Informatica

Informatica® Data Archive
6.4

Data Archive How-To Guide

Informatica Data Archive Data Archive How-To Guide
6.4
June 2015

© Copyright Informatica LLC 2003, 2018

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

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

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

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

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved.Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © 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 (c) 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.

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; and http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html.

This product includes software licensed under the Academic Free License (http://www.opensource.org/licenses/afl-3.0.php), the Common Development and Distribution License (http://www.opensource.org/licenses/cddl1.php) the Common Public License (http://www.opensource.org/licenses/cpl1.0.php), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (http:// www.opensource.org/licenses/bsd-license.php), the 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 Corporation provides this documentation "as is" without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of noninfringement, merchantability, or use for a particular purpose. Informatica Corporation does not warrant that this software or documentation is error free. The information provided in this software or documentation may include technical inaccuracies or typographical errors. The information in this software and documentation is subject to change at any time without notice.

NOTICES

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

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

Publication Date: 2018-06-08

Table of Contents

Preface 10

Informatica Resources. 10

Informatica My Support Portal. 10

Informatica Documentation. 10

Informatica Product Availability Matrixes. 10

Informatica Web Site. 10

Informatica How-To Library. 11

Informatica Knowledge Base. 11

Informatica Support YouTube Channel. 11

Informatica Marketplace. 11

Informatica Velocity. 11

Informatica Global Customer Support. 11

Abstract. xii

Chapter 2: Backing Up and Restoring the Data Vault. 13

Overview. 13

Backup and Restore Guidelines. 13

Offline Backup and Restore. 14

Online Backup and Restore. 15

Abstract. xvi

Chapter 4: Configuring Centera as a Remote Data Vault Store. 17

Overview. 17

Step 1. Install the XAM Libraries. 17

Step 2. Set Up the Target Connection. 18

Step 3. Run the Create Archive Folder Job. 20

Step 4. Add the XAM Connection to the SSA.ini File 20

Step 5. Validate the Connection to Centera. 20

Chapter 6: Configuring Data Archive for a Legacy Source Using Legacy Adapters. 21

Abstract. 21

Configuring Data Archive for a Legacy Source Using Legacy Adapters 21

Chapter 8: Data Archive Seamless Access for PeopleSoft 23

Abstract. 23

Overview. 23

Prerequisites. 24

System Privileges. 24

Object Privileges. 24

Data Archive Seamless Data Access Job. 24
Modifying the tnsnames.ora. 25
PeopleSoft Seamless Access Script. 25
Creating an Application Server Domain. 26
Creating a Process Scheduler (Batch Server). 27
Process Scheduler Setup. 28
Creating a PeopleSoft PIA. 29
Adding Multiple Sites to a Single WebLogic Domain. 33
Appendix A. 35
Script Changes. 35
Appendix B. 37

Chapter 10: Data Archive Transaction Restore API. 39

Abstract. 39
Overview. 39
List of Terms. 39
Solution Overview. 40
Features. 40
Calling the Transaction Restore API. 41
XA_TRX_RESTORE_API Oracle package. 41
Web-based URL API. 46
Security. 47

Chapter 12: Dropping and Truncating Partitions in Data Archive. 49

Abstract. 49
Overview. 49
When to Drop and Truncate Partitions. 50
Entity Configuration. 50
How to Configure an Entity to Truncate or Drop Partitions. 51
How to Choose Partitions. 53
Hard Code the Partition Names. 53
Parameterize the Partition Names. 53
Use the Partitioning Key High Value. 53
How to Drop or Truncate Partitions from Multiple Tables. 55
How to Drop and Truncate Partitions from Multiple Tables within the Same Entity. 56
Interim One - Drop Partitions. 56
Interim Two - Truncate Partitions. 56
Log Messages. 57
Normal Completion. 57
Exception Handling. 58

Chapter 14: How to Create Business Rules to Archive and Purge Transactional Data. 59
Abstract. 59
Overview. 59
Candidate Generation Process. 60
Business Rules. 61
Candidate Generation Steps. 61
Implementing Business Rules. 61

Chapter 16: How to Uninstall Data Archive 5.1. 72
Abstract. 72
Overview. 72
Step 1. Delete the History Instance. 72
Step 2. Delete the Applimation Agent. 73
Step 3. Delete the Source Instance. 73
Step 4. Delete the Agent Schema. 73
Step 5. Create and Run the drop_data_access.sql Script. 74
Step 6. Delete the Applimation Application. 74
Step 7. Drop the amhome Schema and Tablespace. 75

Chapter 18: How to Uninstall Data Archive 5.3. 76
Abstract. 76
How to Uninstall Data Archive 5.3. 76

Chapter 20: How to Use Scripts to Change Database User Passwords in the ILM Repository. 78
Abstract. 78
Overview. 78
Properties File. 79
Changing Database User Passwords. 79

Chapter 22: IBM DB2 Database Connectivity and Setup for Data Archive. 81
Abstract. 81
Overview. 81
Step 1. Set Up IBM DB2 Privileges. 82
Step 2. Bind the Packages on IBM DB2. 82
Step 3. Mine the Source Database and Create Entities. 82
IBM DB2 Connection Properties. 83
Mining the Database. 83
Step 4. Create a Source Connection. 83
Step 5. Create a Target Connection. 84
Step 6. Create an Archive Project. 84

Known Limitations. 84
abstract. lxxxv

Chapter 24: Installing Data Visualization. 86

Installing Data Visualization Overview. 86
Before You Install. 86
Installing Data Visualization. 87

Chapter 26: Integrating Third-Party Schedulers in ILM Engine. 93

Abstract. 93
Overview. 93
Integrating a Third-Party Scheduler. 94
    Replacing the Quartz Scheduler. 94
Using the Third-Party Scheduler API. 94
Integrating External Schedulers. 95
    Step 1: Create an Implementation for the AMScheduler Interface. 95
    Step 2: Register the Implementation. 98
    Step 3: Deploy the Implementation. 98

Chapter 28: Parallel Processing in Data Archive. 99

Abstract. 99
Introduction. 99
Metadata Tuning. 100
Performance-Related Source Repository Attributes. 100
    Use Staging. 100
    Use ROWID for Delete. 101
    Use Oracle Parallel DML for Delete. 101
    Oracle Parallel DML. 101
    Parallel Delete Controlled by Data Archive. 101
Archive Definition Setup. 102
    Analyze Interim. 102
    Delete Commit Interval. 102
    Insert Commit Interval. 102
    Degree of Parallelism. 102
Restore Definition Setup. 103
    Delete Commit Interval. 103
    Insert Commit Interval. 103
    Delete Degree of Parallelism. 103
    Insert Degree of Parallelism. 103
Java Parallelism - Processing Multiple Tables Simultaneously. 104
    Generate Candidates. 104
    Build Staging. 104
    Copy to Staging. 104

Validate Destination. 104
Copy to Destination. 104
Purge Staging. 104
Database Link Setup for Restore Cycles. 104

Chapter 30: Seamless Access Configuration for Siebel Applications. 106

Abstract. 106
Objective. 106
Dedicated Client Mode. 107
Web Client Mode. 108

Chapter 32: Seamless Access Setup for Oracle E-Business Suite. 116

Abstract. 116
Overview. 116
Setting Up Seamless Access Between Informatica Data Archive and Oracle E-Business Suite. . . . 118
Appendix. 127
Sample Script for Archive Only Schema. 127
Sample Script for Combined Schema. 129

Chapter 34: Seamless Access Setup for Oracle R12 in Data Archive 131

Abstract. 131
Apply Oracle Patch 9015132 "ENABLING INVOKERS PROCESSING IN R12". 131
Configure the Seamless Access Users and Responsibilities. 132
Release 12 Applications Tier Configuration for Informatica ILM Seamless Access. 132
Example: AMQUERY GL Administrator Responsibility. 133
References. 134
Abstract. cxxxv

Chapter 36: Using the Data Vault Service JDBC Driver to Connect to the Data Vault. 136

Data Vault Service JDBC Driver Overview. 136
JDBC Driver Requirements. 136
JDBC Driver Installation. 137
ODBC Data Source for the Data Vault. 138
Connecting to the Data Vault. 138

Chapter 38: Using Multiple Engines in an ILM Environment 139

Abstract. 139
Challenge. 139
Description. 140
Workflow Best Practices. 140
Keeping Track of Tomcat Engines. 140
Keeping the ./webapp/file_archive Directories Synchronized. 141

Keeping the ./webapp/xsl Directories Synchronized. 142

Chapter 40: Using PowerExchange ODBC Connections in a Data Archive Retirement Project. 143

Abstract. 143
Overview. 144
Implementation Scenario. 144
Task Flow. 145
Step 1: Install and Configure PowerExchange on z/OS. 145
Install PowerExchange on z/OS. 145
Edit the DBMOVER Configuration Member on the z/OS System. 146
Configure the PowerExchange Listener JCL on z/OS. 146
Start the PowerExchange Listener. 147
Step 2: Install and Configure PowerExchange on the Data Archive Server. 147
Install PowerExchange on the Data Archive Server. 147
Edit the DBMOVER Configuration File on the Data Archive Server. 147
Configure Environment Variables on Linux or UNIX. 148
Step 3: Install and Configure PowerExchange on the Validation Server. 148
Step 4: Install and Configure PowerExchange on Windows. 149
Install PowerExchange on Windows. 149
Edit the DBMOVER Configuration File on the Windows System. 149
Step 5: Create PowerExchange Data Maps. 149
Step 6: Configure PowerExchange ODBC Data Sources on the Data Archive Server. 150
Configure ODBC Data Sources on Linux or UNIX. 151
Configure ODBC Data Sources on Windows. 151
Step 7: Configure PowerExchange ODBC Data Sources on the Validation Server. 152
Step 8: Import z/OS Metadata. 153
Step 9: Define and Run the Retirement Project. 154
Define the Retirement Archive Project. 154
Run the Retirement Project. 154
Step 10: Validate the Retired Data. 154
Abstract. clv

Chapter 42: Discovering Foreign Key Relationships in Enterprise Data Manager. 156

Overview. 156
Airline Ticket Example. 157
Step 1. Configure Profile and Discovery Parameters. 157
Step 2. Start Enterprise Data Manager 159
Step 3. Import Metadata from the Database. 159
Step 4. Discover Table Relationships. 161
Step 5. Define Foreign Keys. 165

Preface

The *Informatica Data Archive How-To Guide* includes articles from the Informatica How-To Library. These articles assume you have knowledge of your operating systems, relational database concepts, and the database engines, flat files, or mainframe systems in your environment.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, you can access the Informatica My Support Portal at http://mysupport.informatica.com.

The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

The site contains product information, user group information, newsletters, access to the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

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

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

Informatica 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. You can access the PAMs on the Informatica My Support Portal at https://mysupport.informatica.com/community/my-support/product-availability-matrices.

Informatica Web Site

You can access the Informatica corporate web site at http://www.informatica.com. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product

and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

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

Informatica Knowledge Base

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

Informatica Support YouTube Channel

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

Informatica Marketplace

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

Informatica Velocity

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

Informatica Global Customer Support

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

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

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at http://www.informatica.com/us/services-and-training/support-services/global-support-centers/.

Abstract

This article explains the backup and restore process for Data Vault in Data Archive.

CHAPTER 2

Backing Up and Restoring the Data Vault

This chapter includes the following topics:

- Overview, 13
- Offline Backup and Restore, 14
- Online Backup and Restore, 15

Overview

You can back up and restore the Data Vault files using the same process you use to back up and restore other file systems in your organization.

The Data Vault Service does not require a specialized backup process. If you have an established file backup and restore process in your organization, you can use the same process to back up and restore the files in a Data Vault.

You can back up the Data Vault files in the following ways:

- Offline backup. The Data Vault Service must be shut down before the start of the backup or restore process.
- Online backup. The Data Vault Service continues to operate during the backup or restore process.

Backup and Restore Guidelines

Use the following guidelines when you back up or restore the Data Vault:

- To ensure that you can recover archived data from data deletion or corruption, back up the Data Vault files after each time you run a Data Vault Loader job. Also back up the ssa.ini and nucleus.ini files when you modify the Data Vault Service configuration.
- To ensure that the metadata in the Data Vault repository provides the correct information about the data files in the Data Vault, back up the Data Vault repository and data files at the same time.
  For example, on UNIX, tar the Data Vault repository and data files into one tarball with one timestamp. When you restore the backup files, the metadata in the Data Vault repository will correctly describe the data files.

- Use the parameters in the nucleus.ini configuration file to determine the location and name of the Data Vault repository. On the server that hosts the Data Vault Service, view the nucleus.ini to get the file and directory information.

Parameter	File	Description
DatabasePath	nucleus.ini	Directory where the Data Vault repository is located.
DATABASE section	nucleus.ini	Name of the Data Vault repository file. The file name is the same as the name of the DATABASE section with an extension of *.n00*. For example, if the nucleus.ini has a database entry of *[DATABASE meta]*, then the name of the Data Vault repository file is *meta.n00*.

- The location of the Data Vault data files is specified in the archive target connection in Data Archive. Log in to Data Archive and view the target connection for the Data Vault you want to back up or restore. The *Data Vault SCT Files Directory* property of the target connection defines the location of the Data Vault data files.

Offline Backup and Restore

If you do not require the Data Vault to be highly available, you can take the Data Vault Service off line before you perform the backup. Use any backup utility supported by your file system to perform an offline backup or restore.

Shut down the Data Vault Service before you start the process.

1. View the configuration files to determine the directories where the Data Vault repository and data files are stored.
2. Shut down the Data Vault Service.

   Shutting down the Data Vault Service also shuts down the Data Vault Agent on the server. If you have installed the Data Vault Agent on multiple servers, you do not need to shut down the Data Vault Agent on other servers.
3. Copy the files to the backup location or restore the files from the backup location.
   - To back up the archive files, use your backup utility to copy the files from the Data Vault Service location to the backup location.
   - To restore the backup files, use your backup utility to copy the files from the backup location to the location of the Data Vault Service specified in the configuration files.
4. Restart the Data Vault Service.

   The Data Vault Agent on the server also starts.

Online Backup and Restore

If you require the Data Vault to be highly available, you can take a snapshot of the Data Vault while the File Archive Service is online and then back up the snapshot.

Use a snapshot utility supported by your file system to take a snapshot of the File Archive repository and data files. Read the documentation for your file system to determine the proper way to use the snapshot utility to back up or restore files.

1. View the configuration files to determine the directories where the File Archive repository and data files are stored.
2. Run the snapshot utility for your file system to back up or restore the File Archive repository and data files.
   - To back up the files, use the snapshot utility to create a snapshot of the File Archive repository and the data files.
     Consult the documentation for your file system and follow the instructions for creating a snapshot.
     When the snapshot process is complete, use your file backup utility to back up the File Archive repository and data files in the snapshot location.
   - To restore the backup files, consult the documentation for your file system and follow the instructions for restoring the File Archive repository and data files from a snapshot.
     When you complete the restore process, the File Archive Service can use the newly restored File Archive repository and data files.

Abstract

This article provides information on how to configure Centera to work as a Data Vault store in Data Archive. The procedure uses the eXtensible Access Method (XAM) API to retrieve data from Centera. This article also provides information on backing up and restoring the Data Vault on Centera.

CHAPTER 4

Configuring Centera as a Remote Data Vault Store

This chapter includes the following topics:

- Overview, 17
- Step 1. Install the XAM Libraries, 17
- Step 2. Set Up the Target Connection, 18
- Step 3. Run the Create Archive Folder Job, 20
- Step 4. Add the XAM Connection to the SSA.ini File , 20
- Step 5. Validate the Connection to Centera, 20

Overview

You can use Centera as a remote Data Vault store in Data Archive.

To create a Data Vault in Centera, complete the following tasks:

1. Install the XAM API libraries on a machine that hosts the File Archive Service.
2. Set up the target connection for an EMC Centera archive store.
3. Run the Create Archive Folder job.
4. Add an entry for the XAM connection section to the SSA.ini file in the File Archive Service.
5. Validate the connection to Centera.

To back up the Data Vault in Centera, use the same backup process that you have established for other files in Centera.

Step 1. Install the XAM Libraries

Install the XAM API libraries on a machine that hosts the Data Vault Service.

You can install the XAM API libraries on a machine where one of the following Data Vault Service components is installed:

- Data Vault Service

- Data Vault Agent
- Data Vault Service Data Archive plug-in

You can download the XAM libraries for your operating system from the EMC download web site. Informatica does not ship the XAM libraries.

The following is an example of the list of XAM API libraries for Windows:

- centera_vim.dll
- FPCore.dll
- fpos64.dll
- fpparser.dll
- FPStreams.dll
- FPUtils.dll
- FPXML.dll
- pai_module.dll
- xam.dll
- xam_toolkit.dll

On UNIX, if the libraries are stored in a location other than the default, you must set the library path in the appropriate system variable, such as LD_LIBRARY_PATH.

Step 2. Set Up the Target Connection

In Data Archive, create a target connection to the Data Vault and set the archive store type to EMC Centera.

The following table describes the properties that you need to set for the target connection:

Property	Description
Connection Name	Archive target name. Use the name to differentiate archive target connections. When you create an archive project and you assign a target, you select the connection name that you define here. You can use the connection name to display a list of all connection types or filter the list of connections by name.
Connection Description	Long text description for the connection name. To manage connections, you can use this field to filter the list of connections.
Connection Type	Type of target connection. The connection type determines the connection properties you need to define. Set the connection type to Data Vault.
Application Version	Business application and application versions that the database connection type supports. The connection type determines the application version available.
Staging Directory	Location where the archive job creates the BCP files.
Number of Rows Per File	Maximum number of rows in the BCP files. When a BCP file reaches the maximum number of rows, the archive job creates another BCP file and adds subsequent rows to the new file.

Property	Description
Data Vault SCT Files Directory	Directory where the Data Vault Service creates the SCT files.
Data Vault Connection Name	Set to the default connection name *FAS*.
Data Vault Folder Name	Name of the folder in the Data Vault where you archive the data. When you run the standalone job to create the archive folder, you provide the target connection. The job takes the folder name that from the target connection and creates the folder using the name you enter here.
Data Vault Host	Host name of the machine where the Data Vault Service is running.
Data Vault Port	Port number used by the Data Vault Agent to connect to the Data Vault Service. Default is 8600.
Data Vault Administration Port	Port number used by the Data Vault Administration Tool and command line program to connect to the Data Vault Service. Default is 8500.
Data Vault User	User name for the Data Vault Service administrator user account to log in to the Data Vault schema.
Data Vault User Password	Password for the Data Vault Service administrator user account.
Confirm Password	Confirmation of the Data Vault user password.
Add-On URL	URL for the Data Vault Service for External Attachments component. Required to convert external attachments from a proprietary format into regular format. Scripts are used to convert the attachments.
Use Mined Source Schema	Indicates whether to use the schema from the source instance mined in the Enterprise Data Manager. If enabled, the Data Vault Loader job uses the mined source schema structure from the Enterprise Data Manager to create the schemas in the Data Vault folder. The job creates the same schema structure in the Data Vault folder and adds the transactional tables to the corresponding schema. The job creates the dbo schema and adds the metadata tables to the dbo schema. If not enabled, the Data Vault Loader job creates the dbo schema in the archive folder. The job adds all transactional tables across all schemas and metadata tables to the dbo schema. **Note:** The Data Vault Loader job uses the schema structure from the source instance that you mined in the Enterprise Data Manager. You may need to re-mine the source instance before you run the Data Vault Loader job if the source connection schema is different from the mined schema. Schema names can vary between instances. For example, you mined a development instance. You exported metadata from the development instance and imported the metadata into the production instance. If the schema names are different in development and production, you must re-mine the production instance.
Archive Store Type	Type of storage for the Data Vault. The store type can be an external storage platform. Set to EMC Centera.
XRI	Extensible Resource Identifier (XRI) that describes the XAM server and the location of the XAM credentials file for the Centera XAM connection.

Step 3. Run the Create Archive Folder Job

Run the Create Archive Folder job to create the folder in the Centera Data Vault store.

The Create Archive Folder job creates the folder in Centera and adds a XAM connection entry to the SSA.ini file in the Data Vault Service plug-in in Data Archive. The job sets the name of the folder and the XAM connection to the name specified by the Data Vault Folder Name property in the target connection. It also adds the location of the Centera server based on the location specified by the XRI property in the target connection.

For example, the Data Vault Folder Name property in the target connection is set to CenteraLoc. The Create Archive Folder job creates a folder named CenteraLoc in the Centera Data Vault store and adds a XAM connection named CenteraLoc to the SSA.ini file in the Data Vault Service plug-in in Data Archive:

```
[XAM_CONNECTION CenteraLoc]
XRI=centera_vim!10.1.42.64?c:\ILM-FAS\ilmcentera1.pea
```

In this example, 10.1.42.64 is the IP address of the Centera server and *ilmcentera1.pea* is the PEA file that stores the user credentials for the Centera server. You specify these details in the XRI property in the target connection based on information provided by the Centera storage administrator.

Step 4. Add the XAM Connection to the SSA.ini File

Add a XAM connection with the Centera information to the SSA.ini file on the machine where the Data Vault Service is installed.

The following is an example of a XAM connection entry in the SSA.ini file for the Data Vault Service:

```
[XAM_CONNECTION CenteraLoc]
XRI=centera_vim!10.1.42.64?c:\ILM-FAS\ilmcentera1.pea
RETENTION_PERIOD=0
```

The XAM connection name and XRI details in the SSA.ini file in the Data Vault Service must match the XAM connection name and XRI details in the SSA.ini file in the Data Vault Service plug-in in Data Archive.

Step 5. Validate the Connection to Centera

Verify the connection to Centera from the Data Vault Service and from the Data Vault Service plug-in in Data Archive.

To validate the connection to the Centera Data Vault store, run the following Data Vault Service administration command:

```
ssadrv -a snia-xam://<DataVaultFolder>/
```

For example, if the folder name in the Centera Data Vault store is CenteraLoc, run the following command:

```
ssadrv -a snia-xam://CenteraLoc/
```

In this command, *snia-xam://CenteraLoc/* is the URL for the connection to the Centera Data Vault store and CenteraLoc is the name of Data Vault folder. CenteraLoc is also the name of the XAM connection in the SSA.ini file.

CHAPTER 6

Configuring Data Archive for a Legacy Source Using Legacy Adapters

This chapter includes the following topics:

- Abstract, 21
- Configuring Data Archive for a Legacy Source Using Legacy Adapters , 21

Abstract

This article describes how to configure Data Archive for a legacy source using legacy adapters for connectivity.

Configuring Data Archive for a Legacy Source Using Legacy Adapters

1. Install the legacy application adapter.
2. Start the legacy application adapter.

   In most cases, you can start the adapter with security turned off.
3. Configure the legacy application adapter.
4. Mine the legacy source tables in the legacy application adapter.
5. Mine the legacy application adapter synonyms in EDM.

   Use the legacy application adapter in the platform list of EDM. (Default username and password: edadb/edadb)
6. Create the staging tables on the source.

   If you cannot create the staging tables on the legacy source, create the staging schema on the same database as AMHOME.

7. Create the staging table on AMHOME.

   This step is only required if you have business rules or if you will create interims for your entity for archiving based on parameters.

8. Create adapter to AMHOME on the legacy application adapter.
9. Mine the staging tables in the legacy application adapter.
10. Create the source repository with "No" for "Use Staging" and "No" for "Parallel Entity Run."

    Leave this property unchecked to ensure that the engine does not create job-specific interims. The engine uses what you created in steps 7-9.

11. Run the archive and purge cycle.

    As long as the where clause for insert and delete is "sql92," you should be fine.

CHAPTER 8

Data Archive Seamless Access for PeopleSoft

This chapter includes the following topics:

- Abstract, 23
- Overview, 23
- Prerequisites, 24
- Data Archive Seamless Data Access Job, 24
- Modifying the tnsnames.ora, 25
- PeopleSoft Seamless Access Script, 25
- Creating an Application Server Domain, 26
- Creating a Process Scheduler (Batch Server), 27
- Process Scheduler Setup, 28
- Creating a PeopleSoft PIA, 29
- Appendix A, 35
- Appendix B, 37

Abstract

This article outlines the steps necessary to create the PeopleSoft seamless access for Data Archive 5.3.

Overview

The purpose of this document is to outline the steps necessary to create the PeopleSoft seamless access for Data Archive 5.3.

The screen captures are taken from PeopleSoft CRM 8.9 with PeopleTools 8.48.17 using WebLogic. Your screens may differ slightly depending on your application, PeopleTools version, and web server.

Prerequisites

Verify that the following steps have been carried out:

- Create a user for the combined access schema (8 characters maximum).
- Create a user for the Data Archive access schema (8 characters maximum).

Both the users must have the following privileges:

System Privileges

- GRANT CREATE SESSION
- GRANT ALTER SESSION
- GRANT ALTER ANY TRIGGER
- GRANT EXECUTE ANY PROCEDURE
- GRANT SELECT ANY DICTIONARY
- GRANT SELECT ANY TABLE - or grant select on all ERP tables; must have synonyms to ERP tables if not the owner of all ERP tables.

Object Privileges

- GRANT INSERT on managed tables
- GRANT DELETE on managed tables
- GRANT UPDATE TABLE on managed tables
- GRANT CREATE DATABASE LINK - database link used to access History Read Only user

Data Archive Seamless Data Access Job

The Data Archive Seamless Data Access job will create the two schemas, and populate them with the appropriate synonyms and views.

To create the Seamless Data Access job:

1. Click Jobs > Schedule a job.
2. Select Standalone Programs and click Add Item.
3. Select SEAMLESS_DATA_ACCESS from the list.

Available fields to be specified are discussed in the following table:

Field	Description
Source Repository	Database with Production data.
Destination Repository	Database with archived data.
Combined Schema Name (and Password)	Schema name for Seamless access across the two Databases.

Field	Description
Query Schema Name (and Password)	Schema for allowing access to archived data.
Combined/Query Schema Location	Specify whether Schemas were created at the Data Source or Data Destination.
Combined/Query Schema Location	Database link name from Data Source to Data Destination

Modifying the tnsnames.ora

The tnsnames.ora file must be modified, on the server where your application server and batch servers are running.

You must create aliases for the COMBINED and ARCHIVE schemas (example schemas). The new aliases point to the original CRM890 instance. This is necessary for the configuration of the application server and process scheduler.

```
# TNSNAMES.ORA Network Configuration File: C:\oracle\ora92\network\admin\tnsnames.ora
# Generated by Oracle configuration tools.

CRM890 =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.70)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = CRM890)
    )
  )

ARCHIVE =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP) (HOST = 192.168.1.70) (PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = CRM890)
    )
  )

COMBINED =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = 192.168.1.70)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SERVICE_NAME = CRM890)
    )
  )
```

PeopleSoft Seamless Access Script

An appropriate PeopleSoft seamless Access Script must be run against your PeopleSoft Instance. The scripts are available on the Informatica Product Download Center website. The script should be run as the

SYSTEM user. You will need to run the script twice: once with COMBINED as the "Combined Access ID" and once using ARCHIVE.

This script will configure PeopleSoft-specific security objects needed for the two Seamless Access schemas, the most important of which are the PSDBOWNER and PSSTATUS tables.

After running the script twice these tables should resemble the following:

PS.PSDBOWNER

```
DBNAME   OWNERID
CRM890   SYSADM
COMBINED COMBINED
ARCHIVE  ARCHIVE
```

CRM890.PSSTATUS

```
VERSION     OWNERID  TOOLSREL  LASTREFRESHDTTM  LASTCHANGEDDTTM  OWNERACCT
UNICODE_ENABLED  DBID
157102      SYSADM   8.48         12-JUN-08        12-JUN-08         1
SERVER
```

COMBINED.PSSTATUS

```
VERSION     OWNERID  TOOLSREL  LASTREFRESHDTTM  LASTCHANGEDDTTM  OWNERACCT
UNICODE_ENABLED  DBID
157102      COMBINED 8.48         12-JUN-08        12-JUN-08         1
SERVER
```

ARCHIVE.PSSTATUS

```
VERSION     OWNERID  TOOLSREL  LASTREFRESHDTTM  LASTCHANGEDDTTM  OWNERACCT
UNICODE_ENABLED  DBID
157102      ARCHIVE  8.48         12-JUN-08        12-JUN-08         1
SERVER
```

As the archived data is by definition read-only, this script makes all PeopleSoft pages in the COMBINED and ARCHIVE schemas "display only" by default.

Note: "Display-only" is a PeopleSoft notion which means, not only the data on a given page cannot be modified, but also disables the action buttons.

This script will also unset the "display only" flag on pages which allow you to click the button to run certain reports. This is a generic script and modifications may need to be made to adapt it to your particular environment.

Creating an Application Server Domain

Create two additional application server domains: one for the combined access schema and one for the archive only schema.

To configure an application server domain, the psadmin program is used. It can be found in the following directory: `$PS_HOME/appserv`

```
---------------------------------------------
Quick-configure menu -- domain: COMBINED
---------------------------------------------
      Features                     Settings
     ==========                   ==========
```

```
  1) Pub/Sub Servers   : Yes   15) DBNAME     :[COMBINED]
  2) Quick Server      : No    16) DBTYPE     :[ORACLE]
  3) Query Servers     : No    17) UserId     :[VP1]
  4) Jolt              : Yes   18) UserPswd   :[VP1]
  5) Jolt Relay        : No    19) DomainID   :[COMBINED]
  6) WSL               : No    20) AddToPATH  :[c:\oracle\ora92\bin]
  7) PC Debugger       : No    21) ConnectID  :[people]
  8) Event Notification: No    22) ConnectPswd:[people]
  9) MCF Servers       : No    23) ServerName :[]
 10) Perf Collator     : No    24) WSL Port   :[7010]
 11) Analytic Servers  : No    25) JSL Port   :[9010]
 12) Domains Gateway   : No    26) JRAD Port  :[9100]
      Actions
     =========
 13) Load config as shown
 14) Custom configuration
  h) Help for this menu
  q) Return to previous menu
HINT: Enter 15 to edit DBNAME, then 13 to load
Enter selection (1-26, h, or q):
The only fields that need particular attention are:
15)  DBNAME        the name of the db alias in the tnsnames.ora file. It will be checked
against the
PSDBOWNER and PSSTATUS tables
17)  UserId    a PeopleSoft user id. Either the super-administrator (PS, VP1) or PSAPPS
which can only start the app server
18)  UserPswd        the PeopleSoft user password. The psadmin program can encrypt it.
19)  DomainID        a label used internally by Tuxedo. There is no reason why it should
differ from
DBNAME
21)  ConnectId        this is always "people" Notice the "l"
22)  ConnectPswd    this is always "people" Notice the "1"
23)  ServerName    this is only for Sybase and Informix customers
```

You will need to create an application server domain for the ARCHIVE schema in the same fashion.

Creating a Process Scheduler (Batch Server)

To run batch jobs on the combined (production + archive) data or the archived data, you will need to create two additional process schedulers. Respectively, they will use the COMBINED and ARCHIVE tns aliases and access the data through the COMBINED and ARCHIVE schemas.

To configure a Process Scheduler, the psadmin program is used. It can be found in the `$PS_HOME/appserv` directory.

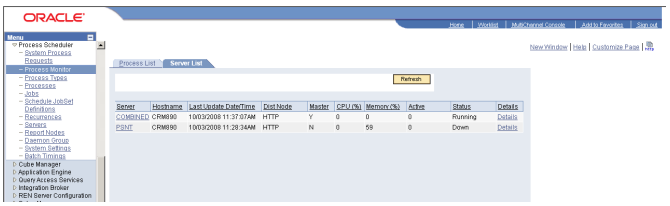
```
------------------------------------------------------------
Quick-configure menu -- Scheduler for Database: COMBINED
------------------------------------------------------------
      Features                      Settings
     ==========                    ==========
  1) Master Schdlr  : Yes     5) DBNAME     :[COMBINED]
  2) App Eng Server : Yes     6) DBTYPE     :[ORACLE]
                              7) PrcsServer :[COMBINED]
                              8) UserId     :[VP1]
                              9) UserPswd   :[VP1]
                             10) ConnectID  :[people]
                             11) ConnectPswd:[people]
                             12) ServerName :[]
      Actions                13) Log/Output Dir:[%PS_SERVDIR%\log_output]
     =========               14) SQRBIN     :[c:\psft\HRMS83\bin\sqr\ORA\binw]
  3) Load config as shown    15) AddToPATH  :[%WINDIR%;%WINDIR%\SYSTEM32]
  4) Custom configuration    16) DBBIN      :[C:\oracle\ora92\bin]
  h) Help for this menu
  q) Return to previous menu
HINT: Enter 5 to edit DBNAME, then 3 to load
```

```
Enter selection (1-16, h, or q):
5)  DBNAME        the name of the db alias in the tnsnames.ora. It will be checked
against the
PSDBOWNER and PSSTATUS tables
7)  PrcsServer        the unique instance name of the Process Scheduler
8)  UserId    a PeopleSoft user id. Either the super-administrator (PS, VP1) or PSAPPS
which can only start the app server
9)  UserPswd        the PeopleSoft user password. The psadmin program can encrypt it.
10)  ConnectId        this is always "people" Notice the "l"
11)  ConnectPswd    this is always "peop1e" Notice the "1"
12)  ServerName    this is only for Sybase and Informix customers
```

Go to the Custom Configuration (choice 4) and provide a unique value for the field PrcsServerName. COMBINED is used in this example.

In the process monitor, you should see both process schedulers. To run a report on the combined data, you have to run the respective report on the COMBINED process scheduler. To run a report on only the archived data, you would need to create an ARCHIVE process scheduler as detailed above.

The Server name in the following screen corresponds to the PrcsServerName field above:



Process Scheduler Setup

A second process scheduler must be configured for the combined access:

DBNAME = AMPROD

PrcsServer = AMPROD - This field is important and every process scheduler should have a unique identifier.

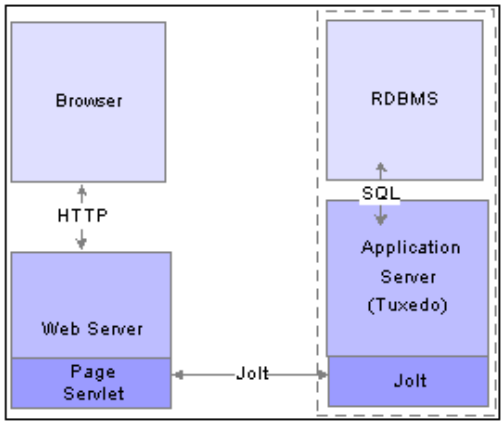
```
-------------------------------------------------------------
Quick-configure menu -- Scheduler for Database: AMPROD
-------------------------------------------------------------
      Features                        Settings
     ==========                      ==========
  1) Master Schdlr  : Yes      5) DBNAME     :[AMPROD]
  2) App Eng Server : Yes      6) DBTYPE     :[ORACLE]
                               7) PrcsServer :[AMPROD]
                               8) UserId     :[PS]
                               9) UserPswd   :[PS]
                              10) ConnectID  :[people]
                              11) ConnectPswd:[peop1e]
                              12) ServerName :[]
      Actions                 13) Log/Output Dir:[%PS_SERVDIR%\log_output]
     =========                14) SQRBIN     :[c:\peoplesoft\HRMS90\bin\sqr\ORA\b
inw]
  3) Load config as shown     15) AddToPATH  :[%WINDIR%;%WINDIR%\SYSTEM32]
  4) Custom configuration     16) DBBIN      :[c:\oracle\ora92\bin]
  h) Help for this menu
  q) Return to previous menu

Enter selection (1-16, h, or q): 4
```

In the process monitor both the process schedulers must appear. Run the report on the AMPROD process scheduler.

Creating a PeopleSoft PIA

The PeopleSoft ERP runs on PeopleSoft Pure Internet Architecture, or PIA which requires the following software elements:

- A relational database management system (RDBMS) server
- An application server – BEA Tuxedo
- A batch server – BEA Tuxedo
- A web server – BEA WebLogic, IBM WebSphere, or Oracle Application Server
- A web browser



It should be noted that what PeopleSoft calls the web server is in fact a J2EE based application server. Basically, only the web server functionalities are used and all of the transaction processing is carried out by Tuxedo. Earlier versions of PeopleSoft supported only Apache servers.

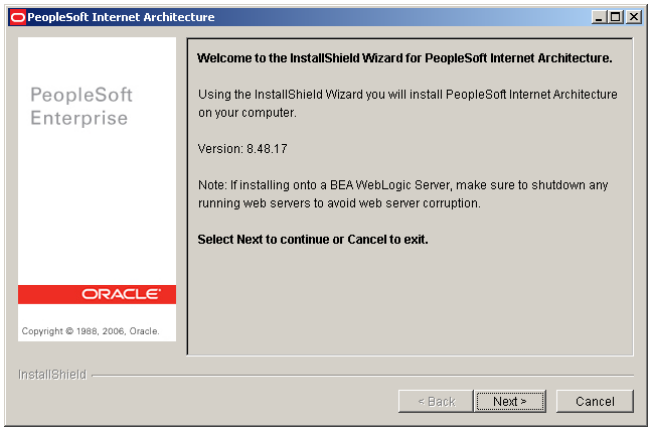
Tuxedo domains communicate with the web server via the JOLT protocol. Each tuxedo domain has a unique port number. However, several tuxedo domains can be specified in the same PIA web configuration for load balancing and failover purposes.

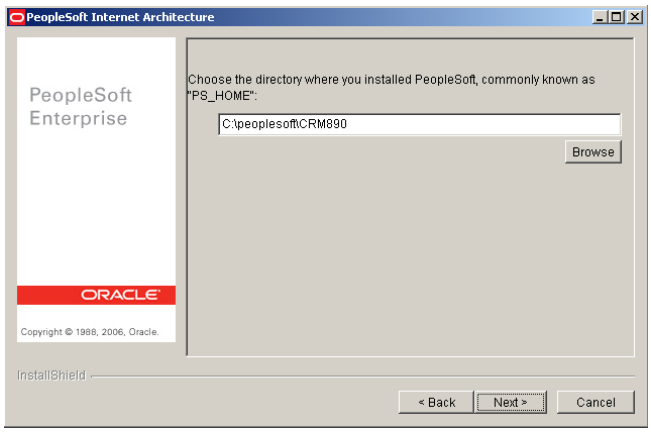
The PIA Installer creates a link between a PeopleSoft site (URL) and a jolt port. (To specify multiple jolt ports for load balancing and failover, the configuration files must be edited manually.)

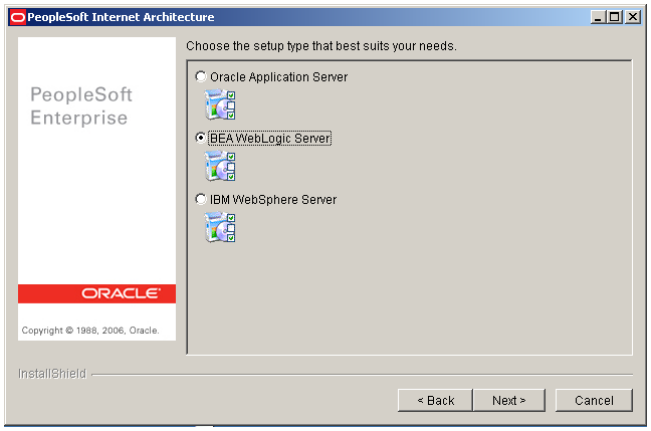
To install a PIA, run the appropriate installer program for your platform from the $PS_HOME\setup \mpinternet (depending on the version of PeopleTools) folder:

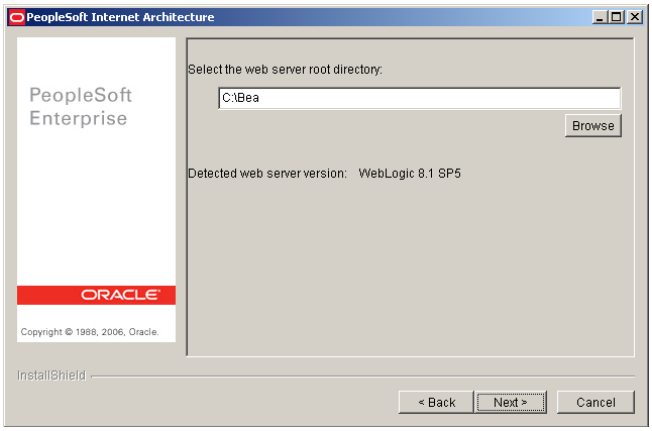
For example, setup.exe.

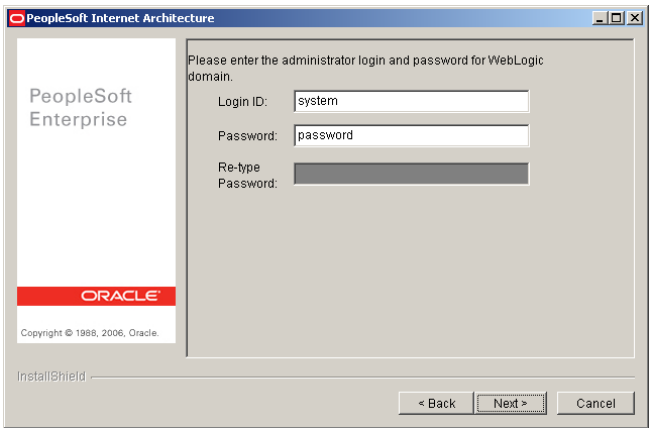
You can create one WebLogic domain per environment (Production, Combined, or Archive) or, in order to limit memory use, the three environments can share the same WebLogic domain. This will be discussed in the next section.

