking View and then click Activate.
7. Click Execute and assign the process chain to a specific SAP BW server.

   If you scheduled the process chain to run immediately, the process chain starts running on the assigned SAP BW server.
8. Optionally, to view the status of the process chain, click Job Overview.

Configuring and Starting the Process Chain from PowerCenter

Use the command line program *pmsaputil* to configure and start the process chain in PowerCenter. You can run the following commands:

- **setparams.** Run to set third-party parameters for the InfoSpoke, which is part of the process chain.

- **startchain.** Run to start the process chain and to optionally set third-party parameters for the InfoSpoke, which is part of the process chain.

If you configure the third-party parameters with the ZPMSENDSTATUS ABAP program, you cannot configure the parameters with *pmsaputil* setparams. *pmsaputil* startchain uses the parameters configured in the ZPMSENDSTATUS ABAP program to start the process chain.

If you configure the third-party parameters with the setparams command, you do not need to enter the parameters again when you run the startchain command.

Tip: To view help for *pmsaputil*, enter *pmsaputil* help setparams or *pmsaputil* help startchain.

Before you run a command, encrypt the password to connect to SAP with *pmpasswd*.

The setparams command uses the following syntax:

```
setparams
<-UserName|-un> user_name
<-PassWord|-pd> password
{[<-HostName|-hn> host_name
<-SystemNumber|-sy> system_number>] |
<-Destination|-ds> dest_entry_in_sapnwrfc_ini}
<-ClientNumber|-cl> client_number
[<-Language|-lg> language]
[<-Trace|-tr> trace]
<-Context|-cn> context
<-FolderName|-fn> folder_name
<-WorkflowName|-wn> workflow_name
[<-SessionName|-sn> session_name]
<-OpenHubDest|-od> openhub_dest
<-ThirdPartyDest|-td> 3rdparty_dest
```

The startchain command uses the following syntax:

```
startchain
<-UserName|-un> user_name
<-PassWord|-pd> password
{[<-HostName|-hn> host_name
<-SystemNumber|-sy> system_number>] |
<-Destination|-ds> dest_entry_in_sapnwrfc_ini}
<-ClientNumber|-cl> client_number
[<-Language|-lg> language]
[<-Trace|-tr> trace]
<-Context|-cn> context
[<-FolderName|-fn> folder_name]
[<-WorkflowName|-wn> workflow_name]
[<-SessionName|-sn> session_name]
[<-OpenHubDest|-od> openhub_dest]
[<-ThirdPartyDest|-td> 3rdparty_dest]
<-ProcessChainName|-pc> processchain_name
```

The following table describes *pmsaputil* setparams and startchain options and arguments:

Options	Argument	Description
-UserName -un	user_name	Required. SAP source system connection user name.
-PassWord -pd	password	Required. Encrypted password. Run *pmpasswd* to encrypt passwords.
-HostName -hn	host_name	Required if you do not enter a destination entry. SAP host name.

Options	Argument	Description
-SystemNumber -sy	system_number	Required if you enter an SAP host name. SAP system number.
-Destination -ds	dest_entry_in_sapnwrfc_ini	Required if you do not enter a host name and system number. Destination entry in the `sapnwrfc.ini` file.
-ClientNumber -cl	client_number	Required. SAP client number.
-Language -lg	language	Optional. Language in which you want to receive messages from the SAP BW system. Use a language code valid for the SAP BW system you are connecting to. If you leave this blank, PowerCenter uses the default language of the SAP system to connect to SAP BW.
-Trace -tr	trace	Optional. Tracks the RFC calls that the SAP system makes. SAP stores the information about the RFC calls in a trace file.
-Context -cn	context	Enter BWOHS for the setparams command. For the startchain command, optionally enter BWOHS.
-FolderName -fn	folder_name	Required for the setparams command. Optional for the startchain command. Name of the repository folder that contains the workflow.
-WorkflowName -wn	workflow_name	Required for the setparams command. Optional for the startchain command. PowerCenter workflow name that you created to extract SAP BW data.
-SessionName -sn	session_name	Optional. Name of the PowerCenter sessions you want to start to extract SAP BW data. Starts the sessions that you configure in the workflow. If you omit this option PowerCenter Integration Service starts all the sessions in the workflow.
-OpenHubDest -od	openhub_dest	Required for the setparams command. Optional for the startchain command. InfoSpoke destination in the SAP system.
-ThirdPartyDest -td	3rdparty_dest	Required for the setparams command. Optional for the startchain command. Third-party destination in the SAP system.
-ProcessChainName -pc	processchain_name	Required. Name of the process chain you want to start. Use in the startchain command only.

Changing the Data Extraction Status

When a data extraction session in PowerCenter fails, you must change the status of the failed session in SAP BW before you can start another session. Set the status of the failed request ID to G.

To change the data extraction status:

1. Go to transaction SE37.

   The Function Builder: Initial Screen appears.

2. In the Function Module field, enter RSB_API_OHS_REQUEST_SETSTATUS and click Test/Execute.

   The Test Function Module: Initial Screen appears.

3. Enter the import parameters and click Execute:

   The following table describes the import parameters:

Import Parameter	Description
Request ID	Request ID of the failed session. You can identify the Request ID of the failed session from the session log or BW monitor.
Status	G, which represents a successful data extraction session.
Message	Description of the change you make.

Viewing Data Extraction Status

You can view the status of data extraction from the Administrator tool and from the BW monitor in the SAP system.

Viewing Data Extraction Status from PowerCenter

The SAP BW Service captures log events that track interactions between PowerCenter and SAP BW. It captures its own log events. It also captures log events when it receives the following information from the SAP BW system and the PowerCenter Integration Service:

- A request from the SAP BW system to start a PowerCenter workflow.
- A message from the PowerCenter Integration Service that it has successfully started a workflow to extract data from SAP BW.
- A message from the PowerCenter Integration Service indicating whether the PowerCenter session failed or succeeded.

When you extract data from SAP BW, you can view SAP BW Service log events in the Administrator tool. On the Log tab, enter search criteria to find SAP BW Service log events. You can also view status of data extraction from the BW monitor.

To view log events about how the PowerCenter Integration Service processes an SAP BW workflow, view the session or workflow log.

Viewing Data Extraction Status from SAP

You can view the data extraction status in SAP from the SAP BW monitor in SAP BW. It shows green for successful data extraction and red for failed data extraction. The SAP BW monitor also shows the following information about data extraction:

- Number of packets sent

- Number of rows in each packet
- Time taken for sending the packets
- Start time and end time

If the data extraction fails, you can start the InfoSpoke again to start the data extraction process. You can start the InfoSpoke again only after the PowerCenter Integration Service completes extracting data from the database table and sends the status to the SAP BW monitor.

Troubleshooting Data Extraction from SAP BW through OHD

The SAP BW Process Chain log reports that the RFC destination is not valid.

The RFC destination specified in the process chain is not accurate. Make sure that the RFC destination for the process chain is valid.

The SAP BW Process Chain log reports that a folder, workflow, or session name is not valid.

The folder, workflow, or session name that you specified when creating the process chain is not accurate. Check the workflow for the folder, workflow, or session name, and update the process chain accordingly.

I successfully extracted data from SAP BW, but the SAP BW Service log includes irrelevant messages about the session.

This issue occurs if the value for the CONTEXT field in the ZPMSENDSTATUS ABAP program is not valid when creating the process chain. Enter OHS for the CONTEXT field in a process chain that extracts data.

Part VIII: Loading Data to SAP BI

This part contains the following chapters:

- Building Components to Load Data into SAP BI, 233
- Building PowerCenter Objects to Load Data into SAP BI, 243
- Loading Data into SAP BI, 250

CHAPTER 20

Building Components to Load Data into SAP BI

This chapter includes the following topics:

- Building Components to Load Data into SAP BI Overview, 233
- Step 1. Create an InfoSource or DataSource, 237
- Step 2. Assign an External Logical System, 242
- Step 3. Activate the InfoSource or DataSource, 242

Building Components to Load Data into SAP BI Overview

To load data into SAP BI, you need to create components in the SAP BI system.

In SAP BI versions earlier than 7.3, you can create an InfoSource, assign it to the PowerCenter logical system that you created in SAP BI, and activate the InfoSource. When you create and activate the InfoSource, you specify the InfoSource type and the transfer method that the PowerCenter workflow uses to write data to the InfoSource.

In an SAP BI version 7.3 system, you can create a 7.x DataSource with the source system as the PowerCenter logical system that you created in SAP BI. You can activate the 7.x DataSource and use it to load data to the Persistent Storage Area (PSA).

In an SAP BI version 7.3 system, you can also create a 7.x InfoSource and associate it with a 7.x DataSource through transformations. You can then use the 7.x DataSource to load data to an InfoProvider.

Note: You can also load data into SAP BW 3.5 or earlier. The steps to load data into SAP BI also apply to SAP BW 3.5 except where noted.

InfoSources for Loading Data

You can write data to InfoSources when you run a PowerCenter workflow. The PowerCenter Integration Service can load data into two types of InfoSources:

- **InfoSource for transaction data.** Load data that changes often and is dependent on master data into an InfoSource for transaction data. For example, you can assign transaction data about sales development to a vendor's master data and use the transaction data to determine the total sales of a vendor.

- **InfoSource for master data.** Load frequently-used data that remains constant over a long period of time into an InfoSource for master data. For example, the master data of a vendor may contain the name, address, and bank account information for the vendor. Create an InfoSource for master data when you want to load data into an SAP BI hierarchy.

DataSources for Loading Data

DataSources contain a logical grouping of fields that describe the metadata of the source. DataSources are used to extract data from a source system and transfer the data to the BI system.

You can import a 7.x DataSource from an SAP BI version 7.3 system as a target in the PowerCenter Designer and include it in a mapping. Create a workflow for the mapping and configure an InfoPackage in the SAP system to load data to the PSA. Configure a Data Transfer Process (DTP) and define a process chain linking the InfoPackage and DTP to transfer the data from the PSA to an SAP target.

You can create the following types of 7.x DataSources:

- DataSource for transaction data
- DataSource for master data text
- DataSource for master data attributes
- DataSource for hierarchies

SAP BI Hierarchy

An SAP BI hierarchy is a tree-like structure that defines classes of information. Each level of the hierarchy represents a different class. A hierarchy displays an SAP BI characteristic, which is a reference object with dimensions. You can create a PowerCenter workflow to load data into an SAP BI hierarchy.

The hierarchy is structured and grouped according to individual evaluation criteria based on the characteristic. The structure at each level of the hierarchy is called a node.

A hierarchy has the following types of nodes:

- **Root node.** The root node is the highest node in the structure and is the origin of all other nodes. The root node represents the hierarchy.
- **Child node.** A child node is a node that is subordinate to another node.
- **Leaf node.** Leaf nodes are the lowest level in the hierarchy. Leaf nodes do not have any successor nodes.

To load data into a hierarchy, create an InfoSource or DataSource for master data in the target SAP BI system. After you create the InfoSource or DataSource, import the hierarchy as a target definition into the Designer. When you import the definition of a hierarchy, the Designer creates a transfer structure with fields that make up the structure of the SAP BI hierarchy.

You can configure the structure of a hierarchy in the InfoSource or DataSource properties. The following settings define the structure of an SAP BI hierarchy:

- **Sorted hierarchy.** A sorted hierarchy defines the sequence of nodes for each level of a hierarchy. When you specify a sorted hierarchy, each child node is ordered in relation to its sibling nodes.
- **Interval hierarchy.** An interval hierarchy specifies a range of characteristic values. You can use one interval to represent multiple leaf nodes. You define an interval in SAP BI by setting a range of values.
- **Time-dependent hierarchy.** A time-dependent hierarchy specifies a range of dates. You can define either the entire hierarchy or the hierarchy structure as time-dependent. When the entire hierarchy is time-dependent, it is valid only in a defined range of dates. When the hierarchy structure is time-dependent, the

nodes in the hierarchy change for specified periods of time while the name and version of the hierarchy remain static. You define a date range in SAP BI by setting a range of dates.

- **Version-dependent hierarchy.** Version-dependent hierarchies have the same name, but different versions.

When you import an InfoSource or DataSource with a hierarchy as an SAP BW target definition, the Designer only creates the fields in the transfer structure definition that are required to replicate the structure of the SAP BI hierarchy.

The following table shows the fields the Designer can create in the SAP BW target definition when you import metadata for a hierarchy definition from SAP BI:

Field Name	Hierarchy Type	Description
NODEID	All hierarchies	Local and unique identifier for the hierarchy node.
INFOOBJECT	All hierarchies	InfoObject that is referenced by the hierarchy node.
NODENAME	All hierarchies	Name of the hierarchy node.
LINK	All hierarchies	Specifies a field as a link node.
PARENTID	All hierarchies	NODEID of the parent node of the hierarchy node.
CHILDID	Sorted hierarchy	NODEID of the first child node of the hierarchy node.
NEXTID	Sorted hierarchy	NODEID of the sibling node following the hierarchy node.
DATEFROM	Time-dependent hierarchy	Start date in a range of dates.
DATETO	Time-dependent hierarchy	End date in a range of dates.
LEAFFROM	Interval hierarchy	Lower limit for an interval node.
LEAFTO	Interval hierarchy	Upper limit for an interval node.
LANGU	All hierarchies	Language.
TXTSH	All hierarchies	Short text.
TXTMD	All hierarchies	Medium text.
TXTLG	All hierarchies	Long text.

If you import different versions of a hierarchy into an SAP BW target definition, the hierarchy name includes the version. If you import a hierarchy with time-dependent values, the hierarchy name contains the date range specified for the hierarchy.

During a PowerCenter workflow, the PowerCenter Integration Service loads the hierarchy into the transfer structure fields.

Transfer Methods for Writing Data to SAP BI

Specify which transfer method you want to use to load data into SAP BI. Specify the transfer method in the SAP BI Administrator when you define the Transfer Rules-Transfer Method and the InfoPackage-Data Target.

You can use the following transfer methods to load data into SAP BI:

- IDoc transfer method
- PSA transfer method

In an SAP BI version 7.3 system, you use the PSA transfer method to load data to the PSA using a 7.x DataSource.

IDoc Transfer Method

Use to synchronously move data from transfer structures to InfoCubes. When you use the IDoc transfer method, the PowerCenter Integration Service loads data into a transfer structure for SAP BI. The IDoc transfer method lets you process a data load while the InfoPackage runs.

PSA Transfer Method

Use when you want to load data into the Persistent Storage Area (PSA) before writing the data to the Operational Data Store (ODS) or InfoCube. When you use the PSA transfer method in SAP BI, the PowerCenter Integration Service writes data to SAP BI, which stores the data in a PSA. SAP BI updates or transforms the data after it has been stored in the PSA.

When you load data to SAP BI 7.3 using a 7.x DataSource, the transfer option is configured to load data to the PSA because the InfoPackage loads data only to the PSA.

When you load data using 3.x data sources, use one of the following PSA transfer options:

- **PSA and then into Data Targets.** Select this option when you want to store data in the PSA before you load the data to data targets such as an ODS, InfoCube, or InfoSource.
- **PSA and Data Targets in Parallel.** Select this option when you want to load data to the PSA and to data targets at the same time to achieve high performance.
- **Only PSA.** Select this option when you want to load data only to the PSA. You can also select the **Update Subsequently in Data Targets** option to load the data to data targets after you load the data to the PSA.
- **Data Targets Only.** Select this option when you want to load data only to data targets such as ODS, InfoCube, or InfoSource.

For optimal performance, configure the transfer option to load data to the PSA. After the source system loads the PSA, you can update the InfoCubes in SAP BW.

Steps to Build Components to Load Data into SAP BI

To build components to load data into SAP BI, complete the following steps in SAP BI:

1. Create an InfoSource or a DataSource based on the version of the SAP BI system. In SAP BI versions earlier than 7.3, you can create an InfoSource. In an SAP BI version 7.3 system, you can create a 7.x DataSource or a 7.x InfoSource.
2. Assign the InfoSource to the PowerCenter logical system.
3. Activate the InfoSource or DataSource.