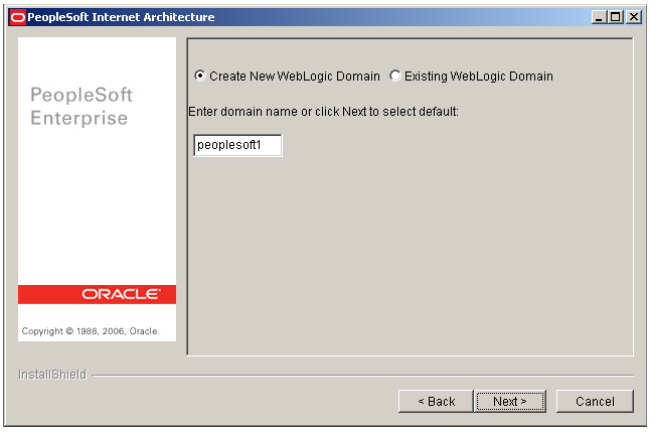


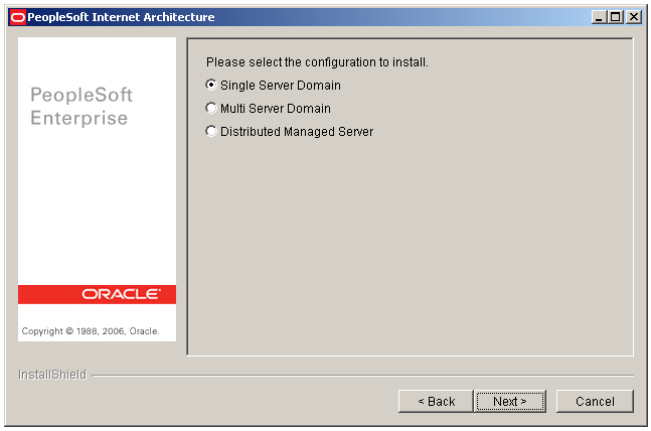


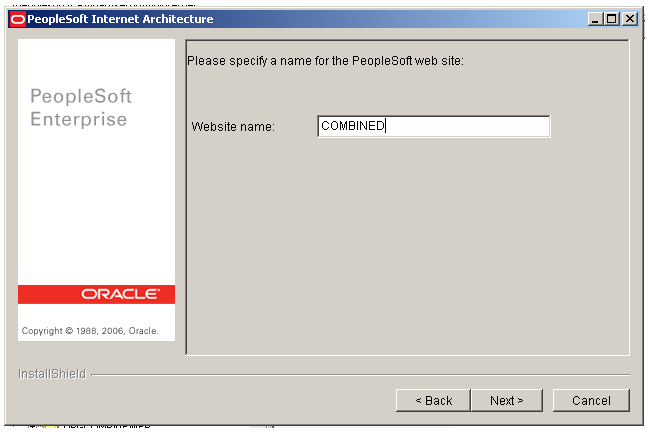


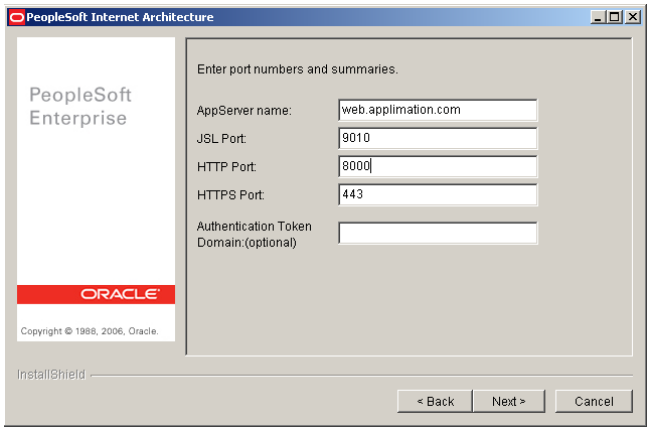

PeopleSoft Internet Architecture
PeopleSoft Enterprise
Select the web server root directory:
C:\Bea
Browse
Detected web server version: WebLogic 8.1 SP5
ORACLE
Copyright © 1988, 2006, Oracle.
InstallShield
< Back
Next >
Cancel

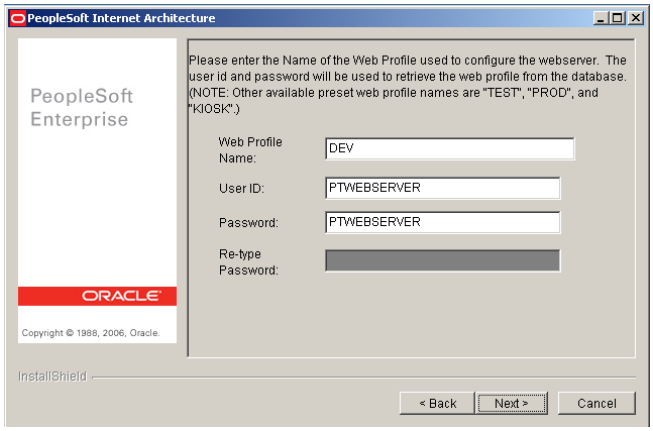

PeopleSoft Internet Architecture
PeopleSoft Enterprise
Please enter the administrator login and password for WebLogic domain.
Login ID:
system
Password:
password
Re-type Password:
ORACLE
Copyright © 1988, 2006, Oracle.
InstallShield
< Back
Next >
Cancel

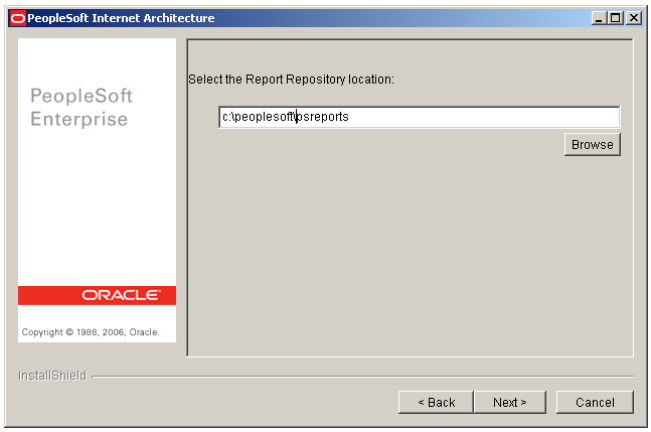

PeopleSoft Internet Architecture
PeopleSoft Enterprise
Create New WebLogic Domain
Existing WebLogic Domain
Enter domain name or click Next to select default:
peoplesoft1
ORACLE
Copyright © 1988, 2006, Oracle.
InstallShield
< Back
Next >
Cancel

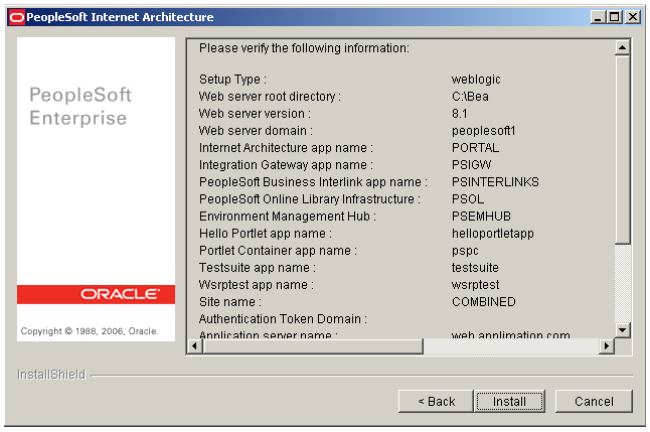

PeopleSoft Internet Architecture
PeopleSoft
Enterprise
ORACLE
Copyright © 1988, 2006, Oracle.
Please select the configuration to install.
Single Server Domain
Multi Server Domain
Distributed Managed Server
InstallShield
< Back
Next >
Cancel


PeopleSoft Internet Architecture
PeopleSoft
Enterprise
ORACLE
Copyright © 1988, 2006, Oracle.
Please specify a name for the PeopleSoft web site:
Website name:
COMBINED
InstallShield
< Back
Next >
Cancel


PeopleSoft Internet Architecture
PeopleSoft
Enterprise
ORACLE
Copyright © 1988, 2006, Oracle.
Enter port numbers and summaries.
AppServer name:
web.applimation.com
JSL Port:
9010
HTTP Port:
8000
HTTPS Port:
443
Authentication Token
Domain:(optional)
InstallShield
< Back
Next >
Cancel







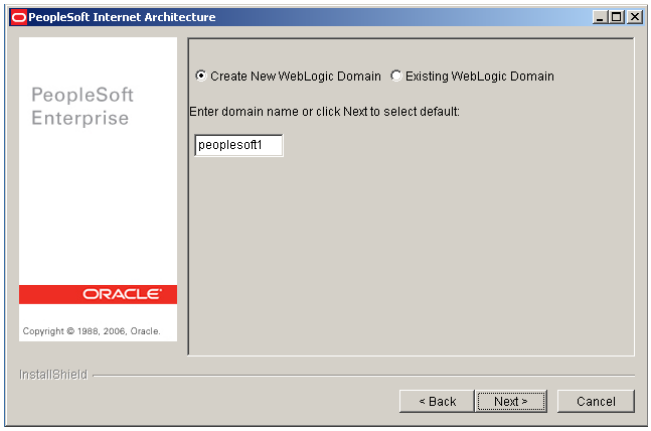
These steps would need to be run again to create the PIA for the ARCHIVE environment.

Adding Multiple Sites to a Single WebLogic Domain

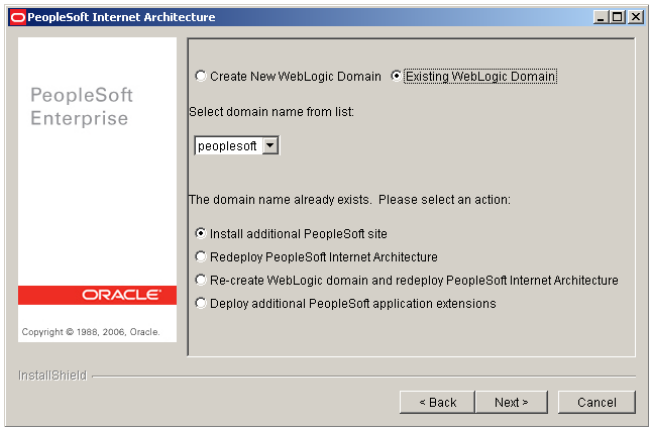
To create a new PIA configuration, a new WebLogic domain may be created, or a new PeopleSoft site may be added to an existing WebLogic domain. The advantage of this latter solution is, running an additional JVM on the production server will not be necessary.

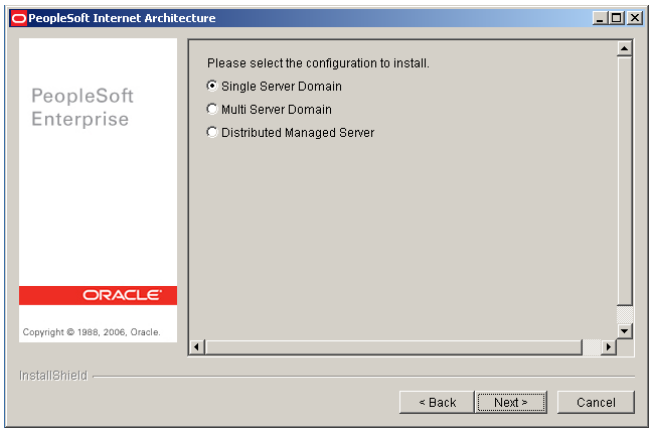
We will assume that the PIA for the production environment has already been created and its URL is: `http://web.applimation.com/CRM890/signon.html`

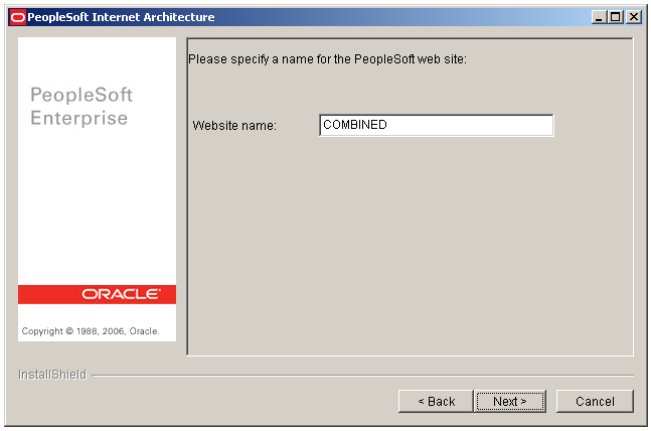
And the JOLT port being used is 9000. Run the installer until you reach the following screen:

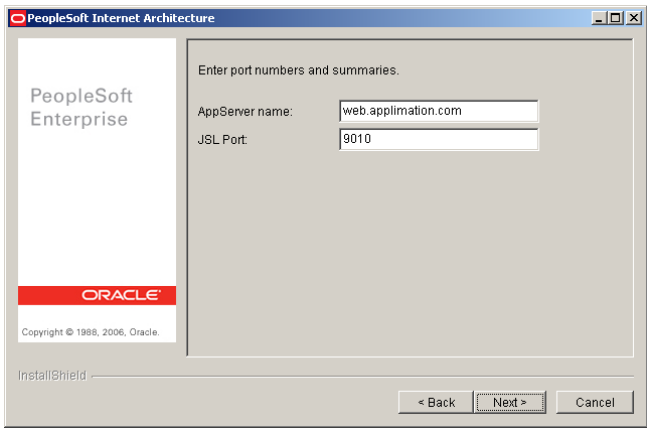


In this screen, specify the existing WebLogic domain which is called PeopleSoft by default:









Specify the site name, the JOLT port, and the application server name. HTTP, HTTP/S ports, and the authentication domain are defined when the WebLogic domain is created.

The resulting URL would be the following:

```
http://web.applimation.com:8000/COMBINED/signon.html
```

And its requests would be routed to the tuxedo domain listening on port 9010.

Appendix A

Following are the modified seamless access data conversion scripts to correct all known issues that were found. The scripts can also be altered to turn on/off a number of security settings. Detailed descriptions of the resolutions can be found at the end of this document.

Script Changes

1. archive-51-ps84-seamless_UVA.sql –University of Virginia
   This script was altered as detailed below:
   - Read Only Access to Pages
   - Able to Access buttons on PO Inquiry Page

- Don’t allow the user the ability to revise nVision parameters
- User can run query to HTML
- User can run query to Excel
- User cannot edit Queries
- User cannot schedule queries
- User cannot run reports from report menu

2. archive-51-ps84-seamless_Hitachi.sql – Hitachi
   This script was altered as detailed below:
   - Read Only Access to Pages
   - Able to Access buttons on PO Inquiry Page
   - Don’t allow the user the ability to revise nVision parameters
   - User can run query to HTML
   - User can run query to Excel
   - User cannot edit Queries
   - User can schedule queries
   - User can run reports from report menu
3. archive-51-ps84-seamless_Standard.sql– Standard 8.4 Script
   This script was altered as detailed below:
   - Read Only Access to Pages
   - Able to Access buttons on PO Inquiry Page
   - Don’t allow the user the ability to revise nVision parameters
   - User can run query to HTML
   - User can run query to Excel
   - User cannot edit Queries
   - User can schedule queries
   - User can run reports from report menu
4. archive-51-ps88-seamless_Standard.sql– Standard 8.8 Script
   This script was altered as detailed below:
   - Read Only Access to Pages
   - Able to Access buttons on PO Inquiry Page
   - Don’t allow the user the ability to revise nVision parameters
   - User can run query to HTML
   - User can run query to Excel
   - User cannot edit Queries
   - User can schedule queries
   - User can run reports from report menu
5. archive-51-ps88-seamless_Standard.sql– Standard 8.9 Script
   This script was altered as detailed below:
   - Read Only Access to Pages
   - Able to Access buttons on PO Inquiry Page

- Don’t allow the user the ability to revise nVision parameters
- User can run query to HTML
- User can run query to Excel
- User cannot edit Queries
- User can schedule queries
- User can run reports from report menu

Appendix B

1. Issue:
   Change user's access rights to "Read Only" for all Pages they currently have access to in Production. This solution assumes that users will not be doing any kind of security or account maintenance in the Seamless Access environment.

   Solution:

   This is controlled by the DISPLAYONLY flag in the PSAUTHITEM table. A value of "1" means the page is display only. A value of "0" states the page is updateable. We modified the view of PSAUTHITEM table to add select statement to return pages that are flagged as display only.

2. Issue:
   Enable user's access to in-page buttons so users can perform drilldown to detail information (i.e. PO Inquiry)

   Solution:

   The PO Inquiry page by definition is a display only page. To allow the user the ability to click on buttons on the page it must not be set as display only. The view PSAUTHITEM will not set this page to display only.

   Future Inquiry Pages:

   For future pages that are by nature inquiry only, the DISPLAYONLY flag in the PSAUTHITEM view should be set to 0 for these pages. A value of "0" states the page is updateable. The item should be added to the 1st and 2nd select statement in the PSAUTHITEM view as noted below where "PO_INQUIRY" is located.

   CREATE OR REPLACE VIEW &&DATAACCESSSID..PSAUTHITEM (CLASSID, MENUNAME, BARNAME, BARITEMNAME, PNLITEMNAME, DISPLAYONLY, AUTHORIZEDACTIONS) AS

```
select
    a1.CLASSID,
    a1.MENUNAME,
    a1.BARNAME,
    a1.BARITEMNAME,
    a1.PNLITEMNAME,
    1 DISPLAYONLY,
    a1.AUTHORIZEDACTIONS
   from &&ACCESSID..PSAUTHITEM a1
   WHERE a1.BARITEMNAME NOT IN ('PROCESSMONITOR','QUERY_MANAGER',
   'PO_INQUIRY')               -- This Allows the user to click on buttons in PO
Inquiry
   and a1.BARNAME NOT IN ('REPORT')
union
select
    a2.CLASSID,
    a2.MENUNAME,
    a2.BARNAME,
```

```
 a2.BARITEMNAME,
 a2.PNLITEMNAME,
 0 DISPLAYONLY,
 a2.AUTHORIZEDACTIONS
from &&ACCESSID..PSAUTHITEM a2
WHERE a2.BARITEMNAME IN ('PROCESSMONITOR','QUERY_MANAGER',
   'PO_INQUIRY')        -- This Allows the user to click on buttons in PO Inquiry
and a2.BARNAME NOT IN ('REPORT')
```

3. Issue:
   Don't allow the user the ability to revise nVision parameters (Scope, Report Requests).

   Solution:

   nVision pages access rights are set to "Read Only" with the resolution for issue 1.

   Note: Report layouts are not controlled via the 3 tier environment. There is no programmatic way to remove the user's ability to create or modify report layouts. They are controlled via 2 tier installation of nVision.

4. Issue:
   Have a way to turn on/off the user's ability to run a query to HTML.

   Solution:

   This is controlled by the flag QRY_OUT_LISTBOX in the table PS_SCRTY_QUERY. We added a new view (PS_SCRTY_QUERY) based off this table. A value of "N" will prevent the user from being able to run a query to HTML. A value of "Y" will give the user the ability to run to HTML.

5. Issue:
   Have a way to turn on/off the user's ability to run a query to Excel.

   Solution:

   This is controlled by the flag QRY_OUT_NVISION in the table PS_SCRTY_QUERY. We added a new view (PS_SCRTY_QUERY) based off this table. A value of "N" will prevent the user from being able to run a query to Excel. A value of "Y" will give the user the ability to run to Excel.

6. Issue:
   Remove user's ability to create/edit Public or Private Queries. Users will still be able to run all queries that they have access to in Production.

   Solution:

   This is controlled by the flag QRY_RUN_ONLY in the table PS_SCRTY_QUERY. We added a new view (PS_SCRTY_QUERY) based off this table. A value of "Y" will prevent the user from being able to edit or create queries. A value of "N" will give the user the ability to create/edit queries.

7. Issue:
   Have a way to turn on/off user's ability to Schedule a query (thru the Process Scheduler)

   Solution:

   There is no way without a customization to the page to remove the "Schedule Query" Link on the query page. To prevent them from scheduling a query we are setting the field AUTHORIZEDACTIONS in the 3rd sql for the PSAUTHITEM to 0. The users will receive a security error saying they are not authorized to access this component which prevents them from scheduling queries.

CHAPTER 10

Data Archive Transaction Restore API

This chapter includes the following topics:

- Abstract, 39
- Overview, 39
- Solution Overview, 40
- Calling the Transaction Restore API, 41

Abstract

This document describes the Data Archive Transaction Restore API. The Transaction Restore API provides the capability to run the Transaction Restore program from an external source.

Overview

This solution provides a mechanism to call the Data Archive Transaction Restore program from external sources.

List of Terms

The following is a list of terms that are used throughout this document:

Transaction Restore

The process that restores a complete transaction from HISTORY back to the SOURCE instance it was originally archived from.

Restore Definition

A Restore Definition is created in the Data Archive user interface. The Restore Definition contains the entity, interim table, and criteria for restoring data from the HISTORY database to the SOURCE database. The criteria can be any of the following:

- Entire HISTORY database
- A single archive cycle
- Selective criteria like single transaction restore. As it relates to Transaction Restore, the Restore Definition contains the criteria to restore a single transaction.

Program Callback

A program callback is a user-defined, independent program that is called by the Transaction Restore API. This program can perform post-restore tasks deemed necessary.

Solution Overview

This section provides an overview of the features and describes the business benefit to each.

Features

The transaction restore API provides the capability to run the Data Archive Transaction Restore program from an external source. The Transaction Restore program API provides all of the necessary details to run the Transaction Restore program. The API can restore a single transaction in real time.

API Parameters

The API needs to receive the following items:

- Definition ID. This represents a pre-created definition that contains the entity and interim table to be used for the transaction restore. This definition is created by the Archive User.
- Parameter String. A series of carat "^" delimited parameters as a single string that include the following:
  - Transaction Criteria. This represents the unique key of the transaction to be restored. The unique key can be either a primary or a composite key, but it should match the unique key defined in the interim table for the entity selected in the Restore Definition.
  - If a Program Callback is being used and the Program Callback requires input parameters, then additional parameter values are to be included in the Transaction Criteria.

When submitted through the Data Archive UI, the Restore Workbench allows the user to select the Entity, the Interim table, and the selection criteria to use to perform the restore. By pre-creating the Restore Definition, the calling program can hard-code the Restore Definition ID, or a new metadata table can be created that maps specific types of transactions to specific Restore Definitions.

A Program Callback feature allows a secondary program to be executed by the Transaction Restore API itself. The Program Callback can be a PL/SQL stored procedure or a Java program.

The calling program is also responsible for passing the selection criteria to the API. In this case the selection criterion is the specific transaction to be restored. Since all of these values are specific to the archive metadata being used, it is the requirement of the calling program to determine these values, and not the purpose of the API itself to determine these values. If a Program Callback is being used the Program

Callback requires input parameters, then the values for these parameters will be appended to the Transaction Restore selection criteria.

The final input parameter will be an indicator to restore the transaction immediately or as part of batch processing to occur later.

API Processing

Immediate Execution

For immediate execution the Transaction Restore API creates the necessary rows in Informatica's job tables to run Transaction Restore job for the given Restore Definition and parameters. The Informatica Job Scheduler processes the job as if it were submitted through the user interface.

Error Handling

If any errors are encountered during the course of processing in the Transaction Restore API, an error code can be passed to the callback program along with any error text that is appropriate. Also, when the API is called, it immediately returns the "JobID" of the job submitted if it was able to schedule successfully (a positive integer). If the schedule is not successful, it returns negative 1 (-1).

Calling the Transaction Restore API

This section contains a feature/function checklist of the items that make up the solution infrastructure. This list is not intended to reiterate all of the functionality of the release. Instead it contains summary statements of the features.

XA_TRX_RESTORE_API Oracle package

The XA_TRX_RESTORE_API package contains several procedures to allow customers to access the Transaction Restore API via Oracle.

Procedure Submit (Without base_url)

If this procedure is used, the customer is expected to edit the package manually, and modify the base_url variable to enter their own base URL (the URL of the Web interface).

Order	Parameter	In/Out	Type
1	defn_id	In	Number
2	param_string	In	Varchar2
3	job_type	In	Varchar2
4	callback	In	Varchar2
5	schedule_id	In	Varchar2

The following table describes the parameters:

Parameter	Description
defn_id	The ID of the Restore Definition that contains the entity and interim table to be used to restore the transaction.

param_string	The column name and value to be used by the Transaction Restore job to restore the transaction. This is the column or columns that comprise the unique key for the transaction. The syntax for the Transaction Key string is:
	`'CLAIM_ID=12345'`
	If a composite key is being used, then the columns are separated by the caret (^) symbol:
	`'CLAIM_ID=12345^CLAIM_TYPE=987'`
	If the value is a string, then PL/SQL syntax for single quotes must be used:
	`'CLAIM_ID=12345^CLAIM_TYPE=''ABC'''`
	If a Program Callback is being used and the program requires input parameters, then those parameter names and values are to be appended to the Transaction Key:
	`'CLAIM_ID=12345^CLAIM_TYPE=987^CALLBACK_ID=12AB'`

job_type	Type of Job to submit (currently only supports "I" for immediate)

callback	The name of the Program Callback to be executed by the Transaction Restore API. This can be an HTTP or a SQL call. The syntax expected of this parameter is:
	`sql://program_callback_name($param_name1,$param_name2,… , $param_nameX)`
	or
	`http://web.page.to.call/index.html?$param_name1&param_name2`
	When using a SQL callback, the sql statement will be executed as the amhome user in the home instance. When formatting this string, it will take any of your parameters (from the parameter string), and it will take these Informatica provided variables:
	$ScheduleID or JobID
	This refers to the Informatica Unique identifier of the job (as it appears in the Web User Interface).
	$ReturnCode
	This refers to the result of the job that was executed.
	- C = Complete
	- E = Error
	$ReturnText
	If the $ReturnCode = E, this will be additional error text from the job.
	Using these variables, you can have the XA_TRX_RESTORE_API call any package (from the amhome schema).
	For example:
	Callback:
	`sql://call my_callback($JobID, ''$ReturnCode'', ''$ReturnText'', ''$my_id'')`
	Resulting Callback:
	`my_callback(111, 'C', '', 'abc')`

schedule_id	Output parameter with the result of the schedule. - -1 = Error scheduling. - Other = The Job ID of the job scheduled.

Procedure Submit (With base_url)

Order	Parameter	In/Out	Type
1	base_url	In	Varchar2
2	defn_id	In	Number
3	param_string	In	Varchar2
4	job_type	In	Varchar2
5	callback	In	Varchar2
6	schedule_id	In	Varchar2

The following table describes the parameters:

Parameter	Description
base_url	The Base URL of your Web User Interface. Example: `http://testamhome.my.com:8080`
defn_id	The ID of the Restore Definition that contains the entity and interim table to be used to restore the transaction.

param_string	The column name and value to be used by the Transaction Restore job to restore the transaction. This is the column or columns that comprise the unique key for the transaction. The syntax for the Transaction Key string is:
	`'CLAIM_ID=12345'`
	If a composite key is being used, then the columns are separated by the caret (^) symbol:
	`'CLAIM_ID=12345^CLAIM_TYPE=987'`
	If the value is a string, then PL/SQL syntax for single quotes must be used:
	`'CLAIM_ID=12345^CLAIM_TYPE=''ABC'''`
	If a Program Callback is being used and the program requires input parameters, then those parameter names and values are to be appended to the Transaction Key:
	`'CLAIM_ID=12345^CLAIM_TYPE=987^CALLBACK_ID=12AB'`

job_type	Type of Job to submit (currently only supports "I" for immediate)

Parameter	Description

callback	The name of the Program Callback to be executed by the Transaction Restore API, typically a SQL procedure call. The syntax expected of this parameter is:
	`program_callback_name($param_name1,$param_name2,… ,$param_nameX)`
	The program will be executed as the amhome user in the home instance. When formatting this string, it will take any of your parameters (from the parameter string), and it will take these Informatica provided variables:
	$ScheduleID or JobID
	This refers to the Informatica Unique identifier of the job (as it appears in the Web User Interface).
	$ReturnCode
	This refers to the result of the job that was executed.
	- C = Complete
	- E = Error
	$ReturnText
	If the $ReturnCode = E, this will be additional error text from the job.
	Using these variables, you can have the XA_TRX_RESTORE_API call any package(from the amhome schema).
	For example:
	Callback:
	`sql://call my_callback($JobID, ''$ReturnCode'', ''$ReturnText'', ''$my_id'')`
	Resulting Callback:
	`my_callback(111, 'C', '', 'abc')`

schedule_id	Output parameter with the result of the schedule. - -1 = error scheduling - Other = The Job ID of the job scheduled.

Function Submit

Order	Parameter	In/Out	Type
1	base_url	In	Varchar2
2	defn_id	In	Number
3	param_string	In	Varchar2
4	job_type	In	Varchar2
5	callback	In	Varchar2

The Return value of this function is the Job ID of the scheduled job.

The following table describes the parameters:

Parameter	Description
base_url	The Base URL of your Web User Interface. Example: `http://testamhome.my.com:8080`
defn_id	The ID of the Restore Definition that contains the entity and interim table to be used to restore the transaction.

param_string	The column name and value to be used by the Transaction Restore job to restore the transaction. This is the column or columns that comprise the unique key for the transaction. The syntax for the Transaction Key string is:
	`'CLAIM_ID=12345'`
	If a composite key is being used, then the columns are separated by the caret (^) symbol:
	`'CLAIM_ID=12345^CLAIM_TYPE=987'`
	If the value is a string, then PL/SQL syntax for single quotes must be used:
	`'CLAIM_ID=12345^CLAIM_TYPE=''ABC'''`
	If a Program Callback is being used and the program requires input parameters, then those parameter names and values are to be appended to the Transaction Key:
	`'CLAIM_ID=12345^CLAIM_TYPE=987^CALLBACK_ID=12AB'`

job_type	Type of Job to submit (currently only supports "I" for immediate)

callback	The name of the Program Callback to be executed by the Transaction Restore API, typically a SQL procedure call. The syntax expected of this parameter is:
	`program_callback_name($param_name1,$param_name2,… ,$param_nameX)`
	The program will be executed as the amhome user in the home instance. When formatting this string, it will take any of your parameters (from the parameter string), and it will take these Informatica provided variables:
	$ScheduleID or JobID
	This refers to the Informatica Unique identifier of the job (as it appears in the Web User Interface).
	$ReturnCode
	This refers to the result of the job that was executed.
	- C = Complete
	- E = Error
	$ReturnText
	If the $ReturnCode = E, this will be additional error text from the job.
	Using these variables, you can have the XA_TRX_RESTORE_API call any package(from the amhome schema).
	For example:
	Callback:
	`sql://call my_callback($JobID, ''$ReturnCode'', ''$ReturnText'', ''$my_id'')`
	Resulting Callback:
	`my_callback(111, 'C', '', 'abc')`

Job ID (returning value of function)	Output parameter with the result of the schedule. - -1 = error scheduling - Other = The Job ID of the job scheduled.

Web-based URL API

QuickTxnRestore.jsp page

An alternative method for calling the API is through an http call (via a URL).

This URL is `<your base URL>/jsp/QuickTxnRestore.jsp`

The following are the Query String Parameters summary:

Parameter	Required
DefinitionID	Yes
Others, List of UserDefined Parameters	Yes
Callback	No

The following table describes the parameters:

Parameter	Description
DefinitionID	The ID of the Restore Definition that contains the entity and interim table to be used to restore the transaction.

Others, List of UserDefined Parameters	As part of the API, you will supply the column name as a parameter name, and the new value that you wish to use.
	The column name and value to be used by the Transaction Restore job to restore the transaction. This is the column or columns that comprise the unique key for the transaction.
	Example:
	`http://testamhome:8080/jsp/ QuickTxnRestore.jsp?DefinitionID=10&CLAIM_ID=12345`
	If a composite key is being used, then you can specify all of the columns.
	Example:
	`CLAIM_ID=12345&CLAIM_TYPE=987`
	Any of the Parameters used can be used as "tokens" for the Callback, for example:
	`$CLAIM_ID or $CLAIM_TYPE`

Parameter	Description

Callback	The name of the Program Callback to be executed by the Transaction Restore API.
	This can be an HTTP or a SQL call. The syntax expected of this parameter is:
	`sql://program_callback_name($param_name1,$param_name2,… ,$param_nameX)`
	or
	`http://web.page.to.call/index.html?$param_name1&param_name2`
	When using a SQL callback, the sql statement will be executed as the amhome user in the home instance.
	When formatting this string, it will take any of your parameters (from the parameter string), and it will take these Informatica provided variables:
	$ScheduleID or JobID
	This refers to the Informatica Unique identifier of the job (as it appears in the Web User Interface).
	$ReturnCode
	This refers to the result of the job that was executed.
	- C = Complete
	- E = Error
	$ReturnText
	If the $ReturnCode = E, this will be additional error text from the job.
	Using these variables, you can have the XA_TRX_RESTORE_API call any package (from the amhome schema).
	For example:
	Callback:
	`sql://call my_callback($JobID, ''$ReturnCode'', ''$ReturnText'', ''$my_id'')`
	Resulting Callback:
	`my_callback(111, 'C', '', 'abc') Resulting Callback: my_callback(111, 'C', '', 'abc')`

Job ID (returning value of function)	Output parameter with the result of the schedule. - -1 = error scheduling - Other = The Job ID of the job schedule

Security

You can create the following optional config file to restrict access to the Transaction Restore API: txnrestore-api.properties.

Save the file to the \WEB-INF directory.

In this file you can supply a parameter called "validHosts", in which you can supply a comma separated list of hostnames, IP Addresses, or Regular Expressions of either. For Example:

```
validHosts=host1, 192.168.168.97, dev.*
```

In the example above, the following machines have access to the Transaction Restore API:

- Any machine named "host1." (The host name is derived from DNS reverse lookup or from the webserver.)
- The machine with IP address 192.168.168.97.
- Any machine with a name that starts with "dev." (The host name is derived from DNS reverse lookup or from the webserver.)

Note: Effective in version 5.3.3 HotFix 2, the txnrestore-api.properties file is deprecated. Modify the validHosts property in the conf.properties file.

Effective in version 5.3.4, the default security changed. By default, only the machines that you configure as valid hosts in the conf.properties file are authorized to call the JavaServer page API. Previously, if you did not configure valid hosts, then any IP address or host could call the JavaServer page API.

CHAPTER 12

Dropping and Truncating Partitions in Data Archive

This chapter includes the following topics:

- Abstract, 49
- Overview, 49
- When to Drop and Truncate Partitions, 50
- Entity Configuration, 50
- How to Configure an Entity to Truncate or Drop Partitions, 51
- How to Choose Partitions, 53
- How to Drop or Truncate Partitions from Multiple Tables, 55
- How to Drop and Truncate Partitions from Multiple Tables within the Same Entity, 56
- Log Messages, 57

Abstract

This article describes how to configure Data Archive entities to drop and truncate partitions.

Overview

This document describes how to configure a Data Archive entity to drop or truncate partitions from a source table in an Oracle database. You can drop and truncate partitions in Oracle databases that have partitioning enabled. Drop or truncate partitions instead of issuing the DELETE statement against the table. Dropping or truncating table partitions replaces the delete step of the archive or purge process. When you drop or truncate partitions, you remove or disable the delete step for that table. Data Archive quickly removes the data from the source table with little overhead. Dropping or truncating partitions helps improve the Delete from Source performance of the archive and purge process.

When you truncate a partition, Data Archive removes all data from the partition. After you truncate a partition, you can add data to the partition. When you drop a partition, Data Archive permanently removes all data from the partition and removes the partition from the table.

When to Drop and Truncate Partitions

Drop and truncate partitions if the source table has partitioning enabled and you want to archive all records in a partition. If you do not plan to add data to the partition, drop the partition. If you plan to add data to the partition, truncate the partition.

Entity Configuration

Data Archive uses the entity configuration, such as table owner, table name, and partition name, to drop or truncate partitions. In the Enterprise Data Manager (EDM), you add an interim table to an entity to track the partitions that you want to drop or truncate. If you create an entity, configure the entity as a normal entity that archives or purges data from transaction tables.

For example, the following figure shows a normal Data Archive entity that has steps that perform candidate generation, archive, and purge:

General Information | Parameters | Steps

Order	Action	Interim table	Entity table	Procedure	Business Ru...	Enabled
1	Drop Interim Table	XC_347_ARC53_ST1				☑
2	Create Interim Table	XC_347_ARC53_ST1				☑
3	Insert Interim Table	XC_347_ARC53_ST1				☑
4	Insert into Archive T...	XC_347_ARC53_ST1	SUBSRPTN_DIM_HIST			☑
5	Delete Apps tables	XC_347_ARC53_ST1	SUBSRPTN_DIM_HIST			☑

The following figure shows a normal Data Archive entity that has an interim table that identifies the transactions that are eligible to archive or purge:

General Information | Tables | Default Columns | Business Rules | Indexes | Report Statements

Order	Name	Type	Length	PK	Select clause
1	ID	Number		☑	A.ID
2	PURGEABLE_FLAG	Char	1	☐	
3	STATS_DATE	Date		☐	A.TRX_DATE
4	ORG_ID	Number	15	☐	

Distinct	From clause	Where clause
☐	SUBSRPTN_DIM_HIST A	A.TRX_DATE <= :p_archive_date

The following figure shows a normal Data Archive entity that has managed tables that are linked to the interim table to archive and purge data:

General Information | Tables | Default Columns | Business Rules | Indexes | Report Statements

Order	Name	Insert statement	Delete statement	Update statement
1	SUBSRPTN_DIM_HIST	A.id IN (SELECT X.id FROM...	A.id IN (SELECT X.id FROM...	

How to Configure an Entity to Truncate or Drop Partitions

1. Remove or disable the Delete from Table steps from the entity.

   The following figure shows the Remove Delete Apps Table step:

   General Information | Parameters | Steps

Order	Action	Interim table	Entity table	Procedure	Business Ru...	Enabled
1	Drop Interim Table	XC_347_ARC53_ST1				☑
2	Create Interim Table	XC_347_ARC53_ST1				☑
3	Insert Interim Table	XC_347_ARC53_ST1				☑
4	Insert into Archive T...	XC_347_ARC53_ST1	SUBSRPTN_DIM_HIST			☑

2. Add an interim table to identify the partitions that you want to process.

   The interim table identifies the partitions that you want to process. This interim table must have the following columns:

   - TABLE_OWNER
   - TABLE_NAME
   - PARTITION_NAME
   - PURGEABLE_FLAG

   The interim table is populated with the table_owner, table_name, and partition_name from ALL_TAB_PARTITIONS. The following figure shows an interim table with the required columns:

   General Information | Tables | Default Columns | Business Rules | Indexes | Report Statements

Order	Name	Type	Length	PK	Select clause
1	TABLE_OWNER	Varchar2	30	☐	A.TABLE_OWNER
2	TABLE_NAME	Varchar2	30	☐	A.TABLE_NAME
3	PARTITION_NAME	Varchar2	30	☐	A.PARTITION_NAME
4	PURGEABLE_FLAG	Char	1	☐	'Y'
5	STATS_DATE	Date		☐	
6	ORG_ID	Number	15	☐	

Distinct	From clause	Where clause
☐	all_tab_partitions A	A.table_name='TABLE_WITH_PARTITIONS' and A.table_o...

   The following figure shows the select statement:

   From clause

```
all_tab_partitions A
```

   Where clause

```
A.table_name='TABLE_WITH_PARTITION' and A.table_owner='APPS'
```

   OK | Cancel

3. Add an entity step to process the partition. In the Steps tab, add a new row to call the partition program logic. Configure the following settings for the step:

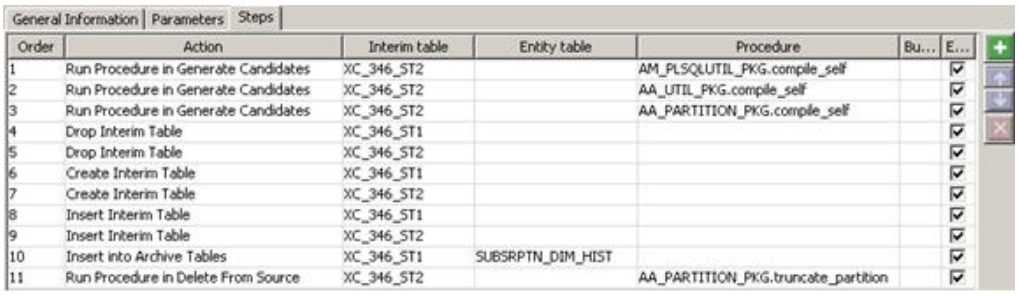
   - Order. Move this step after all "Insert into Archive Tables" steps.

- Action. Choose "Run Procedure in Delete from Source."
- Interim Table. Choose the interim table that you created in Step 1.
- Procedure. Enter AA_PARTITION_PKG.truncate_partition to truncate the partitions or AA_PARTITION_PKG.drop_partition to drop the partitions.

4. Add the following steps to compile the AA_PARTITION_PKG and its dependent packages, AA_UTIL_PKG, and AM_PLSQLUTIL_PKG for the Staging User.

Step	Option	Action
Step	Option	Action
Step 1	Order	Move this step to Order 1.
Step 1	Action	Choose Run Procedure in Generate Candidates.
Step 1	Interim table	Choose the interim table that you created in Step 1.
Step 1	Procedure	Enter AM_PLSQLUTIL_PKG.compile_self
Step 2	Order	Move this step to Order 2.
Step 2	Action	Choose Run Procedure in Generate Candidates.
Step 2	Interim table	Choose the interim table that you created in Step 1
Step 2	Procedure	Enter AA_UTIL_PKG.compile_self.
Step 3	Order	Move this step to Order 3.
Step 3	Action	Choose Run Procedure in Generate Candidates.
Step 3	Interim table	Choose the interim table that you created in Step 1.
Step 3	Procedure	Enter AA_PARTITION_PKG.compile_self.

The following figure shows the required steps:


General Information | Parameters | Steps

Order	Action	Interim table	Entity table	Procedure	Bu...	E...
1	Run Procedure in Generate Candidates	XC_346_ST2		AM_PLSQLUTIL_PKG.compile_self		☑
2	Run Procedure in Generate Candidates	XC_346_ST2		AA_UTIL_PKG.compile_self		☑
3	Run Procedure in Generate Candidates	XC_346_ST2		AA_PARTITION_PKG.compile_self		☑
4	Drop Interim Table	XC_346_ST1				☑
5	Drop Interim Table	XC_346_ST2				☑
6	Create Interim Table	XC_346_ST1				☑
7	Create Interim Table	XC_346_ST2				☑
8	Insert Interim Table	XC_346_ST1				☑
9	Insert Interim Table	XC_346_ST2				☑
10	Insert into Archive Tables	XC_346_ST1	SUBSRPTN_DIM_HIST			☑
11	Run Procedure in Delete From Source	XC_346_ST2		AA_PARTITION_PKG.truncate_partition		☑

How to Choose Partitions

The above example inserts all partitions for a table into the interim table. Typically, you will not want to drop or truncate all partitions in the table. By using the functionality inherent in Candidate Generation, you can control the partitions you want to process. You can control the partitions by one of the following methods:

- Hard code the partition names.
- Parameterize the partition names.
- Use the partitioning key high value to determine if a partition qualifies for processing.

Hard Code the Partition Names

The simplest way to control the partitions that get processed by the entity is to hard code the partition names:

```
A.table_name='TABLE_WITH_PARTITIONS' and A.table_owner='APPS' and partition_name =
'PARTITION_TO_BE_DROPPED'
```

However, this is not re-usable and requires that you update the entity every time you run the archive or purge definition.

Parameterize the Partition Names

Alternatively, you can parameterize the partition names:

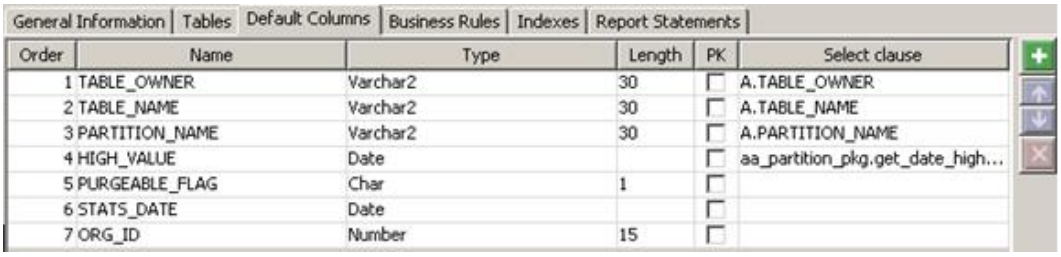
```
A.table_name='TABLE_WITH_PARTITIONS' and A.table_owner='APPS' and partition_name IN
(:p_partition_name_list)
```

While the entity does not need to be updated, the archive definition does need to be updated prior to every run.

Use the Partitioning Key High Value

The most effective set up of the partition interim table, therefore, is to use the partitioning key high value to determine if a partition qualifies for processing. The high value for each partition can be stored in the interim table, and then you can build business rules to validate the high value of the partition meets the archive criteria.

For example, the following figure shows how you can store the high value in the interim table if the partitioning key is a date column:



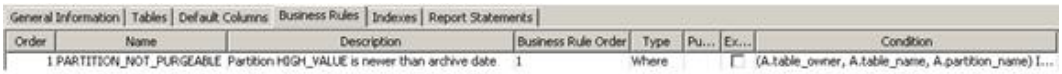
Order	Name	Type	Length	PK	Select clause
1	TABLE_OWNER	Varchar2	30	☐	A.TABLE_OWNER
2	TABLE_NAME	Varchar2	30	☐	A.TABLE_NAME
3	PARTITION_NAME	Varchar2	30	☐	A.PARTITION_NAME
4	HIGH_VALUE	Date		☐	aa_partition_pkg.get_date_high...
5	PURGEABLE_FLAG	Char	1	☐	
6	STATS_DATE	Date		☐	
7	ORG_ID	Number	15	☐	

The following figure shows the select clause:

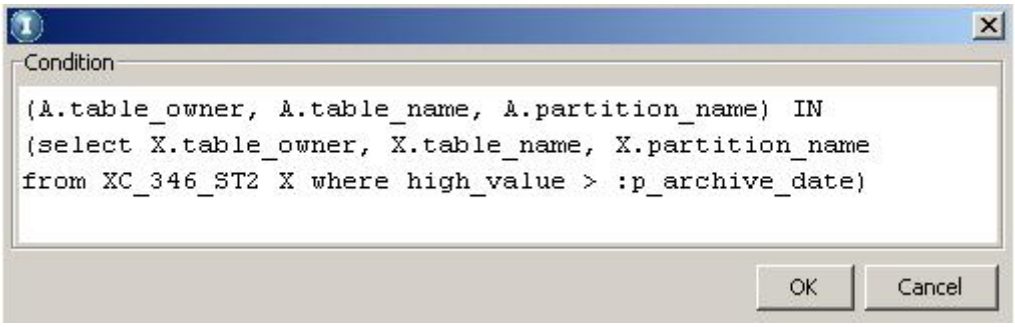
Note that the high value is populated by using a function, AA_PARTITION_PKG.GET_DATE_HIGH_VALUE. The high_value column in the ALL_TAB_PARTITIONS table is a LONG data type. This limits the ability to query and evaluate the high_value column directly from ALL_TAB_PARTITIONS. The function retrieves the high value and translates the value into a date. The high value of a partition may be:

```
TO_DATE(' 2007-12-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
```

The function translates this value into 01-DEC-2007. The following figure shows how you can then build a business rule to validate that the partition key is less than the archive date:

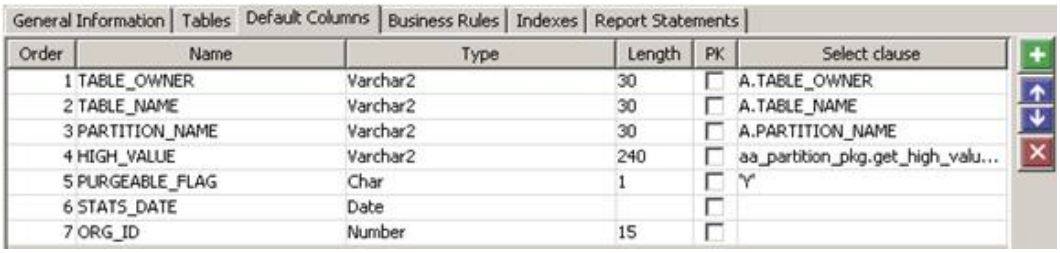


The following figure shows the condition:



This rule disqualifies any partitions where the high value date is greater than the archive date.

If the partition key is based on a varchar or number column, high value can be retrieved and stored in the interim table using the function, AA_PARTITION_PKG.GET_HIGH_VALUE.



The following figure shows the select clause:

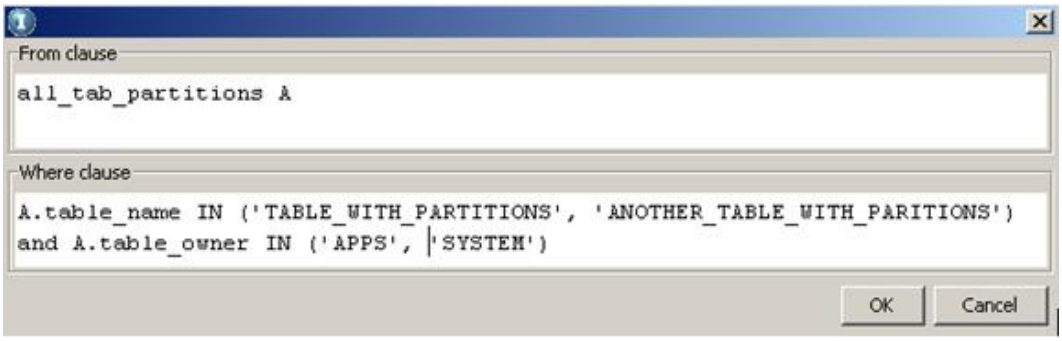
You can now build a business rule to validate the high value that meets the archive or purge criteria. The following syntax illustrates this:

```
(A.table_owner, A.table_name, A.partition_name) IN (select X.table_owner, X.table_name,
X.partition_name from XC_346_ST2 X where high_value < :p_archive_partition_key_value)
Where :p_archive_partition_value is a value determined by the user at run-time.
(A.table_owner, A.table_name, A.partition_name) IN (select X.table_owner, X.table_name,
X.partition_name from XC_346_ST2 X where high_value between :p_range_low
and :p_range_high )
(A.table_owner, A.table_name, A.partition_name) IN (select X.table_owner, X.table_name,
X.partition_name from XC_346_ST2 X where high_value IN ('AR','AP','GL')
```

How to Drop or Truncate Partitions from Multiple Tables

To drop or truncate partitions from multiple tables, use one interim table. Since the interim table is populated with the table_owner, table_name, and partition_name, you can insert partitions from multiple tables into this single interim table.

The following figure shows the select statement:



Note that there is only one entity step to drop or truncate the partitions. The AA_PARTITION_PKG procedure truncates or drops partitions for all tables in the interim table where purgeable_flag='Y'.

The following figure shows the entity steps:

General Information | Parameters | Steps

Order	Action	Interim table	Entity table	Procedure	Bu...	E...
1	Run Procedure in Generate Candidates	XC_346_ST2		AM_PLSQLUTIL_PKG.compile_self		☑
2	Run Procedure in Generate Candidates	XC_346_ST2		AA_UTIL_PKG.compile_self		☑
3	Run Procedure in Generate Candidates	XC_346_ST2		AA_PARTITION_PKG.compile_self		☑
4	Drop Interim Table	XC_346_ST1				☑
5	Drop Interim Table	XC_346_ST2				☑
6	Create Interim Table	XC_346_ST1				☑
7	Create Interim Table	XC_346_ST2				☑
8	Insert Interim Table	XC_346_ST1				☑
9	Insert Interim Table	XC_346_ST2				☑
10	Insert into Archive Tables	XC_346_ST1	SUBSRPTN_DIM_HIST			☑
11	Insert into Archive Tables	XC_346_ST1	SUBSRPTN_DIM_HIST_BAK			☑
12	Run Procedure in Delete From Source	XC_346_ST2		AA_PARTITION_PKG.truncate_partition		☑

How to Drop and Truncate Partitions from Multiple Tables within the Same Entity

If a single entity requires that partitions from one table or from a set of tables are dropped and partitions from another table or set of tables are truncated, then create two interim tables. One interim table manages the dropping of partitions. The second interim table manages the truncating of partitions.

Interim One - Drop Partitions

The following figure shows an entity with an interim table that drops partitions:

The following figure shows the entity steps:

The interim table XC_345_ARC53_ST2 is used to track the tables with partitions that will be dropped. The interim table is selected for the Run Procedure step that executes the Drop Partition logic `AA_PARTITION_PKG.DROP_PARTITION`.

Interim Two - Truncate Partitions

The following figure shows an entity with an interim table that truncates partitions:

The following figure shows the entity steps:

The interim table XC_345_ARC53_ST3 is used to track the tables with partitions that will be truncated. The interim table is selected for the Run Procedure step that executes the Truncate Partition logic `AA_PARTITION_PKG.TRUNCATE_PARTITION`.

The following figure shows the entity steps:

General Information | Parameters | Steps

Order	Action	Interim table	Entity table	Procedure	Business ...	Enabled
1	Run Procedure in Generate Candi...	XC_345_ARC53_ST2		AM_PLSQLUTIL_PKG.compile_self		☑
2	Run Procedure in Generate Candi...	XC_345_ARC53_ST2		AA_UTIL_PKG.compile_self		☑
3	Run Procedure in Generate Candi...	XC_345_ARC53_ST2		AA_PARTITION_PKG.compile_self		☑
4	Drop Interim Table	XC_345_ARC53_ST1				☑
5	Drop Interim Table	XC_345_ARC53_ST2				☑
6	Drop Interim Table	XC_345_ARC53_ST3				☑
7	Create Interim Table	XC_345_ARC53_ST3				☑
8	Create Interim Table	XC_345_ARC53_ST1				☑
9	Create Interim Table	XC_345_ARC53_ST2				☑
10	Insert Interim Table	XC_345_ARC53_ST1				☑
11	Insert Interim Table	XC_345_ARC53_ST2				☑
12	Insert Interim Table	XC_345_ARC53_ST3				☑
13	Insert into Archive Tables	XC_345_ARC53_ST1	SUBSRPTN_DIM_HIST			☑
14	Insert into Archive Tables	XC_345_ARC53_ST1	SUBSRPTN_DIM_HIST_BAK			☑
15	Run Procedure in Delete From Sou...	XC_345_ARC53_ST2		AA_PARTITION_PKG.drop_partition		☑
16	Run Procedure in Delete From Sou...	XC_345_ARC53_ST3		AA_PARTITION_PKG.truncate_partition		☑

Log Messages

AA_PARTITION_PKG logs messages to the Delete From Source job log during processing. Both Job Completion and Exception Handling messages are logged by the package.

Normal Completion

If the dropping or truncating of partitions is successful, the log shows the tables and partitions that were processed.

The following code shows an example of the log file:

```
30-Jun-2010 10:19:05CDT  Calling staging api {call AA_PARTITION_PKG.drop_partition(?,?)}
30-Jun-2010 10:19:06CDT   Dropping Partition ....: P20071101 for table
ARC53_STAGING.SUBSRPTN_DIM_HIST
30-Jun-2010 10:19:06CDT   Dropping Partition ....: P20071201 for table
ARC53_STAGING.SUBSRPTN_DIM_HIST
30-Jun-2010 10:19:06CDT   Dropping Partition ....: P20080101 for table
ARC53_STAGING.SUBSRPTN_DIM_HIST
30-Jun-2010 10:19:06CDT  Drop Partition Completed for Table ....: SUBSRPTN_DIM_HIST

30-Jun-2010 14:23:48CDT  Calling staging api {call
AA_PARTITION_PKG.truncate_partition(?,?)}
30-Jun-2010 14:24:35CDT   Truncating Partition ....: P20071201 for table
ARC53_STAGING.SUBSRPTN_DIM_HIST
30-Jun-2010 14:25:22CDT   Truncating Partition ....: P20080101 for table
ARC53_STAGING.SUBSRPTN_DIM_HIST
30-Jun-2010 14:25:22CDT  Truncate Partition Completed for Table ....: SUBSRPTN_DIM_HIST
```

If there are no partitions to be dropped or truncated, the job completes successfully and there are no tables or partitions listed in the log.

The following code shows an example of the log file:

```
30-Jun-2010 11:31:18CDT  Calling staging api {call AA_PARTITION_PKG.drop_partition(?,?)}
30-Jun-2010 11:31:18CDT  Drop Partition Completed for Table ....:

30-Jun-2010 11:31:19CDT  Calling staging api {call
AA_PARTITION_PKG.truncate_partition(?,?)}
30-Jun-2010 11:31:19CDT  Truncate Partition Completed for Table ....:
```

Exception Handling

AA_PARTITION_PKG ensures that all data in the partition to be dropped or truncated has been archived into a staging table. If a row in the partition is not archived, an error is thrown and written to the log.

The following log shows the error message:

```
30-Jun-2010 10:37:38CDT  Calling staging api {call
AA_PARTITION_PKG.truncate_partition(?,?)}
30-Jun-2010 10:37:39CDT  ERROR: Not all rows archived from partition:
ARC53_STAGING.SUBSRPTN_DIM_HIST PARTITION (P20071101)
30-Jun-2010 10:37:39CDT  Number of Rows in Partition: 3
30-Jun-2010 10:37:39CDT  Number of Rows Not Archived: 3
```

CHAPTER 14

How to Create Business Rules to Archive and Purge Transactional Data

This chapter includes the following topics:

- Abstract, 59
- Overview, 59
- Candidate Generation Process, 60

Abstract

One of the main features of the Data Archive product is the Candidate Generation functionality. Candidate Generation identifies the transactions that can be archived and purged. Additionally, it identifies the transactions that are ineligible for purge. Candidate Generation uses business rules to determine if a transaction can or cannot be archived and purged.

This document explains the different ways to build business rules to meet your business requirements to archive and purge your older, inactive transactional data.

Overview

Candidates are determined by the data retention policy. For example, a customer may decide to archive everything older than 3 years. So, candidate generation will identify all transactions older than 3 years. These are "candidates" for archive/purge.

Once the candidates have been identified, the next step is to determine if a candidate can be archived and purged - that is, does the candidate pass the business rule(s) for archiving and purging. The end result of Candidate Generation is that it provides a list of transactions that can be archived and purged. This list is used by the engine to archive (copy to stage) and purge (delete from source) the data.

The power of candidate generation is that it tells us WHY a transaction cannot be archived and purged - that is, it tells us what business rules a transaction failed. This helps us in a few ways. In particular we focus on

rules where a majority of transactions failed. For example, if 80% of invoices failed the INVOICE IS UNPAID rule, then we can surmise:

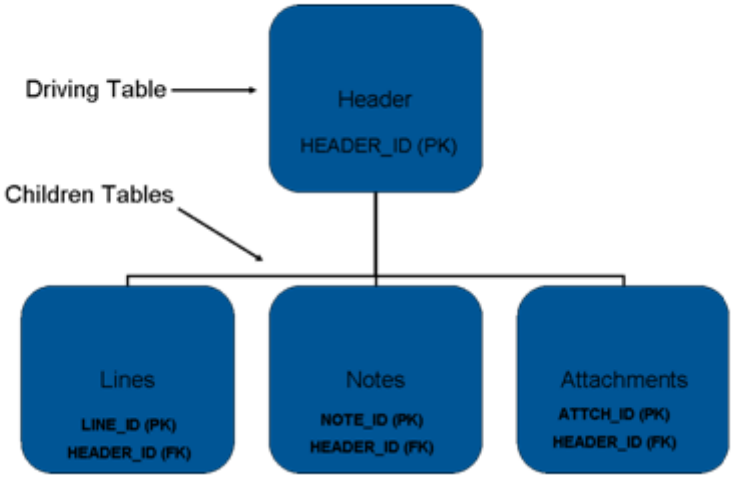
- Maybe this data was converted incorrectly.
- Maybe the customer doesn't PAY invoices in this system.
- Maybe a patch was applied that adversely affected the PAYMENT functionality.
- Maybe the PAYMENT functionality was implemented incorrectly for this timeframe and not corrected until later.

Identifying these types of causes allows us to MODIFY the rule in lieu of modifying the data. Our preference is to never modify data to pass a business rule.

Candidate Generation Process

The previous section stated that the end result of Candidate Generation is a "list of transactions that can be archived and purged". First, what is a transaction? A transaction is a functional business object like an Invoice or a Payment or a Sales Order. In a relational database, the transaction is represented as a series of tables. If we think of a database entity relationship diagram, there is a table at the "top" of the model. This is the driving table. In a relational database, "children" tables are related to the driving table through "foreign key" relationships.

The diagram below illustrates how a transaction may be represented in a relational database:



Second, the list of candidate transactions is actually a table in the database. This table is called an "interim table". The table contains the primary key (PK) of the driving table. Using the PK, Data Archive can identify the related pieces of the transaction (e.g., the Lines, Notes and Attachments in the diagram above). Data Archive uses the interim table to perform the archive and purge of all the transaction tables.

Finally, candidate generation also tells why transactions cannot be archived and purged. This information is also stored in the interim table. For each transaction the interim table tracks what business rules a transaction has passed or failed. The results of the business enforcement are tracked as columns in the interim table.

An interim table may be constructed as follows:

HEADER_ID	PURGEABLE_FLAGHAS_INVALID_PAY_STATUS	HAS_NEGOTIABLE_CHECKS

Business Rules

A business rule for Data Archive is the technical representation of a functional requirement. If the rule is the INVOICE IS NOT CLOSED, then there is a field in a table that tells us that an invoice is not closed. Note that rule is INVOICE IS NOT CLOSED as opposed to INVOICE IS CLOSED. Business rules in Data Archive identify transactions that cannot be archived and purged. So the business rules are looking for transaction that FAIL rules. This is important to remember when creating business rules.

As mentioned previously, the enforcement of business rules are tracked in columns in the interim table. That is, each business rule is represented as a column in the interim table. If a sample business rule is INVOICE IS NOT CLOSED, the column in the interim table may be "INVOICE_IS_NOT_CLOSED". We also just described how business rules identify transactions that cannot be archived or purged. If a transaction fails the business rule, then the business rule column is updated with a 1. If a transaction passes the business rule, then the business rule column is updated with a 0. A business rule, therefore, is a SQL UPDATE statement. The UPDATE statement updates transactions in the interim table according to the business rule criteria. Specifically, the UPDATE statement affects transactions that have failed the business rule.

Candidate Generation Steps

The candidate generation process performs the following steps:

1. Create Interim Table.
2. Insert Interim Table with candidate transactions based on archive criteria (e.g., invoice_date < 12/31/2005).
3. Update Interim Table with results of business rule.

The business rule columns in the interim table are updated separately. That is, each business rule updates its business rule column separately. Consequently, business rules tend to be mutually exclusive, and the results of one business rule do not affect the results of another business rule.

Implementing Business Rules

Business Rule Types

When using EDM business rules can be constructed using one of five methods:

- WHERE
- SET
- INLINE
- LOOP
- FULL

In this section we will describe how to create a business rule using each type, and why one type would be used versus another.

To implement business rules you must first determine what the functional business requirement is, and then identify what the technical representation of that requirement is. The technical representation of the requirement is a SQL statement.

In our example the business rule is INVOICE IS NOT CLOSED. The technical representation of this requirement may be:

```
select invoice_header_id from ap_invoice_header where invoice_status <> ' CLOSED';
```

Our sample interim table would have the following columns:

INVOICE_HEADER_ID	PURGEABLE_FLAGINVOICE_IS_NOT_CLOSED

We will use this example to describe how to use the different condition types.

The WHERE Condition Type

Using our sample business rule, the SQL UPDATE statement for the WHERE condition type would be executed as:

```
UPDATE interim A
SET INVOICE_NOT_CLOSED = 1
WHERE A.invoice_header_id IN
(SELECT H.invoice_header_id
FROM ap_invoice_header H
WHERE H.invoice_status <> ' CLOSED');
```

In this case the technical representation of the business rule is the sub-select statement that is part of the WHERE clause for the UPDATE statement. This is an example of the "WHERE" condition type when defining business rules in EDM.

However, note that the sub-select queries the entire AP_INVOICE_HEADER table. This has two implications: one, this could be a performance issue to query the entire invoice header table; and two, it is not necessary to query the entire invoice header table, as we are only interested in the invoices that are in the interim table. As a best practice, the SQL UPDATE statement would be written as:

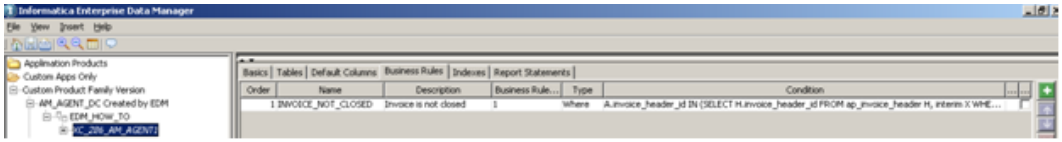
```
UPDATE interim A
SET INVOICE_NOT_CLOSED = 1
WHERE A.invoice_header_id IN
(SELECT H.invoice_header_id
FROM ap_invoice_header H, interim X
WHERE H.invoice_header_id=X.invoice_header_id
AND H.invoice_status <> ' CLOSED');
```

By joining to the interim table we can focus the sub-select query on the ap_invoice_header transactions that are candidates for archive/purge.

Therefore, the business rule INVOICE IS NOT CLOSED for a WHERE condition type would be written as:

```
A.invoice_header_id IN
(SELECT H.invoice_header_id
FROM ap_invoice_header H, interim X
WHERE H.invoice_header_id=X.invoice_header_id
AND H.invoice_status <> ' CLOSED')
```

The WHERE condition type, therefore, represents the WHERE clause portion of the UPDATE statement. This part of the statement is entered into EDM into the Condition field:



The WHERE condition is the simplest condition type to use, and it should be used at all times whenever possible.

The Set Condition Type

The SQL UPDATE statement for the SET condition type is written as follows:

```
UPDATE interim A
SET INVOICE_NOT_CLOSED = (SELECT DECODE(H.invoice_status,'CLOSED',0,1)
FROM ap_invoice_header H
WHERE H.invoice_header_id=A.invoice_header_id
AND H.invoice_status <> ' CLOSED');
```

The SET condition type is the SET clause of the UPDATE SQL statement. In EDM this is written as:

```
(SELECT DECODE(H.invoice_status,'CLOSED',0,1)
FROM ap_invoice_header H
WHERE H.invoice_header_id=A.invoice_header_id
AND H.invoice_status <> ' CLOSED')
```

There are two things to note about how this SQL statement is constructed. First, a DECODE is used to enforce the rule. DECODE (in Oracle, CASE using ANSI SQL) allows you to use IF-ELSE logic. IF invoice_status = 'CLOSED', then set INVOICE_NOT_CLOSED to 0, which means it passed the rule. ELSE if invoice_status equals any other value, then set INVOICE_NOT_CLOSED to 1, which means it failed the rule. Second, a join to the interim table is included to ensure each row in the interim is updated - H.invoice_header_id=A.invoice_header_id.

The main difference between SET and WHERE is that SET updates every single row in the interim table whereas WHERE updates only the rows returned by the condition statement. For this reason the WHERE condition type is typically used in lieu of the SET condition type.

The INLINE Condition Type

The INLINE condition type allows you to evaluate a condition in the driving table without having to run an UPDATE WHERE statement (which is extra processing) or without adding a restrictive Where clause statement on INSERT. For example, our sample rule checks the invoice_status field in the ap_invoice_headers table. The ap_invoice_headers table is also our driving table, which is used to populate the interim table with the list of candidates eligible for archive/purge. We could enforce this business rule as we have already demonstrated with the WHERE or SET condition types. Or we could enforce the business requirement upon insert into the interim table:

```
INSERT INTO INTERIM A
(INVOICE_HEADER_ID)
SELECT invoice_header_id
FROM ap_invoice_headers H
WHERE invoice_date < :p_period_to
AND invoice_status='CLOSED'
```

This would populate the interim table with all invoice header transactions that had an invoice_status of CLOSED, which meets the business requirement. This also allows us to not have to run an UPDATE statement using the WHERE or SET condition types, thereby eliminating the need for a potentially time-consuming SQL statement. However, this does not identify all of the invoice header transactions that do not have an invoice status of CLOSED. It is useful information to know what transactions failed a rule. By adding business

requirements to the WHERE clause of the INSERT statement, the ability to obtain this information is not possible.

However, using the INLINE condition type, we can take advantage of the INSERT statement to meet the business requirement while also providing the exception reporting capability that is critical to Candidate Generation. The INLINE condition type is written as follows:

```
INSERT INTO INTERIM A
(INVOICE_HEADER_ID, INVOICE_NOT_CLOSED)
SELECT invoice_header_id , DECODE(H.invoice_status,'CLOSED',0,1)
FROM ap_invoice_headers H
WHERE invoice_date < :p_period_to
```

The INLINE condition type is added to the SELECT statement used to INSERT into the interim table. Like the SET condition type, the INLINE condition type uses a DECODE (or CASE) statement to evaluate the column value and selects a 0 or 1 accordingly. In EDM the INLINE condition is written as:

```
DECODE(H.invoice_status,'CLOSED',0,1)
```

Note that the invoice_status column is aliased with an H. It is good practice to alias the columns in the INLINE in case there are multiple tables in the INSERT FROM clause. This will ensure you are looking at the correct column in the correct table.

The LOOP Condition Type

The loop business type allows for recursive validation to be enforced. That is, the same business rule logic can be applied over and over again. The reason you would want to loop a business rule over and over again is to deal with transaction chains, which describe many to many relationships between transactions. An example of a one to many relationship would be header table to lines table. One header record has many lines. An example of many to many would be the order to cash or procure to pay transaction chains:

- One req can create many POs
- Many reqs can create one POs
- Many reqs can create many POs

The same many-to-many relationship is true for POs-Invoices and Invoice-Payments. So, a complex business process like Procure to Pay can involve many transactions:

Many reqs create many POs, many POs create many Invoices, many Invoices create many Payments.

For some transaction chains, it is necessary to keep chains together from an archive and purge perspective. Although these are discrete transactions with its own set of business requirements, the business application requires the full chain to be present to function properly. If parts of the chain are missing, the business application cannot operate as expected. The loop rule type, therefore, exists solely to keep transaction chains together - either they get archived together or they stay in production together. The logic of the loop business rules identifies "links" in the chain that cannot be purged. The recursive looping is necessary to identify all the links in the chain that cannot be purged.

The first step in the transaction chain process is to build an interim table that pieces the transaction chain together. In this example we are using the Procure to Pay transaction chain. Individual interim tables will exist for each discrete transaction: Req, PO, Voucher, Payment. Each discrete transaction will have its own set of rules to pass. Once business rules have been enforced for the discrete transactions, then you build the transaction chain interim table. The transaction chain interim is built through a combination of the discrete transaction interim tables and the source table cross-reference tables. Examples of a cross-reference table would be PS_VOUCHER_PAYMENT_XREF in Peoplesoft or AR_RECEIVABLE_APPLICATIONS_ALL in Oracle Apps.

An example of a Procure to Pay transaction chain interim table may be:

TRX CHAIN INTERIM

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10A	100A	1000A	Y
1A	10B	100A	1000A	N
1B	10A	100B	1000A	Y
1B	10A	100C	1000B	Y
1C	10C	100D	1000D	Y

When the transaction chain interim is inserted, the purge_flag is set at that time. Therefore, each link in the chain must be marked as purgeable in the individual interim tables for the purge_flag for the chain to be marked as 'Y'.

Let's look at the example above to see if we can piece a chain together.

- Req 1A created PO 10A.
- PO 10A is matched to Voucher 100A.
- Voucher 100A was paid by Payment 1000A.
- Req1 1A also created PO 10B which is also matched to Voucher 100A.
- We also see that PO 10A was sourced from Req 1B and it was matched to VCHR 100B and VHCR 100C.

So if one of these discrete transactions cannot be archived and purged, all of the related transactions cannot be archived and purged as well. In the example above let's say PO 10B is not purgeable. Since 10B is not purgeable, that "row" of the chain is not purgeable. What is the effect of this part of the chain on the rest of the chain? How will the LOOP business type help us here? Let's step through the recursive logic to see how the LOOP condition statements will navigate through a transaction chain.

As previously stated, the first step in the transaction chain logic is to build the transaction chain interim. Each row in the interim lists every related transaction.

TRX CHAIN INTERIM

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10A	100A	1000A	Y
1A	10B	100A	1000A	N
1B	10A	100B	1000A	Y
1B	10A	100C	1000B	Y
1C	10C	100D	1000D	Y

Like all of the other Business Rule types, the condition statement for the LOOP type affects a single column in the transaction chain interim table. Therefore, a LOOP business rule exists for each column in the transaction chain interim. The LOOP business rule is constructed as such:

```
UPDATE XA_PO_CHAIN_INTERIM
   SET purgeable_flag='N'
 WHERE A.PO_ID IN (SELECT X.PO_ID FROM XA_PO_CHAIN_INTERIM X
                    WHERE X.purgeable_flag ='N')
   AND A.purgeable_flag = 'Y'
```

The logic of the rule is simple - find POs where the purgeable_flag is 'N' and update rows where that PO has a purgeable_flag = 'Y'. So let's look at the chain again. Which row would this statement update? It would update 0 rows. Why? Because it's looking for other rows for PO 10B that have a purgeable flag of Y. There aren't any more rows for PO 10B.

TRX CHAIN INTERIM

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10A	100A	1000A	Y

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10B	100A	1000A	N
1B	10A	100B	1000A	Y
1B	10A	100C	1000B	Y
1C	10C	100D	1000D	Y

This is the condition statement for Reqs. It is the same logic but for Reqs instead of POs.

```
UPDATE XA_PO_CHAIN_INTERIM
   SET purgeable_flag='N'
 WHERE A.REQ_ID IN (SELECT X.REQ_ID FROM XA_PO_CHAIN_INTERIM X
                    WHERE X.purgeable_flag ='N')
   AND A.purgeable_flag = 'Y'
```

Which rows would this statement update? It would update the first row which has Req 1A and purgeable_flag='Y'. So that purgeable flag changes from Y to N.

TRX CHAIN INTERIM

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10A	100A	1000A	N
1A	10B	100A	1000A	N
1B	10A	100B	1000A	Y
1B	10A	100C	1000B	Y
1C	10C	100D	1000D	Y

Now let's look at vouchers. Again, the logic is the same - find vouchers where the purgeable_flag is 'N' and update rows where that voucher has a purgeable_flag = 'Y'. Which row would this statement update?

```
UPDATE XA_PO_CHAIN_INTERIM
   SET purgeable_flag='N'
 WHERE A.VCHR_ID IN (SELECT X.VCHR_ID FROM XA_PO_CHAIN_INTERIM X
                     WHERE X.purgeable_flag ='N')
   AND A.purgeable_flag = 'Y'
```

It would update 0 rows. Why? Because it's looking for other rows for Voucher 100A that have a purgeable flag of Y. There aren't any more rows for Voucher 100A that have a purgeable flag = Y.

TRX CHAIN INTERIM

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10A	100A	1000A	N
1A	10B	100A	1000A	N
1B	10A	100B	1000A	Y
1B	10A	100C	1000B	Y
1C	10C	100D	1000D	Y

For Payments the same logic is used - find payments that are not purgeable and mark other rows as not purgeable where those payments are purgeable. Which rows would this statement update?

```
UPDATE XA_PO_CHAIN_INTERIM
   SET purgeable_flag='N'
 WHERE A.pymnt_ID IN (SELECT X.pymnt_ID FROM XA_PO_CHAIN_INTERIM X
                      WHERE X.purgeable_flag ='N')
   AND A.purgeable_flag = 'Y'
```

It would update the third row which has Payment 1000A and purgeable_flag='Y'

At this point the LOOP logic has completed its first iteration. Now it "loops" and runs all 4 updates again. It keeps doing this until all 4 update statements update 0 rows. So in the second iteration, what rows will be updated?

The Req update will update row 4. Req 1B has a purgeable flag of 'Y'

TRX CHAIN INTERIM

REQ	PO	VCHR	PYMNT	PURGE_FLAG
1A	10A	100A	1000A	N
1A	10B	100A	1000A	N
1B	10A	100B	1000A	N
1B	10A	100C	1000B	N
1C	10C	100D	1000D	Y

Will the remaining LOOP statement update any rows? No. Will the LOOP logic loop again? Yes because the first LOOP update updated 1 row. So now we do a third iteration. Will any rows be updated in the 3rd iteration? No. Will the LOOP logic stop after the 3rd iteration? Yes because all 4 LOOP statement updated 0 rows.

So when the loop logic runs, it runs all of the updates one after the other. It loops back to the top and runs them again. It keeps looping until all updates update 0 rows. The number of iterations is an indication of the complexity of the transaction chain. The higher the number of iterations means that the number of transactions that are related to each other is greater. This means that one non-purgeable transaction can lead to thousands and thousands of transactions from being not purgeable.

Creating Loop Metadata

The loop condition statements themselves are simple to construct. The examples used above can be used in any transaction chain interim. The LOOP condition type represents the WHERE clause of an UPDATE statement. In this respect it operates exactly like the WHERE condition type. The difference is that the LOOP condition type will be repeated many times whereas the WHERE condition type is executed only once.

The key to building LOOP metadata is building the transaction chain interim itself. The end result of any transaction chain interim is to have a row of data that shows the related transactions. The example we have been using is Procure to Pay, which involves requisitions, purchase orders, vouchers/invoices and payments/checks. The complexity of the source application will dictate how complicated it will be to construct the transaction chain interim.

An example of a complicated transaction chain metadata is the Matched PO/AP entity for Oracle Applications. As we go through this example, you will begin to see how LOOP metadata is constructed.

In the Procure to Pay transaction chain there are many transaction relationships:

- Req to PO
- PO to Receipt
- PO to Invoice
- Invoice to Check

In Oracle AP and PO transaction relationships are defined at the line level except for Invoice to Check which uses a cross reference table. So the first step in building LOOP metadata is to identify where the relationships are managed.

- Req to PO: The PO Line is stored in the Requisition Line. From a technical perspective the relationship is defined as:

  ```
  PO.PO_LINE_LOCATIONS_ALL.LINE_LOCATION_ID =
  PO.PO_REQUISITION_LINE_ALL.LINE_LOCATION_ID
  ```

- PO to Receipt: The PO ID is stored in the Receipt Line. The technical representation of this relationship is:

  ```
  PO.PO_HEADERS_ALL.PO_HEADER_ID = PO.RCV_SHIPMENT_LINES.PO_HEADER_ID
  ```

- PO to Invoice: The PO Line is stored on the Invoice Line. From a technical perspective the relationship is defined as:

```
PO.PO_DISTRIBUTIONS_ALL.PO_DISTRIBUTION_ID =
AP.AP_INVOICE_DISTRIBUTIONS_ALL.PO_DISTRIBUTION_ID
```

- Invoice to Check: A cross reference table provides the relationship between what checks paid what invoices:

```
AP.AP_INVOICE_PAYMENTS_ALL.INVOICE_ID
AP.AP_INVOICE_PAYMENTS_ALL.CHECK_ID
```

These relationships highlight how complex this transaction chain is. To populate an interim table with each transaction in the chain in a single query is too complex. There are too many tables involved. Adding to the complexity is the likelihood of these tables being high volume tables. Finally, there can be "broken" links in the chain. A check can pay an invoice that is matched to a PO. It can also pay an invoice that is not matched. A receipt can contain items for a PO that is matched to an Invoice and for a PO that is not matched to an invoice. The same is true for requisitions - they can be promoted to a PO that is matched to an invoice and to a PO that is not matched to an invoice. In addition, an invoice might be paid by two checks - one that is a candidate for purge, and one that is not a candidate for purge (where candidate for purge means it does not exist in the individual transaction interim table).

For these reasons the transaction chain is broken down into multiple steps. Each transaction relationship in the chain is addressed separately. The process for building the Matched AP/PO transaction chain is:

1. Identify Invoice-Check relationships in interim table 1.
2. Identify Invoice-PO relationships in interim table 2.
3. Identify PO-Receipt relationships in interim table 3.
4. Identify PO-Requisition relationships in interim table 4.
5. Combine results from interim tables 1-4 into a single transaction chain interim table.

So to build the transaction chain interim table, four other interim tables are used. To complicate matters even more, the relationships have to be looked at two ways - an invoice might be paid by more than 1 check; a check might pay more than 1 invoice; therefore, when identifying a relationship it must be examined by both sides. So the four steps are:

1. Identify Invoice-Check relationships in interim table 1.
   a. Identify all checks that paid invoices which are candidates for purge.
   b. Identify all invoices that were paid by checks which are candidates for purge.
2. Identify Invoice-PO relationships in interim table 2.
   a. Identify all POs matched to the invoices which are candidates for purge.
   b. Identify all invoices matched to POs which are candidate for purge.
3. Identify PO-Receipt relationships in interim table 3.
   a. Identify all receipts matched to POs which are candidate for purge.
   b. Identify all POs matched to receipts which are candidate for purge.
4. Identify PO-Requisition relationships in interim table 4.
   a. Identify all requisitions matched to POs which are candidate for purge.
   b. Identify all POs matched to requisitions which are candidate for purge.
5. Combine results from interim tables 1-4 into a single transaction chain interim table.

Technically, the metadata is constructed using the below statements.

Invoice Check Relationships:

```
INSERT INTO XA04149_PO4 (INVOICE_ID, CHECK_ID, INV_PURGEABLE_FLAG)
SELECT ip.invoice_id, ip.check_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO3','INVOICE_ID', ip.invoice_id)
FROM ap.ap_invoice_payments_all ip
WHERE invoice_id in (select x.invoice_id from XA04149_PO16 x);

INSERT INTO XA04149_PO4 (INVOICE_ID, CHECK_ID, INV_PURGEABLE_FLAG)
SELECT ip.invoice_id, ip.check_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO3','INVOICE_ID', ip.invoice_id)
FROM ap.ap_invoice_payments_all ip
WHERE check_id in (select ip.check_id from ap.ap_invoice_payments_all ip, XA04149_PO3 i
where i.invoice_id = ip.invoice_id);
```

First, note the use of the function - xa_function_pkg.get_purgeable_flag_number. This function determines if the transaction is purgeable or not. So, each INVOICE-CHECK combination is being marked as purgeable or not purgeable. This function is part of the archive engine and can be used in custom entities.

Second, note that the select does not necessarily retrieve invoices or checks that have been identified as candidates for purge. That is, they may not exist in the interim table because they did not meet the policy scoping criteria (e.g., the invoice date may fall outside the retention period). The purpose of the select statements is to identify all INVOICE-CHECK combinations regardless of both invoice and check being identified as a candidate for purge.

Third, note that the first INSERT retrieves INVOICE-CHECK combinations based on the invoices identified as candidates for purge (that is, the invoices that do exist in the invoice interim table). The second INSERT retrieves INVOICE-CHECK combinations based on the checks that paid the invoices identified as candidates for purge. This covers both sides of the INVOICE-CHECK relationship.

Examination of the other transaction relationships will show a similar construct.

Invoice PO Relationships

```
INSERT INTO XA04149_PO8 (PO_HEADER_ID, INVOICE_ID, PO_PURGEABLE_FLAG, PURGEABLE_FLAG)
SELECT D.po_header_id, i.invoice_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO12', 'PO_HEADER_ID',
d.po_header_id), 'N'
FROM PO.po_distributions_all D, AP.ap_invoice_distributions_all I
WHERE I.invoice_id in (SELECT X.invoice_id FROM XA04149_PO4 X )
AND D.po_distribution_id = I.po_distribution_id;
INSERT INTO XA04149_PO8 (PO_HEADER_ID, INVOICE_ID, PO_PURGEABLE_FLAG, PURGEABLE_FLAG)
SELECT D.po_header_id, i.invoice_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO12',
'PO_HEADER_ID',d.po_header_id), 'N'
FROM PO.po_distributions_all D, AP.ap_invoice_distributions_all I
WHERE D.po_header_id in (SELECT X.po_header_id FROM XA04149_PO12 X )
AND D.po_distribution_id = I.po_distribution_id;
```

The same three principles exist here as they did for the INVOICE-CHECK relationship. First, the xa_function_pkg is used to identify the purgeability of the PO-INVOICE combination. Second, both statements retrieve PO-INVOICE combination without regard for both transactions being candidates for purge. Third, PO-INVOICE combinations are being retrieved for both PO candidates and INVOICE candidates.

PO Requisition Relationship

```
INSERT /*+ APPEND PARALLEL(XA04149_PO15,8) */ INTO XA04149_PO15 (po_header_id,
requisition_header_id, req_purgeable_flag)
SELECT /*+ FULL(pll) PARALLEL(pll,4) */ pll.po_header_id, prl.requisition_header_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO14',
'REQUISITION_HEADER_ID',prl.requisition_header_id)
FROM PO.po_line_locations_all pll, PO.po_requisition_lines_all prl
WHERE prl.line_location_id = pll.line_location_id and prl.requisition_header_id in
(select requisition_header_id from XA04149_PO14);

INSERT /*+ APPEND */ INTO XA04149_PO15 (po_header_id, requisition_header_id,
req_purgeable_flag)
```

```
SELECT pll.po_header_id, prl.requisition_header_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO14',
'REQUISITION_HEADER_ID',prl.requisition_header_id)
FROM PO.po_line_locations_all pll, PO.po_requisition_lines_all prl
WHERE prl.line_location_id = pll.line_location_id and pll.po_header_id in (select
po_header_id from XA04149_PO12);
```

The PO-REQUISITION relationship uses the same approach as the INVOICE-CHECK and PO-INVOICE relationships. First, the xa_function_pkg is used to identify the purgeability of the PO-REQUISITION combination. Second, both statements retrieve PO-REQUISITION combinations without regard for both transactions being candidates for purge. Third, PO-REQUISITION combinations are being retrieved for both PO candidates and REQUISITION candidates.