After you build and activate the components, you can import the InfoSources or DataSources into PowerCenter and build mappings.

Step 1. Create an InfoSource or DataSource

InfoSources are equivalent to target tables in the SAP BI operational data store. The logical system you created for PowerCenter in SAP BI populates the InfoSource with data.

DataSources are equivalent to target tables in the PSA layer. DataSources define the metadata of the source system and are used to transfer data to the SAP BI system.

Creating a 7.x InfoSource in SAP BI 7.3

When you load data to SAP BI 7.3, you can create a 7.x InfoSource to connect multiple 7.x DataSources to a target and all the DataSources have the same business rules. You use the 7.x DataSources as the source when you create a transformation in SAP BI to load data to an InfoProvider.

1. In the Data Warehousing Workbench, click **InfoSources.**
2. Right-click the InfoSources folder, and select **Create application component**.
3. Enter the application component parameters and click **Check**.

   The following table describes the application component parameters:

Parameter	Description
Application Comp	Organizes logical systems.
Long Description	Description of the application component.

   The application component appears in the Workbench.
4. Right-click the application component and select **Create InfoSource**.

   The **Create InfoSource** dialog box appears.
5. Enter the InfoSource parameters and click **Check**.

   The following table describes the InfoSource parameters:

Parameter	Description
InfoSource	Name of the InfoSource.
Long description	Description of the InfoSource.
Copy from	Leave this section blank.

Creating an InfoSource in SAP BI 7.0

When you load data to SAP BI 7.0, you can create one of the following objects:

- InfoSource
- 3.x InfoSource

Creating an InfoSource

Create an InfoSource to use as target when you create a transformation in SAP BI to load data to an InfoProvider.

1. In the Data Warehousing Workbench, click **InfoSources.**
2. Right-click the InfoSources folder, and select **Create application component**.
3. Enter the application component parameters and click **Check**.

   The following table describes the application component parameters:

Parameter	Description
Application Comp	Organizes logical systems.
Long Description	Description of the application component.

   The application component appears in the Workbench.
4. Right-click the application component and select **Create InfoSource**.
5. In the **Create InfoSource** dialog box, choose the InfoSource type.

   Choose **Direct update of Master Data** to create an InfoSource with a hierarchy.
6. Click **Check**.

   The InfoSource appears in the Data Warehousing Workbench.

Creating a 3.x InfoSource

You can create a 3.x InfoSource when you want to load data to the PSA using an InfoPackage. SAP BI creates a data source when you activate the 3.x InfoSource. You use the 3.x data source as the source when you create a transformation in SAP BI to load data to an InfoProvider or OHS.

1. In the Data Warehousing Workbench, click **InfoSources.**
2. Right-click the InfoSources folder, and select **Create application component**.
3. Enter the application component parameters and click **Check**.

   The following table describes the application component parameters:

Parameter	Description
Application Comp	Organizes logical systems.
Long Description	Description of the application component.

   The application component appears in the Workbench.
4. Right-click the application component and select **Additional Functions** > **Create InfoSource 3.x**.

   The **Create InfoSource** dialog box appears.
5. Select **Flexible Update in Any Data Target (Except Hierarchies)** and enter the InfoSource parameter information and click **Check**.

The following table describes the InfoSource parameters:

Parameter	Description
InfoSource	Name of the InfoSource.
Long description	Description of the InfoSource.
Template-InfoSource	Leave this option blank.

The InfoSource appears in the Data Warehousing Workbench.

Creating an InfoSource in SAP BW 3.5 or Earlier

Use the following procedure to create an InfoSource in SAP BW 3.5 or earlier.

1. In the Administrator Workbench, click **InfoSources**.
2. Right-click the InfoSources folder, and select **Create application component**.
3. Enter the following information and click **Check**:

   The following table describes the application connection parameters:

Parameter	Description
Application Comp	Organizes logical systems.
Long Description	Description of the application component.

   The application component appears in the Workbench.
4. Right-click the application component and select **Create InfoSource**.
5. In the **Create InfoSource** dialog box, choose the InfoSource type.

   Choose **Direct update of Master Data** to create an InfoSource with a hierarchy.
6. Click **Check**.

   The InfoSource appears in the Administrator Workbench.

Creating a 7.x DataSource in SAP BI 7.3

You can create a 7.x DataSource when you want to load data to the PSA using an InfoPackage. You can then configure a DTP to transfer the data from the PSA to an SAP target.

1. Define PowerCenter as a logical system in the SAP BI 7.3 system.
2. Create an SAP BW Service using the Administrator tool and enable the service.
3. In the Data Warehousing Workbench, click **DataSources**.
4. Right-click the DataSources folder, and select **Create application component**.
5. Enter the application connection parameters and click **Check**:

The following table describes the application connection parameters:

Parameter	Description
Application Comp	Organizes logical systems.
Long Description	Description of the application component.

The application component appears in the Workbench.

6. Right-click the application component and select **Create DataSource**.

   The **Create DataSource** dialog box appears.

7. Enter the DataSource parameters and click **Check**.

   The following table describes the DataSource parameters:

Parameter	Description
DataSource	Name of the DataSource.
Source system	Name of the source system that populates the DataSource with data.
Data Type DataSource	Type of the DataSource that you want to create. You can create the following types of DataSources: - DataSource for transaction data - DataSource for master data text - DataSource for master data attributes - DataSource for hierarchies

The DataSource appears in the Data Warehousing Workbench.

8. Click the **Fields** tab and add InfoObjects to the DataSource.

   The InfoObjects you use in the DataSource display as ports in the PowerCenter targets.

9. Select the data format options for the InfoObjects.

The following table describes the data format options:

Parameter	Description
Internal Format	Indicates if the data from the source is in a secure internal format. When you select this option, SAP does not verify or convert the format. Therefore, if the source does not return data in an internal format, there could be data inconsistencies or errors in the loading process.
External Format	Indicates if the data from the source is in an external format. When you select this option, SAP converts the data into an internal format before it passes the data to the transformation.
Check for Internal Format	Indicates if the SAP system must verify whether the data from the source system is in an internal format. Select this option if the data from the source is in an internal format, but it must be verified before being passed to the transformation. Select this option to ensure data conformity.

10. After creating metadata, click **Check** to return to the Data Warehousing Workbench.

Configuring Hierarchy Structure

After creating an InfoSource, you can include an InfoObject with a hierarchy in the InfoSource. After associating the InfoObject containing a hierarchy with an InfoSource, you can configure the hierarchy structure in the InfoSource properties.

To create an InfoSource with a hierarchy in SAP BI, specify that the InfoObject you want to include in the InfoSource will be used in a hierarchy. On the Hierarchy tab of the InfoObject details window, ensure that “with hierarchies” is selected. You assign this InfoObject to the InfoSource when you create the InfoSource. To configure and use an SAP BI hierarchy, create an InfoSource for master data.

After you create the InfoSource, select the InfoSource properties to configure the hierarchy structure.

1. Double-click the InfoSource.
2. Select **Transfer_Structure/Transfer_Rules**.
3. Enter values for the **Source System** and the **DataSource** options.

   Ensure the value for the DataSource option is a hierarchy.
4. Click **Hier. Structure** to enter the name for the hierarchy.

   You can select additional properties to configure the structure of the hierarchy.
5. Save the InfoSource.

Step 2. Assign an External Logical System

After you create an InfoSource, you need to associate it with the external logical system you created for PowerCenter in SAP BI. You also need to add metadata to the InfoSource.

1. In the Data Warehousing Workbench for SAP BI 7.0, right-click the 3.x InfoSource and select **Assign DataSource**. In the Administrator Workbench for SAP BW 3.5 or Data Warehousing Workbench for SAP BI 7.0, right-click the InfoSource and select **Assign DataSource**.
2. Select the external logical system that you created for PowerCenter and click **Check**.
3. Add InfoObjects to the InfoSource.

   The InfoObjects you use in the InfoSource display as ports in the PowerCenter targets.
4. After creating metadata, click **Check** to return to the Administrator Workbench or Data Warehousing Workbench.

Step 3. Activate the InfoSource or DataSource

After you create an InfoSource or a DataSource and assign it to an external logical system, you must activate the InfoSource or DataSource to load data to SAP BI.

The InfoSource contains the metadata used as the basis for the transfer and communication structure. When you activate the InfoSource, you also maintain the transfer rules and communication structure. Transfer rules must be active for PowerCenter to load data into the transfer structure.

The PowerCenter Integration Service loads data to the PSA based on the DataSource. To activate a DataSource, select the DataSource and click the **Activate** button. When you activate the DataSource, SAP generates a PSA table in the entry layer of BI. You can then load the data into the PSA and create a DTP to load the data from the PSA to an SAP target.

1. In the Administrator Workbench for SAP BW 3.5 or Data Warehousing Workbench for SAP BI 7.0, right-click the InfoSource and select **Change**.
2. Select the InfoObjects and move them into the communication structure.
3. Click the **Activate** button.
4. Click the **Transfer rules** tab.
5. Select the transfer method and click **Activate**:

   The following table describes the transfer methods:

Method	Description
IDoc	Use IDoc to synchronously move data from transfer structure to InfoCube.
PSA	Required for SAP BI 7.0. Use PSA to load data into the PSA.

Note: If you want to populate an InfoCube in SAP BW 3.5, also define it in the Administrator Workbench. Define the update rules to update the InfoCube from the transfer structure.

CHAPTER 21

Building PowerCenter Objects to Load Data into SAP BI

This chapter includes the following topics:

- Building PowerCenter Objects to Load Data into SAP BI Overview, 243
- Step 1. Import an InfoSource or a DataSource, 243
- Step 2. Create a Mapping, 245
- Filtering Data to Load into SAP BI, 245

Building PowerCenter Objects to Load Data into SAP BI Overview

Based on the version of the SAP BI system, you can create and activate an InfoSource or a DataSource. You can use the Designer to import a 3.x data source or a 7.x data source as an SAP BW target definition. You can then include the SAP BW target definition in a mapping to load data into SAP BI.

When you create a mapping, you can configure the mapping to filter data before loading the data into an SAP BI target. Filtering data increases session performance when the selection occurs during the extraction process, minimizing the amount of records loaded into SAP BI.

To filter data, you configure a data selection in the **Data Selection** tab of the InfoPackage. You then configure the Source Qualifier or Filter transformation in the mapping to use mapping parameters to represent the data selection entry you configured in SAP BI.

Step 1. Import an InfoSource or a DataSource

In the Target Designer, you can connect to an SAP BI data source, browse its contents, and import selected transfer structures as targets.

1. From the Target Designer, click **Targets** > **Import from SAP BW**.
2. In the **Import SAP BW Metadata** dialog box, enter the login information.

The following table describes the login information:

Field	Description
Connect String	Required. DEST parameter defined in the client `sapnwrfc.ini` file for a connection to a specific SAP application server. When you import an InfoSource or a DataSource for the first time, the Designer reads the `sapnwrfc.ini` file and displays the first DEST entry as the connect string. After the first time you import an InfoSource or a DataSource, the Designer stores and displays the DEST entry used for the previous import. Select the DEST parameter that specifies the SAP BI source system from which you want to import InfoSources or DataSources.
User Name	Required. SAP BI user name.
Password	Required. Password for the user name.
Client	Required. SAP BI client number.
Language	Required. The language in which you want to receive messages from the SAP BI system while connected through this dialog box. Use a language code valid for the SAP BI system you are connecting to. If you leave this blank, PowerCenter uses the default language of the SAP system to connect to SAP BI.
DataSource Version	The version of the data source that you want to import. You can import a 3.x data source or a 7.x data source.

3. Click **Connect** to view the available InfoSources or DataSources.
4. From the list of transfer structures, find the InfoSource or DataSource you want to import.

   If you import an InfoSource or DataSource from the Master Transfer List, you can select to import Attribute and Text InfoSources or DataSources.
5. Select the InfoSources or DataSources you want to import.
   - Hold down the **Shift** key to select blocks of sources.
   - Hold down the **Ctrl** key to make non-contiguous selections within a folder.
   - Use the **Select All** button to select all tables.
   - Use the **Select None** button to clear all selections.
6. Click **Add To Import List**.
7. To view the list, click **View Import List**.

   The **Import List** dialog box appears.
8. To remove items from the list that you do not want to import, select the item and click **Delete**.
9. Click **Close** to close the **Import List** dialog box.

   You cannot include multiple InfoSources or DataSources with the same name from different source systems when you select InfoSources or DataSources to import. Import multiple InfoSources or DataSources with the same name from different source systems separately.
10. When the Import List is complete, click **OK**.

    The InfoSource or DataSource definitions display as target tables in the Target Designer. SAP BI may add /BIC/ to the InfoObject name.

    Tip: If the target displays the correct InfoSource or DataSource name, but no ports display, make sure you correctly followed the steps to create an InfoSource or a DataSource.

Step 2. Create a Mapping

When you import InfoSources or DataSources into the Target Designer, the PowerCenter repository stores them as target definitions. You can use these target definitions in mappings that load data into SAP BI.

The following restrictions apply to building mappings with InfoSource and DataSource targets:

- You cannot use SAP BI as a lookup table.
- You can only use one transfer structure for each mapping.
- You cannot execute stored procedures in an SAP BI target.
- You cannot build update strategy in a mapping. SAP BI supports only inserts. It does not support updates or deletes. You can use an Update Strategy transformation in a mapping, but the PowerCenter Integration Service attempts to insert all records, including those marked for update or delete.

Filtering Data to Load into SAP BI

When you want to filter data before loading into an SAP BI target, you need to configure the InfoPackage with data selections. After you configure the InfoPackage for data selection, you create a mapping in the PowerCenter Designer to filter data. Configure the mapping with mapping parameters that reference the data selections you have specified in the InfoPackage. Use the mapping parameters to represent the SAP BI data selection entry in the InfoPackage.

After you configure SAP BI and PowerCenter to filter data, you can start an SAP BI workflow to load data into SAP BI. When the SAP BI Scheduler sends a request to the PowerCenter Integration Service to start a workflow, it includes the SAP BI data selection entries as part of the request. The SAP BW Service converts the SAP BI data selection entries into the PowerCenter transformation language and writes the values that define the data selection entry into a temporary parameter file. It uses the SAP BI request ID as the name for the parameter file. For example, if the request ID is REQU_2AME24K7YDXL2DMA2YC0ZP9CM, the SAP BW Service creates a temporary parameter file of the same name.

When the PowerCenter Integration Service extracts data from a source system to load into SAP BI, it uses the temporary parameter file to evaluate the mapping parameters that you specified in the mapping to filter data. The SAP BW Service deletes the temporary parameter file after the workflow ends. During the workflow, you can view each data selection entry specified in the InfoPackage in the session log.

You specify the location for the parameter file when you create the SAP BW Service.

Related Topics:

- "Step 2. Configure an InfoPackage" on page 253

Filtering Data from a Relational Source

When you create a mapping to filter data from a relational source to load into SAP BI, perform the following tasks:

- **Ensure that the name of each source field is between three and nine characters.** Otherwise, the PowerCenter Integration Service does not apply the data selection entry that you configure in SAP BI when it extracts data from the relational source.
- **Create a mapping parameter called $$BWFILTERVAR to represent the data selection entries you entered in the InfoPackage.**

- **Use the mapping parameter in a filter condition to filter data from the source.** You enter a filter condition in the Source Qualifier transformation for a relational source.
- **Make sure the part of the field name after "/BIC/" in the SAP BW target definition matches the name of the field in the source definition.** For example, if the source field is named "LocationID," the target field must be named "/BIC/LocationID."

For example, you want to extract data from an Oracle source and load it into SAP BI. You want to filter data such that the PowerCenter Integration Service only extracts records where EmpID is between 2222 and 9999 and DeptID equals 804. You configure the InfoPackage accordingly for data selection.