PO Shipment Relationship

```
INSERT /*+ APPEND */ INTO XA04149_PO13 (po_header_id, shipment_header_id,
rcpt_purgeable_flag)
SELECT A.po_header_id, A.shipment_header_id,
xa_function_pkg.get_purgeable_flag_number('XA04149_PO12', 'PO_HEADER_ID',A.po_header_id)
FROM po.rcv_shipment_lines A
WHERE A.shipment_header_id in (select shipment_header_id from po.rcv_shipment_lines sl,
XA04149_PO12 x where sl.po_header_id = x.po_header_id );
```

The PO-RECEIPT relationship is treated differently than the other relationships. Receipts are not tracked separately as candidates for purge. They are part of a PO transaction, and consequently, they are candidates for purge through the PO only. Therefore, PO-RECEIPT combinations are retrieved only for PO candidates - that is, all receipts for PO candidates are identified as part of the transaction chain. However, the statement above still uses the xa_function_pkg to determine purgeability of each PO-RECEIPT combination, and the statement retrieves PO-RECEIPT combinations without regard for both transactions being candidates for purge.

The Transaction Chain Interim

Once all of the combination pairs have been identified, these combinations can be put together into one interim table - the transaction chain interim table. The statements below show how the previous interim tables are combined into a single transaction chain interim table. There are two INSERT statements used to reduce the complexity of the SQL statement which also results in improved performance.

```
INSERT /*+ APPEND */ INTO XA04149_PO10 (PO_HEADER_ID, REQUISITION_HEADER_ID,
purgeable_flag, purge_PO_HEADER_ID, purge_REQUISITION_HEADER_ID)
SELECT r.po_header_id, r.requisition_header_id,
decode(r.req_purgeable_flag,'Y',decode(nvl(x.po_header_id,-1),-1,null,x.purgeable_flag),n
ull) flag, r.po_header_id, r.requisition_header_id
FROM XA04149_PO12 x, XA04149_PO15 r
WHERE r.po_header_id = x.po_header_id (+) UNION SELECT DISTINCT null, null,
s.po_header_id, null, s.shipment_header_id, decode(s.rcpt_purgeable_flag,'Y',
decode(nvl(x.po_header_id,-1),-1,null,x.purgeable_flag),null) flag, null, null,
s.po_header_id, null, s.shipment_header_id FROM XA04149_PO12 x, XA04149_PO13 s where
s.po_header_id = x.po_header_id (+);

INSERT /*+ APPEND PARALLEL(XA04149_PO10,8) */ INTO XA04149_PO10 (INVOICE_ID, CHECK_ID,
PO_HEADER_ID, purgeable_flag, purge_INVOICE_ID, purge_CHECK_ID, purge_PO_HEADER_ID)
SELECT /*+ full(i) cardinality(i,1) parallel(i,#) */ DISTINCT i.invoice_id, i.check_id,
a.po_header_id, decode(matched_to_po_flag,
0,decode(inv_purgeable_flag,'Y',po_purgeable_flag,null),inv_purgeable_flag) flag,
i.invoice_id, i.check_id, a.po_header_id
FROM XA04149_PO4 i, XA04149_PO8 a
WHERE i.invoice_id = a.invoice_id (+) union select /*+ parallel (a,#) */ distinct
a.invoice_id, null, a.po_header_id, null, null,
decode(a.purgeable_flag,'Y',a.purgeable_flag,null) flag, a.invoice_id, null,
a.po_header_id, null, null from XA04149_PO8 a where not exists (select 1 from
XA04149_PO4 b where b.invoice_id=a.invoice_id);
```

The end result is table that looks the below:

TRX CHAIN INTERIM

REQ	PO	VCHR	VCHR	PURGE_FLAG
1A	10A			Y
1A	10B			N
1B	10A			Y
1B	10A			Y
1C	10C			Y
	10A	100A		Y
	10B	100A		N
	10A	100B		Y
	10A	100C		Y
	10C	100D		Y
		100A	1000A	Y
		100A	1000A	N
		100B	1000A	Y
		100C	1000B	Y
		100D	1000D	Y

Stepping through the LOOP rules again, the end result of the transaction chain interim is the same as the initial example in the previous section:

TRX CHAIN INTERIM

REQ	PO	VCHR	VCHR	PURGE_FLAG
1A	10A			N
1A	10B			N
1B	10A			N
1B	10A			N
1C	10C			Y
	10A	100A		N
	10B	100A		N
	10A	100B		N
	10A	100C		N
	10C	100D		Y
		100A	1000A	N
		100A	1000A	N
		100B	1000A	N
		100C	1000B	N
		100D	1000D	Y

Although the chain spans multiple rows, the transaction chain of 1C-10C-100D-1000D is still purgeable in the above interim table.

The final step of the transaction chain logic is to update the transaction interim tables with the results of the transaction chain interim:

```
NOT EXISTS (SELECT 1 FROM XA04149_PO10 X WHERE X.purge_po_header_id = A.po_header_id AND
X.purgeable_flag = 'Y')
NOT EXISTS (SELECT 1 FROM XA04149_PO10  X WHERE X.purge_invoice_id = A.invoice_id AND
X.purgeable_flag = 'Y')
NOT EXISTS (SELECT 1 FROM XA04149_PO10 X WHERE X.purge_requisition_header_id =
A.requisition_header_id AND X.purgeable_flag = 'Y')
```

CHAPTER 16

How to Uninstall Data Archive 5.1

This chapter includes the following topics:

- Abstract, 72
- Overview, 72
- Step 1. Delete the History Instance, 72
- Step 2. Delete the Applimation Agent, 73
- Step 3. Delete the Source Instance, 73
- Step 4. Delete the Agent Schema, 73
- Step 5. Create and Run the drop_data_access.sql Script, 74
- Step 6. Delete the Applimation Application, 74
- Step 7. Drop the amhome Schema and Tablespace, 75

Abstract

This article describes how to uninstall Data Archive 5.1 and the application history from your server.

Overview

Data Archive is uninstalled when a server replaces an older server or when the product is no longer needed on the server. To move to a newer version of Data Archive, you can choose to upgrade an existing installation. Before you uninstall Data Archive, ensure that you have administrator privileges.

Step 1. Delete the History Instance

To delete a history instance, you create a **Restore Informia Archive History** through the ILM web based user interface.

1. Log in to Applimation user interface and navigate to**Administrator** > **ERP Instances**.
2. Click **Edit** for the destination instance to list the Applimation Agent, Informia Archive, and Informia Archive History.

3. Click **Edit** for Informia Archive History and then click **Remove**. A confirmation page appears.
4. In the confirmation page, click **Yes** to create the **Restore Informia Archive History** job.
5. To view the status of this job, navigate to**Jobs** > **View Current Jobs**.

Step 2. Delete the Applimation Agent

To delete the Applimation Agent on the target instance, create a **Restore Applimation Agent** job through the ILM web based user interface.

1. Navigate to**Administrator** > **ERP instances** and click **Edit** for the destination instance.
2. Click **Edit** for the Applimation Agent and then click **Remove** to set a job to restore the Applimation Agent.

   The **Restore Applimation Agent** job restores the Applimation Agent schema from the destination ERP instance.
3. To view the status of the Restore Applimation Agent job, click**Jobs** > **View Current Jobs**.

Step 3. Delete the Source Instance

To delete the source instance, you create a **Restore Informia Archive** job through the ILM web based user interface.

1. Navigate to **Administrator** > **ERP instances** and click **Edit** for the source instance. The ERP instance workbench screen appears.
2. Edit the Informia Archive component.
3. Click **Remove** to create the **Restore Informia Archive** job that restores Informia Archive.

Step 4. Delete the Agent Schema

To delete the Applimation Agent's schema from the source instance, set up a **Restore Applimation Agent** job from the ILM web based user interface.

1. Navigate to **Administrator** > **ERP instances** and click **Edit** for the source instance.
2. Edit the Applimation Agent and then click **Remove** to set a job to restore the Applimation Agent.
3. To view the status of this job, click **Jobs** > **View Current Jobs**.

   The web page lists the status of the Restore Applimation Agent job which restores the Applimation Agent schema from the source ERP instance.

Step 5. Create and Run the drop_data_access.sql Script

Create and run a script on the source instance to drop the amarchive and amquery schemas.

Access the source instance and create the drop_data_access.sql script with the following code:

```
set head off
set feed off
set pages 0

spool drop_data_access_objs.sql

select 'DROP SYNONYM '||owner||'."'||object_name||'";'
from dba_objects
where owner in ('AMARCHIVE','AMQUERY')
and object_type = 'SYNONYM';

select 'DROP VIEW '||owner||'."'||object_name||'";'
from dba_objects
where owner in ('AMARCHIVE','AMQUERY')
and object_type = 'VIEW';

select 'DROP PACKAGE '||owner||'."'||object_name||'";'
from dba_objects
where owner in ('AMARCHIVE','AMQUERY')
and object_type = 'PACKAGE';

spool off

PROMPT "Dropping objects, this will take a while ..."

@drop_data_access_objs.sql

PROMPT "Dropping Users Cascade, not much longer ..."

drop user AMARCHIVE cascade;

drop user AMQUERY cascade;

PROMPT "Data Access Schemas have been Dropped Successfully"
```

Step 6. Delete the Applimation Application

To delete the Applimation Application, delete the Applimation installation directory on the target server.

1. Stop the Tomcat service before you remove the Applimation Application. To stop the Tomcat service, access the target machine's command prompt.
2. Navigate to the Tomcat bin directory.
3. Run the shutdown command. The Tomcat service stops enabling you to delete the Applimation application.
4. Delete the container from the following directory:

   `/u01/tomcat/apache-tomcat-5.5.15/webapps`

   Note: You can use the rm -r command to delete the web container.
5. Delete the container's XML file from the following directory:

   `/u01/tomcat/apache-tomcat-5.5.15/conf/Catalina/localhost`

Step 7. Drop the amhome Schema and Tablespace

Delete the amhome schema and amhome tablespace on the source database.

Access the SQL prompt of the source database to drop the amhome schema.

The following example lists the commands required for cleaning up the home qa4_qa9203:

```
select * from DBA_USERS where usernmae like 'Q4_9203_4%'
drop user Q4_9203_4_AMHOME cascade on qa4_qa9203

select * from DBA_TABLESPACES where tablespace_name like 'QA_4%'
drop tablespace QA_4_AM_DATA including contents and datafiles
drop tablespace QA_4_AM_INDEX including contents and datafiles
drop tablespace QA_4_AM_STAGE including contents and datafiles
drop tablespace QA_4_ARC_DATA including contents and datafiles
drop tablespace QA_4_ARC_INDEX including contents and datafiles
drop tablespace q4_9203_4_AM_History_D including contents and datafiles
drop tablespace Q4_9203_4_AM_HISTORY_X including contents and datafiles
```

CHAPTER 18

How to Uninstall Data Archive 5.3

This chapter includes the following topics:

- Abstract, 76
- How to Uninstall Data Archive 5.3, 76

Abstract

This article describes how to uninstall Data Archive 5.3.

How to Uninstall Data Archive 5.3

1. Log in to the ILM web application.
2. Restore all data from the history database.
3. To drop all the database users and the database links that you created for data archiving, perform the following steps:
   a. Go to **Administration** > **Manage Connections**.
   b. Select a connection.
   c. Find the attributes for the database link. Drop the database link from that connection.
   d. Find the attribute for the staging user. Drop the staging user and tablespace.
   e. For the destination connection, drop the history user and tablespace.
   f. For the source connection, perform the following steps to drop the seamless access layer:
      - Delete the seamless access App layer based on the application.
      - Drop the seamless access users from the source.
      - Drop the database link that is associated for seamless access.
   g. Repeat steps b through f for each connection.
4. Stop the ILM Server.
5. To drop the ILM repository, perform the following steps:
   - Drop the AMHOME user.

- Drop the data tablespace for the AMHOME user.
- Drop the index tablespace for the AMHOME user.

6. Delete the ILM Server directory.
7. If you archive to file, perform the following steps to uninstall Data Vault Service:
   a. Use the npa_stop_sm command to stop the Data Vault Service.
   b. Navigate to following file:

      ```
      NParchive-dist/conf/sm-config.xml
      ```

   c. Delete the data, metadata, and queue directory defined in the file.
   d. Delete the following directory:

      ```
      NParchive-dist
      ```

   e. If you have the VM image installation of Data Vault Service, stop and delete the VM image.
8. If you use source adapters, follow the uninstallation steps for the adapter.
9. Delete any remaining ILM files, such as installation files and documentation.

CHAPTER 20

How to Use Scripts to Change Database User Passwords in the ILM Repository

This chapter includes the following topics:

- Abstract, 78
- Overview, 78
- Properties File, 79
- Changing Database User Passwords, 79

Abstract

Use an SQL script to update database user passwords for source and target connections in the ILM repository. This article describes how to configure properties and run the scripts that update the passwords in the ILM repository.

Overview

When you change database user or schema passwords, you have to update the passwords for the source or target connection in the ILM repository. You can use an SQL script to update passwords in the ILM repository without logging in to the ILM application. Use the script to change the password for the Admin, Apps, and Staging database schemas and for the Data Vault Service target. The script is available as a batch file or a shell file and is located in the optional directory of the ILM installation.

The script provides a more efficient option to update the ILM repository when company policy requires you to update database user passwords on a regular basis. For example, you must change database user passwords every 20 days.

Properties File

The script uses values in a properties file to connect to the ILM repository and to change the passwords. You create and configure the properties file before you run the script.

The following table describes the required properties for the properties file:

Property	Description
amhome.connection.host	ILM repository host name.
amhome.connection.port	ILM repository port number.
amhome.connection.sid	ILM repository database service name or database name.
amhome.connection.username	ILM repository user name.
amhome.connection.password	ILM repository user password.
amhome.connection.schemaname	Unique database placeholder for tables.
amhome.connection.validate.sql	Script that determines if the installation repository is the ILM repository.
amhome.values.separator	Symbol that indicates the separation of multiple values for the other properties. Use a unique value that is not used in the database names or passwords. For example, use `@@`.
amhome.rep.names	ILM repository database schema name for the Admin, Apps, or Staging schemas. Use the same name as the **Schema Name** field in the Edit Archive Source or Edit Archive Target menu. To enter multiple values, use the separator that you defined for the amhome.values.separator property. For example, enter `SYSTEM@@STAGING` to change the password for the SYSTEM and STAGING database schemas.
amhome.rep.usernames	Login name for the database schema or the Data Vault user. Use the same name as the **Login Name** field in the Edit Archive Source or Edit Archive Target menu. To enter multiple values, use the separator that you defined for the amhome.values.separator property.
amhome.rep.newpasswords	New password for the login name. To enter multiple values, use the separator that you defined for the amhome.values.separator property.

Changing Database User Passwords

Configure the ILM repository connection details and the database user details in a properties file. The script uses the properties file to connect to the ILM repository and to validate the database user login name and

password. If the database user login details are valid, the script updates the ILM repository with the new password.

Note that the ILM application server does not need to be running when you run the script.

1. Navigate to the following directory:

   `<ILM installation directory>/optional`

2. Use the `PasswordChangesSampleProperty` file to create a property file.
3. Add the required properties to the property file.
4. Save and close the property file.
5. From the same directory, run one of the following files:
   - `PasswordChangeEnv.bat` for Windows.
   - `PasswordChangeEnv.sh` for Linux and UNIX.
6. Enter the location of the property file that you created.

   The script executes and creates the `PasswordChangeLog` log file in the same directory.

7. View the log file to show the script outcome.

CHAPTER 22

IBM DB2 Database Connectivity and Setup for Data Archive

This chapter includes the following topics:

- Abstract, 81
- Overview, 81
- Step 1. Set Up IBM DB2 Privileges, 82
- Step 2. Bind the Packages on IBM DB2, 82
- Step 3. Mine the Source Database and Create Entities, 82
- Step 4. Create a Source Connection, 83
- Step 5. Create a Target Connection, 84
- Step 6. Create an Archive Project, 84
- Known Limitations, 84

Abstract

This article provides an overview of how to set up IBM DB2 database connectivity and how to set up Data Archive to archive from IBM DB2 sources. You can configure Data Archive to archive from an application on an IBM DB2 source to a IBM DB2 target or to the Data Vault.

Overview

To set up database connectivity and Data Archive to archive from IBM DB2 sources, complete the following steps:

1. Set up IBM DB2 privileges.
2. Bind the packages on IBM DB2.
3. For custom applications, mine the source database and create entities.
4. Create a source connection for the IBM DB2 source database.
5. Create a target connection for the IBM DB2 target database or for the Data Vault.

6. Create an archive project.

This process is required when you archive from IBM DB2 to a file or to another IBM DB2 database. Note that after you set up the privileges and bind the packages, the Data Archive setup is the same as other databases. No special configuration is required for IBM DB2 sources.

Step 1. Set Up IBM DB2 Privileges

Before you install Data Archive, set up privileges to bind the packages and to access data.

Set up the following privileges:

Bind Packages Privileges

The user that you provide in the standalone job to bind IBM DB2 packages requires specific privileges.

Login User Privileges

The user that you provide to connect or log in to IBM DB2 requires specific privileges to access data. You provide a login user when you define source and target connections and when you define connections to mine data in the Enterprise Data Manager.

For more information, see the *Informatica Data Archive Installation Guide*.

Step 2. Bind the Packages on IBM DB2

After you install Data Archive, bind the packages on IBM DB2.

Before you can use DataDirect JDBC drivers to connect to IBM DB2 sources, you must bind the packages on the source. To bind packages, use the DB2 Bind Package standalone job in Data Archive.

For more information, see the *Informatica Data Archive 6.2 HotFix 1 Administrator Guide*.

Step 3. Mine the Source Database and Create Entities

If you have a custom application on an IBM DB2 database, mine the source database and create entities in the Enterprise Data Manager. If you have a pre-packaged application, such as PeopleSoft, then you can skip this step.

IBM DB2 Connection Properties

Specify connection properties when you create a connection to a IBM DB2 database in the Enterprise Data Manager.

The following table describes the IBM DB2 database connection properties:

Property	Description
Connection Name	Connection name for the source database. Use the name to differentiate between different connections in the Enterprise Data Manager.
Database Type	Database connection type that determines how you connect to the archive source database. The connection type that you choose depends on the database that contains the source data. Choose a compatible connection type for the database and database version. Choose one of the following connection types to connect to an IBM DB2 database: - DB2_Adapter_1. Use to connect to IBM DB2 on z/OS. - DB2_Adapter_UDB. Use to connect IBM DB2 on AIX, Linux, UNIX, or Windows (LUX). - DB2_AS400_Adapter. Use to connect to IBM DB2 on AS/400.
Database Host	IP address of the source application database server.
Database Port	Port of the source application database server.
Service Name	Unique identifier or system identifier for the source application database server.
User Name	Database user name.
Password	Password for the database user name.

Mining the Database

Mine the IBM DB2 database that you want to archive from and create entities for the tables that you mined.

1. Open the Enterprise Data Manager.
2. From the Data Archive tab, select the custom product family version.
3. Click **File** > **Mine Database**.

   The **Connect to Mined Database** dialog box appears.
4. Enter the IBM DB2 connection properties.
5. Mine the tables that you want to archive.
6. Create entities.

   After you connect to the database, you can see a list of all available objects within the connected database. Use the objects to create entities.

Step 4. Create a Source Connection

Before you create an archive project, create a source connection for the IBM DB2 source database.

For more information, see the *Informatica Data Archive Administrator Guide*.

Step 5. Create a Target Connection

Before you create an archive project, create a target connection for the IBM DB2 target database or for the Data Vault.

For more information, see the *Informatica Data Archive Administrator Guide*.

Step 6. Create an Archive Project

Create an archive project. Include the source and target connections that you created in the previous steps.

Before you create and run an online archive project that archives data to a IBM DB2 database, you may need to complete prerequisite steps. The prerequisites are required only if IBM DB2 database administrators do not allow external applications to create objects in IBM DB2 databases. In this case, you must disable table creation in the archive job and manually create tables in the source and target databases.

For more information, see the *Informatica Data Archive User Guide*.

Known Limitations

The following known limitation applies to Data Archive versions 5.3.5 to 6.1.1.

Additional entity configuration is required for PeopleSoft and Siebel applications on IBM DB2. For pre-packaged PeopleSoft and Siebel accelerators, you must change all entity parameters that have DATE or TIME datatypes. Copy the entity and change all DATE and TIME datatypes to VARCHAR.

The CR for this known limitation is 268545.

abstract

This article describes how to install Data Visualization after you installed Data Archive 6.1.1.

CHAPTER 24

Installing Data Visualization

This chapter includes the following topics:

- Installing Data Visualization Overview, 86
- Before You Install, 86
- Installing Data Visualization, 87

Installing Data Visualization Overview

Use the Data Archive installer to install Data Visualization.

You can use Data Visualization to create reports based on archived data in the Data Vault. You can install Data Visualization at the same time you install Data Archive or at a later time.

To install Data Visualization after you have installed Data Archive 6.1.1, run the Data Archive installer and choose the Data Visualization option.

Before You Install

Before you install Data Visualization, perform the following tasks.

- ❑ Get the Data Archive advanced license key.
- ❑ Verify that the machine that hosts the ILM Engine has Java version JDK J2SE 7.0 (1.7).

  Note: To install JDK, download JDK from either of the following locations:

  - On IBM AIX operating systems, download JDK from the following IBM site: http://www.ibm.com/developerworks/java/jdk
  - On all other operating systems, download JDK from the following Oracle site: http://www.oracle.com/technetwork/java/javase/downloads/index.html
- ❑ Verify that the `JAVA_HOME` environment variable is set to the JDK root directory.
  For example, the value for the `JAVA_HOME` environment variable is: `C:\Program Files (x86)\Java\JDK1.7_21`.
- ❑ Verify that the `Path` environment variable contains the path to the JDK `bin` directory.
  For example, the value for the `Path` environment variable is: `C:\Program Files (x86)\Java\JDK1.7_21\bin`.

- If the machine that hosts the ILM Engine is not connected to the internet, perform the following steps:
  1. From the ILM installation folder, extract the `Visualization.zip` file.
  2. Open the `WEB-INF` folder.
  3. Open the `web.xml` file in edit mode.
  4. Delete the following line from the `web.xml` file:

     ```
     <!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3//
     EN" "http://java.sun.com/dtd/web-app_2_3.dtd">
     ```
  5. Save the `web.xml` file.
  6. Compress the contents of the `Visualization` folder and save as `Visualization.zip` in the ILM installation folder.
     The `Visualization.zip` file replaces the original `Visualization.zip` file in the ILM installation folder.

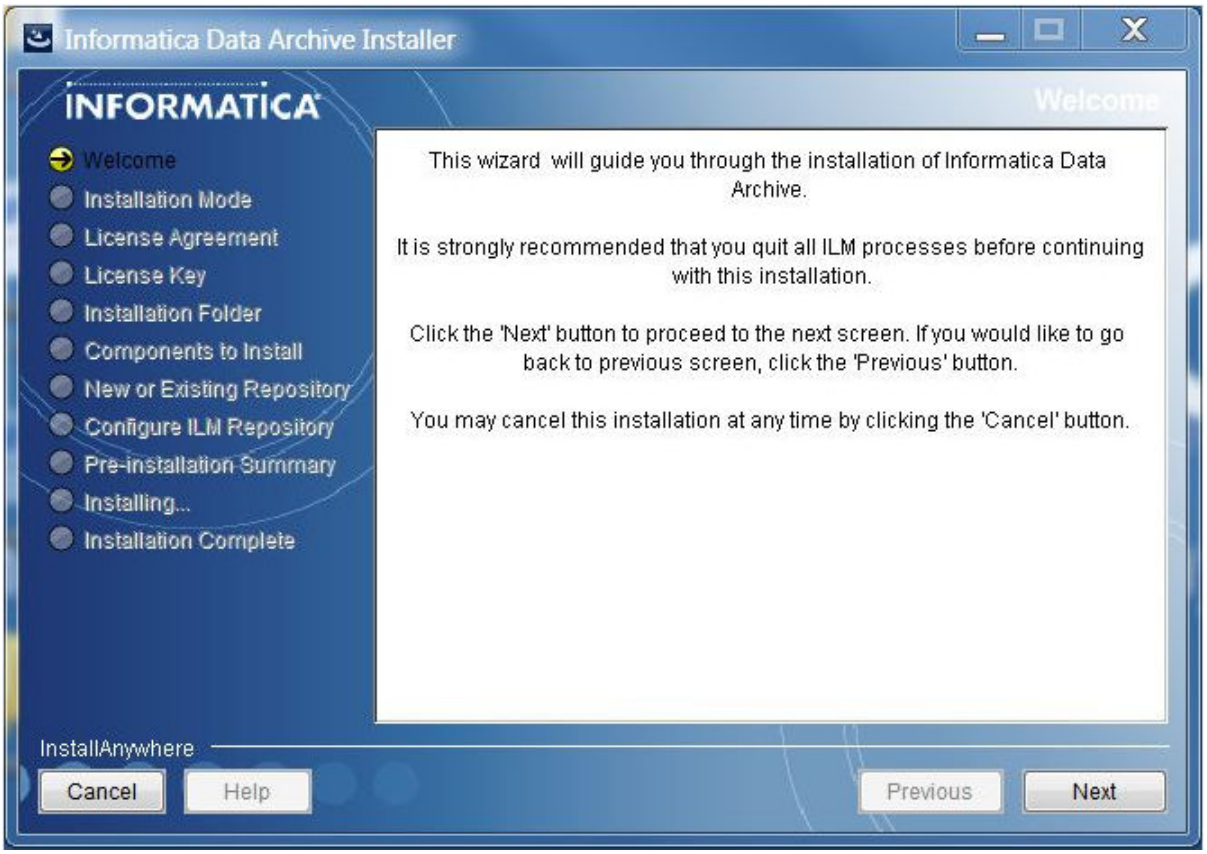
Installing Data Visualization

Use the Data Archive installer to install Data Visualization.

1. Go to the directory where you extracted the files from `Informatica_ILM_611.zip` or `Informatica_ILM_611_<hotfix>.zip`.

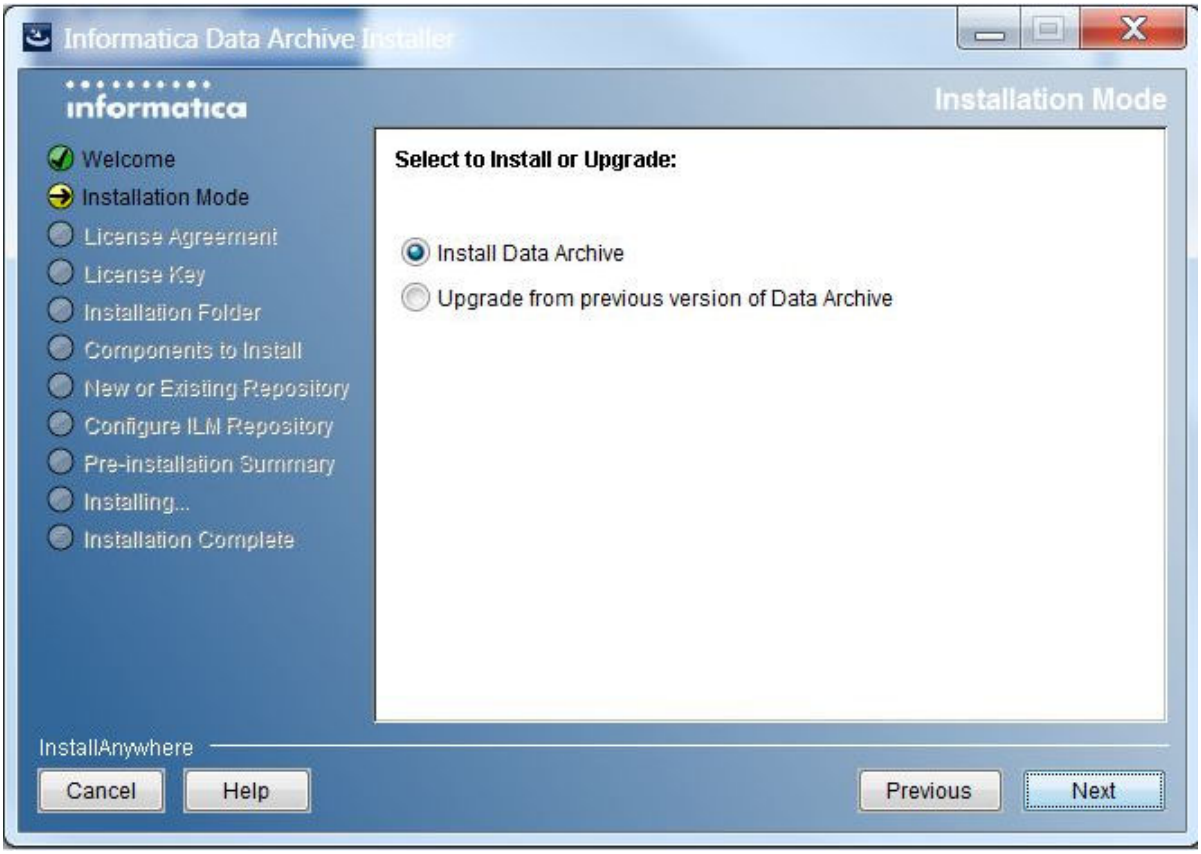
2. Double-click `ILMInstaller.exe`.

   The **ILM Installer** dialog box appears.



3. Click **Next**.

   The **Installation Mode** page appears.

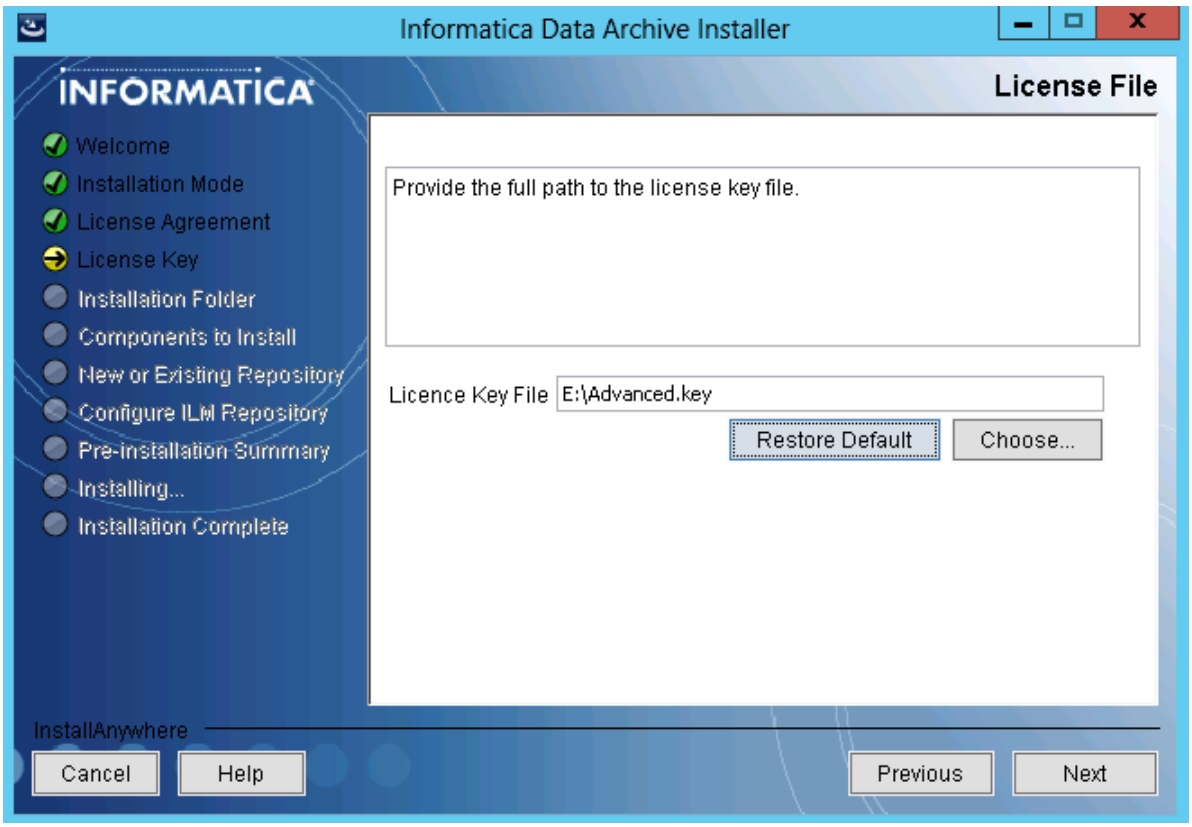


4. Select **Install Data Archive**, and click **Next**.

   The **License Agreement** page appears.

5. Accept the terms of the License Agreement and click **Next**.

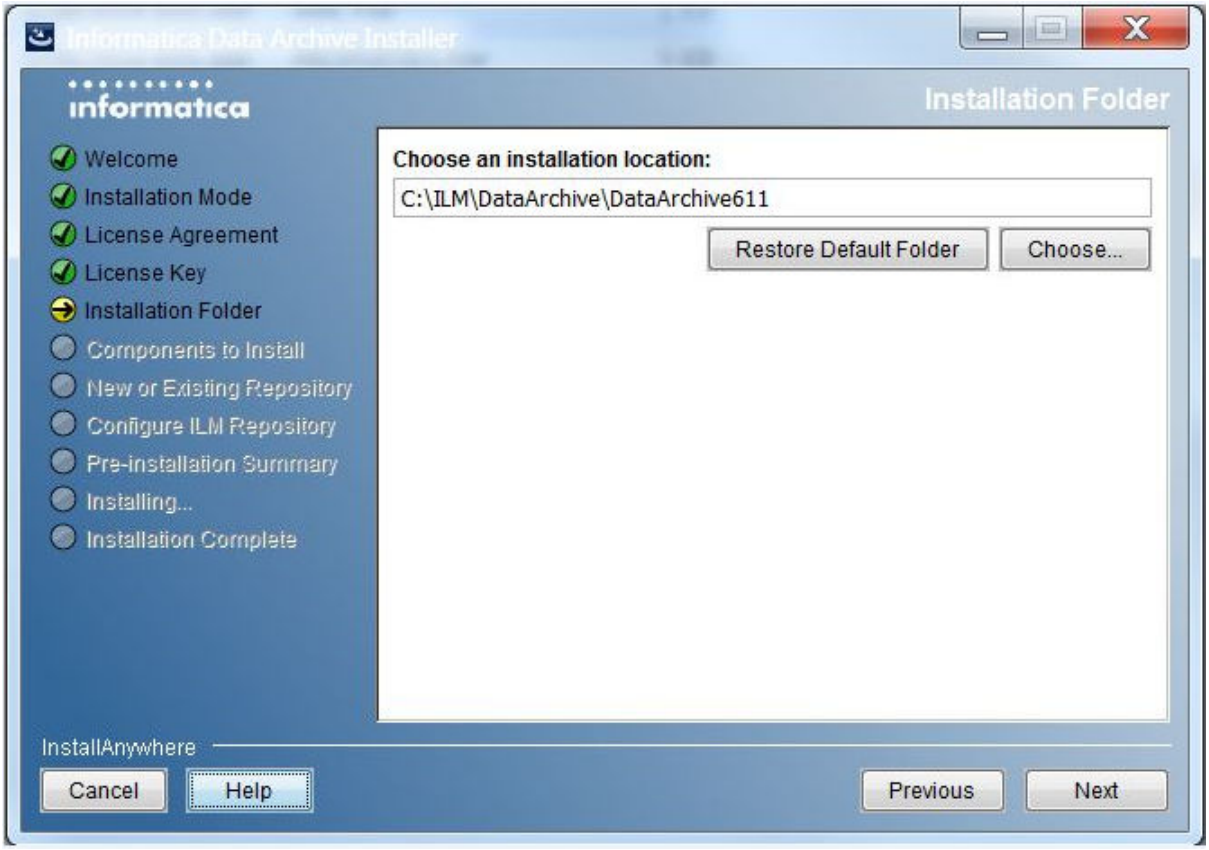
   The **License File** page appears.



6. Click **Choose**, and browse to the location of the advanced license key file.

7. Click **Next**.

   The **Installation Folder** page appears.

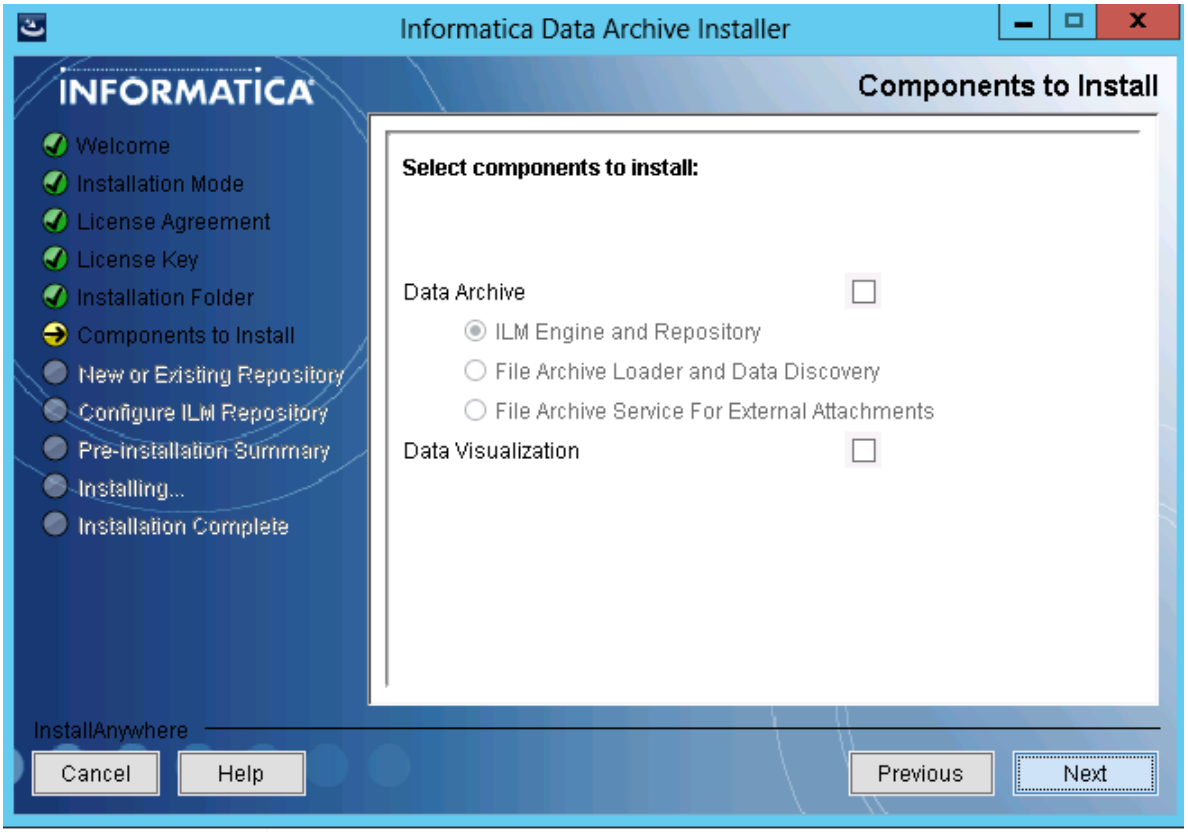


8. Click **Choose**, and browse to the Data Archive 6.1.1 installation folder.

   For example: `C:\ILM\DataArchive\DataArchive611`

9. Click **Next**.

   The **Components to Install** page appears.



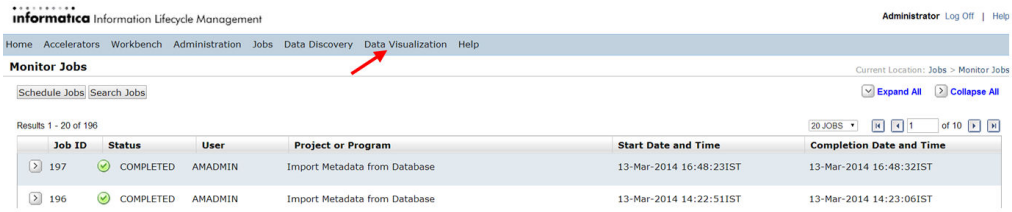
10. Select **Data Visualization**, and click **Next**.

    The **Installing ILM** page appears. When the installation completes, the **Installation Complete** page appears.

11. Click **Done** to close the installer.

12. Log in to Data Archive as a user with the Administrator, Report Designer, or Report Viewer role.

13. Verify that the **Data Visualization** option appears in the main menu.



14. Click **Data Visualization** > **Reports and Dashboards** to start creating reports.

    For more information, see the *Informatica Data Archive Users Guide*.

CHAPTER 26

Integrating Third-Party Schedulers in ILM Engine

This chapter includes the following topics:

- Abstract, 93
- Overview, 93
- Integrating a Third-Party Scheduler, 94
- Using the Third-Party Scheduler API, 94
- Integrating External Schedulers, 95

Abstract

ILM Engine utilizes the Quartz scheduler as the default scheduler. You can also use third-party schedulers. This article shows you how to integrate the third-party schedulers with ILM Engine.

Overview

ILM Engine utilizes the Quartz scheduler as the default scheduler. If your organization uses a third-party scheduler, you can continue to use the third-party schedule while maintaining all of the scheduling functionality that Data Archive provides. Data Archive provides the following scheduling functionality:

- **Schedule a job.** You can integrate different schedulers and use user-defined schedulers while executing jobs in Data Archive. You must be allowed to specify a scheduler in Data Archive.
- **Terminate a schedule.** You can terminate a repeating schedule.
- **Terminate a job.** You can terminate a job based on the provided scheduler.
- **Resume a job.** The scheduler can resume a specific job from the last commit point.

Integrating a Third-Party Scheduler

Use one of the following options to integrate a third-party scheduler:

- Replacing the Quartz scheduler with a third-party scheduler.
- Use the third-party scheduler API to integrate with the Quartz scheduler.

Consider the following factors when you choose how to integrate the third-party scheduler:

- The third-party scheduler must have a Java API.
- The third-party scheduler integration must meet the rules and guidelines for configuring and executing scheduled jobs in your organization.
- You might need to leverage specific third-party scheduler functionality.

Replacing the Quartz Scheduler

Data Archive includes the Quartz scheduler. Quartz is an open source enterprise job scheduler. You can replace Quartz with a scheduler that you use in your organization.

When you replace the Quartz scheduler with a third-party scheduler, you can view detailed information in the third-party scheduler UI. Also, job definitions and schedules that you create in Data Archive are immediately applied to the third-party scheduler database. As a result, scheduling teams might have less control over scheduling.

Using the Third-Party Scheduler API

Informatica has developed a third-party Scheduler API. You can schedule jobs with a third-party scheduler and pass the scheduling rules to Quartz. The third-party scheduler passes a URL with a Definition ID to Quartz. Quartz returns the job status to the scheduler.