The following table shows the data selection entries you configure in the InfoPackage:

InfoObject	FromValue	ToValue	Datatype
EmpId	2222	9999	NUMC
DeptId	804		NUMC

When the SAP BI Scheduler sends the SAP BW Service a workflow request, the SAP BW Service receives the data selection information for the relational source and writes it to the temporary parameter file. For example, the SAP BW Service writes the following to the temporary parameter file for the data selections:

```
$$BWFILTERVAR="EmpId" >= '2222' AND "EmpId" <= '9999' AND ("DeptId" = '804')
```

During a PowerCenter workflow, the PowerCenter Integration Service uses the value of the $$BWFILTERVAR mapping parameter to filter data from the Oracle source.

Filtering Data from a Flat File Source

When you want to filter data from a flat file source to load into SAP BI, create one or more mapping parameters based on the data selection entries configured in SAP BI. Create a mapping parameter for each value that defines a data selection entry in the InfoPackage.

For example, you have an InfoPackage that contains only one data selection entry. If the data selection entry specifies a value for the FromValue field without specifying a value for the ToValue field, you only create one mapping parameter. If the data selection entry specifies values for both the FromValue and the ToValue fields, you need to create a mapping parameter for each value.

After you create a mapping parameter for each value in the SAP BI data selection entries, use the mapping parameters in a filter condition to filter data from the source. You enter a filter condition in the Filter transformation for a flat file source.

For example, you want to extract data from a flat file source and load it into SAP BI. You want to filter data to extract only records where EmpID is between 1108 and 1129. You configure the InfoPackage accordingly for data selection.

The following table shows the data selection entry you configure in the InfoPackage:

InfoObject	FromValue	ToValue	Datatype
EmpId	1108	1129	NUMC

After configuring the data selection entry in the InfoPackage, create mapping parameters for the values that define the data selection.

The following table shows the mapping parameters you create for the data selection:

Parameter Name	Description
`$$EMPID_FROM_0`	Defines the start value for the range of the data selection entry.
`$$EMPID_TO_0`	Defines the end value for the range of the data selection entry.

Use the mapping parameters in the filter condition of the Filter transformation. For example, to represent the data selection entry you defined in the InfoPackage, enter the following in the filter condition:

```
EmpId >= $$EMPID_FROM_0 AND EmpId <= $$EMPID_TO_0
```

where EmpId represents the EmpId field in the mapping.

When the SAP BI Scheduler sends the SAP BW Service a workflow request, the SAP BW Service receives the data selection information for the flat file source and writes it to the temporary parameter file. For example, the SAP BW Service writes the following to the temporary parameter file for the data selection:

```
$$EMPID_FROM_0=1108
$$EMPID_TO_0=1129
```

During the workflow, the PowerCenter Integration Service uses the temporary parameter file to obtain values for the $$EMPID_FROM_0 and $$EMPID_TO_0 mapping parameters in the data selection entry. The PowerCenter Integration Service then uses the data selection entry to filter data from the source.

Filtering Data from an SAP ECC Source

When you create a mapping to filter data from an SAP ECC source to load into SAP BI, create one or more mapping parameters based on the data selection entries configured in SAP BI. Create a mapping parameter for each value that defines a data selection entry in the InfoPackage.

For example, you have an InfoPackage that contains only one data selection entry. If the data selection entry specifies a value for the FromValue field without specifying a value for the ToValue field, you only create one mapping parameter. If the data selection entry specifies values for both the FromValue and the ToValue fields, create a mapping parameter for each value.

After you create a mapping parameter for each value in the SAP BI data selection entries, use the mapping parameters in a filter condition to filter data from the source. Enter the filter condition in the Application Source Qualifier for an SAP ECC source.

For example, you want to extract data from the MARA SAP ECC table using a dynamic filter and load it into SAP BI. You want to filter data such that the PowerCenter Integration Service only extracts records where MATNR is between MR0842 and MT0727 and BLANZ is equal to 219. You configure the InfoPackage accordingly for data selection.

The following table shows the data selection entries in the SAP BI InfoPackage:

InfoObject	FromValue	ToValue	Datatype
MATNR	MR0842	MT0727	CHAR
BLANZ	219		NUMC

After you configure the data selection entry in the InfoPackage, you create mapping parameters for the values that define the data selection.

The following table shows the mapping parameters you create for the data selection:

Parameter Name	Description
`$$MATNR_FROM_0`	Defines the start value for the range of the data selection entry.
`$$MATNR_TO_0`	Defines the end value for the range of the data selection entry.
`$$BLANZ_FROM_0`	Defines a single value as the data selection entry.

After you create the mapping parameters, use the mapping parameters in a dynamic filter condition to represent the data selection you configured in the InfoPackage. You enter a dynamic filter for an SAP ECC source from the Dynamic Filter tab of the ABAP Program Flow dialog in the Application Source Qualifier properties. When you filter data from an SAP ECC source, the dynamic filter condition must conform to ABAP syntax.

For example, to represent the data selection entries, enter the following dynamic filter:

```
MARA-MATNR >= :$$MATNR_FROM_0 AND MARA-MATNR <= :$$MATNR_TO_0 AND

MARA-BLANZ = :$$BLANZ_FROM_0
```

When the SAP BI Scheduler sends the SAP BW Service a workflow request, the SAP BW Service receives the data selection information for the SAP ECC source and writes it to the temporary parameter file. For example, the SAP BW Service writes the following to the temporary parameter file for the data selection:

```
$$MATNR_FROM_0=MR0842
$$MATNR_TO_0=MT0727
$$BLANZ_FROM_0=219
```

During the workflow, the PowerCenter Integration Service uses the temporary parameter file to obtain values for the $$MATNR_FROM_0, $$MATNR_TO_0, and $$BLANZ_FROM_0 mapping parameters in the data selection entry. The PowerCenter Integration Service then uses the data selection entry to filter data from the source.

Configuring Mapping Parameters for Data Selection

You configure mapping parameters in the Designer by selecting Parameters and Variables from the Mapping menu item in the Mapping Designer.

Mapping Parameter for a Relational Source

To filter data from a relational source to load data into SAP BI, create a mapping parameter called $ $BWFILTERVAR. Configure $$BWFILTERVAR with the String datatype. Each data selection entry in the InfoPackage requires a maximum precision of 1500. Set the precision of $$BWFILTERVAR to 1500 to allow for at least 10 data selection entries.

The following table shows the options you need to use to create the $$BWFILTERVAR mapping parameter:

Option	Description
Name	Enter $$BWFILTERVAR.
Type	Select Parameter.

Option	Description
Datatype	Select String.
Precision	Enter 1500.

Mapping Parameter for a Flat File or SAP ECC Source

When you want to extract data from a flat file or SAP ECC source, create mapping parameters to represent the data selection you configured in SAP BI. Use the following naming convention for the mapping parameters:

```
$$<InfoObjectName>_{From|To}_<number>
```

The following table describes the components of a mapping parameter name for a flat file or an SAP ECC source:

Parameter Component	Description
InfoObject Name	Name of the field or InfoObject from which you want to filter data.
From | To	"From" defines the start of a range or a single value. Use "From" when you specify the FromValue field in the data selection entry. "To" defines the end of a range. Use "To" when you specify the ToValue field in the data selection entry.
number	Distinguishes mapping parameters with similar names created for the same InfoObject. Use 0 for a mapping parameter name created for the first time. Increment the number in the mapping parameter name by one for each subsequent mapping parameter for the same InfoObject. For example, if you have two data selection entries for the EmpID field that specify different values for FromValue, use 0 and 1 for the number parameter component.

For example, you configure an InfoPackage to filter data to extract records where the LocationID equals 24 or 19. Create two mapping parameters to represent the data selection entries in the InfoPackage:

- `$$LocationID_From_0`
- `$$LocationID_From_1`

When you want to filter data from a flat file or SAP ECC source to load data into SAP BI, select the datatype for the mapping parameter based on the datatype of the InfoObject. When you specify the precision for a mapping parameter, make sure that it is the same as the precision defined for the corresponding InfoObject.

The following table shows the datatype you need to use for a mapping parameter based on the datatype of the InfoObject:

InfoObject Datatype	Mapping Parameter Datatype
CHAR	String
NUMC, DEC	Number (with appropriate precision and scale)
DATS, TIMS	Date

CHAPTER 22

Loading Data into SAP BI

This chapter includes the following topics:

- Loading Data into SAP BI Overview, 250
- Step 1. Configure a Workflow to Load Data into SAP BI, 251
- Step 2. Configure an InfoPackage, 253
- Step 3. Configure a Data Transfer Process, 254
- Step 4. Configure a Process Chain to Load Data, 254
- Log Events, 257
- InfoPackage Status, 258
- Recovering a PowerCenter Workflow, 259
- Troubleshooting Data Loading into SAP BI, 259

Loading Data into SAP BI Overview

To load data into SAP BI, configure both the PowerCenter and the SAP BI systems.

To load data to versions earlier than 7.3 using InfoSources, complete the following steps:

1. Configure a workflow to load data into SAP BI.
2. Configure an InfoPackage that associates the PowerCenter session with the InfoSource.
3. Configure a process chain to load the data.

To load data to version 7.3 using 7.x DataSources, complete the following steps:

1. Configure a workflow to load data into SAP BI.
2. Configure an InfoPackage that associates the PowerCenter session with the 7.x DataSource. The InfoPackage loads the data to the PSA.
3. Create a DTP.
4. Configure a process chain linking the InfoPackage and DTP to transfer the data from the PSA to an SAP target.

When the SAP BW Service starts, it communicates with the SAP BI system to register as a server. The SAP BW Service waits for a request from the SAP BI system to start the workflow. When the InfoPackage starts, the SAP BI system communicates with the registered SAP BW Service. The SAP BI system sends the workflow name that the PowerCenter Integration Service schedules to run. The SAP BW Service reads information about the workflow and sends a request to the PowerCenter Integration Service to run the workflow. The PowerCenter Integration Service validates the workflow name in the repository and the

workflow name in the InfoPackage. The PowerCenter Integration Service runs the session and loads the data into SAP BI.

You can configure partitions for a session and sessions for workflow recovery. View log events to help you track the interactions between PowerCenter and SAP BI.

Step 1. Configure a Workflow to Load Data into SAP BI

Before you create a session, create an application connection to the SAP BI system. Schedule an InfoPackage in SAP BI, which starts a PowerCenter workflow. You use the Workflow Monitor to stop an SAP BI session. You cannot start or schedule an SAP BI session in the Workflow Monitor.

Partitions for SAP BI sessions

You can configure partitions for a session when you load data to 7.x sources.

When you enable partitioning for a session, you configure how the PowerCenter Integration Service writes data to the target. You can configure the number of target connections to use and the following partition types when you load data into SAP BI:

- Hash user-keys
- Key range
- Pass-through
- Round-robin

Creating an SAP BI Session

After you create an application connection, create a session. Because SAP BI only supports inserts from PowerCenter, configure sessions to load data to SAP BI targets as insert only.

SAP does not support BI version 7.3 in non-Unicode systems. When you load to an SAP BI version 7.3 system, you must configure the session to run in Unicode mode.

1. In the Task Developer, click **Tasks** > **Create**.
2. Select **Session** for the task type.
3. Enter a name for the task.
4. In the **Mappings** dialog box, select the mapping that you want to use in the session and click **Done**.
5. Double-click the SAP BI session to open the session properties.
6. Click the **Properties** tab.
7. In the general options settings, select **Insert** for the **Treat Source Rows As** property.
8. To configure sessions for workflow recovery, set the recovery strategy to resume from the last checkpoint.
9. Click the **Config Object** tab.
10. In the advanced settings, set the default buffer block size.

For optimum performance, set the default buffer block size to 5 MB to 10 MB. You can also create a reusable session configuration object with the default buffer block size set to 5 MB to 10 MB.

11. Click the **Mapping** tab.
12. Click the **Targets** node and select the connection defined for the SAP BI server.
13. Set the value of the packet size.

    This property determines the size of the packet that the PowerCenter Integration Service sends to SAP BI. The default is 10 MB. The value of the packet size must be equal to or less than the following values:

    - **Packet size configuration in SAP BI.** By default, SAP BI allows packets of 10 MB. The SAP BI administrator can change this packet size configuration.
    - **Available memory on the node where the PowerCenter Integration Service process runs.** When the PowerCenter Integration Service processes an SAP BI session, it stores data in memory until the size equals the value of the packet size. The PowerCenter Integration Service then loads the data to SAP BI as a packet.
14. Click **OK**.

Creating a PowerCenter Workflow for an SAP BI Session

After you configure an SAP BI session, create a PowerCenter workflow to execute the session. The following restrictions apply to workflows containing sessions that load SAP BI targets:

- **The workflow name in the repository must match the workflow name in the InfoPackage.** When you create the InfoPackage, you include the workflow name. The PowerCenter Integration Service validates the workflow name in the repository and the workflow name in the InfoPackage. These names must match exactly, including case.
- **Configure the workflow to run on demand.** You can configure the schedule when you create the InfoPackage in SAP BI. You cannot schedule a PowerCenter workflow to load data into SAP BI.

1. In the Workflow Designer, click **Workflows** > **Create**.
2. In the workflow properties, accept the default workflow name or rename the workflow.

   Make sure the workflow name in the workflow properties matches both the workflow name in the InfoPackage and the SAP BI session name.
3. Select the PowerCenter Integration Service name you configured in the SAP BW Service properties.

   The PowerCenter Integration Service runs the workflow and uses the session to load data into SAP BI.
4. To prepare the workflow for recovery, click **Suspend on Error**.
5. On the **Scheduler** tab of the workflow properties, click the right side of the **Scheduler** field to edit scheduling settings for the scheduler.

   The **Edit Scheduler** dialog box appears.
6. Click the **Schedule** tab.
7. Select **Run on Demand** for **Run Options**.
8. Click **OK** to exit the scheduler.
9. Click **OK** to exit the workflow properties.
10. Add the session you created to the workflow.

You can only include one session in the workflow. Ensure that the session name is identical to the workflow name.

Step 2. Configure an InfoPackage

An InfoPackage is the SAP BI mechanism for scheduling and running ETL jobs. The InfoPackage defines the target InfoSource or DataSource, and the source system. You can also enter a data selection entry in the InfoPackage to select data from the source system.

Creating and Scheduling an InfoPackage

Create and schedule an InfoPackage in SAP BI 7.0 and 7.3 with the Data Warehousing Workbench. Create and schedule an InfoPackage in SAP BW 3.5 or earlier with the Administrator Workbench.

1. In the Administrator Workbench or Data Warehousing Workbench, click the **InfoSources** or **DataSources** tab.
2. Locate the InfoSource or DataSource.
3. Perform one of the following tasks based on whether you use an InfoSource or a DataSource:
   - Under the InfoSource, right-click the source system.
   - Right-click the DataSource.
4. Select **Create InfoPackage** and enter a description for the InfoPackage. Click **Save**.

   The **Scheduler (Maintain InfoPackage)** window appears.
5. Click the **3rd Party Selections** tab.
6. Click **Refresh**.
7. Enter the domain, service, and session details.

   The following table describes the domain, service, and session details:

Property	Description
Domain Name for DI Service	Name of the PowerCenter domain for the PowerCenter Integration Service that runs the workflow.
Data Integration Service Name	Name of the PowerCenter Integration Service that runs the workflow.
Name of Folder Containing Workflow	Name of the PowerCenter folder containing the workflow.
Workflow Name	PowerCenter workflow name.
Session Name	PowerCenter session name. If you enter a session name, the PowerCenter Integration Service runs only this session in the workflow. If you do not enter a session name, the PowerCenter Integration Service runs the entire workflow. You must enter a session name if you are filtering data from a relational source before loading it into an SAP BI target.

8. In the **Processing** tab, select a data transfer option. For 7.x DataSources, the **Only PSA** data transfer option is selected by default because the InfoPackage loads the data only to the PSA.
9. In SAP BW 3.5 or earlier, click the **Scheduling Info** tab. In SAP BI 7.0 and 7.3, click the **Schedule** tab.
10. In SAP BI 7.0, click **Start to run the InfoPackage immediately**. In SAP BI 7.3, click **Start Data Load Immediately** . In SAP 7.0 and 7.3, you can also click **Jobs** to schedule it for a specific time.

Setting a Data Selection Entry to Filter Data

When you want to load filtered data into an SAP BI target, set data selection entries in the InfoPackage. Before defining data selection entries, specify the InfoObjects from which you want to filter data.

Under the **Transfer Structure/Transfer Rules** section of the InfoSource, click the **Data Source/Transfer Structure** tab and select the **Selection** check box for each InfoObject for which you want to filter data. Under the **Fields** tab of the DataSource, select the **Sel. Option** check box for each InfoObject for which you want to filter data.

1. In the Administrator Workbench or Data Warehousing Workbench, click the **InfoSources** or **DataSources** tab.
2. Open the InfoPackage for the InfoSource or DataSource in which you want to include a data selection entry.
3. Select the **Data Selection** tab.
4. Enter values in the **FromValue** and **ToValue** fields for the InfoObjects you want to filter.

Step 3. Configure a Data Transfer Process

When you use 7.x DataSources, the InfoPackage loads data to the PSA. To load the data from the PSA to an SAP target, you must configure a Data Transfer Process (DTP).

1. In the Administrator Workbench or Data Warehousing Workbench, click the **DataSources** tab.
2. Locate the 7.x DataSource that you used to load the data in the PSA.
3. Right-click the 7.x DataSource and select **Create Data Transfer Process**.

   The **Create Data Transfer Process** dialog box appears. The source of the DTP is set to the 7.x DataSource that you used for loading the data in the PSA.
4. Define the target of the DTP. Select the object type and enter the name of the SAP target to which you want to load the data from the PSA.
5. Save and activate the DTP. The SAP system generates a default transformation between the source and target of the DTP.

   Note: You can also right-click the SAP target to which you want to load the data and then create a DTP. The target of the DTP is set to the target that you selected for loading the data. You must then define the source of the DTP as the 7.x DataSource that you used to load the data in the PSA.

Step 4. Configure a Process Chain to Load Data

To load data into SAP BI, configure a process chain that links the following components:

- InfoPackage process
- Data Transfer Process
- Additional loading processes
- ZPMSENDSTATUS ABAP program

Note: You need to insert a Data Transfer Process in the process chain only when you load to SAP BI version 7.3 using 7.x DataSources.

The InfoPackage and loading processes process the data. The ABAP program sends status messages to the SAP BW Service. The SAP BW Service sends these messages to the PowerCenter Log Manager.

After inserting the ABAP program into the process chain, create a variant for the program. A variant is an SAP BI structure that contains parameter values that SAP BI passes during program execution.

When you load data to SAP BI 7.3 using a 7.x DataSource, the transfer option is configured to load data to the PSA because the InfoPackage loads data only to the PSA.

When you load data using 3.x data sources, use one of the following PSA transfer options:

- **PSA and then into Data Targets.** Select this option when you want to store data in the PSA before you load the data to data targets such as an ODS, InfoCube, or InfoSource.
- **PSA and Data Targets in Parallel.** Select this option when you want to load data to the PSA and to data targets at the same time to achieve high performance.
- **Only PSA.** Select this option when you want to load data only to the PSA. You can also select the **Update Subsequently in Data Targets** option to load the data to data targets after you load the data to the PSA.
- **Data Targets Only.** Select this option when you want to load data only to data targets such as ODS, InfoCube, or InfoSource.

For faster performance, configure the transfer option to load data to the PSA. After the source system loads the PSA, you can update the InfoCubes in SAP BW.

The process chain may contain a single InfoPackage that loads data to the PSA only or to a data target only. Insert the ZPMSENDSTATUS ABAP program after the InfoPackage to send the status to the SAP BW Service.

The process chain can also contain an InfoPackage that loads data to the PSA and additional processes that load the data to data targets. Insert the ZPMSENDSTATUS ABAP program after each loading process to ensure that the SAP BW Service receives status information at each point in the process chain.

To load data into SAP BI, complete the following steps to configure the process chain:

1. Create the process chain and insert the start process.
2. Insert an InfoPackage process.

   Note: When you insert an InfoPackage process for a 7.x DataSource, the DTP associated with the 7.x DataSource is also inserted.
3. Insert the ZPMSENDSTATUS ABAP program.

Creating the Process Chain and Inserting the Start Process

When you create the process chain and insert the start process, you also schedule the process chain.

1. From the Administrator Workbench or Data Warehousing Workbench, click **SAP Menu** > **Administration** > **RSPC - Process Chains**.

   The **Process Chain Maintenance Planning View** window appears.
2. Click **Create**.

   The **New Process Chain** dialog box appears.
3. Enter a unique name for the process chain and enter a description.
4. Click **Enter**.

   The **Insert Start Process** dialog box appears.
5. Click **Create**.

The **Start Process** dialog box appears.

6. Enter a unique name for the start process variant and enter a description.
7. Click **Enter**.

   The **Maintain Start Process** window appears.
8. Click **Change Selections** to schedule the process chain.

   The **Start Time** window appears.
9. To schedule the process chain to run immediately after you execute it, click **Immediate**.
10. Click **Save**.
11. In the **Maintain Start Process** window, click **Cancel**.
12. In the **Insert Start Process** dialog box, click **Enter**.

    The start process appears in the **Process Chain Maintenance Planning View** window.

Inserting an InfoPackage Process

Insert a process for the InfoPackage you created in SAP BI.

1. In the **Process Chain Maintenance Planning View** window, click **Process Types**.
2. From the **Process Types** menu, click **Load Process and Post-Processing** > **Execute InfoPackage**.

   The **Insert Execute InfoPackage** dialog box appears.
3. For the **Process Variants** field, click the **Browse** button to select the InfoPackage you created.
4. Click **Enter**.

   The InfoPackage process appears in the **Process Chain Maintenance Planning View** window.

   Note: When you insert an InfoPackage process for a 7.x DataSource, the DTP associated with the 7.x DataSource is also inserted.
5. Click the start process description and drag to link the start process to the InfoPackage process.

Inserting the ZPMSENDSTATUS ABAP Program

Before you can insert the ZPMSENDSTATUS ABAP program into a process chain, you must have imported the program into SAP BI.

1. In the **Process Chain Maintenance Planning View** window, click **Process Types**.
2. From the **Process Types** menu, click **General Services** > **ABAP Program**.

   The **Insert ABAP Program** dialog box appears.
3. Click **Create**.

   The **ABAP Program** dialog box appears.
4. Enter a unique name for the ABAP program process variant and enter a description.
5. Click **Enter**.

   The **Process Maintenance: ABAP Program** window appears.
6. In the **Program Name** field, click the **Browse** button to select the ZPMSENDSTATUS ABAP program.
7. Click **Change** next to the **Program Variant** field.

   The **ABAP: Variants - Initial Screen** window appears.
8. Click **Create**.

9. In the **ABAP: Variants** dialog box, enter a name for the ABAP variant and click **Create**.

   The **Maintain Variant** window appears.
10. For the **DEST** field, select the name of the RFC destination.
11. For the **INFPARAM** field, enter one of the following options:
    - PSA if the previous process loaded to the PSA.
    - Data Target if the previous process loaded to a data target.
12. For the **CONTEXT** field, enter BW LOAD.
13. For the **INFOPAK** field, enter the technical name of the InfoPackage.

    For example, ZPAK_439OS93K56GKQT7HQT5TFV1Z6.
14. Click **Save and Exit** in the **Maintain Variant** window.
15. Click **Save and Exit** in the **ABAP Variants** window.
16. Click **Save and Exit** in the **Process Maintenance: ABAP Program** window.
17. Click **Enter** in the **Insert ABAP Program** dialog box.

    The ABAP program appears in the **Process Chain Maintenance Planning View** window.
18. Click the InfoPackage process description and drag to link the InfoPackage process to the ABAP program.

    When prompted, click **Successful condition**.

    Note: When you use a 7.x DataSource to load to the PSA, you must link the InfoPackage process to the Data Transfer Process and then link the Data Transfer Process to the ABAP program.
19. Optionally, insert additional loading processes into the process chain.

    Use the instructions in "Inserting an InfoPackage Process" on page 256.
20. Insert the ZPMSENDSTATUS program after each loading process.
21. In the **Process Chain Maintenance Planning View** window, click **Checking View** and then click **Activate**.
22. Click **Execute** and assign the process chain to a specific SAP BI server.

    If you scheduled the process chain to run immediately, the process chain starts running on the assigned SAP BI server.
23. Optionally, to view the status of the process chain, click **Job Overview**.

    The **Simple Job Selection** window appears.
24. Enter selection criteria to specify which process chains you want to monitor and click **Execute**.

    The **Job Overview** window appears.
25. Select the BI_PROCESS_ABAP job and click **Job Log**.

    The **Job Log Entries** window appears. It includes an entry about the status of the PowerCenter workflow that the process chain was configured to start.

Log Events

The SAP BW Service captures log events that track interactions between PowerCenter and SAP BI. It also captures log events when it receives the following information from the SAP BI system and the PowerCenter Integration Service:

- A request from the SAP BI system to start a PowerCenter workflow.
- A message from the PowerCenter Integration Service that it has successfully started a workflow to load data into SAP BI.
- A message from the PowerCenter Integration Service indicating whether the PowerCenter session failed or succeeded.
- Status information from the ZPMSENDSTATUS ABAP program in the SAP BI process chain that loads data to SAP BI.

When you load data into SAP BI, you can view SAP BW Service log events in the following locations:

- **Administrator tool.** On the Log tab, enter search criteria to find SAP BW Service log events.
- **SAP BI Monitor.** In the Monitor - Administrator Workbench window, you can view log events that the SAP BW Service captures for an InfoPackage included in a process chain that loads data into SAP BI.

To view log events about how the PowerCenter Integration Service processes an SAP BI workflow, view the session or workflow log.

Set the environment variable PMTOOL_DATEFORMAT to customize the date format returned by the SAP BW Service for log events. When you set the environment variable, the SAP BW Service validates the string before it writes a date to the log. If the date is not valid, the SAP BW Service uses the default date display “DY MON DD HH24:MI:SS YYYY.”

Viewing SAP BW Service Log Events in the SAP BI Monitor

You can use the SAP BI Monitor to view log events that the SAP BW Service captures for an InfoPackage included in a process chain that loads data into SAP BI. SAP BI pulls the messages from the SAP BW Service and displays them in the monitor. The SAP BW Service must be running to view the messages in the SAP BI Monitor.

To view SAP BW Service log events in the SAP BI Monitor:

1. From the Administrator Workbench or Data Warehousing Workbench, click Monitor.

   The Monitor - Administrator Workbench window appears.
2. Select an InfoPackage.
3. Click Goto > Logs > Non-SAP System Extraction Log.

   The Third Party System Log dialog box appears, displaying log events that the SAP BW Service captured for the InfoPackage.

InfoPackage Status

The SAP BI Monitor displays the InfoPackage status. The status in the SAP BI monitor is green, yellow, or red depending on the status of the PowerCenter session. Green indicates that the data loaded successfully. Yellow indicates that the data loaded with warnings. Red indicates that the failed to load.

The SAP BI Monitor indicates status in the following cases:

- The Integration Service did not load any rows to SAP BI, because the source system contained no data. However, the PowerCenter session completed successfully. The SAP BI Monitor indicates a status of green, yellow, or red, depending on the "Traffic Light Color for Empty Requests" settings in the InfoPackage. Default is yellow.
- The Integration Service failed and did not load any rows to SAP BI. The SAP BI Monitor indicates a status of red.
- The Integration Service failed after loading some data into SAP BI. The SAP BI Monitor indicates a status of red.
- If a session enabled for recovery fails, the SAP BI Monitory indicates a status of yellow. The SAP BI Monitor status changes to green after data loads successfully during the recovery run.

Recovering a PowerCenter Workflow

If an SAP BI session enabled for recovery fails, you can use the Workflow Manager or Workflow Monitor to recover the PowerCenter workflow. When you recover a workflow, the PowerCenter Integration Service can resume the session that failed. You enable an SAP BI session for recovery when you configure the session properties.

You can use the Workflow Manager or Workflow Monitor to start PowerCenter workflows that load data into SAP BI in recovery mode only. The SAP BI system starts all normal runs of the PowerCenter workflow.

Troubleshooting Data Loading into SAP BI

The Workflow Manager reports that the session completes successfully, but the SAP BI system reports that the session failed.

This might occur when the PowerCenter Integration Service successfully moves data into the InfoSource, but SAP BI fails to move the data from the InfoSource to the InfoCube. This problem is not related to PowerCenter or PowerExchange for SAP BI. The problem is related to the SAP BI server. See the SAP BI documentation.

I cannot start an InfoPackage.

The most common reasons for a failed connection are:

- **The `sapnwrfc.ini` file is improperly configured.** Verify that the PROGRAM_ID defined for the connection to the RFC server program matches the Program ID for the external source in SAP BI.
- **The SAP BW Service is not running.**
- **The SAP BW Service is running, but the PowerCenter Integration Service is not running.** The InfoPackage is launched and sends a request to the SAP BW Service. The SAP BW Service, in turn, sends a request to the PowerCenter Integration Service to start the session. If the PowerCenter Integration Service is not running, you might see the following message:

```
There was a problem connecting to the Integration Service [Error Number <error>].
Retrying...
```

The error number embedded in the message originates from the operating system.

The InfoPackage aborts if the SAP BW Service does not connect to the PowerCenter Integration Service immediately.

If you have problems starting an InfoPackage, test the connection using the Administrator Workbench.

To test the SAP BW Service connection:

- From the Administrator Workbench or Data Warehousing Workbench, click the Source Systems tab.
- Right-click the source system and select Change.
- Click the Test Connection button.
- The RFC Connection Test returns a status screen that indicates the status and description of the connection.

When I try to start a workflow containing an SAP BI session in the Workflow Manager, nothing happens.

You cannot use the Workflow Manager to start or schedule a PowerCenter workflow with an SAP BI session. You need to configure the workflow to run on demand. Create an InfoPackage in the SAP BI system to schedule the workflow containing an SAP BI session.

I need to stop a workflow containing an SAP BI session.

To stop an SAP BI workflow, stop the PowerCenter workflow using *pmcmd* or in the Workflow Monitor. You cannot stop an InfoPackage in SAP BI.

The InfoPackage starts in SAP BI, but no message appears in the log for the PowerCenter session.

This may occur if you have more than one SAP BW Service in the same environment using the same Program ID. When you have multiple SAP BW Services in the same environment using the same Program ID, the SAP BW Service which was started first receives the requests from the SAP BI system.

When you do not see a message in the Administrator tool or SAP BI Monitor log for an Infopackage that starts in SAP BI, verify if there are other SAP BW Services connected to the SAP BI system. Check the log for the other SAP BW Services to see if the InfoPackage started.

I ran a session to load data into SAP BI. However, the session status reported by SAP BI is not the same as the session status reported by the PowerCenter Integration Service.

Status messages coming from SAP BI are not transported correctly to the PowerCenter Integration Service when the load is successful but has zero rows.

In SAP BI, you can set the Traffic Light Color options to indicate success if there is no data. SAP BI sends a status message of success to the PowerCenter Integration Service when a load is successful but has zero rows.

The SAP BW Service starts my filter session run, but the log includes an error message “Error in opening parameter file.”

This only occurs on Windows. The permissions on the directory that contains the parameter file are not set correctly. Enable the SAP BW Service appropriate read and write permissions for the parameter file directory.

I ran a session that successfully loaded data into SAP BI, but the Administrator tool log includes irrelevant messages about the session.

This may occur if you entered an invalid value for the CONTEXT field in the ZPMSENDSTATUS ABAP program when creating the process chain. You must enter BW LOAD for the CONTEXT field in a process chain that loads data.