To utilize the third-party Scheduler API, complete the following steps:

1. Set up an Archive Definition in the Data Archive UI, but do not create a schedule within the UI for this job. For each Archive Definition that is created there will be a Definition ID number, which will be part of the URL that is passed from the third-party Scheduler.
2. Wrap the Retrieve Current Status Program in a script so that it continues to be called until the completed code is returned.

The third-party schedule API is easy to deploy.

Other than adding the third-party scheduler to pass the URL that contains the Definition ID, you do not need to make other changes. Also, you can set up the archive definitions, while the scheduling team can maintain control over the set up and execution of the scheduled jobs. However, s because scheduling occurs outside of Quartz, information such as Future Run is not visible in the Data Archive.

Integrating External Schedulers

To integration with the external scheduler, complete the following steps:

1. Create an implementation for AMScheduler Interface.
2. Register the implementation.
3. Deploy the implementation.

Step 1: Create an Implementation for the AMScheduler Interface

Enter a short description of your concept here (optional).

Create a java class implementation for the interface com.applimation.schedule.AMScheduler with the following name:

```
com.applimation.schedule.<scheduler name>.impl.AM<scheduler name>SchedulerImpl
```

For example:

```
com.applimation.schedule.quartz.impl.AMQuartzSchedulerImpl
com.applimation.schedule.cron.impl.AMCronSchedulerImpl
```

The Interface expects the following methods:

- Scheduler setup
- Scheduler start/stop

You also need to schedule a definition and retrieve the current status.

Scheduler Setup Methods

Implement the following methods for scheduling definitions and for resuming and terminating the Informatica Archive jobs:

```
/**
* This method should create a schedule for the job. It run the job, it should call
 * com.applimation.schedule.AMScheduleHandler.runJob api
     *
     * @param schedulerBean
     * schedulerBean should have the parameters required for to schedule the job
     * 1 SchedulerType
     * 2 scheduleDate
     * 3 TimeZone
     * 4 scheduleNotificationFlag;
     * 5 scheduleNotificationEmail;
     * 6 List<Integer> definitionIDs;
     * 7 scheduleRepeatFlag;
     * 8 SchedulerOccurrenceType
     * 9 scheduleOccurrenceValue;
     * 10 scheduleEndDate;
     * 11 SchedulerRepeatType
     * 12 scheduleRepeatValue
     * 13 scheduleOccurrenceFlag;
     * 14 notificationEmail;
     * 15 SchedulerNotificationType
     * 16 priority;
     *
     * @param scheduleId
     * @param amuId
     * @return Date (calculated endDate for schedule if number of occurrence are given )
     * @throws Exception
     */
public Date scheduleJob(SchedulerBean schedulerBean, int scheduleId, int amuId) throws
Exception;
```

```
    /**
     * To resume a paused or errored job it should call the
 *  com.applimation.schedule.AMScheduleHandler.resumeJob api
     * @param jobId
     * @param amuId
     * @throws Exception
     */
    public void scheduleResumeJob(int jobId,int amuId) throws Exception;

    /**
     * To terminate the job, it should call
 *  com.applimation.schedule.AMScheduleHandler.resumeJob api
* @param jobId
     * @param amuId
     * @throws Exception
     */
    public void scheduleTerminateJob(int jobId,int amuId) throws Exception;


    /**
     * This method should terminate the repeating schedule
     */
    public boolean terminateSchedule(final int scheduleId, int amuId) throws Exception;

    /**
     * This method should get the next fire time for this schedule
     * @param scheduleId
     * @param amuId
     * @return
     * @throws Exception
     */
    public Date getNextFireTime(int scheduleId, int amuId) throws Exception;
```

Scheduler Start and Stop Methods

Implement the following methods if the Applimation application server starts and stops the external scheduler. If the control is not required, you can have the null implementation.

```
/**
     * This method should start the scheduler
     * @throws Exception
     */
    public void startScheduler() throws Exception;

    /**
     * @param configFile
     * @throws Exception
     */
    public void startScheduler(String configFile) throws Exception;

    /**
     * This method should shut down the scheduler
     *
     * @throws Exception
     */
    public void shutDownScheduler() throws Exception;

    /**
     * should set the properties configuration file of the scheduler
     *
     */
    public String getConfigFile();

    /**
     *
     * should get the properties configuration file of the scheduler
     */
    public void setConfigFile(String configFile);
```

```
/**
 * set the configuration file location
 * @param location
 */
public void setConfigFileLocation(String location);

/**
 * get the configuration file location
 *
 * @return
 */
public String getConfigFileLocation();
```

Scheduling a Definition

Set up an Archive Definition in Data Archive UI. However, you still need to create a schedule for the job. For each Archive Definition that you create, a Definition ID is part of the URL that is passed from the third-party Scheduler.

For example:

```
<base url>/jsp/RunDefinition.jsp?DefinitionID=<Definition ID>
```

Input: ID of the definition that you want to run.

Output: Schedule ID.

Retrieving the Current Status

You need to include the following program in the schedule job within the third-party scheduler. The program retrieves the current status. Wrap the program in a script so that it is called until the completed code is returned.

For example:

```
<cbase url>/jsp/GetJobStatus.jsp?JobID=<Job ID>
```

Input: Job ID that is in the UI or that is returned from the schedule API.

Output: One of the following codes:

Code	Description
C	Complete
T	Terminated
P	Paused
R	Running
E	Error
W	Warning

Or, the following message appears when there is an invalid Job ID or another error occurs:

```
Unable to get status for Job ID <Job ID>.
```

Step 2: Register the Implementation

To register the implementation, open the $APPLIMATION_HOME/webapps/applimation/WEB-INF/schedule.xml and locate the amScheduler bean entry. Update the bean entry to point to the new class implementation.

For example:

```
<bean id ="amScheduler"
class="com.applimation.schedule.quartz.impl.AMQuartzSchedulerImpl" singleton="true">
<property name="configFile" value="quartz.properties" />
</bean>
```

Step 3: Deploy the Implementation

Create a JAR for the implementation with any assisting classes and copy them to $APPLIMATION_HOME/webapps/applimation/WEB-INF/lib folder.s

CHAPTER 28

Parallel Processing in Data Archive

This chapter includes the following topics:

- Abstract, 99
- Introduction, 99
- Metadata Tuning, 100
- Performance-Related Source Repository Attributes, 100
- Archive Definition Setup, 102
- Restore Definition Setup, 103
- Java Parallelism - Processing Multiple Tables Simultaneously, 104
- Database Link Setup for Restore Cycles, 104

Abstract

This article describes the parallel processing capabilities of Data Archive 5.3. To improve archiving performance in Data Archive, you can use parallel processing in SQL statements. You can also process multiple tables simultaneously using java parallelism/threading.

Introduction

To improve archiving performance in Data Archive, you can use parallel processing in SQL statements. Data Archive parallelism utilizes the Parallel Query and Parallel DML Oracle database features. It invokes them by adding hints to the SQL statements it executes. You can also process multiple tables simultaneously using java parallelism/threading.

Metadata Tuning

To enable parallelism, complete the following tasks:

1. Set the relevant Degree of Parallelism attribute.
2. Add parallel hints to Data Archive metadata.

For example, to archive the PO_DISTRIBUTIONS_ALL table with parallelism, add the following parallel hint to the insert statement metadata for the table:

```
A.po_header_id IN (SELECT /*+ FULL(X) CARDINALITY(X, 1) PARALLEL(X, #) */
X.po_header_id FROM XA_4149_PO_HEADERS_INTERIM X WHERE X.purgeable_flag = 'Y')
```

The pound sign (#) is a placeholder that is replaced at runtime with the corresponding Degree of Parallelism attribute. This variable enables the same metadata to work with different degrees of parallelism, such as the Test and Production environments.

A Data Archive insert statement executes in parallel if you set the Insert Degree of Parallelism and add a parallel hint metadata where clause. Using the where clause above the engine constructs an insert statement as follows:

```
INSERT /*+ APPEND PARALLEL (Z, 8) */ INTO AM_STAGE.AA_3666113 Z
SELECT /*+ PARALLEL (A, 8) */ A.*, A.ROWID FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE
A.po_header_id IN (SELECT /*+ FULL(X) CARDINALITY(X, 1) PARALLEL(X, 8) */
X.po_header_id FROM XA_4149_PO_HEADERS_INTERIM X WHERE X.purgeable_flag = 'Y')
```

If the metadata does not include a PARALLEL(X, #) hint, the Data Archive engine does not process the table in parallel. Also, no parallel hint is added to the insert clause of the statement. This allows you to control which tables the Data Archive engine archives in parallel.

The FULL and CARDINALITY hints ensure that the optimizer uses the proper execution plan when selecting the data to archive. The FULL hint suggests that Oracle does a full table scan on the interim table. The CARDINALITY hint tells the optimizer that the subselect only returns one row, which of course is not accurate, but it ensures that the ERP table is accessed using an index.

It is important to tune the statement properly for the parallel processing to work efficiently. Parallel and other hints might be required in multiple places in the metadata statement text, often combined with other hints, to be effective and improve performance. If you need help tuning one or more statements, contact Informatica Global Customer Support.

The Data Archive engine can also execute business rule update statements in parallel. The implementation is the same. The Update Degree of Parallelism attribute controls the parallelism for these update statements.

Data Archive does not require metadata changes to support delete parallelism. If the Delete Degree of Parallelism attribute is greater than one, the Data Archive engine adds the necessary parallel hints automatically. However, it only processes tables without a primary key metadata constraint in parallel if the Use Oracle Parallel DML For Delete source repository attribute is enabled and the metadata delete statement contains a parallel hint.

Performance-Related Source Repository Attributes

Use Staging

If enabled, the Data Archive engine temporarily copies the data to be archived to staging tables on the ERP instance. Enable this option to increase performance.

Use ROWID for Delete

If enabled, the Data Archive engine stores the ROWID of the source rows in the staging tables and the Delete from Source step deletes rows using the ROWID. This is the fastest way to delete rows in Oracle and the recommended setting.

Use Oracle Parallel DML for Delete

The way Data Archive processes delete statements during the Delete from Source step is controlled by the Use Oracle Parallel DML for Delete source repository attribute. If this attribute is enabled, Data Archive uses Oracle's parallel DML. Otherwise, the Data Archive engine controls delete parallelism

Oracle Parallel DML

Delete parallelism is achieved by using Oracle's parallel query/DML features. These features are invoked by adding parallel hints to the delete statements that are executed. This method is the fastest way in Oracle to delete data. The following example shows a typical delete statement using Oracle parallelism:

```
DELETE /*+ PARALLEL(A,8) */ FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE (ROWID, PO_DISTRIBUTION_ID) IN (
SELECT /*+ CARDINALITY(X,1) PARALLEL(X,8) */ APPLIMATION_ROW_ID,
PO_DISTRIBUTION_ID FROM AM_STAGE.AA_3666113 X)
```

Parallel Delete Controlled by Data Archive

Data Archive spawns and controls multiple delete processes. Each delete process is a Java thread, where each worker processes a range of rows. This method is slower than Oracle's parallelism, but does not lock the table. Therefore, use this option when you schedule archive cycles during business hours.

The following are sample delete statements executed by parallel worker threads:

```
DELETE FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE (ROWID, PO_DISTRIBUTION_ID) IN (
SELECT /*+ CARDINALITY(X,1) */ APPLIMATION_ROW_ID, PO_DISTRIBUTION_ID FROM
AM_STAGE.AA_3666113_T1 X WHERE APPLIMATION_ROW_NUM BETWEEN 1 AND 100000)

DELETE FROM PO.PO_DISTRIBUTIONS_ALL A
WHERE (ROWID, PO_DISTRIBUTION_ID) IN (
SELECT /*+ CARDINALITY(X,1) */ APPLIMATION_ROW_ID, PO_DISTRIBUTION_ID FROM
AM_STAGE.AA_3666113_T1 X WHERE APPLIMATION_ROW_NUM BETWEEN 1000001 AND 200000)
```

Consider the following factors before you use Oracle parallel DML:

- Oracle parallelism during delete operations requires an exclusive lock on the table that is deleted, preventing other users from updating the same table. Read access on these tables is still possible. Use Oracle parallelism if archive cycles are scheduled outside of business hours when no users need write access to the ERP system.
- The source or ERP tables need to support parallel DML. Tables that you created in an Oracle 8/8i database, which you upgraded to 9i/10g, do not support parallel DML. Use the query below to check the relevant table property:

  ```
  select
   a.property,
   decode(bitand(a.property,536870912), 0, 'DISABLED', 'ENABLED') pdml_enabled
  from sys.tab$ a, dba_objects b
  where a.obj# = b.object_id
  and b.object_name = '&table_name' and b.owner = '&owner';
  ```

If a table does not support parallel DML, Oracle provides a means to enable it. For more information, contact Informatica Global Customer Support.

Archive Definition Setup

Analyze Interim

This attribute controls when optimizer statistics are gathered for interim tables during Generate Candidates. Analyze the interim for optimal table and index statistics for the Oracle Cost Base Optimizer.

Select one of the following options:

- After Insert. The interim table is analyzed after data is inserted.
- After Insert and Update. The interim table is analyzed after every insert and update statement.
- After Update. The interim table is analyzed after business rule update statements.
- None. No statistics are gathered.

Delete Commit Interval

This attribute controls the commit frequency during delete operations during the Delete from Source step. If the value is 0 or if the table processed does not have a primary key metadata constraint defined, then the delete is executed as a single statement and no commit interval is used.

Insert Commit Interval

This attribute controls the commit frequency during insert operations when you cannot use an "INSERT AS SELECT" type statement.

You cannot use Insert as Select in the following cases:

- The table has a LONG column.
- If the Database Link to Source destination repository attribute is not set during the Copy to Destination step.
- If the table processed has a user-defined type column, such as WF_EVENT_T, and the database version is Oracle 9i or lower during the Copy to Destination step.

The Generate Candidates and Copy to Staging steps use Insert as Select type statements except when the table contains a LONG column. Undo usage is minimal during these steps, because the Data Archive engine adds an APPEND hint to the insert statement. In this case, Oracle does a direct path data load above the table high water mark.

Degree of Parallelism

In general, the possible degree of parallelism depends on the following factors:

- The number of CPUs available on the database server. Use the following query to find out about the number of CPUs available to an Oracle instance:

  ```
  select value from v$parameter where name = 'cpu_count';
  ```

- The setting of the parallel_max_servers initialization parameter. Use the following query to check the current setting:

  ```
  select value from v$parameter where name = 'parallel_max_servers';
  ```

- Set this parameter to at least twice the value of the degree of parallelism chosen. For example, to execute a delete statement with a degree of 8, Oracle needs to spawn 16 parallel threads, where 8 threads work on the necessary select to retrieve the rows and 8 perform the delete.

Delete Degree of Parallelism

The degree of parallelism used for delete operations during the Delete from Source step.

Insert Degree of Parallelism

The degree of parallelism used for insert operations during the Generate Candidates, Copy to Staging, and Copy to Destination steps.

Update Degree of Parallelism

The degree of parallelism used for update operations during the Generate Candidates step.

Restore Definition Setup

Delete Commit Interval

Unused. Restore cycles issue a single delete statement. Because single commit delete statements possibly delete large volumes of data, size the UNDO tablespace on the history database accordingly.

Insert Commit Interval

This attribute controls the commit frequency during insert operations when you cannot use an Insert as Select statement.

You cannot use Insert as Select in the following cases:

- The table has a LONG column.
- The Database Link to Source destination repository attribute is not set during the Copy to Destination step.
- The table processed has a user defined type column, such as WF_EVENT_T, and the database version is Oracle 9i or lower during the Copy to Destination step.

Delete Degree of Parallelism

Currently unused.

Insert Degree of Parallelism

The degree of parallelism used for insert operations during the Generate Candidates, Copy to Staging, and Copy to Destination steps.

Java Parallelism - Processing Multiple Tables Simultaneously

The Data Archive engine can process multiple tables simultaneously depending on the step executed and the value of the corresponding Degree of Parallelism attribute.

The maximum number of concurrent processes or threads depends on the value of the informia.maxActiveAMThread property. Set this property in the conf.properties file located in the web server root directory. Default is 10. This property is a system-wide setting. It does not apply to a single archive cycle. It limits the total number of java threads the Data Archive engine executes in parallel.

Use the following URL to see all active and pending threads in the Data Archive thread queue manager:

```
http://<hostname>:port/jsp/tqm.jsp
```

Generate Candidates

This step executes in step order.

Build Staging

This step processes multiple tables simultaneously up to the maximum number of threads allowed.

Copy to Staging

This step processes the tables sequentially in step order and uses Oracle parallel DML if the insert degree of parallelism is greater than one. However, you can set the insert degree to one to enable java parallelism. Data Archive processes multiple tables simultaneously up to the maximum number of threads allowed.

Validate Destination

This step processes multiple tables simultaneously up to the maximum number of threads allowed.

Copy to Destination

This step processes the tables sequentially in step order and uses Oracle parallel DML if the insert degree of parallelism is greater than one. However, you can enable Java parallelism by setting the insert degree to one. Data Archive processes multiple tables simultaneously up to the maximum number of threads allowed.

Restore cycles process the Copy to Destination step sequentially in step order.

Purge Staging

This step processes multiple tables simultaneously up to the maximum number of threads allowed.

Database Link Setup for Restore Cycles

To increase performance, Data Archive can use database links during Restore cycles. The Data Archive engine copies data using Insert as Select SQL statements. However, since the Copy to Destination step

during Restore possibly inserts into heavily indexed ERP tables, you need to properly size the UNDO tablespace on the ERP instance.

To set up the restore database links:

1. Connect to the ERP source instance as the application user.
2. Create a private database link that points to the staging user on your history instance.
3. Edit the Restore Destination Repository and enter the name of the database link created in the Database Link to Source attribute.
4. Connect to the history instance as the history agent user and create a private database link that points to the application user on the ERP instance.
5. Edit the Restore Source Repository and enter the name of the database link created in step 4 in the Database Link to Production (Restore only) attribute.

CHAPTER 30

Seamless Access Configuration for Siebel Applications

This chapter includes the following topics:

- Abstract, 106
- Objective, 106
- Dedicated Client Mode, 107
- Web Client Mode, 108

Abstract

You can configure the seamless Access Layer in the Siebel application using Dedicated Client Mode and Web Client Mode. This article explains how to connect the Siebel application by configuring the seamless Access Layer.

Objective

This document describes how to configure the seamless Access Layer in the Siebel application.

Use the Siebel application through the combined user (AMQUERY by default) which is created during the configuration of the Data Access Layer.

You can visualize data through to the Siebel application in the following ways:

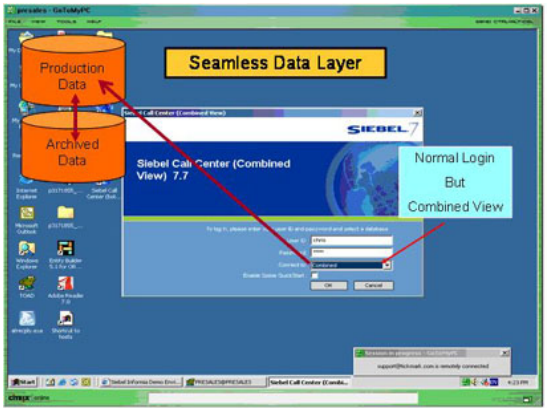
- Dedicated Client Mode
  - Install the client software.
  - Change the configuration file to connect to the Siebel instance.
- Web Client Mode.
  - Install Siebel Gateway Name Server, Enterprise Server, Siebel Server, and Siebel Web Server Extension.
  - Make sure that the new Siebel Server repository is same as the Production instance.
  - Create user "AMCOMB" on the Production database.

- Use the following syntax:

```
Grant SSE ROLE to AMCOMB
Grant TBLO_ROLE to AMCOMB
Grant select all to SADMIN on AMCOMB
```

- Change the Object Manager parameters and Server Datasource profile parameters.

Dedicated Client Mode



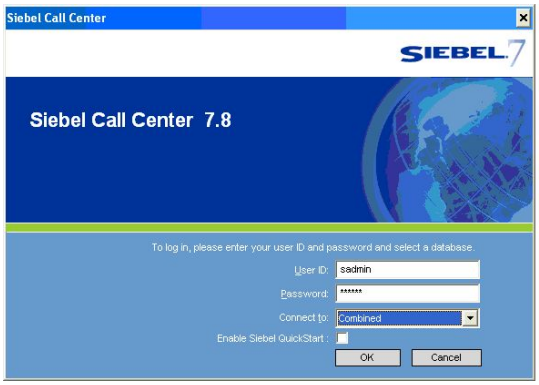
Install the Siebel Web Client software.

To connect change the following file:

C:\Program Files\Siebel\7.8\web client\BIN\ENU\uagent.cfg

```
[DataSources]
Local                    = Local
Sample                   = Sample
ServerDataSrc            = Server
ApplimDataSrc            = Combined
GatewayDataSrc           = Gateway

[ServerDataSrc]
Docked                   = TRUE
ConnectString            = $(ConnectString)
TableOwner               = $(TableOwner)
DLL                      = sscddcli.dll
SqlStyle                 = DB2
MaxCachedCursors         = 16
MaxCachedDataSets        = 16
ReverseFillThreshold     = 100
CaseInsensitive          = FALSE
InsensitivityFactor      = 2
FileSystem               = C:\PROGRA~1\Siebel\7.8\WEBCLI~1\bin\att
GatewayAddress           = sblvm01
EnterpriseServer         = Sieb78
CurrentSQLID             = Startup
MaxCursorSize            =
PrefetchSize             =
ChartServer              = localhost:8001
ChartImageFormat         = png

[ApplimDataSrc]
Docked                   = TRUE
ConnectString            = SEADB78 (SID of the Siebel Instance)
TableOwner               = AMQUERY
```

```
DLL                     = sscdo90.dll
SqlStyle                = OracleCBO
MaxCachedCursors        = 16
MaxCachedDataSets       = 16
ReverseFillThreshold    = 100
CaseInsensitive         = FALSE
InsensitivityFactor     = 2
FileSystem              = C:\PROGRA~1\Siebel\7.8\WEBCLI~1\bin\att
GatewayAddress          = sblvm01
EnterpriseServer        = Sieb78
CurrentSQLID            = Startup
MaxCursorSize           =
PrefetchSize            =
ChartServer             = localhost:8001
ChartImageFormat        = png
```



Click Start > All Programs > Siebel Web Client 7.8 > Siebel Call Center - ENU

Connect to the application using the SADMIN user.

If you cannot log in, then you have to execute as SYSTEM on the Siebel instance the following queries:

```
GRANT SELECT ANY TABLE TO SSE_ROLE;
GRANT SSE_ROLE TO AMQUERY;
```

Web Client Mode

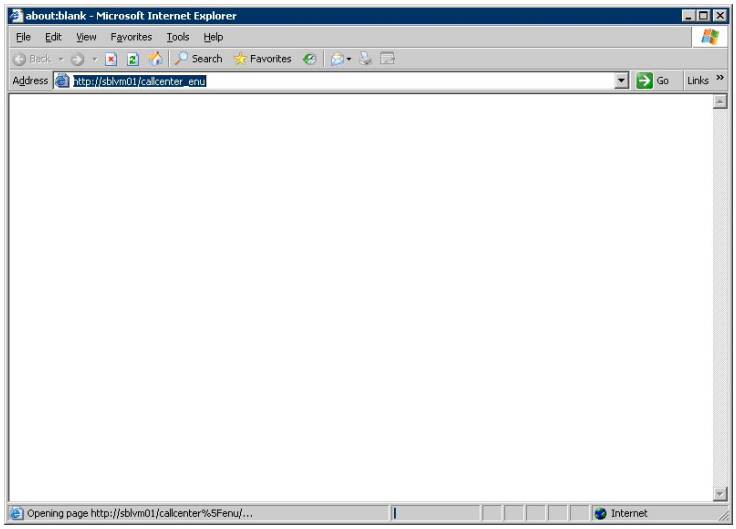
This section describes how to configure the Siebel Call Center Application to connect on the combined schema in the Siebel Instance.

Before you configure the Web Client mode, you must configure the Seamless Access Layer using the Archive UI. For more information about how to configure, see *SeamLessDataAccess-Steps.pdf*

Log in the Administrator - Server Configuration.

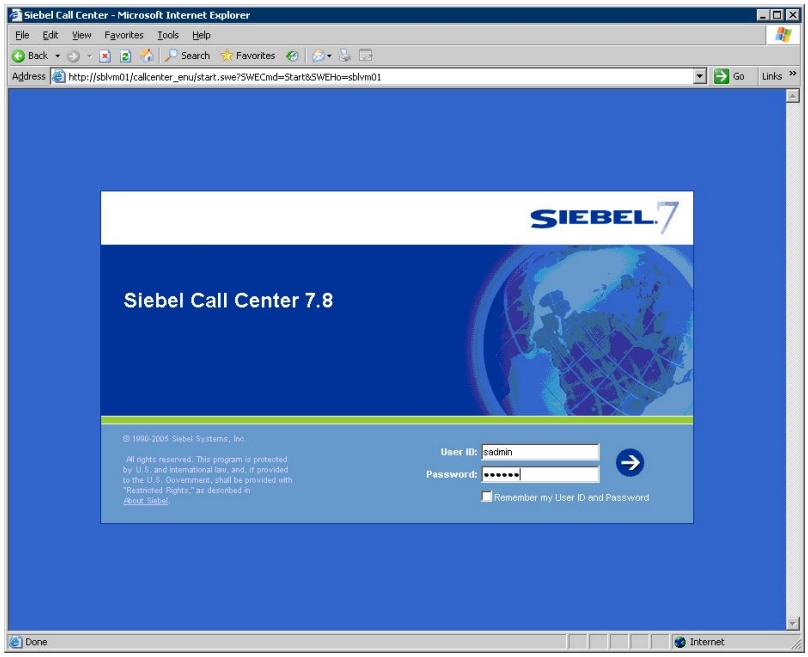
To start the Siebel configuration, launch the Internet Explorer and go to the following URL:

http://Servername/callcenter_enu

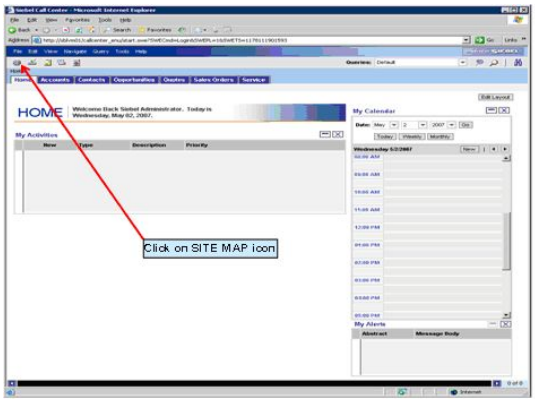


When the login screen appears, login as the SADMIN user.

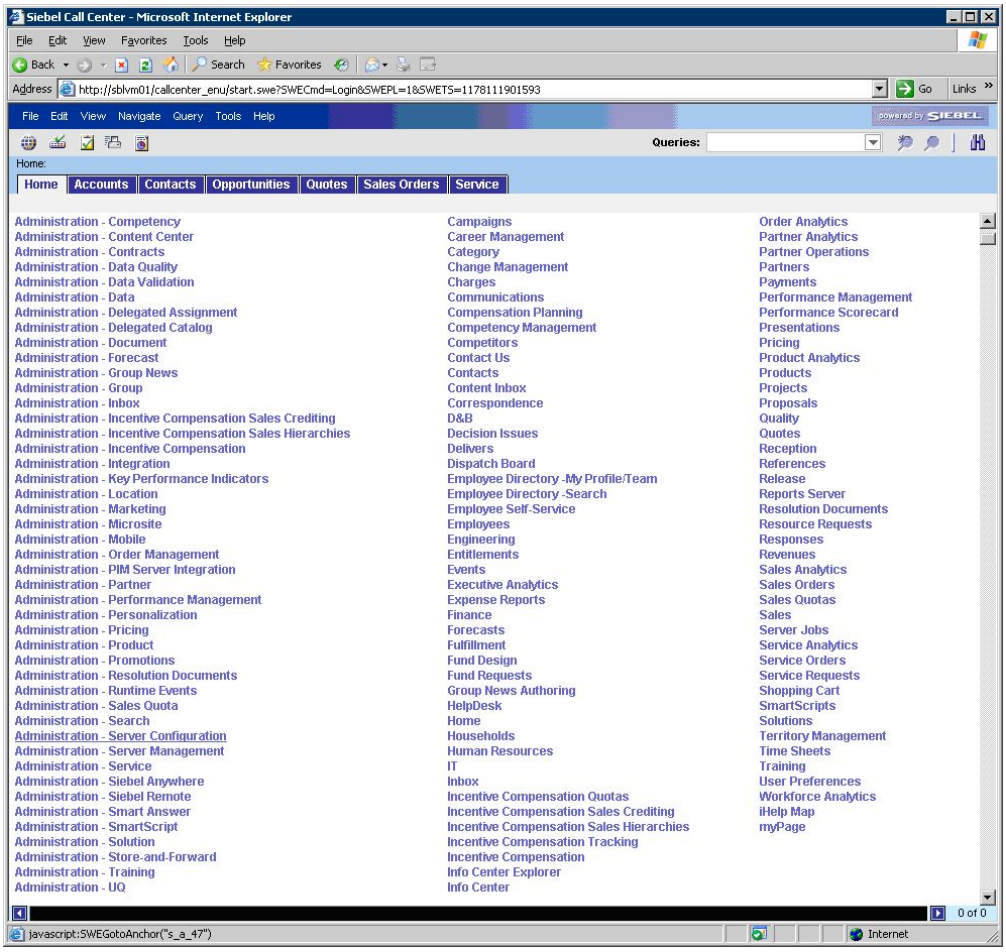
As SADMIN, you can configure the Object Manager and the Server DataSource profile configuration.



Click the Site Map icon to see all applications links.

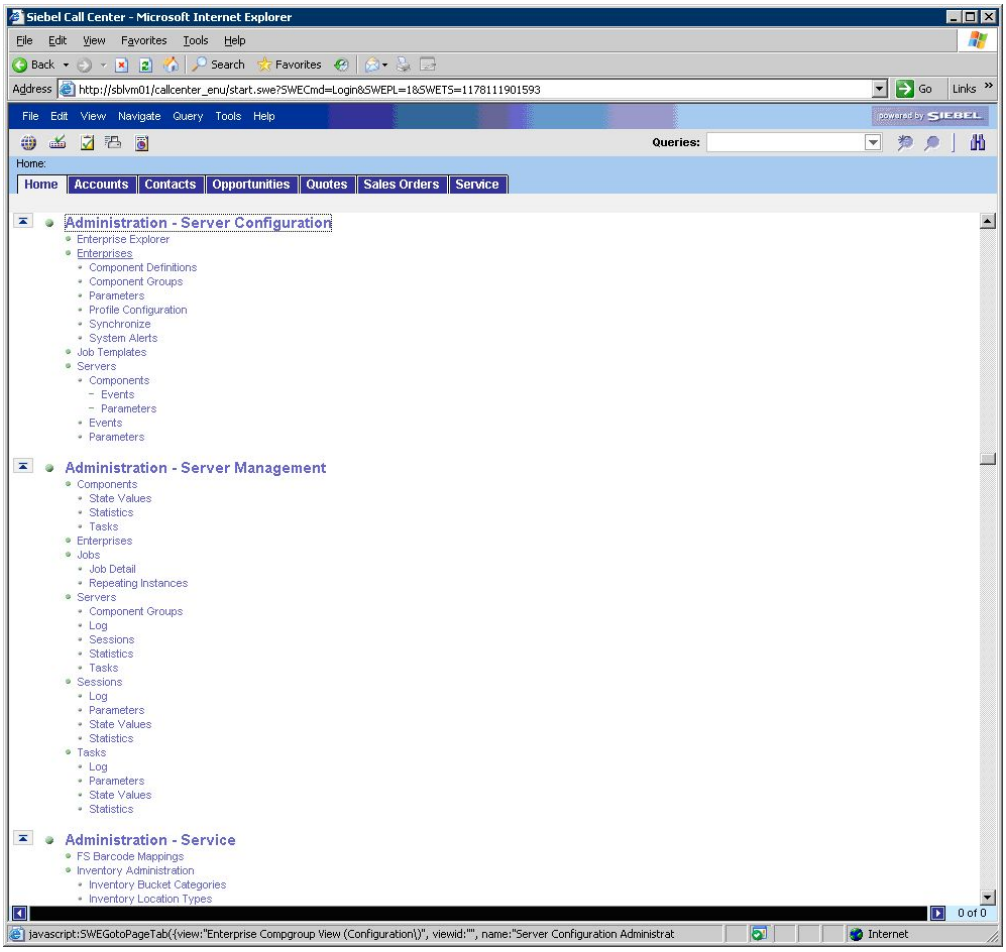


After you click the SITE MAP icon, the following screen appears:

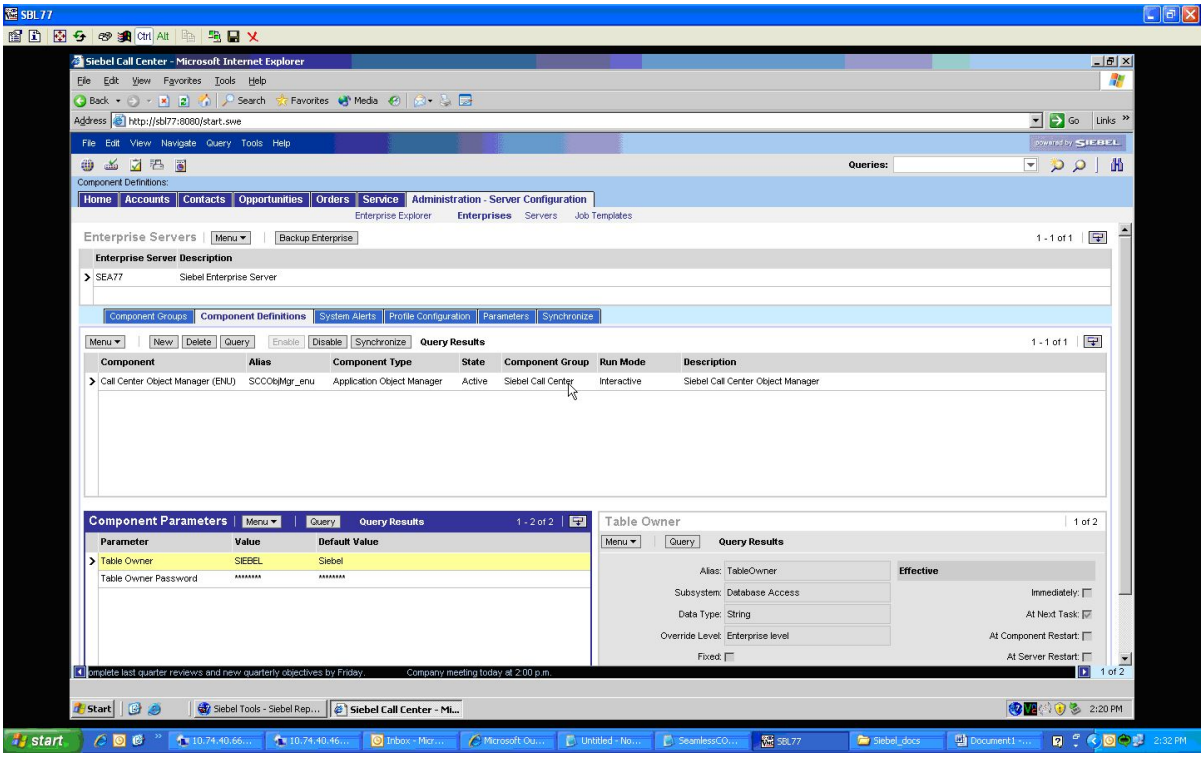


Click Administration - Server Configuration to get the list of Siebel configuration screens.

Click the Enterprises link to open the server configuration screen.



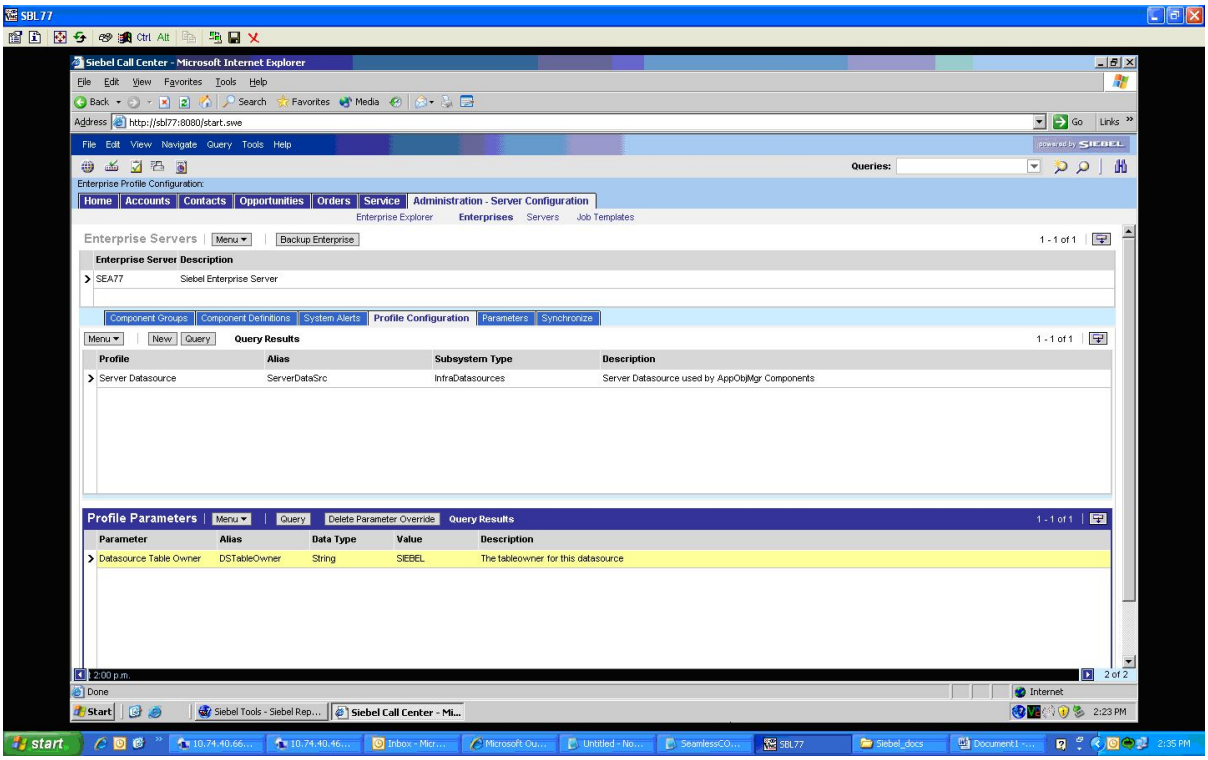
The following figure shows where you can configure the respective Object Manager and the ServerData source profile:



In the component parameters section of the Object Manager, update the parameter values. The following table shows the parameters to update:

Parameter	Value
Password	SADMIN
UserName	SADMIN
Table Owner	AMCOMB
Table Owner Password	AMCOMB

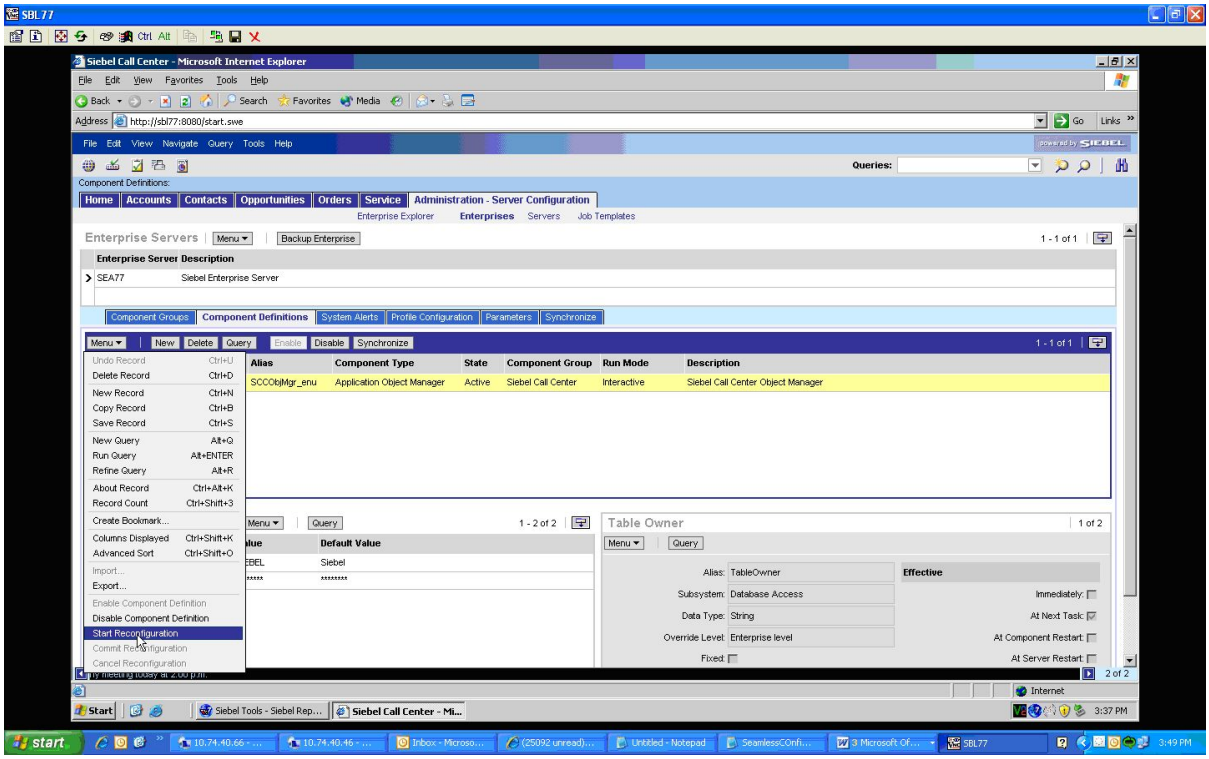
On the Profile Configuration tab, change the profile parameters for "Server Datasource profile":



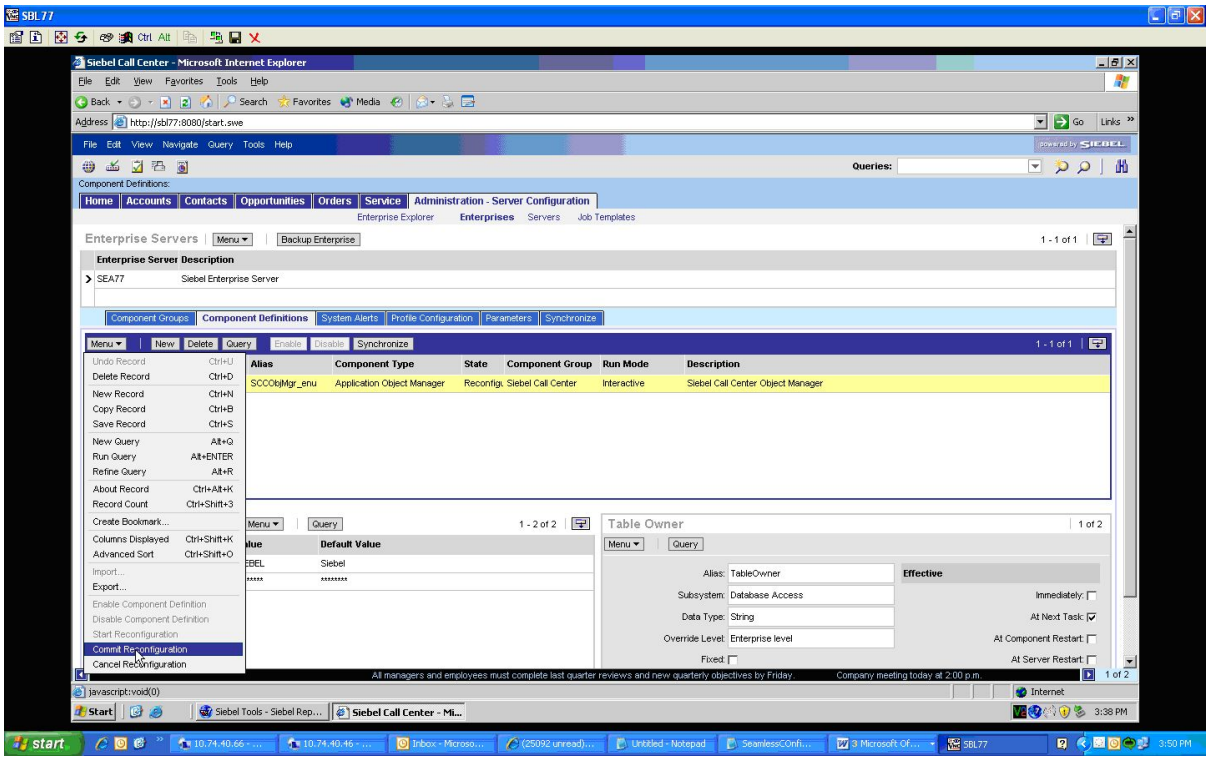
Use the values listed in the following table for profile parameters:

Parameter	Value
Data source Connect String	SID of your prod Siebel instance
Default username password	Siebel
Datasource Primary Enterprise Server	Siebel
Privileged username password	AMCOMB
Datasource Table Owner	AMCOMB

Reconfigure the Siebel Server after specifying the values for profile parameters. Click the Menu Button in the Component Definitions tab.