I ran a session to load data into SAP BI. However, the session failed with the following error in the session log:

```
WRITER_1_*_1>WRT_8025 Error in BW Target. [===>SAPSendDone Failed.
SAP system exception raised.
key = RFC_ERROR_SYSTEM_FAILURE
message = &INCLUDE INCL_INSTALLATION_ERROR
```

This may occur if the Packet Size session property is higher than the packet size configured in the SAP BI system or higher than the available memory on the node where the PowerCenter Integration Service process runs. Decrease the value of the Packet Size property and run the session again.

APPENDIX A

Data Type Reference

This appendix includes the following topics:

- SAP Data Types, 262
- PowerExchange for SAP NetWeaver and SAP Data Types, 263
- PowerExchange for SAP BW and SAP Data Types, 266

SAP Data Types

The following table lists the data types that are available in the SAP NetWeaver and SAP BW systems:

SAP Data Type	Type	Range and Description
ACCP	Date	Posting period of 6 positions, the format is YYYYMM. In input and output, a point is inserted between year and month, so the template of this data type has the form '____.__'.
CHAR	Text	Character string with maximum length of 255. If longer fields are required, use LCHR.
CLNT	Text	Client fields. Always has 3 positions.
CUKY	Text	Currency key of 5 positions containing the possible currencies referenced by CURR fields.
CURR	Numeric	Currency field with a maximum length of 31 positions. Equivalent to a DEC amount field. A CURR field must reference a CUKY field. For P type, only 14 digits are allowed after the decimal point.
DATS	Date	8-position date field. The format is YYYYMMDD.
DEC	Numeric	Maximum length of 31 positions. Counter or amount field with a decimal point, sign, and commas separating thousands. For P type, only 14 digits are allowed after the decimal point.
FLTP	Numeric	Floating point number of 16 positions including decimal places.
INT1	Numeric	1-byte integer between 0 and 255. 3 positions. Not supported for PowerExchange for SAP NetWeaver BW

SAP Data Type	Type	Range and Description
INT2	Numeric	2-byte integer between -32,767 to 32,767, only used for length fields; positioned immediately in front of LCHR and LRAW. With INSERT or UPDATE on the long field, the database interface enters the length used in the length field and the length is set at 5 positions.
INT4	Numeric	4-byte integer between -2,147,483,647 and 2,147,483,647. The length is set at 10 positions.
LANG	Text	Language key, field format for special functions of 1 position.
LCHR	Text	Long character string with a minimum length of 256 characters. Must be at the end of transparent table and must be preceded by a length field INT2.
LRAW	Binary	Limited support. Long byte string with a minimum of 256 positions. Must be at the end of transparent table and must be preceded by a length field of type INT2.
NUMC	Text	Long character field of arbitrary length with a maximum length of 255 positions. You can enter only numbers.
PREC	Binary	Precision of a QUAN field of 2 positions. Not supported in PowerExchange for SAP NetWeaver.
QUAN	Text	Quantity field with a maximum length of 31 positions. Points to a unit field with the UNIT format. For P type, only 14 digits are allowed after the decimal point.
RAW	Binary	Uninterrupted sequence of bytes with a maximum length of 255 positions. If longer fields are required, use LRAW.
SSTR	String	Short character string with a maximum variable length of 1333 characters. Supported when you read data from SAP tables through ABAP.
TIMS	Date	Time field (HHMMSS) of 6 positions, the display format is HH.MM.SS.
UNIT	Text	Units key of 2 or 3 positions, field containing the allowed quantity units referenced by QUAN fields.
VARC	Text	Variable length character string, requires an INT2 length field. Not supported in SAP 3.0 and later.

PowerExchange for SAP NetWeaver and SAP Data Types

PowerExchange for SAP NetWeaver uses the following data types in mappings:

- Native data types. Native data types are data types specific to the source and target databases or flat files. They appear in non-SAP ECC source definitions and target definitions in the mapping.

- SAP data types. SAP data types appear in the SAP definitions in the mapping. SAP performs any necessary conversion between the SAP data types and the native data types of the underlying source database tables.
- Transformation data types. Transformation data types are generic data types that PowerCenter Integration Service uses during the transformation process. They appear in all the transformations in the mapping.

When you connect an SAP source definition to a source qualifier, the Designer creates a port with a transformation data type that is compatible with the SAP data type. The PowerCenter Integration Service converts SAP data types to transformation data types. The PowerCenter Integration Service passes all transformation data types to the target, and the target converts them to the native data types.

The following table compares SAP data types and the transformation data types:

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
ACCP	Date/time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D.
CHAR	String	1 to 104,857,600 characters Fixed-length or varying-length string.
CLNT	String	1 to 104,857,600 characters Fixed-length or varying-length string.
CUKY	String	1 to 104,857,600 characters Fixed-length or varying-length string.
CURR	Decimal	Precision 1 to 28 digits, scale 0 to 28
DATS	Date/time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
DEC	Decimal	Precision 1 to 28 digits, scale 0 to 28
FLTP	Double	Precision 15, scale 0
INT1	Small Integer	Precision 5, scale 0
INT2	Small Integer	Precision 5, scale 0
INT4	Integer	Precision 10, scale 0
LANG	String	1 to 104,857,600 characters Fixed-length or varying-length string.
LCHR	String	1 to 104,857,600 characters Fixed-length or varying-length string.
LRAW	Binary	Limited support in PowerExchange for SAP NetWeaver
NUMC	Decimal or Double	Precision 1 to 28 digits, scale 0 to 28
PREC	Binary	Not supported in PowerExchange for SAP NetWeaver
QUAN	Decimal	Precision 1 to 28 digits, scale 0 to 28

SAP Data Type	Transformation Data Type	Range for Transformation Data Type
RAW	Binary	Uninterrupted sequence of bytes with a maximum length of 255 positions.
SSTR	String	1 to 104,857,600 characters Fixed-length or varying-length string.
TIMS	Date/time	Jan 1, 0001 A.D. to Dec 31, 9999 A.D. Precision to the nanosecond.
UNIT	String	1 to 104,857,600 characters Fixed-length or varying-length string.
VARC	String	1 to 104,857,600 characters Fixed-length or varying-length string.

Overriding Data Types in the Application Source Qualifier

Although the PowerCenter Integration Service converts most SAP data types successfully, you might need to override the Application Source Qualifier properties for the following data types:

- Date and number data types
- Binary data types
- CHAR, CUKY, and UNIT data types

When the ABAP program extracts from SAP, it stores all data, including dates and numbers, in character buffers. You might want to override the NUMC, ACCP, and DATS data types in the Application Source Qualifier to ensure full precision of dates and numbers.

NUMC

NUMC is a numeric string that supports more positions than any of the PowerCenter numeric data types. It holds only unsigned numeric strings with a maximum length of 255. The Application Source Qualifier converts NUMC to Decimal. You can also configure a Source Qualifier to convert this data type to Double.

By default, the PowerCenter Integration Service treats all Decimal ports as Double and maintains precision up to 15 digits. If NUMC has up to 28 digits, you can enable high precision in the session properties to maintain precision. If NUMC has more than 28 digits, the PowerCenter Integration Service converts NUMC to double even when you enable high precision.

Therefore, if you extract more than 28 digits, and you want to maintain full precision, you can change the NUMC data type to String in the Application Source Qualifier. However, you cannot perform numeric calculations on strings.

Because SAP does not store signs with NUMC data, do not use negative filter conditions in the Application Source Qualifier for NUMC columns. SAP does not recognize the negative conditions and treats all comparisons for NUMC columns as positive.

ACCP and DATS

ACCP and DATS are date data types that support zero values. PowerCenter does not support zero values in the Date/Time transformation data type. ACCP and DATS are date data types in SAP that PowerCenter converts to the Date/Time transformation data type. Data in these fields is stored internally as a character

string and may not correspond to a valid date. For example, a column might store a string of zeros. If SAP passes zeros to a Date/Time column in the Application Source Qualifier, the PowerCenter Integration Service converts the zeros to NULL and continues processing the records. However, the PowerCenter Integration Service rejects all other rows with invalid dates and writes an error to the session log.

If you want the PowerCenter Integration Service to process these rows, change the data type in the Application Source Qualifier to String and pass the row to an Expression transformation. You can write an expression using the IS_DATE function to test strings for valid dates and the TO_DATE function to convert valid strings to dates. You can also use the TO_DATE function to convert the invalid strings to an arbitrary date, such as the current date, so the PowerCenter Integration Service does not skip the row.

Binary Data Types

PowerCenter provides limited support for the binary data types RAW and LRAW. RAW holds binary data up to 255 bytes. LRAW holds binary data with a minimum of 256 bytes. PowerCenter can move binary data to a relational target, but it cannot transform it. PowerCenter cannot move binary data to flat file targets.

To move binary data, connect the RAW or LRAW column to a compatible binary column in the target definition. You can pass the binary data through other transformations, but you cannot perform mapping logic on the binary data.

For example, connect a RAW column from the SAP ECC source to the Application Source Qualifier. The Application Source Qualifier uses the Binary transformation data type. You can then pass the binary column to binary columns in other transformations and finally to a RAW column in Oracle. The SAP RAW data type in SAP is compatible with the Oracle RAW data type. If you apply mapping logic to the binary data type, the session fails.

PowerCenter does not provide support for the binary data type PREC. You can connect PREC columns to an Application Source Qualifier, but if you connect them to other transformations or to a target definition, the ABAP code generation fails.

CHAR, CUKY, and UNIT Data Types

PowerCenter treats the SAP data type CHAR as VARCHAR. PowerCenter trims trailing blanks for CHAR, CUKY, and UNIT data so you can compare SAP data with other source data.

SAP stores all CHAR data with trailing blanks. The TreatCHARasCHARonRead PowerCenter Integration Service property determines whether the PowerCenter Integration Service keeps the trailing blanks. When you set the property to No, the PowerCenter Integration Service treats SAP CHAR data as VARCHAR data and trims the trailing blanks. The PowerCenter Integration Service also trims trailing blanks for CUKY and UNIT data. You can compare SAP data with other source data without having to use the RTRIM function.

If you have mappings that include blanks when comparing an SAP column with other data, you might not want the PowerCenter Integration Service to trim the trailing blanks. To configure the PowerCenter Integration Service to keep the trailing blanks in CHAR data, set the TreatCHARasCHARonRead PowerCenter Integration Service property to Yes.

PowerExchange for SAP BW and SAP Data Types

The PowerCenter Integration Service moves data from source to target based on PowerCenter transformation data types. The PowerCenter Integration Service loads data into SAP BW based on the SAP BW target data types.

The following table lists the SAP data types supported by PowerExchange for SAP BW:

SAP BW	Binary	Date/Time	Decimal	Double, Real	Integer, Small Integer	String, Nstring, Text, Ntext
ACCP	No	Yes	No	No	No	Yes
CHAR	No	Yes	Yes	Yes	Yes	Yes
CLNT	No	Yes	Yes	Yes	Yes	Yes
CUKY	No	Yes	Yes	Yes	Yes	Yes
CURR	No	No	Yes	Yes	Yes	Yes
DATS	No	Yes	No	No	No	Yes
DEC	No	No	Yes	Yes	Yes	Yes
FLTP	No	No	Yes	Yes	Yes	Yes
INT2	No	No	Yes	Yes	Yes	Yes
INT4	No	No	Yes	Yes	Yes	Yes
LANG	No	Yes	Yes	Yes	Yes	Yes
LCHR	No	Yes	Yes	Yes	Yes	Yes
NUMC	No	No	Yes	Yes	Yes	Yes
QUAN	No	No	Yes	Yes	Yes	Yes
TIMS	No	Yes	No	No	No	Yes
UNIT	No	Yes	Yes	Yes	Yes	Yes
VARC	No	Yes	Yes	Yes	Yes	Yes

The PowerCenter Integration Service transforms data based on the PowerCenter transformation data types. The PowerCenter Integration Service converts all data to a CHAR data type and puts it into packets of 250 bytes, plus one byte for a continuation flag. SAP BW receives data until it reads the continuation flag set to zero. Within the transfer structure, SAP BW then converts the data to the SAP data type.

SAP only supports the following data types in transfer structures assigned to BAPI source systems, such as PowerCenter:

- CHAR
- CUKY
- CURR
- DATS
- NUMC
- TIMS
- UNIT

All other data types result in the following error in SAP BW:

```
Invalid data type <data type name> for source system of type BAPI.
```

Date/Time Data Types

The transformation Date/Time data type supports dates with precision to the nanosecond.

Binary Data Types

SAP BW does not allow you to build a transfer structure with binary data types. Therefore, you cannot load binary data from PowerCenter into SAP BW.

Numeric Data Types

PowerCenter does not support the INT1 data type.

For numeric data types such as CURR, DEC, FLTP, INT2, INT4, and QUAN, the PowerCenter Integration Service uses the precision of the SAP data type to determine the length of the data loaded into SAP BW. For example, if you try to load a value of -1000000000 into an SAP BW field with the INT4 data type, the PowerCenter Integration Service skips the row. This is because the INT4 data type supports data up to 10 bytes in length, but the -1000000000 value uses 11 bytes.

The PowerCenter Integration Service does not truncate extraneous bytes when loading data that exceeds the length allowed by the field data type. If a row contains data that exceeds the length allowed by the field data type in SAP BW, the PowerCenter Integration Service skips the row and writes the skipped row and a corresponding error message to the session log.

Writing to SAP BW Date Columns

The PowerCenter Integration Service converts strings stored in the PowerCenter default date format of MM/DD/YYYY HH24:MI:SS.US to date values before loading data into SAP BW. If strings are not in the default date format, use TO_DATE to convert them to dates.

DATS

You can pass any string, text, or date/time value to a DATS column. The PowerCenter Integration Service converts data to the YYYYMMDD format.

The following table describes how the PowerCenter Integration Service converts the data if you pass strings to a DATS column:

Source Data	Converts to
‘12/30/1998 5:15:59.123 AM’	19981230
‘02/01/1996’	19960201
‘05/05/1998 02:14:08’	19980505
‘Jul 18 99’	Error
‘09/10/49’	Error

Source Data	Converts to
'01-21-51'	Error
'10023'	Error
'Jan151999'	Error

The following table describes how the PowerCenter Integration Service converts dates if you pass strings to a DATS column:

Source Data	Converts to
12/08/98	19981208
04/12/52	20520412
03/17/49	19490317
11/22/1998	19981122
May 2 1998 5:15AM	19980502
1998/21/06 12:13:08	19980621

TIMS

You can pass any string, text, or date/time value to a TIMS column. The PowerCenter Integration Service converts the time portion of the string or date to the HHMMSS format.

The following table describes how the PowerCenter Integration Service converts the data if you pass a string to a TIMS column:

Source Data	Converts to
'10/31/98 03:15:08PM'	Error
'09/23/1998'	000000
'08/15/1998 09:55:06'	095506
'02/01/1998 14:22:44'	142244

The following table describes how the PowerCenter Integration Service converts the data if you pass date to a TIMS column:

Source Data	Converts to
12/08/98	000000
04/12/52 3:00:56PM	150056

Source Data	Converts to
11/22/1998	19981122
05/01/1998 12:24:18	122418

APPENDIX B

Code Pages and Unicode Support

This appendix includes the following topics:

- Language Code Selection, 271
- Code Page Selection, 272
- Supported Code Pages, 273
- Processing Unicode Data, 274

Language Code Selection

SAP supports many languages, but the system may be configured to support only a subset of the languages. When you configure the application connection to connect to the mySAP or SAP BW system, you may have to specify the language code of the system.

The following table shows which application connections require a language code:

Application Connection	Language Code Required
SAP_ALE_IDoc_Reader	No
SAP_ALE_IDoc_Writer	Yes
SAP BAPI/RFC Interface	Yes
SAP BW	Yes
SAP BW OHS	Yes
SAP R/3	Yes
FTP, SFTP	No

The language code you select affects the following PowerExchange for SAP NetWeaver tasks:

- **Importing SAP metadata in the Designer.** The Designer imports metadata in the language you specify. The mySAP system returns messages to the Designer in the language you specify.
- **Installing ABAP program.s** The mySAP system returns messages to the Designer in the language you specify.