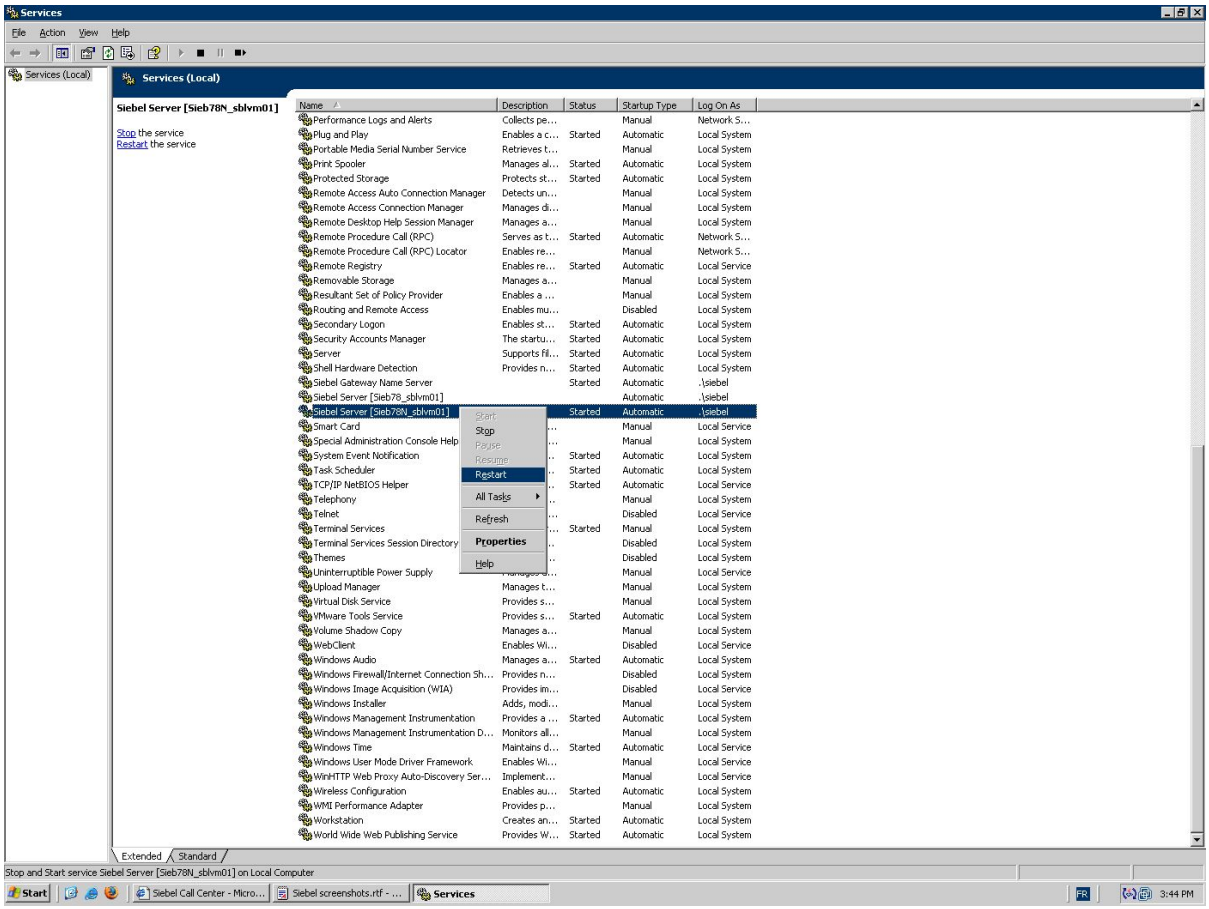


In this menu, click Start Reconfiguration menu item. Click the menu button again and then click 'Commit Reconfiguration' as shown in the following figure:



Restart the Siebel Server.

Click Start -> Programs -> Administrative Tools -> Services to launch the Services windows application.



Right-click Siebel Server and then click Restart.

CHAPTER 32

Seamless Access Setup for Oracle E-Business Suite

This chapter includes the following topics:

- Abstract, 116
- Overview, 116
- Setting Up Seamless Access Between Informatica Data Archive and Oracle E-Business Suite, 118
- Appendix, 127

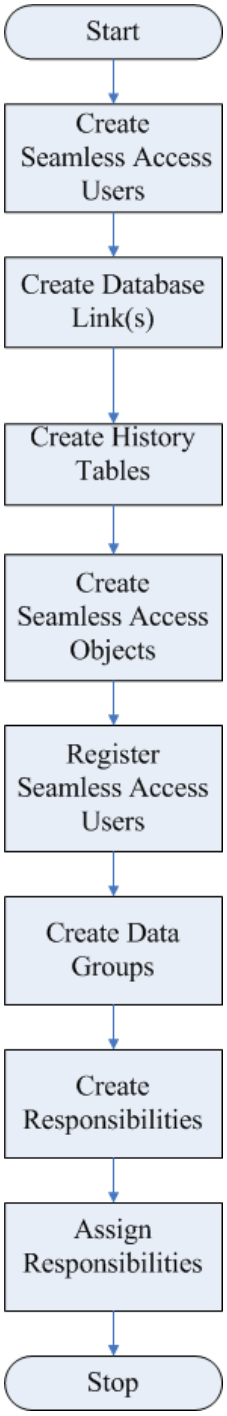
Abstract

This article explains how to setup seamless access between Informatica Data Archive and Oracle E-Business Suite.

Overview

You can setup seamless access between Informatica Data Archive and Oracle E-Business Suite.

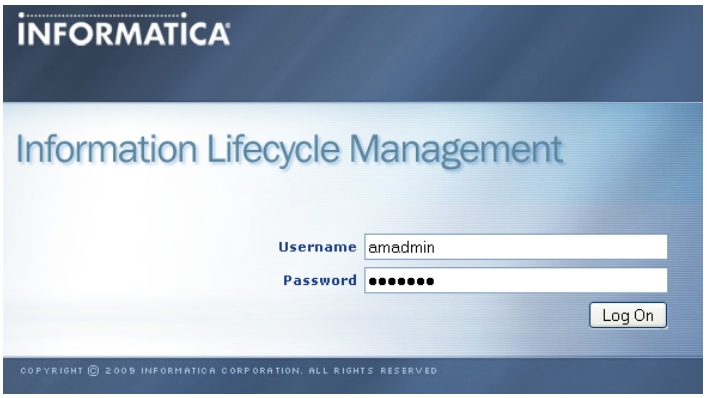
The following figure explains the steps to set seamless acess between Informatica Data Archive and Oracle E-Business Suite:



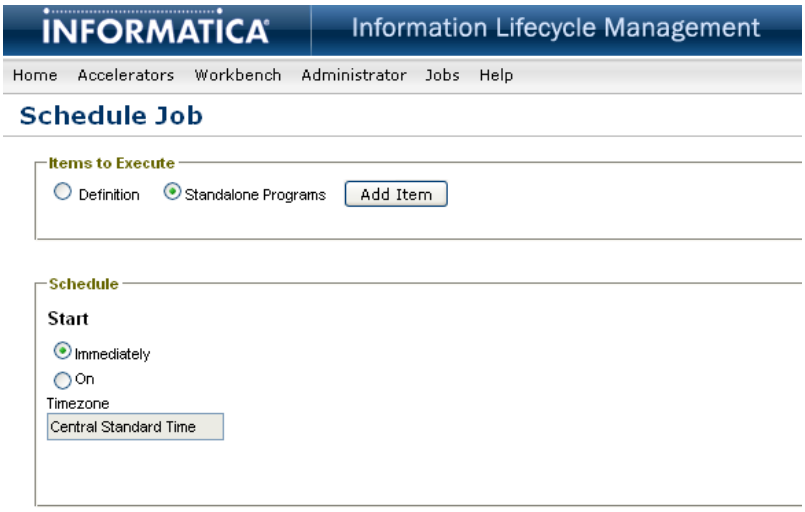
To create seamless access users, see sample scripts in appendix. There is an Archive only seamless access schema and a Combined seamless access schema. The scripts create the tables, synonyms, and views required for seamless access.

Setting Up Seamless Access Between Informatica Data Archive and Oracle E-Business Suite

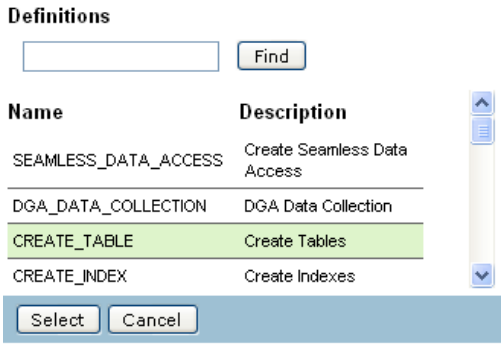
1. Log in to Informatica Data Archive.



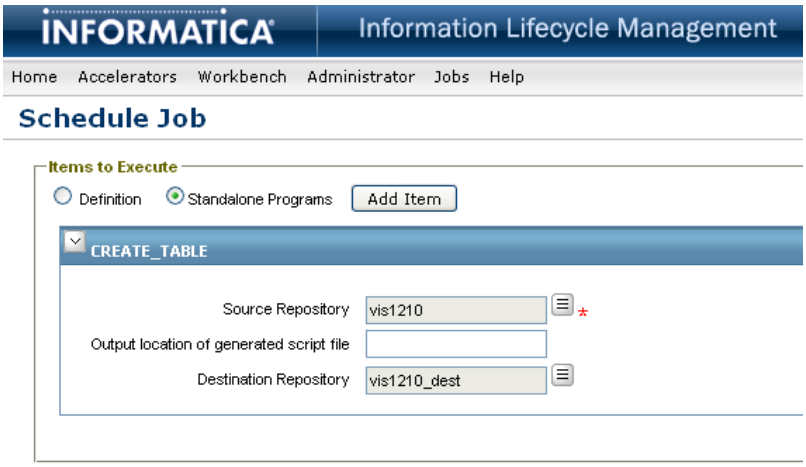
2. Click **Schedule Jobs**.



3. Choose the CREATE_TABLE definition and click **Select**.



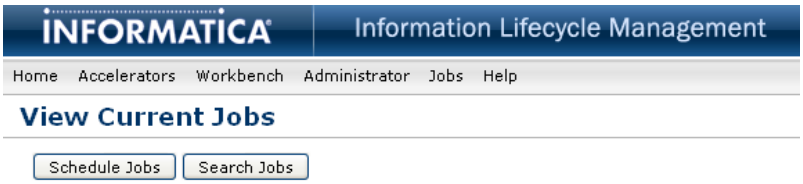
4. Choose the source and destination repository from the drop-down list.



5. Click **Schedule** to run the job. You can navigate to**Jobs** > **View Current Jobs** to monitor the progress of the job.

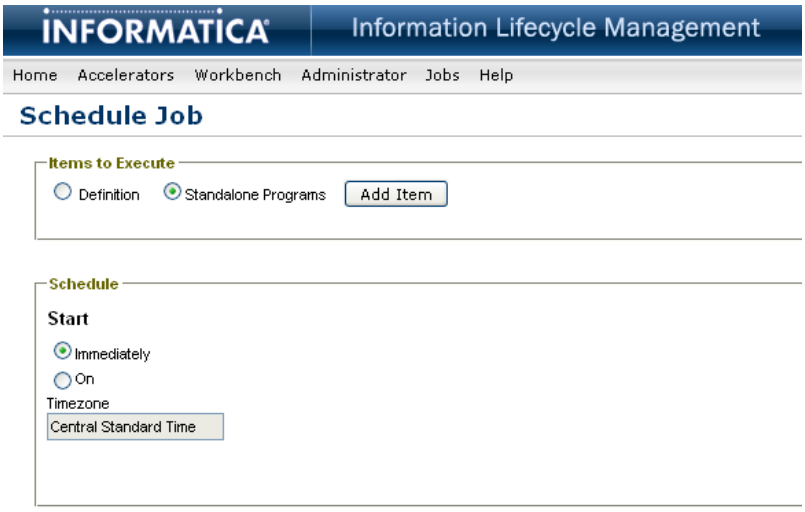
   The job creates a table in the history schema of the destination repository for every table processed by any entity defined in the product family version of your source E-Business Suite repository. If you have added custom tables to any of the entities, the job creates the custom tables. If you add custom tables later, run the job again. If you apply an Oracle patch to your E-Business Suite instance and modify any of the tables created by the job, you need to be run the job again.
   Wait until the CREATE_TABLE job completes before moving to the next step.

6. Click **Schedule Jobs**.

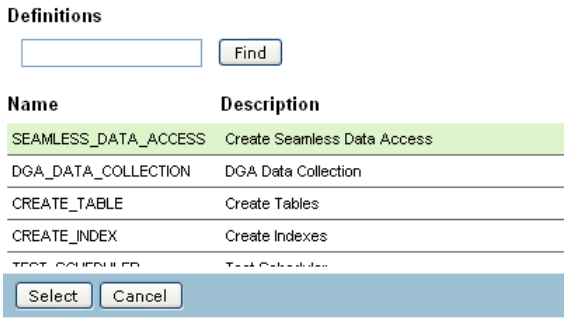


   There are synonyms to most of the objects in the APPS schema for both the Archive Only and Combined seamless access schemas. For all of the tables created in the previous step there are views. In the archive only schema the view selects from the history table across a database link. In the combined schema the view selects from the history table across a database link and union that with a select from the APPS table. We call these views base level seamless access views. We also create archive only and combined schema views for APPS views that reference these base views, they are called dependent views.

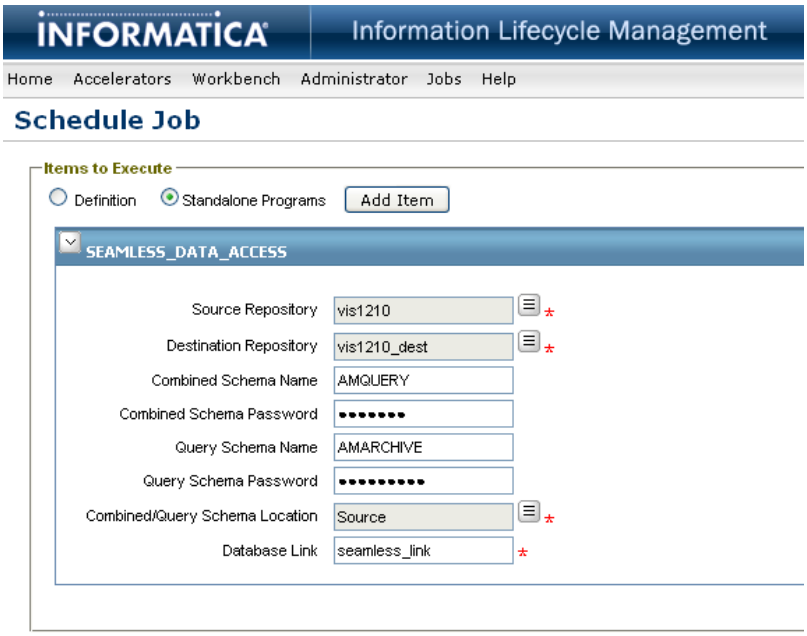
7. Click **Add Item**.



8. Choose the SEAMLESS_DATA_ACCESS definition and click **Select.**



9. Fill in all the information.



10. Choose the Source and Destination repository from the drop-down list.

Combined schema name is AMQUERY in this example and in the example script in the appendix. Query schema name is AMARCHIVE in this example and in the example script in the appendix.

11. Choose the Combined/Query schema location as Source from the drop-down list.

    Database Link is a public link or two private links, one owned by the archive only schema and one to the combined schema, that connects to the history public user on the destination repository.

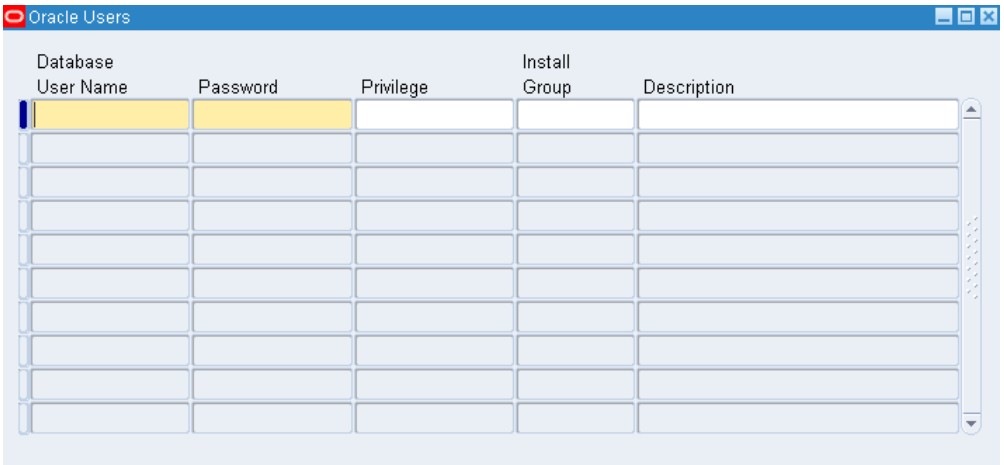
12. Click **Schedule**. You can navigate to**Jobs** > **View Current Jobs** to monitor the progress of the job.

    Wait until the job completes before moving to the next step.

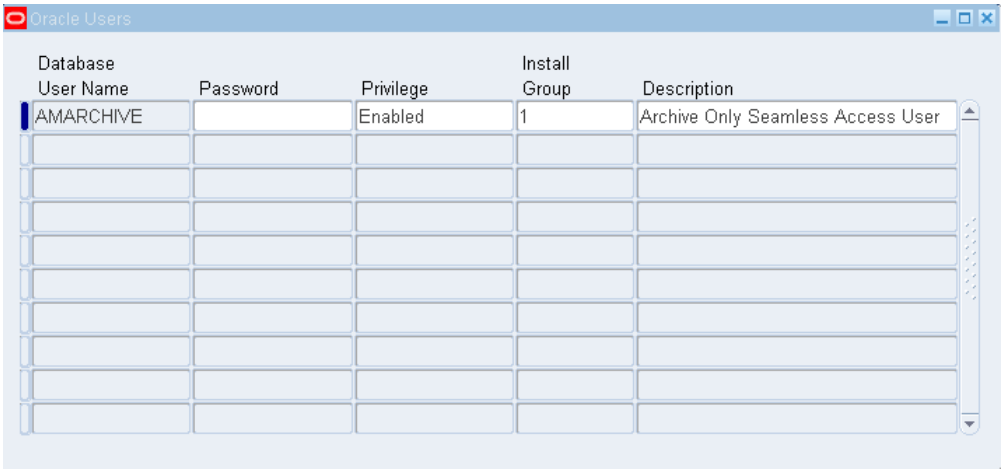
    Now that the seamless access objects have been created it is time for the APPS portion of the setup. It includes registering the seamless access schemas, creating data groups for the seamless access schemas, and creating responsibilities based on those data groups.

13. Log into your Oracle E-Business Suite and choose the System Administrator responsibility.

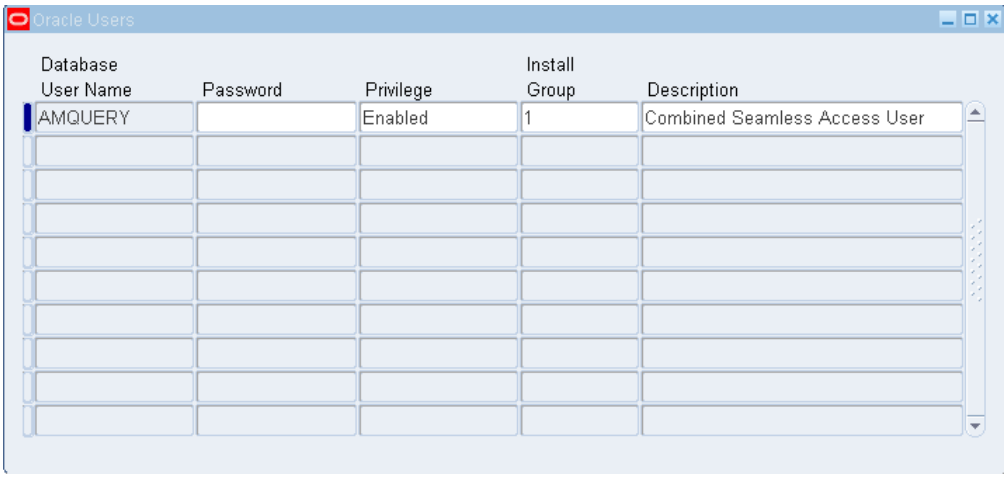
14. Navigate to **Security** > **ORACLE** > **Register.**.



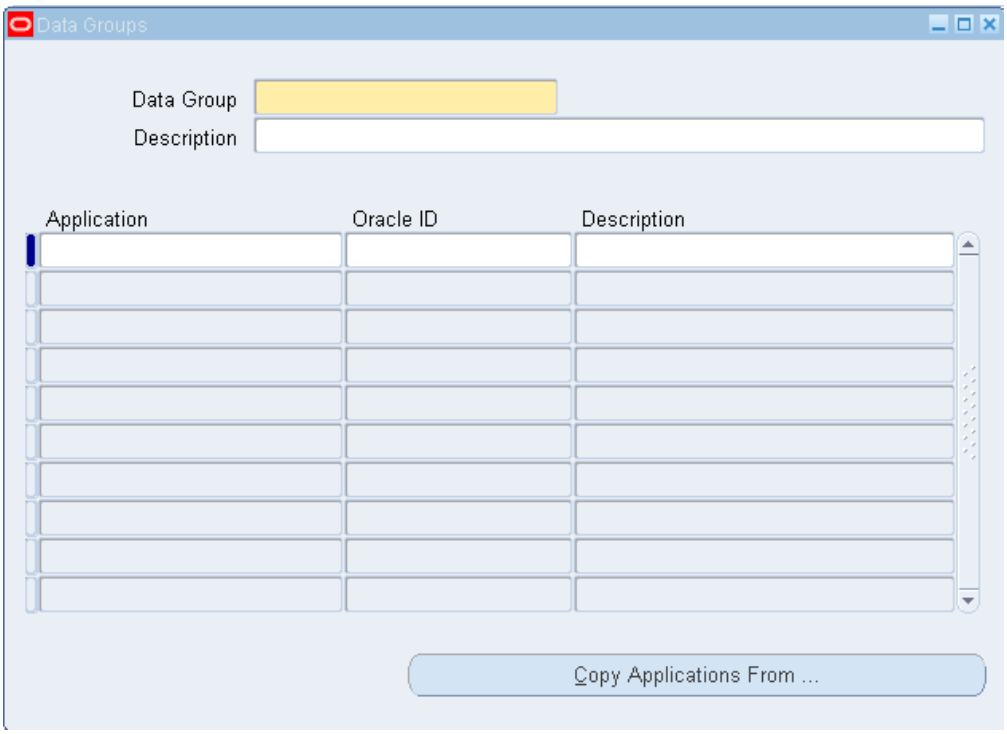
15. Register the archive only seamless access user.



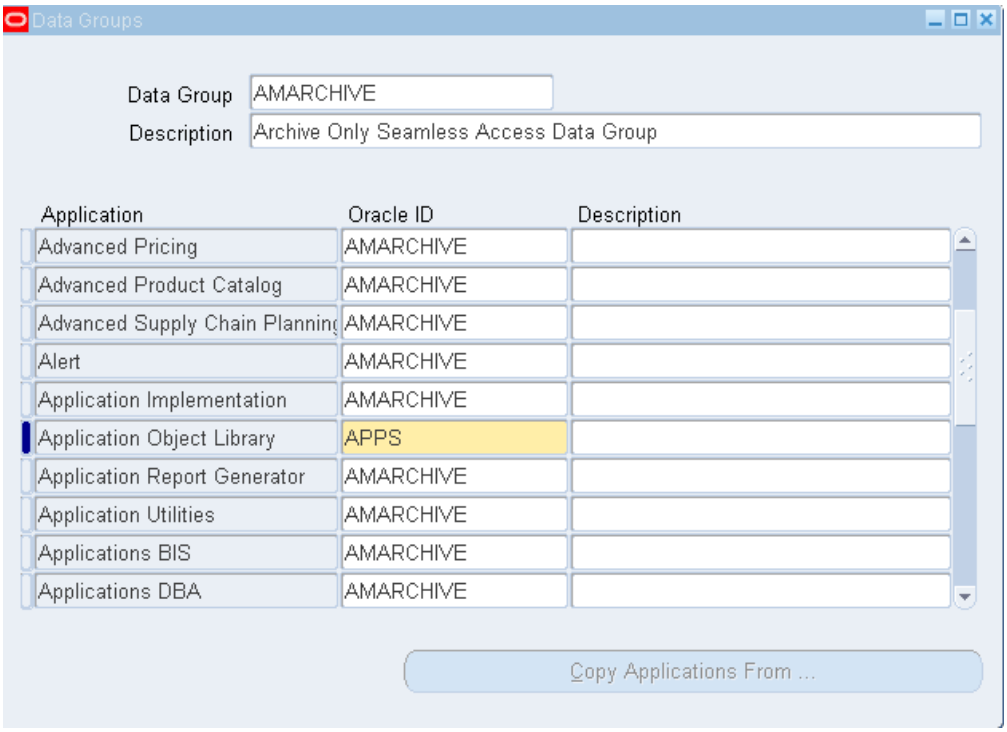
16. Register the combined seamless access user.



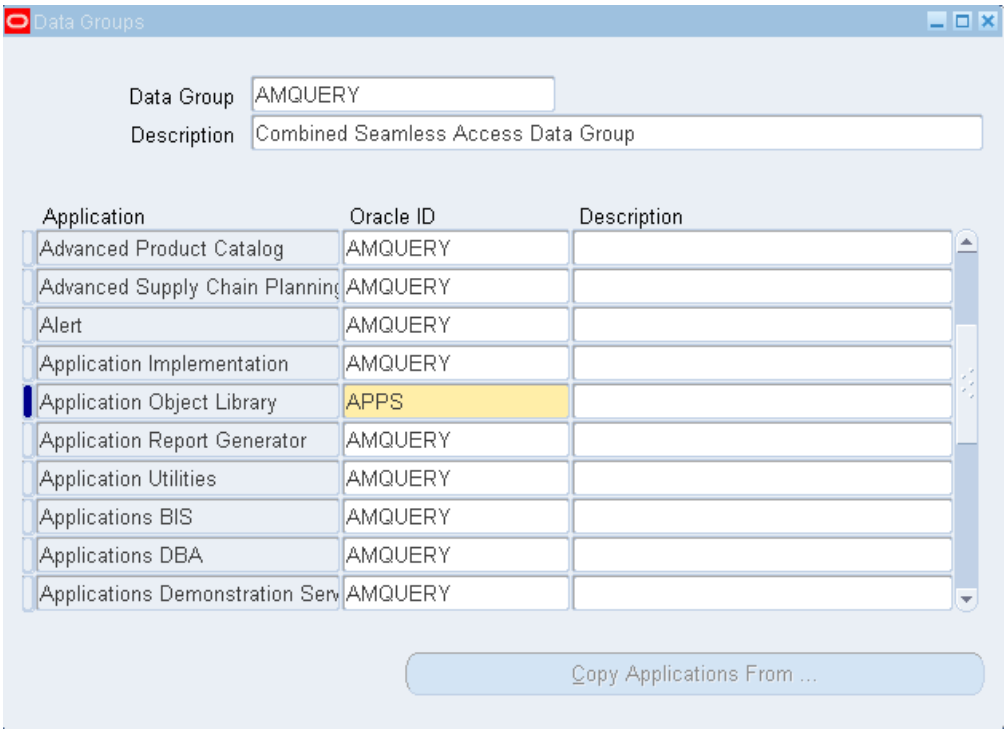
17. Navigate to **Security** > **ORACLE** > **Data Group**.



18. Enter the Data Group, Description, and hit the Copy Applications From button. Choose the Standard data group and click OK. Change the Oracle ID for all of the Applications except "Application Object Library" for the archive only data group to AMARCHIVE.



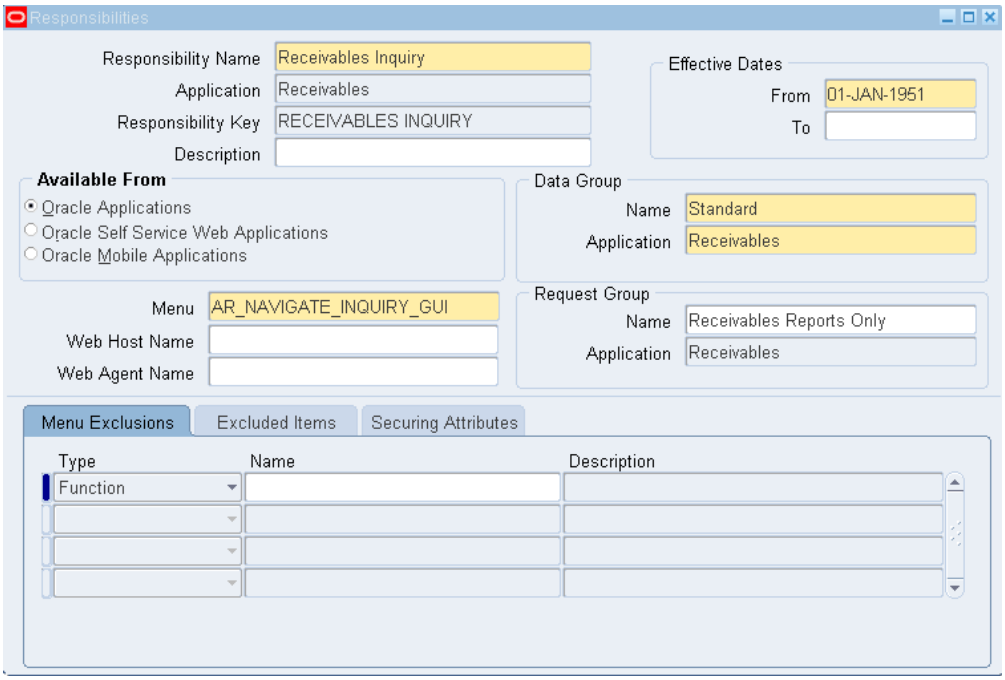
19. Perform the same steps for the combined data group and change the Oracle ID for all of the Applications except "Application Object Library" to AMQUERY.



You are now ready to create archive and combined versions of your select inquiry only responsibilities. Seamless access is only intended as a method of looking at data or reporting so choose inquiry responsibilities necessary to give access to the archived data or a combination of the archived and non-

archived data for users that will require access to the older archived data. If something needs to be changed then the data will have to be restored and then archived again later after making the changes in one of your standard responsibilities. If the number of responsibilities is large you can download the ARC_FNDLOAD.zip file for a more automated responsibility creation process. The zip file includes a readme with instructions for that process and it will not be covered here.

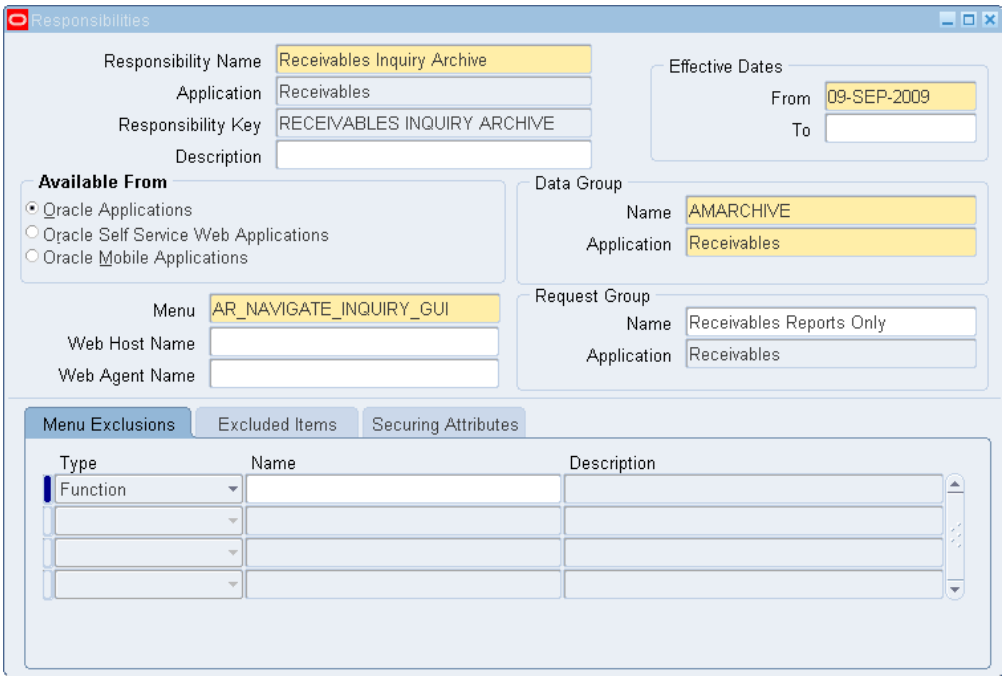
20. Navigate to **Security** > **Responsibility** > **Define** and query up the standard version of the inquiry responsibility you wish to clone.



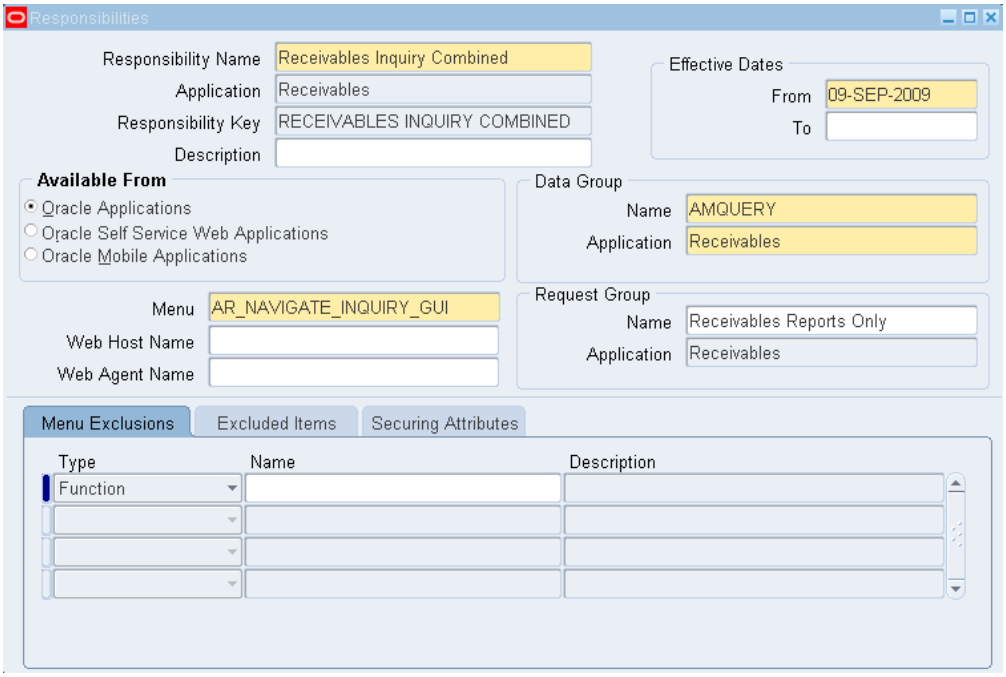
Make a note of all the defined values including Menu Exclusions, Excluded Items, and Securing Attributes if any are defined.

21. Navigate to **Security** > **Responsibility** > **Define** and create a new archive only responsibility with the following differences from the original responsibility.

22. Choose a new responsibility name, responsibility key, and choose the AMARCHIVE data group instead of the standard data group. Don't forget to include the Menu Exclusions, Excluded Items, and Securing Attributes if any are defined.

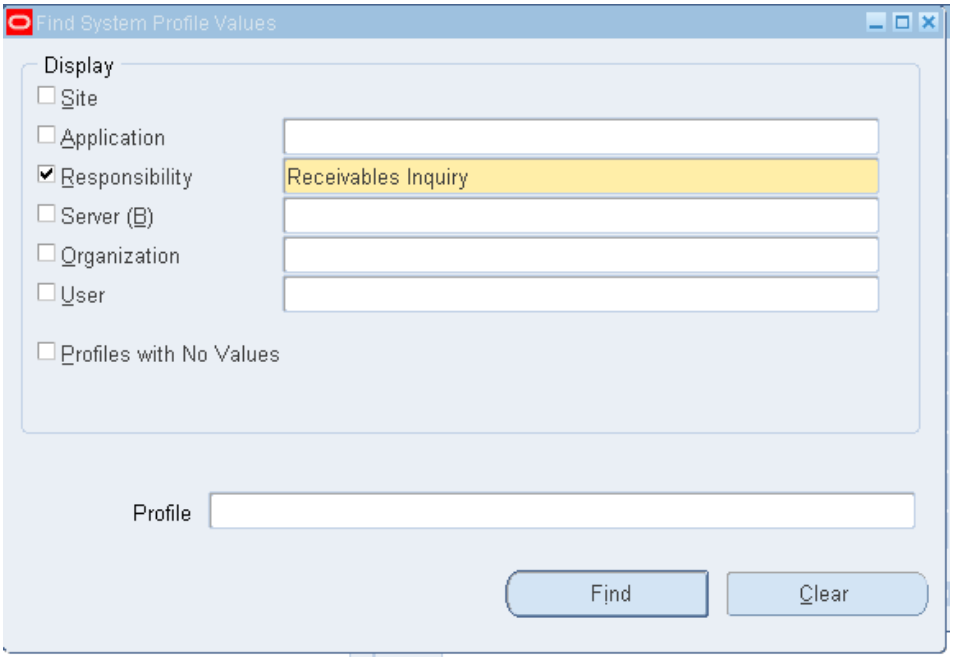


23. Navigate to **Security** > **Responsibility** > **Define** and create a new combined responsibility with the following differences from the original responsibility.
24. Choose a new responsibility name, responsibility key, and choose the AMQUERY data group instead of the standard data group. Don't forget to include the Menu Exclusions, Excluded Items, and Securing Attributes if any are defined.

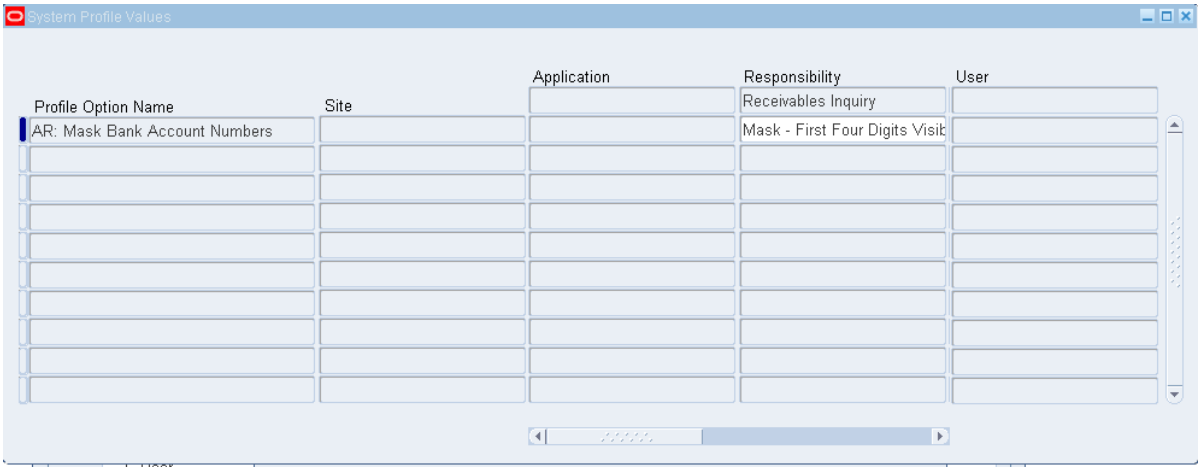


The next step is the check for any profile options that are set for the original responsibility so the same options can also be set for the archive and combined versions.