- **Running sessions.** The ABAP program extracts data in the language you specify in the application connection. The mySAP system also returns session log and server messages in the language you specify. When you configure an application connection, you also select a code page.

The language code you select affects the following PowerExchange for SAP BW tasks:

- **Importing InfoSource definitions.** The SAP BW system returns messages to the Designer in the language you specify.
- **Running sessions.** The SAP BW system returns session log and server messages in the language you specify. When you configure a database connection, you also select a code page.

SAP substitutes the default language of the mySAP or SAP BW system under the following conditions:

- You leave the language code blank.
- You specify a valid language code, but the system does not support that language.

When you run a session with an SAP R/3 application connection and you specify a code page other than UTF-8, SAP substitutes the default language of that particular system under these conditions.

Code Page Selection

You must select a code page for each application connection. Use the following rules and guidelines when you select a code page:

- **The code page of the application connection must be compatible with the type of data in SAP.** For example, if you extract Unicode data from SAP, set the application connection code page to UTF-8.
- **The code page of the application connection must be a subset of the code page for the corresponding PowerCenter Integration Service process.** To prevent data inconsistencies, ensure that the application connection code page is a subset of the code page for the corresponding PowerCenter Integration Service.
- **If you configure the PowerCenter Integration Service for code page validation, the SAP R/3 application connection must use a code page that is a subset of the PowerCenter Integration Service code page.** If you configure the PowerCenter Integration Service for relaxed code page validation, you can select any code page supported by PowerCenter for the source database connection. When you use relaxed code page validation, select compatible code pages for the source and target data to prevent data inconsistencies. For Unicode, select UTF-8.
- **The data movement mode for the PowerCenter Integration Service must be compatible with the application connection code page.** For example, if the code page is UTF-8, set the data movement mode for the PowerCenter Integration Service to Unicode.
- **If you want to use ABAP to read data from SAP, do not select a Unicode code page to read data from a non-Unicode system.** The PowerCenter Integration Service might not correctly read the data.
- **PowerCenter does not validate the code page and data movement mode for the SAP_ALE_IDoc_Reader application connection.** To prevent data inconsistencies, ensure that the code page for this application connection is compatible with the data in SAP and ensure that the PowerCenter Integration Service is running in the right data movement mode.

Supported Code Pages

The code pages SAP supports differs depending on whether the SAP system runs on an operating system that uses a Unicode SAP RFC library.

The following table lists the name, description, and ID for supported code pages for each language when the SAP system runs on an operating system that uses a Unicode SAP RFC library:

Name	Description	ID
ISO-8859-2	ISO 8859-2 Eastern European	5
ISO-8859-3	ISO 8859-3 Southeast European	6
ISO-8859-4	ISO 8859-4 Baltic	7
ISO-8859-5	ISO 8859-5 Cyrillic	8
ISO-8859-6	ISO 8859-6 Arabic	9
ISO-8859-7	ISO 8859-7 Greek	10
ISO-8859-8	ISO 8859-8 Hebrew	11
ISO-8859-9	ISO 8859-9 Latin 5 (Turkish)	12
Latin1	ISO 8859-1 Western European	4
MS874	MS-DOS Thai, superset of TIS 620	874
MS932	MS Windows Japanese, Shift-JIS	2024
MS936	MS Windows Simplified Chinese, superset of GB 2312-80, EUC encoding	936
MS949	MS Windows Korean, superset of KS C 5601-1992	949
MS1250	MS Windows Latin 2 (Central Europe)	2250
MS1251	MS Windows Cyrillic (Slavic)	2251
MS1252	MS Windows Latin 1 (ANSI), superset of Latin1	2252
MS1253	MS Windows Greek	2253
MS1254	MS Windows Latin 5 (Turkish), superset of ISO 8859-9	2254
MS1255	MS Windows Hebrew	2255
MS1256	MS Windows Arabic	2256
MS1257	MS Windows Baltic Rim	2257
MS1258	MS Windows Vietnamese	2258
UTF-8	UTF-8 encoding of Unicode	106

Name	Description	ID
UTF-16_OppositeEndian	UTF-16 encoding of Unicode (Opposite Platform Endian)	10004
UTF-16_PlatformEndian	UTF-16 encoding of Unicode (Platform Endian)	10003
UTF-16BE	UTF-16 encoding of Unicode (Big Endian)	1200
UTF-16LE	UTF-16 encoding of Unicode (Lower Endian)	1201

The following table lists the name, description, and ID for supported code pages when the SAP system runs on an operating system that uses a non-Unicode SAP RFC library:

Name	Description	ID
ISO-8859-2	ISO 8859-2 Eastern European	5
ISO-8859-5	ISO 8859-5 Cyrillic	8
ISO-8859-7	ISO 8859-7 Greek	10
ISO-8859-8	ISO 8859-8 Hebrew	11
ISO-8859-9	ISO 8859-9 Latin 5 (Turkish)	12
Latin1	ISO 8859-1 Western European	4
MS874	MS-DOS Thai, superset of TIS 620	874
MS932	MS Windows Japanese, Shift-JIS	2024
MS936	MS Windows Simplified Chinese, superset of GB 2312-80, EUC encoding	936
MS949	MS Windows Korean, superset of KS C 5601-1992	949
MS950	MS Windows Traditional Chinese, superset of Big 5	950

Processing Unicode Data

A Unicode SAP system encodes data it reads or writes through RFC in UTF-16. The PowerCenter Integration Service reads UTF-16 data from a Unicode SAP system and writes UTF-16 data to a Unicode SAP system if you select UTF-8 as the application connection code page.

The PowerCenter Integration Service processes SAP Unicode data in one session or multiple sessions, depending on the operating system of the PowerCenter Integration Service. If SAP provides Unicode libraries for the operating system of the PowerCenter Integration Service, you can process Unicode data in a single session. If SAP does not provide Unicode libraries for the operating system of the PowerCenter Integration Service, you need to process Unicode data with multiple sessions.

The SAP BW Service must run on an operating system that uses a Unicode SAP RFC library.

Processing Unicode Data with a Single Session

If the PowerCenter Integration Service runs on an operating system for which SAP provides Unicode libraries, use Unicode for the data movement mode to increase session performance. SAP provides Unicode libraries for the following operating systems:

- AIX (64-bit)
- Linux
- Solaris (64-bit)
- Windows

If the logical system you created for PowerCenter uses a connection for the RFC server program, configure the logical system to communicate in Unicode mode.

Processing Unicode Data with Multiple Sessions

SAP does not provide Unicode libraries for the following operating systems:

- AIX (32-bit)
- Solaris (32-bit)

To process Unicode data with non-Unicode libraries, route the data into a separate session for each required code page and use a different code page for each session.

Processing Unicode Data in ABAP

If the SAP system is a Unicode system, you can specify characters across several code pages in a single filter in the ABAP program flow dialog box.

Processing Unicode Data with Different Code Pages

If you need to process data with different code pages, route the data into a separate session for each required code page. You can create multiple application connections and assign a different code page to each one. Assign each application connection to the session based on the code page requirement.

For example, you want to run a session to extract SAP Unicode data on AIX 32-bit. The SAP data contains German and Japanese data. Because SAP does not provide Unicode libraries for AIX 32-bit connection, you need to process the German data and Japanese data separately. Create different sessions for the Japanese and German data. Assign a Japanese code page to the application connection for the Japanese data and a German code page to the application connection for the German data.

The following figure shows how to use different code pages to process the German and Japanese data:

APPENDIX C

Glossary

ABAP

Proprietary language used by SAP. The Designer generates and installs ABAP code on the application server to extract SAP data.

ABAP join syntax

Join syntax available on SAP 4.x systems. You can generate the ABAP program using ABAP join syntax if the mapping contains only transparent tables and you connect to an SAP 4.x system.

ABAP program variable

A variable in the ABAP code block or static filter condition. ABAP program variables can represent SAP structures, fields in SAP structures, or values in the ABAP program.

ABAP type variable

An ABAP program variable that represents a value in the ABAP program.

ALE (Application Link Enabling)

An SAP technology to exchange business data between interconnected programs across various platforms and systems.

application server

Part of the three-tiered architecture of the SAP system. PowerCenter makes all requests through the application server.

background process

A work process on the application server that the PowerCenter Integration Service uses to run file mode sessions in the background, or batch mode.

BAPI

Business Application Programming Interface. An SAP programming interface to access SAP from external applications that support the RFC protocol.

branch

The structure in the hierarchy that connects the nodes, extending from the root node to the leaf nodes.

buffers

Shared memory area on the application server that holds query results.

business content

SAP business content is a collection of metadata objects that can be easily integrated with other applications and used for analysis and reporting.

cluster table

A table on the application server that does not have a one-to-one relationship with the related table on the database server.

code block

Additional ABAP code you can add to the ABAP program. You can create code blocks in the ABAP Program Flow dialog box in the Application Source Qualifier.

CPI-C

Common Program Interface-Communications. The communication protocol that enables online data exchange and data conversion between the SAP system and PowerCenter. CPI-C is used for stream mode sessions.

database server

Part of the three-tiered architecture of the SAP system. This server contains the underlying database for SAP.

database views

Views on the application server that are based on views of transparent tables on the database server. You can extract from a database view in the same way you extract from a transparent table.

Data Migration Interface

See *SAP DMI*on page 280.

DataSource

A set of logically-related fields that are used to extract data from a source system and transfer the data to the SAP BW system.

Data Transfer Process

An SAP process used to transfer data from source objects to target objects in SAP BW.

design-time transports

Transports you install and use in the development environment.

detail table

The SAP table that joins with the hierarchy. The detail table provides the data for the detail range associated with the leaf nodes in the hierarchy.

dialog process

A work process on the application server that runs file mode sessions in the foreground.

dynamic filter

A filter in the Application Source Qualifier to reduce the number of rows the ABAP program returns. The Designer stores the dynamic filter information in the repository. The dynamic filter condition is not a part of the ABAP program.

exec SQL

Standard SQL that accesses the physical database. Use exec SQL to access transparent tables and database views.

file mode

Use the file mode extraction method to extract SAP data to a staging file. File mode sessions use RFC.

FROM_VALUE

The beginning range of values for the leaf node of the hierarchy. The PowerCenter Integration Service uses either this column or the TO_VALUE column to join with the detail table.

functions

Generic modules in the SAP system. You insert SAP functions in the ABAP program to extract source data.

gateway process

A work process on the application server that implements the CPI-C protocol for stream mode sessions.

hierarchy

Tree-like structure of metadata that defines classes of information.

IDoc

An Intermediate Document (IDoc) is a hierarchical structure that contains segments. Each IDoc segment contains a header and data rows. You import IDoc source definitions in the Source Analyzer.

IDoc transfer method

A transfer method within the SAP BW system used to synchronously move data from transfer structures to InfoCubes.

inbound IDoc

An IDoc a third party application, such as the PowerCenter Server, sends to SAP.

InfoCube

A self-contained dataset in the SAP Business Information Warehouse created with data from one or more InfoSources.

InfoPackage

The SAP BW mechanism for scheduling and running ETL jobs. You create an InfoPackage in the SAP BW system to specify the scheduling of SAP BW workflows and data requests from PowerCenter.

InfoSource

A collection of data in the SAP Business Information Warehouse that logically belongs together, summarized into a single unit.

leaf nodes

The lowest nodes in the structure of the hierarchy. These nodes are keyed to the detail table containing data.

logical unit of work (LUW)

Contains a set of functions that perform a given task in SAP. When you use an RFC/BAPI mapping to process data in SAP, use TransactionID values to define the commit points for an LUW. LUW you define must make a function call for each function you use in the mapping.

nested loop

The syntax that Open SQL uses to extract data. You can generate the ABAP program using nested loop by selecting the Force Nested Loop option in the Application Source Qualifier.

nodes

The structure at each level of the hierarchy. The highest level node is called the root node. The lowest level nodes are called leaf nodes. Other level nodes are called nodes.

non-uniform hierarchy

A hierarchy that has different numbers of nodes across the branches.

open SQL

SQL written in ABAP used to query tables on the application server. Use open SQL to access database views and transparent, pool, and cluster tables. When you join two or more sources in one Application Source Qualifier, Open SQL uses a nested loop to extract data.

package

Structure in the SAP system that holds objects in the same development project. PowerCenter creates the ZERP development class and holds all the PowerCenter objects. Package is referred to as development class in SAP 46C and earlier.

pool table

A table on the application server that does not have a one-to-one relationship with the related table on the database server.

presentation server

The top layer of the SAP three-tiered architecture. The presentation server is usually a PC or terminal that end-users access to enter or query into the SAP system.

process chain

A process chain is used to extract data from SAP BW. It links an InfoSpoke that extracts the data and writes it to an SAP transparent table or file and the ZPMSENDSTATUS ABAP program that calls the SAP BW Service. A process chain also helps you identify the point of failure in case of a system failure.

PSA transfer method

A transfer method within the SAP BW system used to load data into the Persistent Storage Area (PSA) before writing the data to the Operational Data Store (ODS) or InfoCube.

qualifying table

The last table selected in the join order that you use to preface a join condition override.

RFC

Remote Function Call. A standard interface to make remote calls between programs located on different systems. PowerCenter uses RFC each time it connects to the SAP system.

root node

The highest node in the structure of the hierarchy. It is the origin of all other nodes.

run-time transports

Transports you install in the development environment and then deploy to test and production environments.

SAP BI

An SAP system containing the BW Enterprise Data Warehouse (SAP BW). PowerExchange for SAP NetWeaver BW allows you to extract data from or load data into an SAP BW system.

SAP BW Service

An application service that listens for RFC requests from SAP BW and initiates workflows to extract from or load to SAP BW.

SAP DMI

SAP Data Migration Interface. Use the SAP interface to migrate data from legacy applications, other ERP systems, or any number of other sources into SAP.

sapnwrfc.ini

Connectivity file that enables PowerCenter to initiate RFC and CPI-C with the SAP and BW systems.

SetID

A value that provides a unique identifier for each hierarchy.

single-dimension hierarchy

A hierarchy with only one associated detail table. PowerCenter supports single-dimension hierarchies.

static filter

A filter in the Application Source Qualifier to reduce the number of rows the ABAP program returns. The Designer writes the static filter condition into the ABAP program as a WHERE clause.

stream mode

Use the stream mode extraction method to extract SAP data to buffers. Data from the buffers is streamed to the PowerCenter Integration Service.

structure field variable

A variable in the ABAP program that represents a structure field in an SAP structure. Structures are virtual tables defined in the SAP dictionary. A structure can have many fields.

structure variable

A variable in the ABAP program that represents a structure in the SAP system. Structures are virtual tables defined in the SAP dictionary.

TO_VALUE

The end range of values for the leaf node of the hierarchy. The PowerCenter Integration Service uses either this column or the FROM_VALUE column to join with the detail table.

tp addtobuffer

Transport system command used to add transport requests to a buffer prior to importing them into the SAP system.

tp import

Transport system command used to import the transport requests into the SAP system.

transparent table

A table on the application server that has a table of matching structure on the database server.

transport

SAP system used to transfer development objects from one system to another. Use the transport system when you install the PowerCenter development objects on SAP. See also *design-time transport*on page 277 and *run-time transport*on page 280.

tRFC

Transactional RFC. An SAP method of guaranteeing that RFCs are executed only once.

uniform hierarchy

A hierarchy with the same number of nodes in each of the branches.

variant

A BW structure that contains parameter values that the SAP BW system passes during program execution.

work process

A process on the application server that executes requests. All PowerCenter requests for data extraction go through a work process.