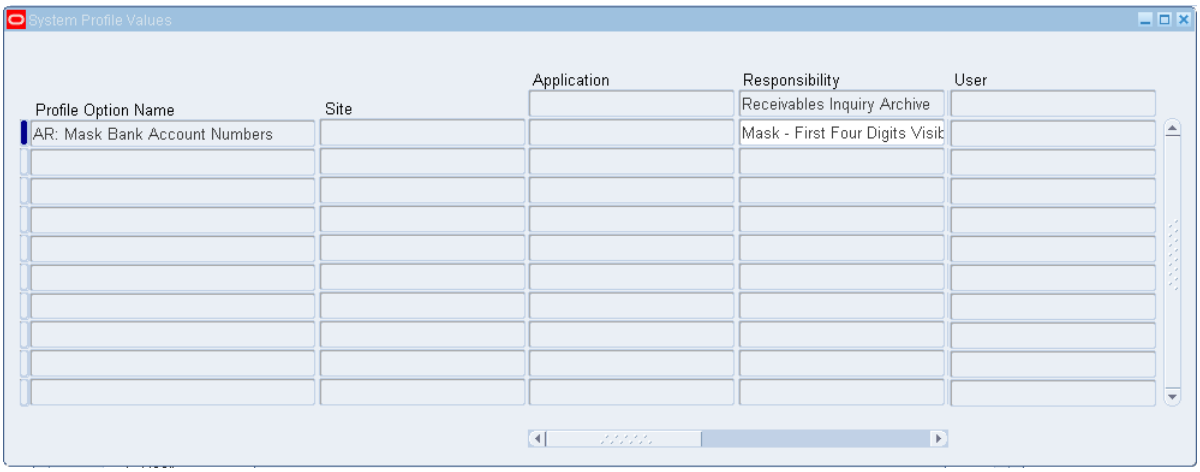
25. Navigate to **Profile** > **System**. Uncheck the box next to Site. Check the box next to Responsibility and find the original responsibility. Uncheck the box next to Profiles with No Values.



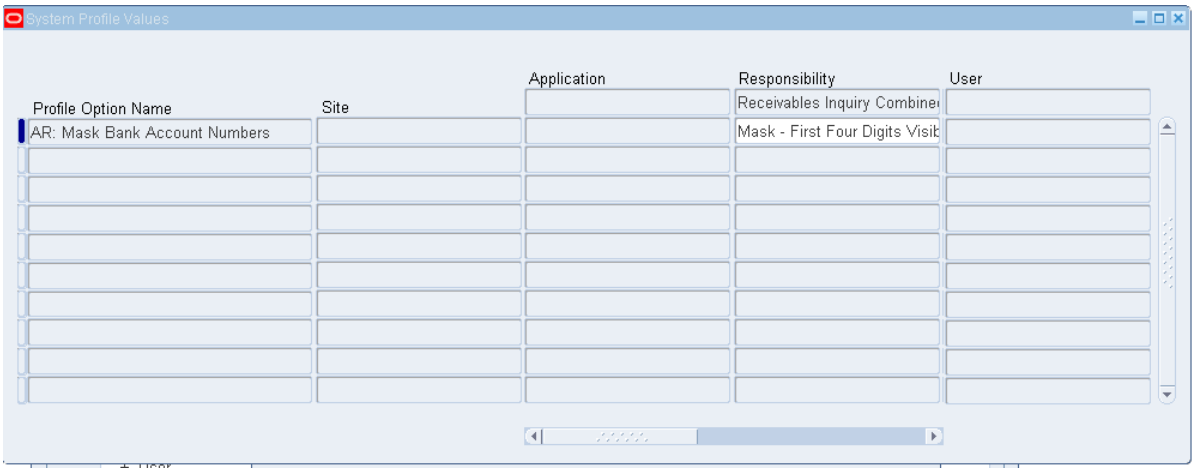
26. Click **Find** to see if any profile options are set.



27. Add any profile options set at the responsibility level to the archive version of the responsibility if required or desired.



28. Add any profile options set at the responsibility level to the combined version of the responsibility if required or desired.



29. Perform the same steps for all the responsibilities requiring an archive and combined version.
30. Assign the new responsibilities to users.

Appendix

The seamless access users are intended to be APPS like schemas with all the permissions of the APPS schema. The sample scripts are intended as example scripts. There may be additional grants required and some of the example grants may not be required. Check the data dictionary for exact grants to your APPS schema.

Sample Script for Archive Only Schema

Use the following sample script to create Archive only schema.

```
CREATE USER AMARCHIVE
  IDENTIFIED BY password
  DEFAULT TABLESPACE ARC53_STAGE_D
  TEMPORARY TABLESPACE TEMP
```

```
PROFILE DEFAULT
ACCOUNT UNLOCK;

GRANT SELECT_CATALOG_ROLE TO AMARCHIVE;
GRANT CTXAPP TO AMARCHIVE;
GRANT HR_REPORTING_USER TO AMARCHIVE WITH ADMIN OPTION;
GRANT CONNECT TO AMARCHIVE;
GRANT AQ_ADMINISTRATOR_ROLE TO AMARCHIVE WITH ADMIN OPTION;
GRANT HR_SECURE_USER TO AMARCHIVE WITH ADMIN OPTION;
GRANT OLAP_USER TO AMARCHIVE;
ALTER USER AMARCHIVE DEFAULT ROLE ALL;
GRANT CREATE ANY TRIGGER TO AMARCHIVE;
GRANT DROP ANY PROCEDURE TO AMARCHIVE;
GRANT CREATE DATABASE LINK TO AMARCHIVE;
GRANT CREATE ANY SEQUENCE TO AMARCHIVE;
GRANT DROP ANY VIEW TO AMARCHIVE;
GRANT UPDATE ANY TABLE TO AMARCHIVE;
GRANT ALTER ANY TABLE TO AMARCHIVE;
GRANT CREATE TABLE TO AMARCHIVE;
GRANT CREATE ANY CONTEXT TO AMARCHIVE;
GRANT CREATE ANY DIRECTORY TO AMARCHIVE;
GRANT CREATE TRIGGER TO AMARCHIVE;
GRANT GRANT ANY ROLE TO AMARCHIVE;
GRANT SELECT ANY SEQUENCE TO AMARCHIVE;
GRANT CREATE ANY TABLE TO AMARCHIVE;
GRANT DROP ANY MATERIALIZED VIEW TO AMARCHIVE;
GRANT ALTER ANY MATERIALIZED VIEW TO AMARCHIVE;
GRANT ALTER ANY TRIGGER TO AMARCHIVE;
GRANT ALTER ANY ROLE TO AMARCHIVE;
GRANT ALTER ANY CLUSTER TO AMARCHIVE;
GRANT ALTER ANY OUTLINE TO AMARCHIVE;
GRANT DROP PUBLIC SYNONYM TO AMARCHIVE;
GRANT CREATE CLUSTER TO AMARCHIVE;
GRANT SELECT ANY TABLE TO AMARCHIVE;
GRANT DROP ANY TABLE TO AMARCHIVE;
GRANT SELECT ANY DICTIONARY TO AMARCHIVE;
GRANT DROP ANY OUTLINE TO AMARCHIVE;
GRANT CREATE ANY OUTLINE TO AMARCHIVE;
GRANT DROP ANY CONTEXT TO AMARCHIVE;
GRANT EXECUTE ANY PROCEDURE TO AMARCHIVE;
GRANT CREATE PUBLIC DATABASE LINK TO AMARCHIVE;
GRANT CREATE PUBLIC SYNONYM TO AMARCHIVE;
GRANT DROP ANY INDEX TO AMARCHIVE;
GRANT DROP ANY CLUSTER TO AMARCHIVE;
GRANT COMMENT ANY TABLE TO AMARCHIVE;
GRANT ALTER SYSTEM TO AMARCHIVE;
GRANT CREATE TYPE TO AMARCHIVE;
GRANT DROP ANY TRIGGER TO AMARCHIVE;
GRANT ALTER ANY PROCEDURE TO AMARCHIVE;
GRANT DROP ANY ROLE TO AMARCHIVE;
GRANT ALTER ANY SEQUENCE TO AMARCHIVE;
GRANT ALTER ANY INDEX TO AMARCHIVE;
GRANT CREATE ANY CLUSTER TO AMARCHIVE;
GRANT GLOBAL QUERY REWRITE TO AMARCHIVE;
GRANT CREATE ANY MATERIALIZED VIEW TO AMARCHIVE;
GRANT DROP ANY SEQUENCE TO AMARCHIVE;
GRANT CREATE ROLE TO AMARCHIVE;
GRANT ADMINISTER DATABASE TRIGGER TO AMARCHIVE;
GRANT ALTER SESSION TO AMARCHIVE;
GRANT ANALYZE ANY TO AMARCHIVE;
GRANT CREATE ANY INDEX TO AMARCHIVE;
GRANT CREATE ANY PROCEDURE TO AMARCHIVE;
GRANT CREATE ANY SYNONYM TO AMARCHIVE;
GRANT CREATE ANY VIEW TO AMARCHIVE;
GRANT CREATE PROCEDURE TO AMARCHIVE;
GRANT CREATE SEQUENCE TO AMARCHIVE;
GRANT CREATE SESSION TO AMARCHIVE;
GRANT CREATE SYNONYM TO AMARCHIVE;
GRANT CREATE VIEW TO AMARCHIVE;
GRANT DELETE ANY TABLE TO AMARCHIVE;
GRANT DROP ANY SYNONYM TO AMARCHIVE;
```

```
  GRANT EXECUTE ANY PROCEDURE TO AMARCHIVE WITH ADMIN OPTION;
  GRANT EXECUTE ANY TYPE TO AMARCHIVE;
  GRANT INSERT ANY TABLE TO AMARCHIVE;
  GRANT LOCK ANY TABLE TO AMARCHIVE;
  GRANT SELECT ANY DICTIONARY TO AMARCHIVE WITH ADMIN OPTION;
  GRANT SELECT ANY TABLE TO AMARCHIVE WITH ADMIN OPTION;
  GRANT UNLIMITED TABLESPACE TO AMARCHIVE;
  GRANT EXECUTE ON APPS.EMCASO_FREIGHT_MGR_PKG TO AMARCHIVE;
  GRANT SELECT ON APPS.EMCASO_QUOTE_HEADERS_ALL_V TO AMARCHIVE;
  GRANT EXECUTE ON APPS.FND_FLEX_SERVER TO AMARCHIVE;
  GRANT EXECUTE ON APPS.WF_EVENT TO AMARCHIVE;
  GRANT EXECUTE ON SYS.DBMS_ALERT TO AMARCHIVE;
  GRANT EXECUTE ON SYS.DBMS_LOCK TO AMARCHIVE;
  GRANT EXECUTE ON SYS.DBMS_SPACE_ADMIN TO AMARCHIVE;
  GRANT SELECT ON SYS.LINK$ TO AMARCHIVE;
  GRANT SELECT ON SYS.USER$ TO AMARCHIVE;
  GRANT EXECUTE_CATALOG_ROLE TO AMARCHIVE;
  GRANT PUBLIC TO AMARCHIVE;
  GRANT RESOURCE TO AMARCHIVE;
```

Sample Script for Combined Schema

Use the following script to create a Combined schema.

```
CREATE USER AMQUERY
  IDENTIFIED BY password
  DEFAULT TABLESPACE ARC53_STAGE_D
  TEMPORARY TABLESPACE TEMP
  PROFILE DEFAULT
  ACCOUNT UNLOCK;

  GRANT SELECT_CATALOG_ROLE TO AMQUERY;
  GRANT CTXAPP TO AMQUERY;
  GRANT HR_REPORTING_USER TO AMQUERY WITH ADMIN OPTION;
  GRANT CONNECT TO AMQUERY;
  GRANT AQ_ADMINISTRATOR_ROLE TO AMQUERY WITH ADMIN OPTION;
  GRANT HR_SECURE_USER TO AMQUERY WITH ADMIN OPTION;
  GRANT OLAP_USER TO AMQUERY;
  ALTER USER AMQUERY DEFAULT ROLE ALL;
  GRANT CREATE ANY TRIGGER TO AMQUERY;
  GRANT DROP ANY PROCEDURE TO AMQUERY;
  GRANT CREATE DATABASE LINK TO AMQUERY;
  GRANT CREATE ANY SEQUENCE TO AMQUERY;
  GRANT DROP ANY VIEW TO AMQUERY;
  GRANT UPDATE ANY TABLE TO AMQUERY;
  GRANT ALTER ANY TABLE TO AMQUERY;
  GRANT CREATE TABLE TO AMQUERY;
  GRANT CREATE ANY CONTEXT TO AMQUERY;
  GRANT CREATE ANY DIRECTORY TO AMQUERY;
  GRANT CREATE TRIGGER TO AMQUERY;
  GRANT GRANT ANY ROLE TO AMQUERY;
  GRANT SELECT ANY SEQUENCE TO AMQUERY;
  GRANT CREATE ANY TABLE TO AMQUERY;
  GRANT DROP ANY MATERIALIZED VIEW TO AMQUERY;
  GRANT ALTER ANY MATERIALIZED VIEW TO AMQUERY;
  GRANT ALTER ANY TRIGGER TO AMQUERY;
  GRANT ALTER ANY ROLE TO AMQUERY;
  GRANT ALTER ANY CLUSTER TO AMQUERY;
  GRANT ALTER ANY OUTLINE TO AMQUERY;
  GRANT DROP PUBLIC SYNONYM TO AMQUERY;
  GRANT CREATE CLUSTER TO AMQUERY;
  GRANT SELECT ANY TABLE TO AMQUERY;
  GRANT DROP ANY TABLE TO AMQUERY;
  GRANT SELECT ANY DICTIONARY TO AMQUERY;
  GRANT DROP ANY OUTLINE TO AMQUERY;
  GRANT CREATE ANY OUTLINE TO AMQUERY;
  GRANT DROP ANY CONTEXT TO AMQUERY;
  GRANT EXECUTE ANY PROCEDURE TO AMQUERY;
  GRANT CREATE PUBLIC DATABASE LINK TO AMQUERY;
```

```
GRANT CREATE PUBLIC SYNONYM TO AMQUERY;
GRANT DROP ANY INDEX TO AMQUERY;
GRANT DROP ANY CLUSTER TO AMQUERY;
GRANT COMMENT ANY TABLE TO AMQUERY;
GRANT ALTER SYSTEM TO AMQUERY;
GRANT CREATE TYPE TO AMQUERY;
GRANT DROP ANY TRIGGER TO AMQUERY;
GRANT ALTER ANY PROCEDURE TO AMQUERY;
GRANT DROP ANY ROLE TO AMQUERY;
GRANT ALTER ANY SEQUENCE TO AMQUERY;
GRANT ALTER ANY INDEX TO AMQUERY;
GRANT CREATE ANY CLUSTER TO AMQUERY;
GRANT GLOBAL QUERY REWRITE TO AMQUERY;
GRANT CREATE ANY MATERIALIZED VIEW TO AMQUERY;
GRANT DROP ANY SEQUENCE TO AMQUERY;
GRANT CREATE ROLE TO AMQUERY;
GRANT ADMINISTER DATABASE TRIGGER TO AMQUERY;
GRANT ALTER SESSION TO AMQUERY;
GRANT ANALYZE ANY TO AMQUERY;
GRANT CREATE ANY INDEX TO AMQUERY;
GRANT CREATE ANY PROCEDURE TO AMQUERY;
GRANT CREATE ANY SYNONYM TO AMQUERY;
GRANT CREATE ANY VIEW TO AMQUERY;
GRANT CREATE PROCEDURE TO AMQUERY;
GRANT CREATE SEQUENCE TO AMQUERY;
GRANT CREATE SESSION TO AMQUERY;
GRANT CREATE SYNONYM TO AMQUERY;
GRANT CREATE VIEW TO AMQUERY;
GRANT DELETE ANY TABLE TO AMQUERY;
GRANT DROP ANY SYNONYM TO AMQUERY;
GRANT EXECUTE ANY PROCEDURE TO AMQUERY WITH ADMIN OPTION;
GRANT EXECUTE ANY TYPE TO AMQUERY;
GRANT INSERT ANY TABLE TO AMQUERY;
GRANT LOCK ANY TABLE TO AMQUERY;
GRANT SELECT ANY DICTIONARY TO AMQUERY WITH ADMIN OPTION;
GRANT SELECT ANY TABLE TO AMQUERY WITH ADMIN OPTION;
GRANT UNLIMITED TABLESPACE TO AMQUERY;
GRANT EXECUTE ON APPS.FND_FLEX_SERVER TO AMQUERY;
GRANT EXECUTE ON APPS.WF_EVENT TO AMQUERY;
GRANT EXECUTE ON SYS.DBMS_ALERT TO AMQUERY;
GRANT EXECUTE ON SYS.DBMS_LOCK TO AMQUERY;
GRANT EXECUTE ON SYS.DBMS_SPACE_ADMIN TO AMQUERY;
GRANT SELECT ON SYS.LINK$ TO AMQUERY;
GRANT SELECT ON SYS.USER$ TO AMQUERY;
GRANT EXECUTE_CATALOG_ROLE TO AMQUERY;
GRANT PUBLIC TO AMQUERY;
GRANT RESOURCE TO AMQUERY;
```

CHAPTER 34

Seamless Access Setup for Oracle R12 in Data Archive

This chapter includes the following topics:

- Abstract, 131
- Apply Oracle Patch 9015132 "ENABLING INVOKERS PROCESSING IN R12", 131
- Configure the Seamless Access Users and Responsibilities, 132
- Release 12 Applications Tier Configuration for Informatica ILM Seamless Access, 132
- References, 134

Abstract

This article provides more details on Oracle R12 seamless access setup and configuration in Data Archive.

Apply Oracle Patch 9015132 "ENABLING INVOKERS PROCESSING IN R12"

In Oracle E-Business Suite 11.5.x, PL/SQL packages owned by APPS were created with invoker rights, resulting in the execution of the program units with the privileges of the caller. However, with R12, Oracle changed the package security model to definer rights. Privileges of the package owner, also known as the definer, are used at execution time.

The fact that definer rights not only dictate the privileges, but also are used to resolve object references, prevents the seamless data access layer from functioning correctly. Customers therefore need to download and apply Oracle Patch 9015132 "ENABLING INVOKERS PROCESSING IN R12" to revert the package security model back to invoker rights.

Configure the Seamless Access Users and Responsibilities

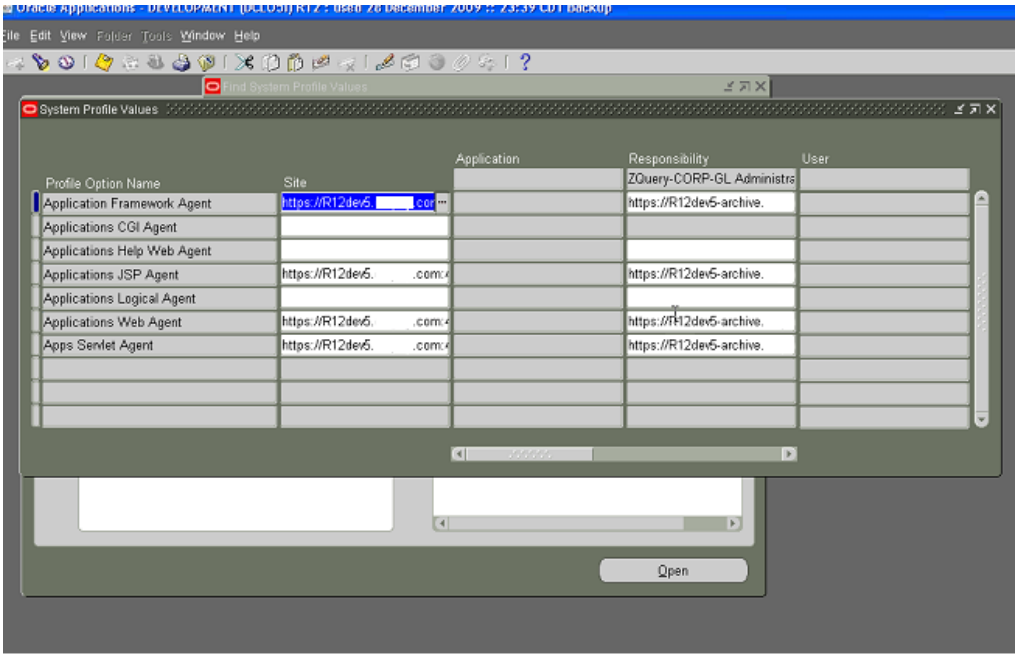
1. Register the Seamless Access Users (Security -> ORACLE -> Register).
2. Create Seamless Access Data Groups (Security -> ORACLE -> Data Group).
3. Define Seamless Access Responsibilities (Security -> Responsibility - Define).

Release 12 Applications Tier Configuration for Informatica ILM Seamless Access

1. Set up a dedicated Application tier node to service Seamless Access users.
   This node can either be on a dedicated server or on an existing server using a shared application tier file system.
2. Install or point to a shared APPL_TOP.
3. Configure and enable all relevant services on the dedicated node.
   - Application Framework Agent (APPS_FRAMEWORK_AGENT)
   - Applications JSP Agent (APPS_JSP_AGENT)
   - Applications Web Agent (APPS_WEB_AGENT)
   - Apps Servlet Agent (APPS_SERVLET_AGENT)

Example: AMQUERY GL Administrator Responsibility

1. Set all agent related profiles, such as Application Framework Agent, to point to the archive node https://r12dev-amquery.informatica.com:443.



2. Configure the DBC file on the archive nodes (located in $INST_TOP/appl/fnd/12.0.0/secure). Change the following profile options:

```
FNDNAM=AMQUERY
FND_JDBC_USABLE_CHECK=true
FND_JDBC_PLSQL_RESET=true
FND_JDBC_CONTEXT_CHECK=true
```

3. Create the new concurrent manager for Java concurrent requests on the seamless node:
   - Create a new concurrent manager on the forms Define Concurrent Managers from the System Administrator responsibility (set the same setup as the Standard Concurrent manager).
   - Click the Specialization Rules button and include the concurrent programs you want to run on this concurrent manager.
   - Query the Standard concurrent manager on the forms Define Concurrent Managers from the System Administrator responsibility.
   - Click the Specialization Rules button and exclude the concurrent programs you included in the previous step for the new concurrent managers.
   - Bounce the concurrent managers.
4. AMQUERY & COMBINED password should be same as APPS schema password.
   For example:

```
FNDCPASS apps/<apps pwd> 0 Y system/manager ORACLE AMQUERY <APPS PWD>
FNDCPASS apps/<apps pwd> 0 Y system/manager ORACLE COMBINED <APPS PWD>
```

References

- Metalink Note 380489.1, Using Load-Balancers with Oracle E-Business Suite Release 12 Section 2.2: Using Hardware Load Balancers with Multiple Web Entry Points.
- Sharing The Application Tier File System in Oracle E-Business Suite Release 12 [ID 384248.1] - Section 4: Adding a Node to a Shared Application Tier File System.
- Using AutoConfig to Manage System Configurations in Oracle E-Business Suite Release 12 [ID 387859.1]
- 3.12 Configure Parallel Concurrent Processing section - (Doc ID 388577.1)

Abstract

This article provides information about how to use the Data Vault Service JDBC driver to connect to a Data Vault in Data Archive.

CHAPTER 36

Using the Data Vault Service JDBC Driver to Connect to the Data Vault

This chapter includes the following topics:

- Data Vault Service JDBC Driver Overview, 136
- JDBC Driver Requirements, 136
- JDBC Driver Installation, 137
- ODBC Data Source for the Data Vault, 138
- Connecting to the Data Vault, 138

Data Vault Service JDBC Driver Overview

The Data Vault Service provides a JDBC driver to connect to the Data Vault.

The Data Vault Service JDBC driver is a Type 2 driver. It uses the Data Vault Service ODBC driver to connect to the Data Vault. You must install the Data Vault Service ODBC and JDBC drivers on the machine that hosts the client tool you plan to use to connect to the Data Vault.

JDBC Driver Requirements

The Data Vault Service JDBC driver requires Java and the Data Vault Service ODBC driver.

Before you use the JDBC driver, verify that Java is installed on the machine where you plan to connect to the Data Vault and set the environment variables for the class and library path.

Java

Verify that you have the correct Java version installed. The JDBC driver requires the 32-bit or 64-bit Java version 1.6, based on the ODBC driver version you use.

CLASSPATH

Set the Java CLASSPATH environment variable to the path and file name of the JDBC driver. For example:

```
Set CLASSPATH=/usr/FAS/jdbc/nucleus.jar
```

Library path

Set the library path environment variable to the directory where the ODBC driver is installed. For example, to set the library path for 32-bit Java:

```
Set LD_LIBRARY_PATH=/usr/FAS/odbc32
```

or

```
Set LIBPATH=/usr/FAS/odbc32
```

JDBC Driver Installation

To use the Data Vault Service JDBC driver from a client machine, install the Data Vault Service ODBC Driver component. The ODBC driver component contains the ODBC and JDBC drivers that you need to connect to the Data Vault.

Use the Data Vault Service installer to install the JDBC driver. The ODBC and JDBC drivers are shipped with a number of Data Vault Service components. If you install a component that contains the ODBC and JDBC drivers, you do not need to install the drivers separately.

The following Data Vault Service components contain the JDBC driver:

Data Vault Service

To use the Data Vault, you install the Data Vault Service component. The Data Vault Service component includes the ODBC and JDBC drivers.

The Data Vault Service installer copies the JDBC driver to the following folder:

```
<DataVaultServiceDirectory>/jdbc
```

Data Vault Service ODBC driver

To connect to the Data Vault from a client machine, install the ODBC driver. The ODBC driver installation files includes the JDBC driver.

The Data Vault Service installer copies the JDBC driver to the following folder:

```
<DataVaultServiceODBCDriverDirectory>/jdbc
```

Data Vault Service Data Archive plug-in

To use the Data Vault with Data Archive, you install the Data Vault Service Data Archive plug-in component on the machine that hosts the Data Archive ILM Engine. The Data Vault Service Data Archive plug-in includes the ODBC and JDBC drivers. You can use the drivers to connect to the Data Vault from the Data Archive server.

The Data Vault Service installer copies the JDBC driver to the following folder:

```
<DataArchiveDirectory>/webapp/file_archive/jdbc
```

The Data Vault Service JDBC driver is in the file named `nucleus.jar`.

ODBC Data Source for the Data Vault

When you create the ODBC data source for the Data Vault, use the name of the Data Vault folder in the target connection as the name of the database in the ODBC data source.

In Data Archive, when you create the target connection to the Data Vault, you specify the name of the Data Vault folder to use for the archived data. When you archive data to the Data Vault, Data Archive uses the Data Vault folder name specified in the target connection to create the database in the Data Vault.

When you create the ODBC data source for the Data Vault, set the Database property to the name of Data Vault folder of the archived data you want to access.

Connecting to the Data Vault

Use an ODBC data source for the JDBC connection to the Data Vault.

The class name for the JDBC driver is *nucleus.Driver*.

1. Create an ODBC data source for the Data Vault.

   On Windows, you can use the ODBC Data Source Administrator to set up a data source. On UNIX, set up a data source in the odbc.ini.

2. Include the DSN in the JDBC URL: `jdbc:nucleus:// <DSN>`

   DSN is the name of the ODBC data source to use to connect to the Data Vault.

3. Add an entry for the connection to the Data Vault Service configuration file *nucleus.ini*.

The following sample code shows how to use the Java DriverManager class to connect to a database in the Data Vault with the JDBC driver:

```
String db_connect_string = "jdbc:nucleus://FAS";
String db_userid = "dba";
String db_password = "dba";
Class.forName("nucleus.Driver");
conn = DriverManager.getConnection(db_connect_string, db_userid, db_password);
```

For this example, the odbc.ini and nucleus.ini files have the following entries :

odbc.ini

```
[FAS]
Driver=/usr/FAS/odbc/libsando.so
Description = Informatica ILM FAS 6.1 ODBC Driver
Connection = fas
Database = Archive_ SEP_2012
UID = DBA
```

nucleus.ini

```
[Connection fas]
HOST=localhost
PORT=8500
```

CHAPTER 38

Using Multiple Engines in an ILM Environment

This chapter includes the following topics:

- Abstract, 139
- Challenge, 139
- Description, 140
- Workflow Best Practices, 140
- Keeping Track of Tomcat Engines, 140
- Keeping the ./webapp/file_archive Directories Synchronized, 141
- Keeping the ./webapp/xsl Directories Synchronized, 142

Abstract

You can use multiple Tomcat engines to maximize resources for ILM during data archive for application retirement. This article recommends the best practices to track jobs running on the different engines and to synchronize the Data Vault directories and report style sheets for the different engines.

Challenge

Having a single ILM Tomcat engine puts a limit on the resources, such as the number of threads and heap space, that Data Archive can use. To maximize resources, you can install multiple Tomcat engines that point to the same ILM repository. This configuration allows you to have the maximum number of extracts (Copy to Destination) running at the same time on the same machine. However, this configuration also presents the challenge of keeping track of which engine is running each job.

Description

An environment with multiple ILM Tomcat engines requires only installation. All instances of the user interface will share the same ILM home repository.

To set up multiple Tomcat engines for ILM, complete the following steps:

- Run the installer to create the initial Tomcat engine and ILM Home schema in the database. For this example, the installation path of the initial engine is /app/dev/ilm/ILM_Engine.
- Go to the /app/dev/ilm/ILM_Engine directory and zip up the contents of the directory, including subdirectories.
- Determine how many engines you want to run and create a directory for each. For example, to run two additional engines, you can create the directories /app/dev2/ilm/ILM_Engine and /app/dev2/ilm/ILM_Engine.
- Copy the zip file you created in step 2 to each of the new directories you created in step 3 and extract the files.
- For each new instance, change the port number for the Tomcat engine in the conf.properties file. Edit each conf.properties file and search for the property informia.applicationPort. By default, the initial engine uses port number 8080.You can increment the port number by one for each additional engine. For example, use 8081 and 8082 for two additional engines. Verify that the port numbers are not used by other applications.
- To start each Tomcat engine, run the startApplimation.sh script. Verify that users can access all engines.

Workflow Best Practices

When you set up multiple Tomcat engines for ILM, you need to set up workflows differently.

Use the following guidelines when you set up workflows in an ILM environment with multiple Tomcat engines:

- Keep track of which Tomcat engine runs a job.
- Keep all ./webapp/file_archive directories synchronized.
- Keep all ./webapp/xsl directories synchronized.

Keeping Track of Tomcat Engines

Keep track of which instance runs each job, in case you later need to view the logs or to troubleshoot or resume a job.

To keep track of the instances, follow these best practices:

- In ILM 5.3.3 and previous versions, incorporate the instance in the project name to associate the job with the instance. For example, use a project name such as AR_ACME_PROD1 where PROD1 is the instance that the job is running on.

  Effective in ILM 5.3.4, use the informia.appendHostAndPortInLog parameter in the conf.properties file to identify the host name and port number of the instance where the job is running. Edit the conf.properties file for each instance and set the parameter to informia.appendHostAndPortInLog=Y. ILM appends the host name and port number to the timestamps in the logs. The configuration applies to all Data Archive, Data Masking, and Data Subset jobs and all standalone jobs.

- To see which instance has the least load for a new job, go to each instance and check the Thread Queue Manager. The queue manager can be found at http://URL:port/jsp/tqm.jsp. This page shows you how many threads are running and pending for each instance. You can pick the least active instance to begin the new job.

  The following is an example of the tqm.jsp output:

  ```
  Thread Status
  ThreadId=86 ThreadName=SCHEDULEJOB$86$174_WYETHIDR_IDR_201011 174_WYETHIDR_IDR_201011
  Status=Running GroupId=SCHEDULEJOB$86 ParentIds=null
  Priority=1
  ThreadId=148 ThreadName=JOBSTEP$148$Copy To Destination Status=Running GroupId=JOB$86
  ParentIds=null Priority=0
  ThreadId=33906 ThreadName=33$RES_FACT$FLAT_FILE Status=Running GroupId=STEP$148
  ParentIds=null Priority=1
  ThreadId=33914 ThreadName=33$SAMP_BRIDGE$FLAT_FILE Status=Running
  GroupId=STEP$148 ParentIds=null Priority=1
  ```

  Each thread that has "status=Running" will count against the total thread count allocated through the amMaxServerThreads parameter. For example, if you have the parameter set to 30, then, for this instance, 27 threads are left for other processes.

  The best instance to start a new job is one where the tqm.jsp returns the following status:

  ```
  Thread Status
  No threads in queue
  ```

  Effective in ILM 5.3.4, if you set informia.appendHostAndPortInLog parameter=Y to append the host name and port number to the logs, the job logs in View Current Jobs indicate what instance a job is running on. Using this information, you can open the job log for any process in a running state to see which engine has the heaviest load and which one has the lightest load.

Keeping the ./webapp/file_archive Directories Synchronized

To manage retention, synchronize the ./webapp/file_archive directory for all Tomcat engines. When you change retention, ILM looks in the Data Vault directory for the DDL of the driving table. If you archive in one instance and then try to manage retention in a different instance, the DDL file will not be found. To ensure that the DDL file can be found, use symbolic links to tie all engines to one base folder.

If you use multiple Tomcat engines and each engine was installed separately, you must synchronize the Data Vault folder and then create the symbolic links to the base archive directory.

For example, you have three ILM engines Prod1, Prod2, and Prod3 and you want to use the Data Vault directory in Prod1 as the base archive directory.

To synchronize the Data Vault folders, complete the following steps:

1. Go to the ./webapp/file_archive directory in the Prod2 directory.
2. Copy the contents, including all subdirectories, into the ./webapp/file_archive folder in the Prod1 directory.
3. Change the Data Vault directory in Prod2 to be a symbolic link to the Data Vault folder in the Prod1 directory.
4. Go to the ./webapp/file_archive directory in the Prod3 directory.
5. Copy the contents, including all subdirectories, into the ./webapp/file_archive folder in the Prod1 directory.
6. Change the Data Vault directory Prod3 to be a symbolic link to the Data Vault folder in the Prod1 directory.

After you set all engines to archive to the same Data Vault directory, enable the informia.dataDiscoveryMetadataQuery parameter so that the metadata for data discovery would be retrieved from the AMHOME schema. Edit the conf.properties file for each instance and set the parameter to informia.dataDiscoveryMetadataQuery=AMHOME.

Note: Effective in ILM 5.3.4, the informia.dataDiscoveryMetadataQuery parameter is set to AMHOME by default.

Keeping the ./webapp/xsl Directories Synchronized

Make sure that any style sheet that is created for Data Discovery is copied to the ./webapp/xsl folder for all Tomcat engines so that you can access the reports from any of the ILM instances. Use symbolic links to synchronize the style sheet directories of all Tomcat engines with one base directory in the same way you use symbolic links to synchronize the Data Vault directories.

CHAPTER 40

Using PowerExchange ODBC Connections in a Data Archive Retirement Project

This chapter includes the following topics:

- Abstract, 143
- Overview, 144
- Implementation Scenario, 144
- Task Flow, 145
- Step 1: Install and Configure PowerExchange on z/OS, 145
- Step 2: Install and Configure PowerExchange on the Data Archive Server, 147
- Step 3: Install and Configure PowerExchange on the Validation Server, 148
- Step 4: Install and Configure PowerExchange on Windows, 149
- Step 5: Create PowerExchange Data Maps, 149
- Step 6: Configure PowerExchange ODBC Data Sources on the Data Archive Server, 150
- Step 7: Configure PowerExchange ODBC Data Sources on the Validation Server, 152
- Step 8: Import z/OS Metadata, 153
- Step 9: Define and Run the Retirement Project, 154
- Step 10: Validate the Retired Data, 154

Abstract

You can use PowerExchange ODBC connections to connect to z/OS data sources in an Informatica Data Archive retirement project. This article describes the procedures for using PowerExchange ODBC connections in an archive retirement project.

Overview

This article summarizes the steps that are required to use PowerExchange ODBC connections in a Data Archive retirement project.

A retirement archive project copies a set of tables from a source database to an archive database or Data Vault. The source database can then be decommissioned. If necessary, the archived data can be restored by using Information Lifecycle Management (ILM) tools.

To retire IMS or VSAM source data on a z/OS system, you can use PowerExchange ODBC connections that are provided with PowerExchange. The Enterprise Data Manager (EDM) and Data Archive use the JDBC-ODBC bridge to connect to the z/OS source data through the PowerExchange ODBC drivers.

This article assumes that you have installed Informatica Data Archive, PowerCenter, and Data Validation Option (DVO) and are familiar with these products. Familiarity with PowerExchange is also recommended.

Implementation Scenario

A data archive retirement project uses PowerExchange ODBC to access IMS and VSAM data sources on z/OS. This implementation includes the following systems and components:

- A z/OS system with the IMS and VSAM data sources.

  On this system, you will install PowerExchange 9.1.0 HotFix 4 or later.
- A Linux, UNIX, or Windows system that has the following type of storage and software:

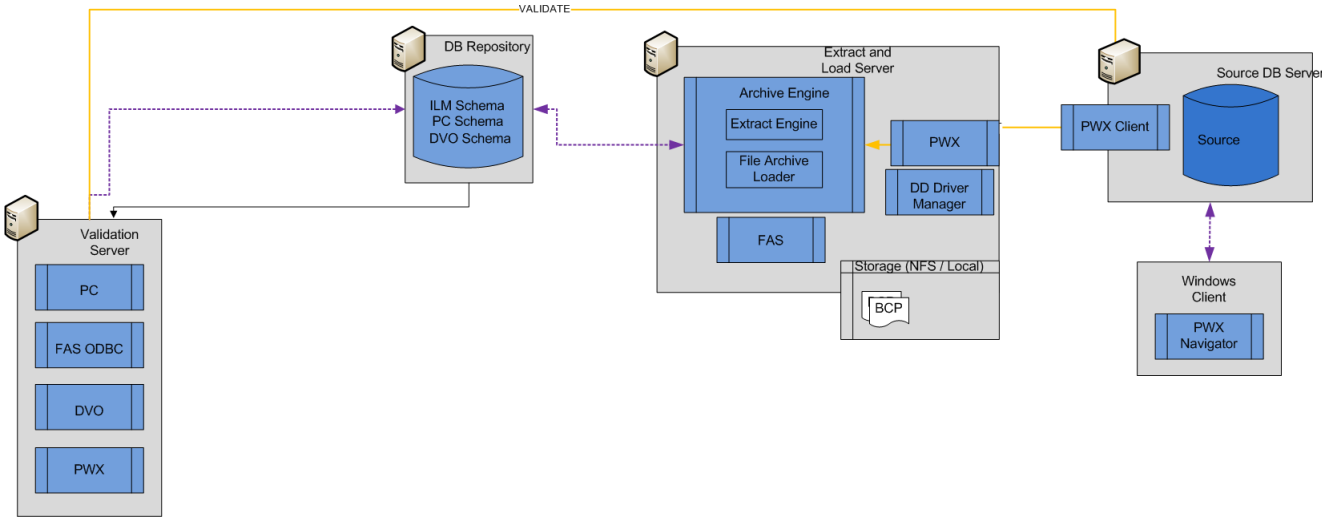
  - Informatica Data Archive engine 5.3.7 or later
  - Data Vault Server
  - NFS or local storage for the encapsulated BCP files

  On this system, you will install PowerExchange 9.1.0 HotFix 4 or later. You will also copy the DataDirect Driver Manager and define PowerExchange ODBC data sources.
- Database repository for ILM, PowerCenter, and DVO schema.
- Validation server with the PowerCenter Domain, PowerCenter Client, FAS ODBC, and DVO installed.

  On this system, you will install PowerExchange and define PowerExchange ODBC data sources.
- A Microsoft Windows system.

  On this system, you will install PowerExchange 9.1.0 HotFix 4 or later and run the PowerExchange Navigator.

The following figure illustrates the implementation. In this figure, PWX represents PowerExchange, PC represents PowerCenter, and DD represents DataDirect.



Task Flow

To use PowerExchange ODBC connections in a Data Archive retirement project, complete the following steps:

1. Install and configure PowerExchange on z/OS.
2. Install and configure PowerExchange on the Data Archive server.
3. Install and configure PowerExchange on the Validation server.
4. Install and configure PowerExchange on Windows.
5. Define data maps for the z/OS data sources.
6. Configure PowerExchange ODBC data sources on the Data Archive server.
7. Configure PowerExchange ODBC data sources on the Validation server.
8. Import z/OS metadata.
9. Define and run the retirement archive project.
10. Validate the retired data.

Step 1: Install and Configure PowerExchange on z/OS

To install PowerExchange on z/OS, run the MVS Installation Assistant. After you install PowerExchange, edit the DBMOVER configuration file and start the PowerExchange Listener.

Install PowerExchange on z/OS

Install Power Exchange on the z/OS system where the source data for the retirement project is located.

To install PowerExchange on z/OS, run the MVS Installation Assistant. For detailed instructions, see the *PowerExchange Installation Guide*. This guide gives details about how access to the z/OS databases and files is controlled.

Edit the DBMOVER Configuration Member on the z/OS System

Edit the DBMOVER configuration member.

Verify that the member includes a LISTENER statement. The LISTENER statement defines the TCP/IP port on which a named PowerExchange Listener process listens for work requests. The default DBMOVER member includes the required LISTENER statement.

In addition to the LISTENER statement, you can include the following statements in the DBMOVER configuration member to minimize data map I/O:

```
DMXCACHE_MAX_MEMORY_MB=20
DM_SUBTASK=R
```

When Data Archive accesses PowerExchange data maps through the PowerExchange Listener, every data map in the DATAMAPS KSDS is opened and processed. The DMXCACHE_MAX_MEMORY_MB and DM_SUBTASK statements configure the PowerExchange Listener on z/OS to use data maps caching and a data maps subtask in read-only mode.

Note: To further reduce data map I/O, limit the number of data maps in the DATAMAPS file. For best results, limit the number of data maps to 100.

For more information about the DBMOVER configuration member, see the *PowerExchange Reference Manual*.

Configure the PowerExchange Listener JCL on z/OS

Configure the PowerExchange Listener JCL before starting the PowerExchange Listener the first time. PowerExchange provides PowerExchange Listener JCL in the following members of the RUNLIB library:

- PWXLSTNR runs PowerExchange as a started task.
- STARTLST runs PowerExchange as a batch job.

Note: When you ran the XIZZZ998 job during installation, the PWXLSTNR member was copied to the PROCLIB library.

1. Verify that the STEPLIB DD concatenation in the PowerExchange Listener JCL includes the load libraries for your source DBMSs.

   Usually, the PowerExchange LOADLIB library occurs before the data-source load libraries in the concatenation.

   The STEPLIB concatenation in the STARTLST member includes the load libraries for all data sources in a proper order.
2. Verify that the PARM statement specifies the node name that is defined in the LISTENER statement of the DBMOVER configuration file.
3. In the REGION parameter of the EXEC statement, enter a region size. The region size indicates the amount of storage to allocate for running the PowerExchange Listener.