INDEX

$$BWFILTERVAR
  SAP BI mapping parameter 245, 248
$TMP
  installing ABAP programs 90

A

ABAP
  generating ABAP 104
  processing Unicode data 275
  SAP transport system 22
ABAP code blocks
  creating 111
  insertion 111
  validation 112
ABAP extraction programs
  selecting for SAP processing mappings 207
ABAP join syntax
  generating 105
  generating ABAP programs 104
  inserting into ABAP programs 111
  joining SAP sources 108
  key relationship 108
  overriding join conditions 110
  sample ABAP programs 108
  sort order example with SAP transparent tables 87
ABAP mappings
  creating a package 60
ABAP program variables
  creating 112
ABAP programs
  authority checks, adding 89
  configuring parameters 100
  copying information 94
  creating code blocks 111
  creating variables 106
  filtering IDoc data 120
  generating and installing 88
  generating local copies 90
  generation modes 104
  handling IDoc control information 120
  handling Unicode systems 90
  importing into SAP NetWeaver BW 52
  incorporating functions 97
  inserting code blocks 111
  inserting functions 99
  inserting functions, rules 101
  installing 90
  installing for versioned mappings 89
  invalid mappings 88
  mapping shortcuts 88
  mapping variables 119
  naming 89
  program flow 106
  program information, viewing 93
ABAP programs *(continued)*
  program modes 88
  purging a mapping 90
  rules for using IDoc 119
  SAP partitioning restriction 129
  troubleshooting program installation 95, 121
  undoing a checkout 90
  uninstalling 39, 94
  uninstalling from versioned mappings 89
  uploading 90
  using multiple outer joins 109
  validating functions 101
  validating the program flow 106
ABAP type variables
  creating 114
  definition 112
ACCP data type
  conversion from SAP 265
active transformations
  BAPI/RFC transformation 161
Administrator tool
  configuring SAP BW Service 50
ALE
  configuring in SAP 30
  configuring the IDoc logical system 136
  definition 135, 145
ALE configuration
  removing 39
ALEREMOTE
  description 34
APO
  using SAP for integration 18
Application Multi-Group Source Qualifier
  outbound IDoc mappings 136
  pipeline partitioning for outbound IDoc sessions 154
application servers
  generating and installing ABAP programs 90
  importing SAP ECC source definitions 72
  importing SAP ECC table definitions 73
  importing SAP ECC target tables 80
Application Source Qualifier
  ABAP program variables 112
  avoiding copies for SAP 121
  configuring for SAP 121
  creating ABAP code blocks 111
  creating SAP 121
  editing ABAP program flows 106
  generating ABAP 104
  generating ABAP join syntax 105
  generating SAP exec SQL 105
  generating SAP open SQL 105
  joining SAP sources 106
  mapping parameters with SAP, using 118
  mapping variables with SAP, using 118
  overriding data types for SAP 265
  rules for SAP join conditions 110

Application Source Qualifier *(continued)*
SAP datatypes 121
source filters for SAP 115
using IDoc sources 119
authority checks
adding to ABAP program 89
creating users in SAP 58
running SAP sessions 123
troubleshooting SAP 132

B

BAPI/RFC errors
error output 169
in BAPI/RFC sessions 173
BAPI/RFC sessions
caches 172
configuring 171
verbose logging 171
BAPI/RFC transformation
caches 172
components 162
creating 165
function input data 168
overview 161
properties 162
session processing 171
troubleshooting 170
basic IDoc type
identifying and verifying 199
BCI
communication settings, deleting 39
BCI listener mapping
basic IDoc type, identifying and verifying 199
BCI_Mappings.xml
importing objects for SAP 196
SAP business content integration 191
binary data types
support for PowerCenter and SAP 266
support for SAP BW 268
blocking data
BAPI/RFC transformation 162
buffer block size
SAP BI sessions 251
business components
creating SAP ECC 79
organizing SAP ECC in the Navigator 78
business content integration
configuration tasks for SAP 195
creating database tables for SAP 197
creating processing workflows for SAP 212
deploying SAP request files 211
importing PowerCenter objects for SAP 196
mappings for SAP 191
PowerCenter Integration Service processing for SAP 194
SAP DataSources 191, 195
SAP integration steps 195
SAP listener workflow 200
SAP logical system 34
SAP logical systems 191
SAP overview 190
SAP processing mappings 202
SAP send request workflows 211
SAP workflows 193
scheduling SAP workflows 213
business names
SAP ECC hierarchies 74
business names *(continued)*
SAP ECC tables and views 73

C

Cache Directory (property)
BAPI/RFC sessions 172
inbound IDoc sessions 156
Cache Size (property)
BAPI/RFC sessions 172
inbound IDoc sessions 156
caches
BAPI/RFC session cache files 172
inbound IDoc session cache files 156
changing parameters
configuring for SAP 100
description for SAP 99
CHAR data types
trimming trailing blanks for SAP 266
child nodes
SAP BI 234
cleanup mappings
SAP business content integration 191
cleanup sessions
creating for SAP 212
cluster tables
generating ABAP programs 104
SAP ECC definitions 73
sort order for SAP 87
code pages
alternate code pages for Unicode SAP data 275
SAP requirements 272
supported for SAP 273
cofiles
description for SAP 56
commit call
user-defined commits for inbound IDoc sessions 155
components
SAP BI, building 236
connections
SAP BI, testing 259
connectivity
importing SAP ECC source definitions 77
importing SAP ECC target definitions 82
installing ABAP programs 90
PowerCenter and SAP integration overview 19
PowerExchange for SAP NetWeaver BW integration overview 21
SAP communication interfaces 21
sapnwrfc.ini file 36
Continue on Error (property)
BAPI/RFC sessions 173
control information
handling IDoc 120
CPI-C
running stream mode SAP sessions 124
SAP communication protocol overview 22
SAP integration process 19
CRM
using SAP for integration 18
CUKY data type
trimming trailing blanks for SAP 266
Custom transformation
thread-specific code 162

D

data extraction
  SAP BW process chain 225
  scheduling for SAP BW 225
data extraction parameters
  selecting for SAP processing mappings 209
data migration
  SAP overview 176
data selection
  SAP BI mapping parameters 248
data selection entry
  SAP BI 254
data transfer process
  configuring 254
data types
  binary support for SAP BW 268
  date/time support for SAP BW 268
  ensuring accurate SAP date and numbers 265
  numeric support for SAP BW 268
  overriding SAP Application Source Qualifier 265
  PowerExchange for SAP NetWeaver 263
  PowerExchange for SAP NetWeaver BW 266
  SAP 262
  SAP BW 262
  SAP CHAR, CUKY, and UNIT trailing blanks 266
  SAP native data types 263
  supported binary for PowerCenter and SAP 266
database servers
  SAP ECC table definitions, importing 73
database views
  extracting from SAP ECC 73
datafiles
  description for SAP 56
DataSources
  activating 242
  activating in SAP 195
  assigning logical system 242
  creating 237
  customizing fields for SAP 196
  importing 243
  SAP BI, loading 234
  SAP overview 191
  SAP processing mappings for hierarchy 204
  SAP processing mappings for non-hierarchy 204
  SAP processing mappings request files 203
  selecting for SAP processing mappings 206
  troubleshooting 216
datatypes
  loading to SAP BW date columns 268
  loading to SAP BW DATS datatype 268
  loading to SAP BW TIMS datatype 269
  SAP, setting 121
date data types
  ensuring accuracy for SAP 265
date formats
  SAP 119
  SAP partitioning 129
dates
  loading to SAP BW date columns 268
DATS data type
  conversion from SAP 265
DATS datatype
  loading to SAP BW 268
definitions
  organizing SAP ECC in the Navigator 78
Delay Between Retries (property)
  inbound IDoc sessions 156
DEST
  generating and installing ABAP programs 90
  importing SAP ECC sources 77
  importing SAP ECC targets 82
detail ranges
  SAP ECC FROM_VALUE 75
  SAP ECC hierarchies 75
  SAP ECC TO_VALUE 75
detail tables
  joining SAP with SAP hierarchies 109
  joining with SAP ECC hierarchies 75
  SAP properties 87
  with hierarchies in SAP sessions 123
development system
  installing and configuring 53
development user
  creating profiles for SAP 58
  profiles to extract data from SAP BW 44
  profiles to extract data from SAP BW 3.5 43
  profiles to load data to SAP BW 45
  profiles to load data to SAP BW 3.5 45
dialog process
  file mode SAP sessions 125
  stream mode SAP sessions 124
directories
  SAP source and staging 128
DMI
  creating mappings for SAP 176
  preparing SAP DMI metadata for import 178
  primary and foreign keys for SAP 177
  SAP DMI Prepare transformation 176
DMI_Prepare_Error_Output group
  processing invalid SAP DMI documents 177
Duplicate Parent Row Handling (property)
  BAPI/RFC sessions 173
  IDoc sessions 156
duplicate rows
  in BAPI/RFC sessions, configuring 173
dynamic filters
  creating for SAP sources 116
  using with SAP 115

E

EDIDC structures
  SAP ECC source definition, importing 76
EDIDD structures
  SAP ECC source definition, importing 76
environment variables
  PMTOOL_DATEFORMAT for SAP BI 257
ERP
  using SAP for integration 18
error handling
  BAPI/RFC sessions, configuring 173
  IDoc sessions 159
error messages
  SAP BI sessions 254
error output
  BAPI/RFC errors 169
error threshold
  in BAPI/RFC sessions, configuring 173
exec SQL
  generating ABAP programs 104
  generating for SAP 105
  inserting into ABAP programs 111
  joining SAP sources 108
  overriding join conditions 110

exec SQL *(continued)*
SAP partitioning 129
sort order example for SAP 86
Extended Syntax Check (property)
IDoc sessions 156

F

file direct
accessing SAP staging files 127
file mode
ABAP program 88
configuring SAP sessions 128
enabling compressed data transfer 125
SAP sessions 123, 125
filter conditions
avoiding negative in SAP 117
creating IDoc 120
SAP, overriding 126
validating IDoc 120
Filter transformation
filtering flat file data for SAP BI 246
filtering data
flat file for SAP BI 246
loading to SAP BI 245
relational for SAP BI 245
SAP BI data selection entry 254
SAP ECC data for SAP BI 247
filters
avoiding negative filter conditions for SAP 117
creating for ABAP 106
multibyte/ISO 8859-1 with Unicode SAP systems 115
specifying for SAP 115
syntax rules for SAP 117
using dynamic with SAP 116
using SAP NUMC columns 117
using static with SAP 117
flat file sources
filtering data for SAP BI 246
SAP BI mapping parameters 249
foreign keys
SAP DMI data 177
FROM_VALUE
SAP ECC detail ranges 75
SAP ECC hierarchy relationship 76
FTP
accessing SAP staging files 128
function input data
BAPI/RFC transformation 168

G

Generate Request ID (property)
inbound IDoc session 156
Generate Transaction (property)
BAPI/RFC transformation 162
generating SQL
ABAP generation modes 104
groups
IDoc 138
viewing status for IDoc 139

H

hierarchies
creating columns for SAP ECC 74
establishing relationships with SAP ECC 76
extracting from SAP ECC 74
generating ABAP programs 104
importing from SAP ECC 74, 77
joining SAP with SAP tables 76, 109
SAP BI 234
SAP BI child nodes 234
SAP BI leaf nodes 234
SAP BI root nodes 234
SAP BI structure types 234
SAP ECC detail ranges 75
SAP ECC FROM_VALUE 76
SAP ECC nodes 75
SAP ECC non-uniform hierarchies 74
SAP ECC overview 74
SAP ECC SetID 74, 75
SAP ECC structure 75
SAP ECC TO_VALUE 76
SAP ECC uniform hierarchies 74
SAP partitioning restriction 129
structures for SAP BI, configuring 241
viewing SAP properties 87
with detail tables in SAP sessions 123
hierarchy DataSources
SAP processing mappings 204

I

idle time
outbound IDoc sessions 153
IDoc
attributes 76
configuring PowerCenter as the logical system 136
control information 76
editing IDoc type 87
filtering data 120
generating ABAP programs 104
generating metadata file 141
handling control information 120
handling outbound sessions 135
importing IDoc definitions 76
importing IDocs with the same name 87
importing metadata 140
joining with SAP tables 110
primary key 146
properties 76
receiving outbound data in real time 135
rules for using in ABAP program flows 119
segments and groups 138
source definitions 119
type 76
validating filter conditions 120
viewing segment and group status 139
IDoc definitions
viewing 76
IDoc sessions
error handling 159
real time 153
troubleshooting 159
IDoc transfer method
loading to SAP BI 236
importing metadata
IDoc from file 140

importing metadata *(continued)*
IDoc from SAP 140
inbound IDoc
document number, passing values to 150
processing invalid IDocs 151
sender partner number, passing values to 150
syntax validation 151
inbound IDoc mapping
configuring 150
inbound IDoc sessions
caches 156
commit call 155
configuring 155, 156
packet size 155
pipeline partitioning 155
validating IDocs 156
InfoPackage
configuring for SAP BI 253
troubleshooting 259
workflow names 252
InfoSources
3.x InfoSources 237
7.x InfoSources 237
activating 242
assigning logical system 242
creating 237
importing 243
SAP BI, loading 233
inner join
sample ABAP program 108
input parameters
configuring for SAP 100
description for SAP 99
Inputs Must Block (property)
BAPI/RFC transformation 162
integration
PowerCenter and SAP overview 18
Integration ID port
BAPI/RFC transformation, connecting 167
integration methods
PowerExchange for SAP NetWeaver 19
PowerExchange for SAP NetWeaver BW 21
interfaces
SAP overview 21
interval hierarchies
SAP BI 234
invalid mappings
ABAP programs 88
Is Partitionable (property)
BAPI/RFC transformation 162
ISO 8849-1
using with Unicode SAP systems 115

J

join condition
rules for SAP 110
specifying for SAP 110
join override
for ABAP 106
joining sources
rules for SAP 110
SAP 106
SAP hierarchies 87
SAP tables and hierarchies 109
SAP tables and IDoc 110
SAP with ABAP join syntax 108
joining sources *(continued)*
SAP with exec SQL 108
SAP with open SQL 107

K

key relationships
importing from SAP ECC 73, 82

L

language codes
selecting for SAP 271
supported SAP 273
LCHR datatype
with Select Distinct in SAP mappings 85
leaf nodes
SAP BI 234
libraries
Unicode for multiple SAP sessions 275
Unicode for single SAP sessions 275
listener mappings
SAP business content integration 191
listener workflows
configuring for SAP 200
running for SAP 193
starting for SAP 200
LMAPITarget
application connection for SAP 198
load balancing
SAP BW Service 50
support for SAP BW 50
load balancing connection
entry in sapnwrfc.ini 36
loading data
SAP BI overview 250
log events
SAP BI, viewing 257
SAP BW, viewing 230
logging
verbose logging for BAPI/RFC calls 171
logical system
assigning to DataSources 242
assigning to InfoSources 242
default background user 34
SAP ALE integration 30
SAP business content integration 34, 191
SAP BW extraction and loading 47
SAP program ID 34, 47
SAP, program ID 31
ZINFABCI program 34
LRAW datatype
limitation for sort order 86

M

mapping parameters
flat file sources for SAP BI 249
relational sources for SAP BI 248
SAP BI data selection 248
SAP ECC sources for SAP BI 249
using with SAP 118
mapping shortcuts
ABAP programs 88
installing ABAP programs 88

mapping variables
sample SAP filter condition 119
SAP date formats 119
using with SAP 118
mappings
creating for SAP BI 245
creating for SAP BW 221
filtering data loaded to SAP BI 243
installing ABAP for versioned mappings 89
outbound IDoc components 136
processing mappings for SAP 202
purging a mapping with an ABAP program 90
SAPALEIDoc source definitions 137
undoing a checkout with an ABAP program 90
using copies of SAP Application Source Qualifiers 121
message recovery
outbound IDoc session recovery cache folders 154
outbound IDoc sessions 153
metadata
generating file for IDoc 141
preparing SAP DMI for import 178
metadata extensions
BAPI/RFC transformation 162
migrating
data into SAP 176
multibyte data
using with Unicode SAP systems 115
mySAP applications
using SAP for integration 18

N

namespace
naming ABAP programs 89
naming conventions
SAP variables 113
NFS Mount
accessing SAP staging files 127
nodes
SAP BI child nodes 234
SAP BI leaf nodes 234
SAP BI root nodes 234
SAP ECC hierarchies 75
SAP ECC leaf 74
SAP ECC root 74
non-hierarchy DataSources
overriding SQL query for SAP 210
SAP processing mappings 204
non-uniform hierarchies
SAP ECC overview 74
NULL Field Representation (property)
inbound IDoc sessions 156
number data types
ensuring accuracy for SAP 265
Number of Retries (property)
inbound IDoc sessions 156
NUMC data type
conversion from SAP 265
NUMC datatype
filtering in SAP 117
numeric data types
support for SAP BW 268

O

open SQL
generating ABAP programs 104
generating for SAP 105
inserting into ABAP programs 111
joining SAP sources 107
overriding join conditions 110
sort order example for SAP 86
operating system profile
SAP sessions 22
Order By Ports (property)
in SAP source definitions 86
Orphan Row Handling (property)
BAPI/RFC sessions 173
IDoc sessions 156
orphan rows
in BAPI/RFC sessions, configuring 173
outbound IDoc
enabling session recovery 143
handling invalid IDoc 136
interpreting data 137
processing invalid IDocs 144
SAPALEIDoc source definitions in mappings 137
source metadata 136
outbound IDoc sessions
configuring 152, 156
idle time 153
message recovery 153
packet counts and pipeline partitioning 153
pipeline partitioning 154
reader time limit 153
recovery cache folders 154
validating IDocs 154
outer join
sample ABAP program 108
using multiple in ABAP programs 109
Output is Deterministic (property)
BAPI/RFC transformation 162
SAP/ALE IDoc Interpreter transformation 143
output parameters
configuring for SAP 100
description for SAP 99

P

package
$TMP 60
$TMP in SAP 90
creating 60
ZERP 56
packet count
outbound IDoc sessions 153
packet size
inbound IDoc sessions 155
parameters
configuring for SAP functions 99
SAP functions 97
viewing for SAP functions 99
partitioning
configuring SAP sessions 132
partner profile
SAP ALE integration 32
SAP business content integration 34
performance
buffer block size for SAP BI 251
IDoc syntax validation 156

performance *(continued)*
outbound IDoc sessions 156
permissions
running SAP sessions 123
Persist the Stage File (property)
description 126
pipeline partitioning
BAPI/RFC transformation 162
configuring SAP sessions 132
in SAP BAPI/RFC sessions 171
inbound IDoc sessions 155
outbound IDoc session recovery cache folders 154
outbound IDoc sessions 154
outbound IDoc sessions and packet counts 153
SAP restrictions 129
stream mode SAP sessions 124
pmsaputil
process chain, starting 227
third-party parameters, setting 227
PMTOOL_DATEFORMAT
environment variable for SAP BI 257
pool tables
generating ABAP programs 104
SAP ECC definitions 73
sort order for SAP 87
PowerExchange for SAP NetWeaver
configuring 24
upgrading 25
PowerExchange for SAP NetWeaver BI
performance 251
PowerExchange for SAP NetWeaver BW
configuring 40
upgrading 41
PowerExchange for SAP NetWeaver Client
registering 30, 43
PREC data
avoiding use for SAP 121
primary keys
SAP DMI data 177
process chain
configuring from PowerCenter 227
configuring SAP BW 225
loading data to SAP BI 254
starting from PowerCenter 227
processing
SAP business content integration 194
processing mappings
creating for SAP 205
executing SQL for relational targets for SAP 210
naming and generating for SAP 210
overriding SQL query for SAP 210
SAP business content integration 191
SAP Data Source request files 203
SAP target definitions 204
troubleshooting 216
update modes for SAP 202
processing sessions
creating for SAP 212
processing workflows
configuring for SAP 213
creating cleanup sessions for SAP 212
creating for SAP 212
creating sessions for SAP 212
running for SAP 193
scheduling for SAP 193
production system
installing and configuring 53
production user
creating profiles for SAP 58
profiles to extract data from SAP BW 45
profiles to extract data from SAP BW 3.5 44
profiles to load data to SAP BW 46
profiles to load data to SAP BW 3.5 46
profiles
creating for mySAP 58
creating for SAP BW 43
creating for SAP BW 3.5 43
program ID
ALE integration 31
SAP business content integration 34
SAP BW extraction and loading 47
SAP logical system 34, 47
SAP, logical system 31
sapnwrfc.ini file 36, 48
program information
copying ABAP 94
Properties tab
BAPI/RFC transformation 162
PSA transfer method
loading to SAP BI 236

R

reader time limit
outbound IDoc sessions 153
real-time sessions
description for BAPI 174
description for SAP IDoc 153
SAP IDoc source-based commit 153
records
updating to include non-key columns 81
recovery
enabling for SAP BI sessions 251
failed SAP BI sessions 259
Reinitialize the Stage File (property)
description 126
relational sources
SAP BI data, filtering 245
SAP BI mapping parameters 248
relational targets
executing SQL for SAP processing mappings 210
request files
configuring for SAP processing mappings 209
deploying for SAP 211
SAP DataSource processing mappings 203
requirements
code pages for SAP 272
Requires Single Thread per Partition (property)
BAPI/RFC transformation 162
RFC
configuring sapnwrfc.ini 36
PowerExchange for SAP NetWeaver BW integration process 21
running file mode SAP sessions 125
SAP communication protocol overview 22
SAP integration process 19
RFC server program connection
entry in sapnwrfc.ini 36
root nodes
SAP BI 234
Row Level Processing (property)
outbound IDoc sessions 156
row-level processing
outbound IDoc sessions 154

RSAPEXP
SAP DMI metadata 178
RSEIDoc3
generating metadata file for IDoc 141
rules
for target updating 81
Runtime Location (property)
BAPI/RFC transformation 162

S

SAP
application platform overview 18
deleting transport objects 39
SAP application server connection
entry in sapnwrfc.ini 48
SAP BI
data selection mapping parameters 248
loading data overview 250
Unicode 274
SAP BI components
building 236
SAP BI hierarchies
structures, configuring 241
SAP BI Monitor
log events 258
log messages 257
SAP BW Service
creating 50
troubleshooting 259
viewing log events 230, 257
SAP DMI Prepare transformation
caches 156
creating 178, 179
editing 180
overview 176
primary and foreign keys 177
SAP ECC source definitions
editing 72
importing 72, 77
importing IDoc definitions 76
importing key relationships 73, 82
troubleshooting 79
SAP ECC sources
filtering data for SAP BI 247
mapping parameters for SAP BI 249
SAP ECC target tables
importing 80
SAP functions
configuring function parameters 99
importing 98
inserting in ABAP program flow 99
insertion rules for ABAP programs 101
parameters 97
parameters defined 99
using with ABAP program 97
validating 101
versioning restriction 96
viewing 99
SAP load balancing connection
entry in sapnwrfc.ini 48
SAP NetWeaver BI 7.0
InfoSources 237
SAP NetWeaver BW
importing the ABAP program 52
SAP NetWeaver RFC SDK libraries
installing 28, 41
SAP RFC server program connection
entry in sapnwrfc.ini 48
SAP systems
ABAP program, generating and uploading 92
importing SAP ECC source definitions 72
importing SAP ECC target tables 80
SAP/ALE IDoc Interpreter transformation
creating 140, 141
editing 143
enabling session recovery 143
outbound IDoc mapping 136
overview 137
SAP/ALE IDoc Prepare transformation
caches 156
CONTROL_SNDPRN port 150
creating 141, 148
DOCNUM port 150
editing 148
SAPALEIDoc
creating target definitions 150
source definitions 136
source definitions in mappings 137
source definitions in outbound IDoc mappings 136
sapnwrfc.ini
configuring 36
connections 36
entries 48
sample entry 37, 48
scheduling
SAP DataSource workflows 213
security
accessing SAP staging files 128
mySAP authorizations 58
SAP BW 3.5 authorizations 43
SAP BW authorizations 43
segments
IDoc 138
viewing status for IDoc 139
select a single row
from SAP sources 85
select distinct rows
from SAP sources 85
SAP LCHR datatype 85
Send IDoc Based On (property)
inbound IDoc sessions 156
send request mappings
SAP business content integration 191
send request workflows
creating for SAP 211
running for SAP 193
scheduling for SAP 193
services file
entries, creating for SAP 39
session properties
SAP 132
session recovery
BAPI/RFC transformation 162
IDoc sessions 153
sessions
configuring file mode SAP sessions 128
configuring SAP 132
configuring SAP BI packet size 251
joining SAP detail tables and hierarchies 123
modes for SAP 123
overview for SAP 123
recovering SAP BI 259
SAP BAPI/RFC overview 171
SAP partitioning restrictions 129

sessions *(continued)*
SAP read permissions 123
SAP source and staging file directories 128
troubleshooting SAP 132
troubleshooting SAP BI 259
troubleshooting SAP BW 231
SetID
SAP ECC hierarchies 74
setparams (pmsaputil)
description 227
SFTP
accessing SAP staging files 128
sort order
in SAP transparent tables 86
SAP 86
SAP LRAW limitation 86
SAP pool and cluster tables 87
SAP transparent tables 86
with SAP cluster tables 87
with SAP pool tables 87
sorted hierarchies
SAP BI 234
source definitions
editing SAP ECC 72
importing from SAP ECC 72, 77
importing hierarchies from SAP ECC 74
importing IDoc definitions 76
importing SAP ECC tables and views 73
organizing SAP ECC in the Navigator 78
SAPALEIDoc 136
sort order for SAP, configuring 86
troubleshooting SAP ECC 79
source files
sample SAP directories 128
source-based commit
IDoc real-time sessions 153
sources
creating SAP Application Source Qualifiers 121
IDoc 119
importing SAP ECC definitions 73
joining SAP 106
joining SAP with ABAP join syntax 108
joining SAP with exec SQL 108
joining SAP with open SQL 107
rules for SAP join conditions 110
SAP ECC hierarchies 74
SAP ECC tables 73
SAP ECC views 73
specific application server connection
entry in sapnwrfc.ini 36
SQL
generating ABAP join syntax 105
generating open SQL 105
generating SAP exec SQL 105
joining SAP sources 106
staging areas
clearing for SAP 212
staging files
accessing SAP files on UNIX 128
accessing SAP files with file direct 127
accessing SAP files with FTP 128
accessing SAP files with NFS mount 127
accessing SAP files with SFTP 128
establishing access for SAP 127
file persistence for SAP 126
preparing the SAP staging directory 128
reusing for SAP 126
sample SAP directories 128
startchain (pmsaputil)
description 227
static filters
creating for SAP sources 117
using with IDoc 120
using with SAP 115
status
SAP BI session, troubleshooting 259
stream mode
ABAP program 88
configuring partitioning for SAP 124
entries in the SAP services file, creating 39
SAP sessions 123, 124
structure field variables
creating for SAP 113
defined for SAP 112
structure variables
creating for SAP 113
defined for SAP 112
SY-DATUM
BAPI/RFC system variable 169
SY-LANGU
BAPI/RFC system variable 169
SY-MANDT
BAPI/RFC system variable 169
SY-MODNO
BAPI/RFC system variable 169
SY-UHOST
BAPI/RFC system variable 169
SY-UNAME
BAPI/RFC system variable 169
SY-UZEIT
BAPI/RFC system variable 169
syntax validation
inbound IDocs 151
outbound IDocs 144
system variables
BAPI/RFC 169
initial values for SAP 115

T

table parameters
configuring for SAP 100
description for SAP 99
table type parameters
XML transformations 168
tables
importing SAP ECC definitions 73, 77
joining SAP with IDoc 110
joining SAP with SAP hierarchies 109
SAP detail 76
SAP ECC descriptions 73
SAP ECC detail 74
target definitions
creating SAPALEIDoc 150
importing from SAP BI 243
target tables
importing from SAP ECC 80
targets
overriding update to include non-key columns 81
rules for updating 81
terminating conditions
SAP IDoc real-time flush latency 153
test system
installing and configuring 53

thread-specific operations
BAPI/RFC transformation 162
Custom transformations 162
time-dependent hierarchies
SAP BI 234
timestamps
troubleshooting for SAP business content integration 216
TIMS datatype
loading to SAP BW 269
TO_VALUE
SAP ECC hierarchy definition 75
SAP ECC hierarchy relationship 76
tp import
development objects, transporting 56
tracing levels
BAPI/RFC transformation property 162
transaction control
BAPI/RFC transformation 162
transactional RFC (tRFC)
with IDocs 135
transfer methods
loading into SAP BI 235
transfer modes
selecting for SAP processing mappings 207
transfer rules
setting for SAP BI 242
Transformation Scope (property)
BAPI/RFC transformation 162
transformations
BAPI/RFC transformation 161
transparent tables
generating ABAP programs 104
SAP ECC definitions 73
sort order for SAP 86
transport programs
running 56
YPMPARSQ 56
transports
SAP overview 22
SAP, deleting 54
SAP, installing 56
SAP, upgrading 54
TreatCHARasCHARonRead
trimming trailing blanks for SAP 266
troubleshooting
BAPI/RFC transformation 170
business content integration 216
extracting data from SAP BW 231
failed IDoc sessions 159
importing SAP ECC source definitions 79
InfoPackage 259
installing ABAP programs 95
loading data into SAP BI 259
SAP authority checks 132
SAP BI session status 259
SAP BI sessions 259
SAP BI workflow, stopping 259
SAP BW Service 259
SAP BW sessions 231

U

Unicode
alternate code pages for SAP 275
libraries for multiple SAP sessions 275
libraries for single SAP sessions 275
SAP support 274
Unicode *(continued)*
uploading ABAP programs 90
Unicode mode
SAP, upgrading 25
uniform hierarchies
SAP ECC examples 74
uninstalling
SAP environment, cleaning 39
transport objects in SAP, removing 39
UNIT data type
trimming trailing blanks for SAP 266
UNIX
accessing SAP staging files 128
update modes
SAP processing mappings 202
update strategy
BAPI/RFC transformation 162
Update Strategy Transformation (property)
BAPI/RFC transformation 162
updating
rules for target updates 81
UTF-16
SAP Unicode data 274

V

validation
ABAP code blocks 112
ABAP program flow 106
IDoc filter conditions 120
IDocs in inbound IDoc sessions 156
processing invalid SAP DMI data 177
SAP functions 101
variables
ABAP program variables 112
ABAP type variables 114
creating ABAP program variables 106
creating SAP structure field variables 113
creating SAP structure variables 113
SAP naming conventions 113
verbose logging
for BAPI/RFC calls 171
version-dependent hierarchies
SAP BI 234
versioned mappings
installing an ABAP program 89
purging a mapping with an ABAP program 90
undoing a checkout with an ABAP program 90
uninstalling an ABAP program 89
versioning
SAP functions 96
views
extracting from SAP ECC 73
importing SAP ECC definitions 73, 77
SAP ECC descriptions 73

W

workflows
creating for SAP BI 251
creating for SAP BW 224
monitoring SAP BW 230
recovering SAP BI 259
running IDoc 159
running SAP BI on demand 252
SAP BAPI/RFC overview 171

workflows *(continued)*
SAP BI name requirement 252
SAP BI, monitoring 257
SAP BI, stopping 259
SAP business content integration 193
stopping SAP BI 259

X

XML Generator transformation
in BAPI/RFC mappings 168
XML Parser transformation
in BAPI/RFC mappings 168
XSD files
BAPI/RFC transformations, generating 165

Y

YPMPARSQ
description 56

Z

ZERP package
description 56
ZINFABCI
logical system 34
ZINFABCI transaction
communication settings, deleting 39
ZPMSENDSTATUS
configuring for data loading 254
configuring for SAP BW data extraction 225
creating a variant for SAP BW 225
description 52