   The sample JCL specifies a region size of 128 MB in the EXEC statement.

   Informatica recommends a region size of 0M, which indicates no storage limit.

   If you omit the REGION parameter, the system default applies.
4. Verify that the TIME parameter in the EXEC statement is set to NOLIMIT. The NOLIMIT setting enables the PowerExchange Listener job to run without a time limit and prevents timeout-related abends that might occur with abend code S322.

5. If the SECURITY statement in the DBMOVER configuration file is set to 1 or 2, APF-authorize the PowerExchange LOADLIB library and any other load libraries in the STEPLIB DD concatenation of the PowerExchange Listener JCL.

For more information, see the *PowerExchange Bulk Data Movement Guide*.

Start the PowerExchange Listener

To start the PowerExchange Listener, issue the MVS START command with the name of the started task. For example:

```
START PWXLSTNR
```

You can also run the PowerExchange Listener as a batch job. However, because it is a long-running task, using an MVS started task is more appropriate.

Step 2: Install and Configure PowerExchange on the Data Archive Server

Install and configure PowerExchange on the same server as Informatica Data Archive.

Install PowerExchange on the Data Archive Server

On the Data Archive server, install PowerExchange 9.1.0 HotFix 4 or later. Use the PowerExchange installation instructions for the operating system type and bit-level. If the Data Archive server runs on a 64-bit machine, use the PowerExchange 64-bit installation executable or tar file. If the Data Archive server runs on a 32-bit machine, use the PowerExchange 32-bit executable or tar file.

If you are installing on Windows, install the PowerExchange ODBC drivers in the Windows registry:

- On Windows 32-bit systems other than Vista, enter the following command at the command prompt:

  ```
  dtlodbci add
  ```

- On Windows Vista, update the dtlodbci.exe file properties to select **Run this program as an administrator**, and then enter the following command at the command prompt:

  ```
  dtlodbci add
  ```

- On 64-bit Windows, update the registry manually.

For detailed installation instructions, see the *PowerExchange Installation Guide*.

Edit the DBMOVER Configuration File on the Data Archive Server

Edit the DBMOVER configuration file.

Include a NODE statement to define the TCP/IP host name and port that PowerExchange uses to contact the PowerExchange Listener on the machine where the source data for the retirement project is located. Use the following format:

```
NODE=({node_name|node1}
      ,TCPIP
      ,host_name
     ,{port|2480}
    [,{send_bufsize|65536}]
    [,{receive_bufsize|65536}]
    [,{send_size|4096}]
```

```
        [,{receive_size|4096}]
        [,receive_timeout]
        [,{SSL|ZOSSSL}]
        [,service_name]
    )
```

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

Configure Environment Variables on Linux or UNIX

If the Data Archive server is running on Linux or UNIX, edit the path and library path environment variables on the Linux or UNIX system to point to the directories where the files and libraries reside.

Also, set the PWX_HOME environment variable so that PowerExchange can locate various run-time components.

The following table lists the Linux and UNIX environment variables that you need to set:

Variable	Platform	Example
PWX_HOME	All	PWX_HOME=/usr/pwxuser/v910
PATH	All	PATH=/usr/pwxuser/v910
LIBPATH	AIX	LIBPATH=/usr/pwxuser/v910
LD_LIBRARY_PATH	Sun	LD_LIBRARY_PATH=/usr/pwxuser/v910
LD_LIBRARY_PATH	Linux	LD_LIBRARY_PATH=/usr/pwxuser/v910
SHLIB_PATH	HP-UX 10/11	SHLIB_PATH=/usr/pwxuser/v910

To verify that the environment variables are correct, log off and log in again.

Note: After you create ODBC data sources on UNIX, you must update the library path and ODBCINI environment variables. For more information, see "Configure ODBC Data Sources on Linux or UNIX" on page 151.

Step 3: Install and Configure PowerExchange on the Validation Server

Install and configure PowerExchange on the Validation server, as follows:

- To install PowerExchange on the Validation server, use the same procedure as for installing PowerExchange on the Data Archive server. See " Step 2: Install and Configure PowerExchange on the Data Archive Server" on page 147.
- Edit the DBMOVER configuration file.

  Include a NODE statement to define the TCP/IP host name and port that PowerExchange uses to contact the PowerExchange Listener on the machine where the source data for the retirement project is located.

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

- On Linux or UNIX, set the PWX_HOME environment variable so that PowerExchange can locate various run-time components. For example:

  ```
  PWX_HOME=/usr/pwxuser/v910
  ```

Step 4: Install and Configure PowerExchange on Windows

On the Windows system on which you will run the PowerExchange Navigator, install Power Exchange and edit the DBMOVER configuration file.

Install PowerExchange on Windows

On the Windows machine, run the PowerExchange 32-bit executable file to install PowerExchange. You must use the Windows 32-bit executable to install the PowerExchange Navigator, even on a Windows 64-bit machine.

For detailed installation instructions, see the *PowerExchange Installation Guide*.

Edit the DBMOVER Configuration File on the Windows System

Edit the DBMOVER configuration file.

Include a NODE statement to define the TCP/IP host name and port that PowerExchange uses to contact the PowerExchange Listener on the machine where the source data for the retirement project is located.

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

Step 5: Create PowerExchange Data Maps

For each IMS and VSAM data source that you want to retire, create a PowerExchange data map.

Before you create the data maps, select to use a two-tier naming convention for the data maps. To use a two-tier naming convention, click **Options > Preferences**, and on the **Data Maps** tab in the **Preferences** dialog box, select the **Use 2-tier names** option.

1. On the **Resources** tab in the **Resource Explorer**, click **Add** > **Data Map**.

2. In the **Name** dialog box, enter the following information:

Option	Description
Schema Name	Enter a user-defined schema name. PowerExchange uses this name as the first part of the data map name, as follows: `schema_name.data_map_name` The name must consist of alphanumeric characters and begin with a letter. Maximum length is 10 characters.
Data Map Name	Enter a user-defined map name. PowerExchange uses this name as the second part of the data map name, as follows: `schema_name.data_map_name` The name must consist of alphanumeric characters and begin with a letter. Maximum length is 10 characters.
Access Method	Select one of the following access methods: - **DL/1 BATCH**. IMS database on z/OS. - **ESDS**. VSAM Entry Sequenced Data Set (ESDS) on z/OS. - **IMS ODBA**. IMS database accessed through a PowerExchange Listener on z/OS. - **KSDS**. VSAM Key Sequenced Data Set (KSDS) on z/OS. - **RRDS**. VSAM Relative Record Data Set (RRDS) on z/OS.
Import Record Definitions	Select this option to import a copybook into the data map. Clear this option to add only the data map. By default, this option is selected.

3. Enter any additional information that is required for the access method that you selected.
4. If you selected the **Import Records Definitions** option, click **Next**.

   If you cleared this option, click **Finish**.
5. If you selected the **Import Records Definitions** option, provide the required information in the dialog boxes.

For more information about creating data maps, see the *PowerExchange Navigator User Guide*.

Step 6: Configure PowerExchange ODBC Data Sources on the Data Archive Server

Configure PowerExchange ODBC data sources on the Linux, UNIX, or Windows system where the Data Archive server and PowerExchange are installed.

Configure ODBC Data Sources on Linux or UNIX

Use this procedure to configure ODBC data sources on Linux or UNIX.

1. Copy the DataDirect Driver Manager from the computer where PowerCenter and DVO are installed to the server where Power Exchange and the Data Archive engine are installed.

   To copy the DataDirect Driver Manager, copy the following directory to the system with the Data Archive engine:

   ```
   PowerCenter_installation_directory/ODBC6.1
   ```

   Tar the entire directory, copy it using FTP, and then untar the directory on the target system.

2. Add the ODBC6.1/lib directory to one of the following environment variables:
   - LD_LIBRARY_PATH on Sun or Linux
   - SHLIB_PATH on HP-UX
   - LIBPATH on AIX

3. Create the odbc.ini file and set the ODBCINI environment variable to point to this file.

   In the odbc.ini file, include the following line for each data source:

   ```
   Compatibility=ILMBRIDGE
   ```

   The following example shows a section of an odbci.ini file for one data source:

   ```
   [nrdb2_mhz8902]
   Driver=/pwx_installation_directory/source/libdtlodbc.so
   Description=dBase
   Database=
   DBType=nrdb2
   Location=zos_listener_name
   DBQual1=
   DBQual2=
   MaxRows=0
   Compress=N
   Encrypt=N
   ConfirmWrite=N
   PaceSize=0
   InterpretASRows=N
   BulkLoad=N
   DeleteTempFiles=0
   LoadOptions=0
   JCLTemplate=
   CTLTemplate=
   ModeType=0
   ModeTime=0
   Time=0
   Space=
   pace=0
   SecSpace=0
   Compatibility=ILMBRIDGE
   ```

4. For HP-UX systems, change the driver name in the odbc.ini file from libdtlodbc.so to libdtlodbc.sl.

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

Configure ODBC Data Sources on Windows

Use this procedure to configure ODBC data sources on Windows.

Before you configure ODBC data sources on Windows, ensure that the PowerExchange ODBC drivers have been added to the Windows registry.

1. Copy the DataDirect Driver Manager from the computer where PowerCenter and DVO are installed to the server where Power Exchange and the Data Archive engine are installed.

To copy the DataDirect Driver Manager, copy the following directory to the system with the Data Archive engine:

```
PowerCenter_installation_directory/ODBC6.1
```

2. To start the 32-bit **ODBC Data Source Administrator** wizard, perform one of the following actions:
   - On a Windows 32-bit system, double-click **Administrative Tools** in the **Control Panel**. In the **Administrative Tools** window, double-click **Data Sources (ODBC)**.
   - On a Windows 64-bit system, enter the following command at the command prompt:

     ```
     %windir%\SysWOW64\odbcad32.exe
     ```

   The **ODBC Data Source Administrator** wizard appears.
3. On the **System DSN** tab, click **Add**.
4. In the **Create New Data Source** dialog box, select the **Informatica PowerExchange** driver from the list of available drivers and click **Finish**.

   The **PowerExchange Data Source** wizard appears.
5. On the **PowerExchange Data Source** tab, enter the following information:
   - At **Location**, enter the name of the PowerExchange Listener, as defined in the NODE statement in the DBMOVER configuration file.
   - At **Type**, select **NRDB2**.
   - At **Local Codepage**, select **UTF-8 encoding of Unicode**.
   - Depending on the data source type that you select, enter values for other available properties.
6. On the **General** tab, enter the following values at **Integration Mask**:

   ```
   ILMBRIDGE,CPOOL
   ```

   ILMBRIDGE sets a number of options that are required by the JDBC-ODBC bridge. CPOOL enables PowerExchange connection pooling.

   For all data source types, you can optionally enter additional information.
7. Click **OK**.

   The ODBC data source appears in the **System Data Sources** list on the **System DSN** tab in the **ODBC Data Source Administrator** wizard.
8. Click **OK**.

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

Step 7: Configure PowerExchange ODBC Data Sources on the Validation Server

Configure PowerExchange ODBC data sources on the Validation server. Use the same procedures that you used to configure the data sources on the Data Archive server. For more information, see " Step 6: Configure PowerExchange ODBC Data Sources on the Data Archive Server" on page 150.

Step 8: Import z/OS Metadata

Use the Enterprise Data Manager (EDM) to import metadata for the z/OS data sources. Perform this procedure for each data source.

1. To start the Enterprise Data Manager, at the Information Lifecycle Management main window, select **Accelerators > Enterprise Data Manager**.
2. To create an application version for the data source, right-click **Custom Apps Only** in the navigator pane, select **New Application Version**, and enter an application name and description.
3. To import PowerExchange metadata for the application version that you created, complete the following tasks:
   a. Highlight the object and click **File > Mine Database**.

      The **Connect to Mined Database** dialog box appears.

   b. If you have already defined a connection, select it, and enter your z/OS user name and password. The remaining connection information is filled in automatically.

      If you are defining a new connection, enter the following information:

      - At **Database Type**, select **CUSTOM_JDBC**.
      - At **JDBC URL**, enter `jdbc:odbc:dsn_name`, where *dsn_name* matches the name in the odbc.ini file.
      - At **Driver**, enter `sun.jdbc.odbc.JdbcOdbcDriver`.
      - Enter your z/OS user name and password.

   c. In the **Database Mining** wizard, select the schema and list of tables for the data source.
   d. At the prompt to mine child tables of the selected tables, click **No**.

      Note: Because the PowerExchange NRDB2 access method does not support foreign key relationships, you cannot mine child tables. By clicking **No**, you ensure that the PowerExchange Listener on z/OS does not spawn multiple subtasks that would return empty result sets.

      If you enabled the ILMBRIDGE or NOFKEYS Compatibility option for the PowerExchange ODBC connection, the PowerExchange Listener does not spawn multiple empty subtasks, regardless of what you select for this option.

      The EDM begins importing the metadata and displays the percentage complete in the **Database Mining Progress** window.
4. To create and configure an entity object for the application version, complete the following tasks:
   a. Create an entity object and interim table.
   b. Insert tables into the entity.
   c. To avoid validation warnings, mark the first table as the driving table.
   d. Add tables to the interim table.
   e. Add Insert and Delete statements for each table.

      For retirement projects, you typically select a true statement such as **1=1** for the Insert statement and a false statement such as **1=2** for the Delete statement.

   f. Set default columns information for the interim table. For the From clause, enter the name of the driving table. For the Where clause, enter **1=1**.
   g. Delete unneeded entity steps. For retirement projects, you can delete all steps except the Insert into Archive Tables steps.
5. Click **Save**, and close the EDM.

For more information about importing metadata into Data Archive, see the *Data Archive Enterprise Data Manager* guide.

Step 9: Define and Run the Retirement Project

In this step, you define and run the retirement archive project. Running the project copies the nonrelational data to a target database or Data Vault.

Define the Retirement Archive Project

To define a retirement archive project, specify the source and target connections, and define the retirement project application details.

1. To define a source connection, select **Administration > New Archive Source** from the **Information Life Cycle** main window, and enter the following information:
   - At **Connection Type**, select **JDBC-ODBC_BRIDGE 1.0**.
   - Enter the **Data Source Name** as defined in the odbc.ini file.
   - For the **Admin**, **Application**, and **Staging** login names and passwords, enter a valid z/OS user ID and password.
   - At **JDBC Fetch Size**, enter 0.
   - Clear the **Use Staging** check box.
2. Define the target connection, if you have not already defined it.
3. Define the retirement project:
   a. Click **Workbench > Manage Retirement Projects**, and then click **New Retirement Project**.
   b. In the **New/Edit Retirement Project** window, complete the application details.
   c. Select the data source and target connection that you just defined.
   d. Click **Add Entity**, and select the entity for the data source that you are retiring.
   e. Click **Next** until the **Manage Execution** window is displayed. Then select **Publish** to save the retirement project.

When you click the **Publish** button, Data Archive lists the project in the **Manage Retirement Projects** page. By publishing the project, you indicate that the project is ready for scheduling.

For more information, see the *Data Archive User Guide*.

Run the Retirement Project

Schedule or run the retirement project that you published in the previous task.

1. Select **Workbench > Manage Retirement Projects**, and select the project.
2. Complete the tasks to schedule the retirement project and confirm the retirement run.

For more information, see the *Data Archive User Guide*.

Step 10: Validate the Retired Data

Use DVO to verify that the data that was copied to the data archive matches the source data on z/OS.

For more information, see the *PowerCenter Data Validation Option Installation and User Guide*.

Abstract

This article describes how to use Enterprise Data Manager to discover and define foreign key relationships between tables. You can use foreign key relationships to create entities that you use to find and archive related data.

CHAPTER 42

Discovering Foreign Key Relationships in Enterprise Data Manager

This chapter includes the following topics:

- Overview, 156
- Airline Ticket Example, 157
- Step 1. Configure Profile and Discovery Parameters, 157
- Step 2. Start Enterprise Data Manager , 159
- Step 3. Import Metadata from the Database, 159
- Step 4. Discover Table Relationships, 161
- Step 5. Define Foreign Keys, 165

Overview

You can use Enterprise Data Manager to discover table relationships and find foreign keys.

Find foreign key relationships in Enterprise Data Manger to create entities. You can use entities to access related information that exists in multiple tables and archive data based on table relationships.

Before you begin profiling in Enterprise Data Manager, verify the following prerequisites:

- The Model Repository Service must be configured and running. For more information, see *Chapter 7: Model Repository Service* of the *Informatica 9.6.0 Application Service Guide*.
- The Data Integration Service must be running and a profiling warehouse must be configured. For more information, see *Chapter 3: Data Integration Service* of the *Informatica 9.6.0 Application Service Guide*.

To discover foreign key relationships, perform the following tasks:

1. Configure the profile and discovery parameters in Data Archive.
2. Start Enterprise Data Manager.
3. Import metadata from the database into Enterprise Data Manager.
4. Discover table relationships.
5. Define foreign keys.

Airline Ticket Example

You have multiple tables that contain information about airline tickets and you want to archive data for tickets that are older than one year.

You have the following tables that contain airline ticket information:

Table	Columns
TICKET_NUMBER	- TICKET_NUMBER - ISSUE_DATE - CUST_ID
TICKET_COUPON	- TICKET_NUBMER - COUPON - FROM_CITY - TO_CITY
TICKET_COUPON_SAVE	- TICKET_NUMBER - COUPON - IS_USED
TICKET_PARTITION	- TICKET_NUMBER - ISSUE_DATE - PASSENGER_NAME
TICKET_PARTITION_TARGET	- TICKET_NUMBER - PASSENGER_NAME - ID_NUM
TICKET_SAVE	- TICKET_NUMBER - ISSUE_DATE - CREDITCARD

Each table has a TICKET_NUMBER column. You use the TICKET_NUMBER columns as foreign keys to relate the tables to the driving table and create an entity. You use the TICKET_NUMBER table as the driving table because it has the ISSUE_DATE column. After you create the entity, you use the ISSUE_DATE column to determine which ticket numbers are older than one year, and you archive the rows in each table that have the old ticket numbers.

To create an entity that contains all the tables with airline ticket information, you use Enterprise Data Manager to find table relationships that you can use as foreign keys. The foreign key relationships that you select determine how Enterprise Data Manager forms the entity.

Step 1. Configure Profile and Discovery Parameters

Configure the profiling and discovery parameters in Data Archive to connect to the Model Repository Service and the profiling warehouse.

1. In Data Archive, select **Administration** > **System Profile**.

   The **Configuration Settings** page appears.

2. Select the **Profiling and Discovery** tab.

3. Configure the following profiling and discovery parameters:

Domain Name

Name of the Informatica domain.

Host

Name of the machine that hosts the master gateway node.

Port

Port on which the Data Integration Service accepts requests.

User Name

Name of the user that has access to the Model Repository Service.

Password

Password for the user that has access to the Model Repository Service.

Authentication Type

Informatica domain authentication method.

Model Repository Service

Name of the Model Repository Service.

Data Integration Service

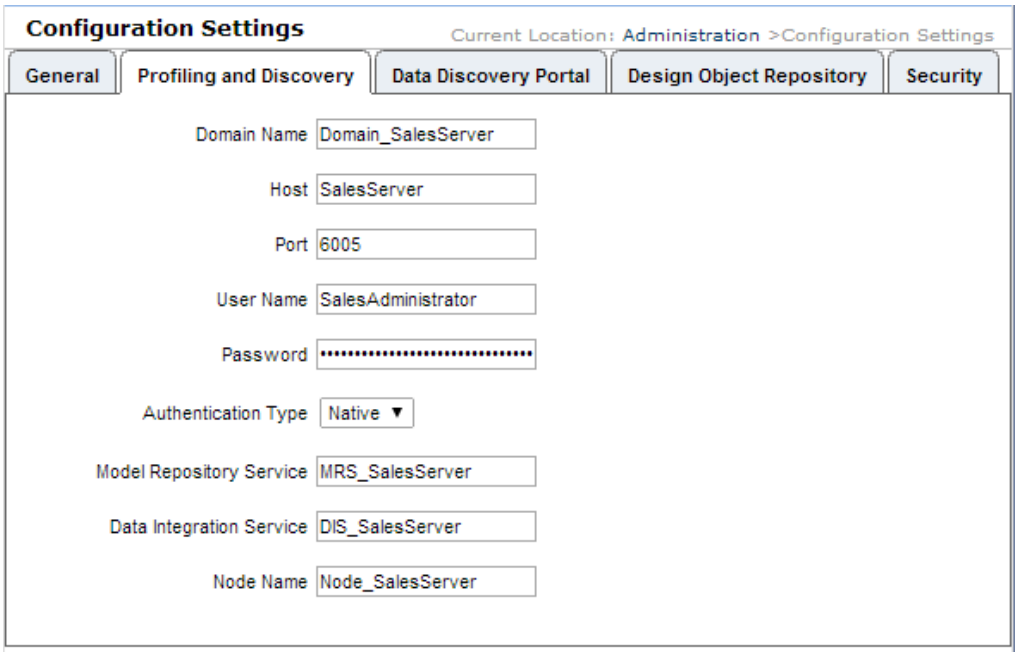
Name of the Data Integration Service.

Node Name

The Informatica domain node on which the Data Integration Service runs.

4. Click **Save**.

The following image shows the **Profiling and Discovery** parameters:



Step 2. Start Enterprise Data Manager

Open Enterprise Data Manger from Data Archive.

1. In Data Archive, select **Accelerators** > **Enterprise Data Manager**.

   The **Save As** window opens.

2. Save `RunEdm.jnlp` on the local machine.

3. To open Enterprise Data Manager, run `RunEdm.jnlp`.

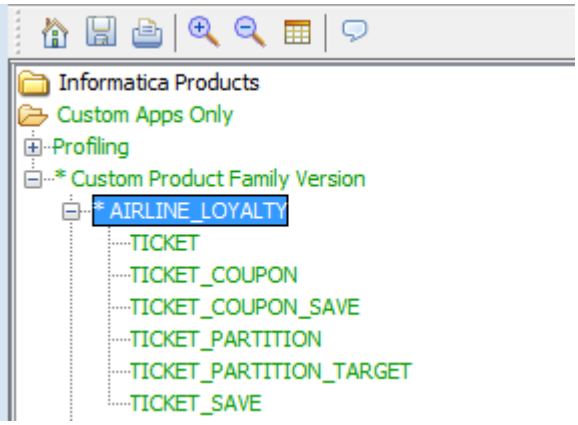
   Note: You must have Java 7 installed on the machine where you run Enterprise Data Manager.

Step 3. Import Metadata from the Database

To import table metadata, define a connection to the database in Enterprise Data Manager.

1. In Enterprise Data Manager, select **View** > **Constraints**.

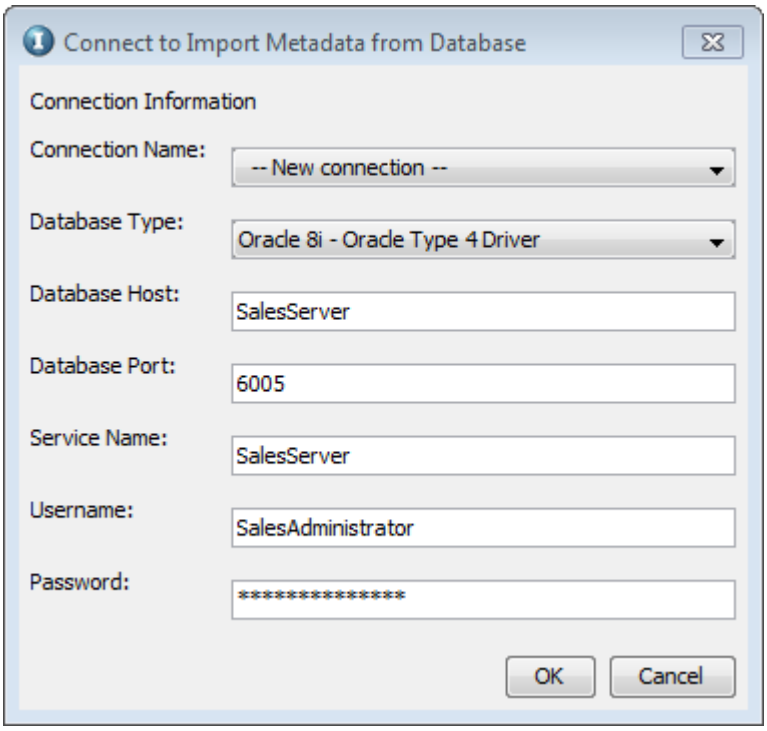
   The constraints entity tree appears.
   The entity tree is organized in a hierarchy. The folder icon represents an application. In the tree, applications contain application versions, schemas, and tables. The following image shows the constraints entity tree:



2. To expand the entity tree, double-click an application.

3. Application versions are directly below the applications in the entity tree. Select an application version and click **File** > **Import Metadata from Database**.

   The **Connect to Import Metadata from Database** window opens.
   The following image shows the connection parameters:



4. In the **Connect to Import Metadata from Database** window, you can use the Connection Name parameter to select a connection that you previously defined, or you can create a connection. Configure the following parameters to create a connection:

   Database Type

   The type of database that you want to connect to.

   Database Host

   The name of the machine that hosts the source data.

   Database Port

   The port number that the source database uses.

   Service Name

   Unique identifier or system identifier for the source database server.

   User Name

   The name of the user that connects to the source database. The user is the owner of the schema that you want to run the profile on. The user must own the tables that you want to run the profile and must have privileges to select table catalogs.

   For Oracle and IBM DB2, the user name is the same as the schema name.

   Password

   The password for the database user name.

5. Click **OK**.

   The **Import Metadata from Database Wizard** opens.

6. Select the schemas that you want to import metadata from. Double-click a schema to move it to the **Selected** box.
7. Click **Next**.
8. Choose one of the following options:
   - **Submit Import Metadata as a Background Job**. Runs the job in the background. If you run the job in the background, you can continue to perform other tasks. Run the job in the background if you have a large volume of metadata to import and to avoid memory errors.

     You must run the job in the background to import metadata from SAP applications.
   - **Continue Import Metadata through EDM**. Runs the job in the foreground. If you run the job in the foreground, you must wait until the job completes to perform another task. Additionally, if you run the job in the foreground and you have a large volume of metadata to import, you might receive memory errors. You might want to run the job in the foreground if you have a low volume of metadata to import.
9. Click **Next**.
10. Select **Update Metadata for Existing Tables**.
11. Click **Finish**.

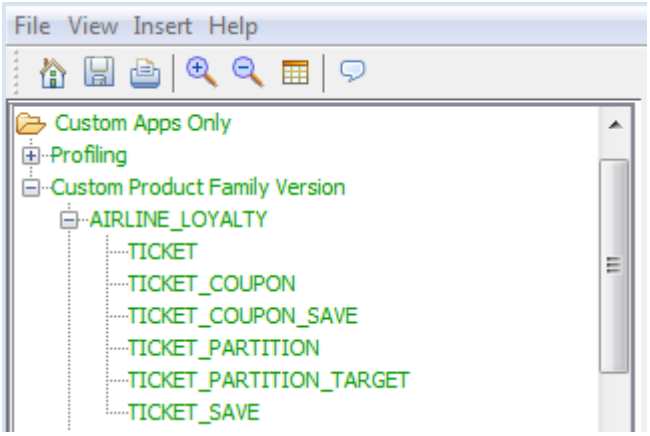
    A message window appears that confirms that the job was successfully submitted.
12. Click **OK** to close the message window.
13. To view the status of the job, select **Jobs** > **Monitor Jobs** in Data Archive. When the job is complete, the job status is Completed.

Step 4. Discover Table Relationships

Define the type of profiling that you want to implement in Enterprise Data Manager.

1. When the import metadata job is complete, select **View** > **Refresh** and expand the entity tree in the **View** > **Constraints** view of Enterprise Data Manager.

   The following image shows the entity tree hierarchy:



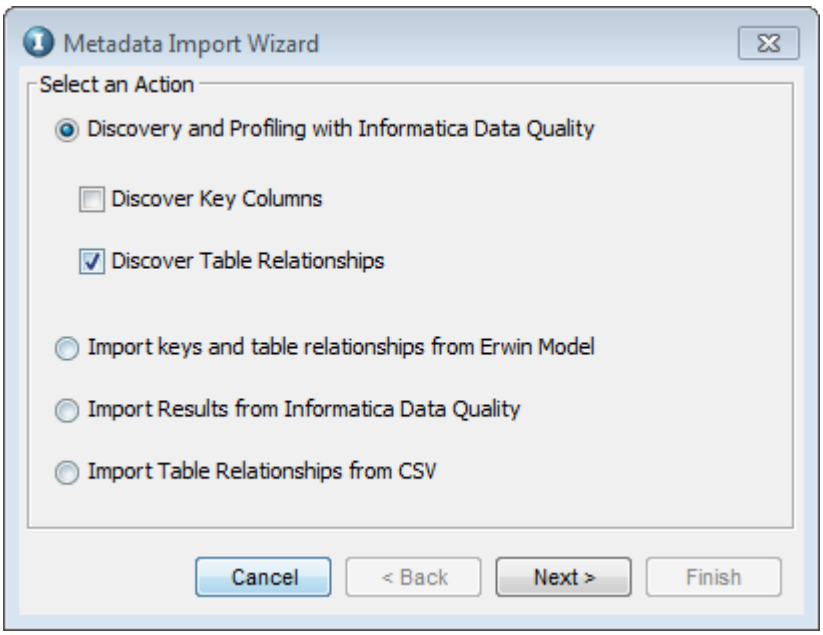
   The Custom Apps Only application contains a Custom Product Family Version application version, an AIRLINE_LOYALTY schema, and TICKET tables.

2. Right-click the schema that you want to run the profile on, and select **Discover Keys and Relationships**. For example, right-click the AIRLINE_LOYALTY schema.

   The **Metadata Import Wizard** opens.

3. In the **Metadata Import Wizard**, select **Discovery and Profiling with Informatica Data Quality** and **Discover Table Relationships**.

   The following image shows the **Metadata Import Wizard** with the correct parameters selected:



4. Click **Next**.

5. In the Connection Name field, select the connection that you defined in Step 3. Optionally, you can define a new connection.

   If you choose to define a new connection, enter the following parameters:

   Database Type

   The type of database that you want to connect to.

   Host

   The name of the machine that hosts the source data.

   Port

   The port number that the source database uses.

   Database Name / SID

   Unique identifier or system identifier for the source database server.

   Source Connection String

   Connect string that determines how the database native client connects to the database. The Data Integration Service uses the connect string to create a connection object to the source database.

The following table describes the service name for each source database:

Source Database	Description	Connect String Syntax
IBM DB2	Database alias name of the remote database configured from the IBM DB2 Connect client.	`dbname`
Microsoft SQL Server	Name of the host of the Microsoft SQL Server database.	`servername@dbname`
Oracle	Full service name or SID for the Oracle database. The TNS name that is configured from the ILM application server to the database server. The name is defined in the application server `TNSNAMES.ORA` file.	`dbname.world`
Data Vault	The ODBC data source name created on the machine hosting Informatica Data Quality.	`dsn`

User Name

The name of the user that connects to the source database. The user is the owner of the schema that you want to run the profile on. The user must own the tables that you want to run the profile on and must have privileges to select table catalogs.

For Oracle and IBM DB2, the user name is the same as the schema name.

Password

The password for the database user name.

6. Optionally, you can click **Advanced Options** to define the following parameters:

Data Types Used in Comparison

Determines whether the profile uses data dictionary data types or generic data types.

Choose one of the following options:

- Inferred Data Types. Select to use generic data types.
- Metadata. Select to use data dictionary data types.

Comparison Case Sensitivity

Determines whether profiling takes into account case differences.

Choose one of the following options:

- Case Sensitive. Select to run case sensitive profiling. The Case Sensitive option can affect job performance.
- Not Case Sensitive. Select to ignore case differences.

Trim Spaces before Comparison

Determines whether the Data Integration Service includes leading or trailing spaces in column data.

Maximum Foreign Keys Returned

The maximum number of inferred columns that the Data Integration Service returns.

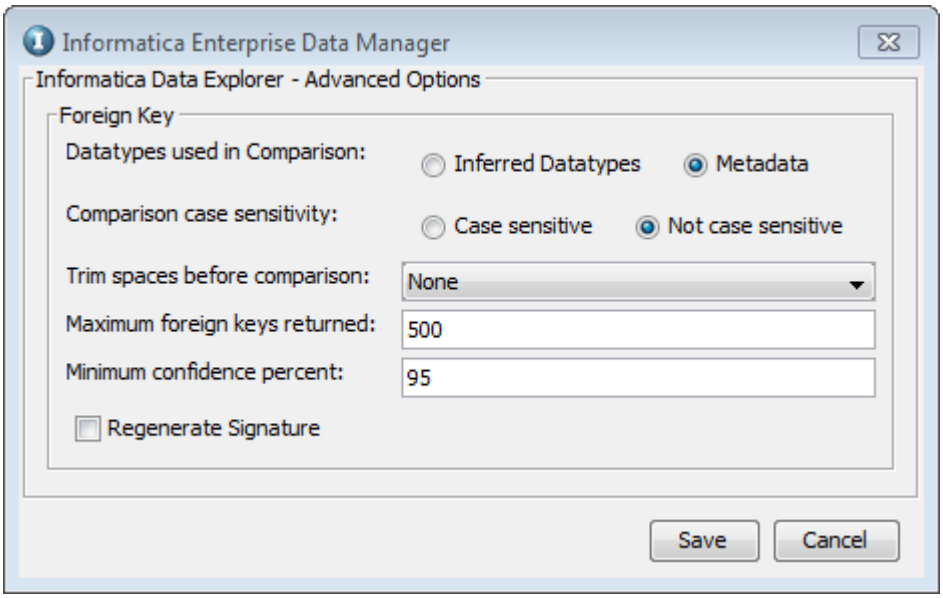
Minimum Confidence Percent

The minimum percentage of key violations that the profile permits for foreign key discovery. For example, if you enter 90, the profile returns foreign keys that have more than 90% accuracy.

Regenerate Signature

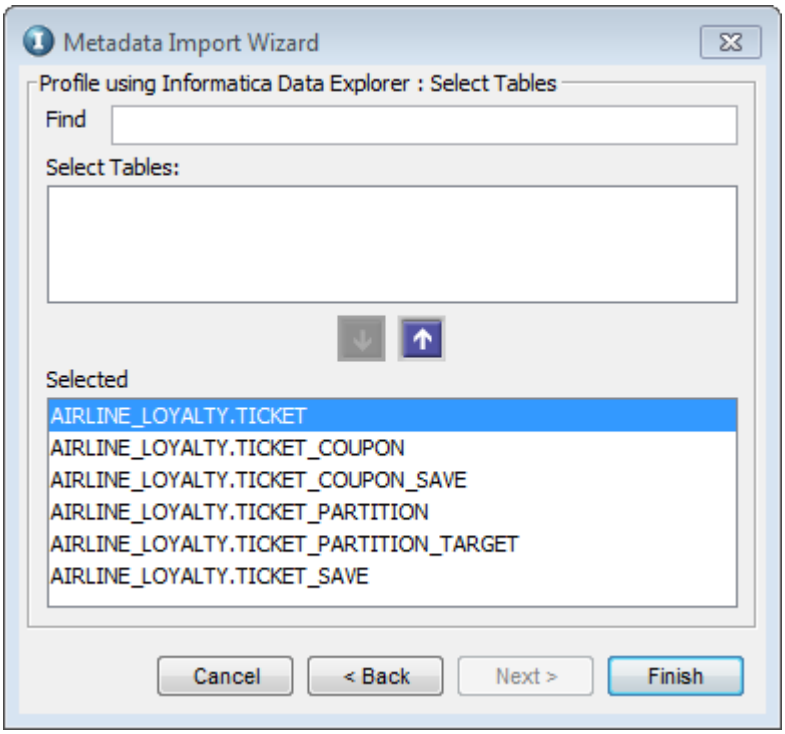
Reloads column signatures if the source data changes.

The following images shows the advanced parameters:



7. Click **Save** to exit the **Advanced Options** window.
8. Click **Next** in the **Metadata Import Wizard**.
9. Select the tables that you want to include in the profiling job. You can double-click a table to move it to the Selected box or you can highlight tables and click the down arrow.

   The following image shows the Metadata Import Wizard with the tables that contain ticket information selected:



10. Click **Finish**.

    A confirmation window appears with the job ID.

11. You can check the status of the job in the **Monitor Jobs** view in Data Archive. When the job is complete, the job status is Completed.

Step 5. Define Foreign Keys

Select the suggested table relationships that you want to use as foreign keys.

1. In the **View** > **Constraints** view of Enterprise Data Manager, right-click the schema that you ran the profiling job on and select **View Suggested Table Relationships**.

   A window appears that shows the suggested table relationships.

2. Select the relationships that you want to define as foreign keys. You want to choose relationships that have a high inclusion percentage.

   In this example, the TICKET table is the driving table and the TICKET_NUMBER column is the primary key for the TICKET table. Therefore, the relationships that have the parent table as TICKET and the parent column as TICKET_NUMBER are selected. The following image shows the suggested table relationships with the relationships selected:

Find Filter

Parent Schema	Parent T...	Parent Column	Child Schema	Child Table	Child Column	Inclusion %
AIRLINE_LOYALTY	TICKET_S...	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_COUPON_S...	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET_S...	TICKET_NUMBER	AIRLINE_LOYA...	TICKET	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET_S...	ISSUE_DATE, PASSEN...	AIRLINE_LOYA...	TICKET_PARTITION...	ISSUE_DATE, P...	100
AIRLINE_LOYALTY	TICKET_S...	CUST_ID, ISSUE_DATE	AIRLINE_LOYA...	TICKET	CUST_ID, ISSU...	100
AIRLINE_LOYALTY	TICKET_S...	ISSUE_DATE, PASSEN...	AIRLINE_LOYA...	TICKET	ISSUE_DATE, P...	100
AIRLINE_LOYALTY	TICKET_S...	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_COUPON	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET_S...	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_PARTITION...	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET_S...	ISSUE_DATE, PASSEN...	AIRLINE_LOYA...	TICKET_PARTITION	ISSUE_DATE, P...	100
AIRLINE_LOYALTY	TICKET_S...	CUST_ID, ISSUE_DATE	AIRLINE_LOYA...	TICKET_PARTITION	CUST_ID, ISSU...	100
AIRLINE_LOYALTY	TICKET_S...	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_PARTITION	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET_C...	COUPON, TICKET_NU...	AIRLINE_LOYA...	TICKET_COUPON	COUPON, TICK...	100
AIRLINE_LOYALTY	TICKET_C...	COUPON, TICKET_NU...	AIRLINE_LOYA...	TICKET_COUPON_S...	COUPON, TICK...	100
AIRLINE_LOYALTY	TICKET	ID_NUM, ISSUE_DATE	AIRLINE_LOYA...	TICKET_PARTITION	ID_NUM, ISSUE...	100
AIRLINE_LOYALTY	TICKET	ISSUE_DATE, PASSEN...	AIRLINE_LOYA...	TICKET_PARTITION	ISSUE_DATE, P...	100
AIRLINE_LOYALTY	TICKET	CUST_ID, ISSUE_DATE	AIRLINE_LOYA...	TICKET_PARTITION	CUST_ID, ISSU...	100
AIRLINE_LOYALTY	TICKET	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_PARTITION	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET	ID_NUM, ISSUE_DATE	AIRLINE_LOYA...	TICKET_PARTITION...	ID_NUM, ISSUE...	100
AIRLINE_LOYALTY	TICKET	ISSUE_DATE, PASSEN...	AIRLINE_LOYA...	TICKET_PARTITION...	ISSUE_DATE, P...	100
AIRLINE_LOYALTY	TICKET	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_PARTITION...	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_SAVE	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET	CUST_ID, ISSUE_DATE	AIRLINE_LOYA...	TICKET_SAVE	CUST_ID, ISSU...	100
AIRLINE_LOYALTY	TICKET	ISSUE_DATE, PASSEN...	AIRLINE_LOYA...	TICKET_SAVE	ISSUE_DATE, P...	100
AIRLINE_LOYALTY	TICKET	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_COUPON_S...	TICKET_NUMBER	100
AIRLINE_LOYALTY	TICKET	TICKET_NUMBER	AIRLINE_LOYA...	TICKET_COUPON	TICKET_NUMBER	100

3. Click **Submit**.

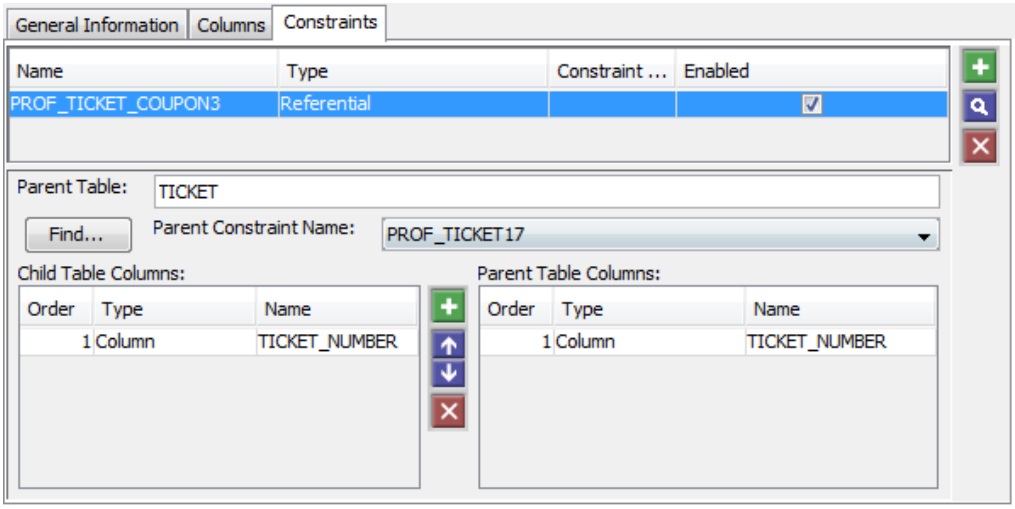
   A window appears that confirms Enterprise Data Manager created the table relationships successfully. Click **OK** to close the window.

4. In the Enterprise Data Manager entity tree, select a table that you included in a foreign key relationship. In the right pane, select the Constraints tab.

   The Constraints tab displays a list of table relationships.

5. In the Constraints tab, select the foreign key relationship that you created. Foreign keys have the Referential relationship type.

   In the lower right pane, you can view the relationship details, such as the child and parent table columns. The following image shows the foreign key relationship details between the TICKET and TICKET_COUPON tables:



6. You can create as many foreign key relationships as required. The foreign keys define the entity that you use to archive data.