



Informatica® Data Archive  
6.5 .1

# Enterprise Data Manager Guide

© Copyright Informatica LLC 2003, 2021

This software and documentation contain proprietary information of Informatica LLC 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 LLC. 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, Informatica Master Data Management, and Live Data Map are trademarks or registered trademarks of Informatica LLC 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 © University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, Inc. All rights reserved. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jqWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. All rights reserved. Copyright © MapR Technologies Inc. All rights reserved. Copyright © Amazon Corporate LLC. All rights reserved. Copyright © Highsoft. All rights reserved. Copyright © Python Software Foundation. All rights reserved. Copyright © BeOpen.com. All rights reserved. Copyright © CNRI. All rights reserved.

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

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

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

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

This product includes Curl software which is Copyright 1996-2013, Daniel Stenberg, <daniel@haxx.se>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://curl.haxx.se/docs/copyright.html>. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

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

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

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

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

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

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at [http://www.boost.org/LICENSE\\_1\\_0.txt](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/hsqldbLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, [http://www.gzip.org/zlib/zlib\\_license.html](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://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/license.html>, <http://www.sqlite.org/copyright.html>, <http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>, <http://www.jdom.org/docs/faq.html>, <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://www.php.net/license/3_01.txt); <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <http://benalman.com/about/license/>; <https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>; <http://www.h2database.com/html/license.html#summary>; <http://jsoncpp.sourceforge.net/LICENSE>; <http://jdbc.postgresql.org/license.html>; <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>; <https://github.com/rantav/hector/blob/master/LICENSE>; <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>; <http://jibx.sourceforge.net/jibx-license.html>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scala-lang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>; <https://aws.amazon.com/asl/>; <https://github.com/twbs/bootstrap/blob/master/LICENSE>; <https://sourceforge.net/p/xmlunit/code/HEAD/tree/trunk/LICENSE.txt>; <https://github.com/documentcloud/underscore-contrib/blob/master/LICENSE>, and <https://github.com/apache/hbase/blob/master/LICENSE.txt>.

This product includes software licensed under the Academic Free License (<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License (<http://www.opensource.org/licenses/cddl1.php>) the Common Public License (<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>), the new BSD License (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

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

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

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

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

#### NOTICES

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

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

Publication Date: 2021-12-21

# Table of Contents

<b>Preface</b> .....	<b>7</b>
Informatica Resources. ....	7
Informatica Network. ....	7
Informatica Knowledge Base. ....	7
Informatica Documentation. ....	7
Informatica Product Availability Matrices. ....	8
Informatica Velocity. ....	8
Informatica Marketplace. ....	8
Informatica Global Customer Support. ....	8
<b>Chapter 1: Introduction to Enterprise Data Manager</b> .....	<b>9</b>
Enterprise Data Manager Overview. ....	9
Data Archive-Specific Functions. ....	9
Third-Party Application Server Support. ....	10
Database Support. ....	10
Installing Enterprise Data Manager. ....	10
Logging in to Enterprise Data Manager. ....	11
Enterprise Data Manager Application Interface. ....	11
Explorer Pane. ....	12
Canvas Pane. ....	14
Details Pane. ....	14
Menu. ....	14
Toolbar. ....	15
Tabbed Navigation. ....	16
Error Messages. ....	16
Status Bar. ....	16
Temporary Save Mechanism. ....	16
<b>Chapter 2: Enterprise Data Manager</b> .....	<b>17</b>
Customizing Data Archive Metadata. ....	17
Customizing Data Archive Metadata Example. ....	18
Basic Operations. ....	20
Metadata Import for SAP Applications. ....	20
Importing Metadata from the Database. ....	21
Adding or Deleting an Application. ....	23
Adding or Deleting an Application Version. ....	23
Adding or Deleting an Application Module. ....	23
Adding or Deleting an Entity. ....	23
Copying and Pasting Modules. ....	23
Creating or Deleting Reporting Statements. ....	23

Creating a List of Values. . . . .	24
Creating a List of Value Variant. . . . .	28
Updating Entity Information. . . . .	28
Copying Archived Entities to Customize. . . . .	29
Copying Standard Entity Metadata for Data Vault Search. . . . .	30
Adding and Deleting Interim Tables. . . . .	30
Inserting and Removing Business Tables. . . . .	30
Business Tables. . . . .	30
Specifying Translation Columns (Independent Archive). . . . .	31
Specifying Integrity Constraints. . . . .	33
Interim Tables. . . . .	33
Entities. . . . .	35
Before You Create an Entity. . . . .	35
Multiple Entity Generation. . . . .	36
Automatic Entity Generation. . . . .	37
Entities for Informix Databases. . . . .	38
Retirement Entity Generation. . . . .	38
Troubleshooting Entities. . . . .	41
Exporting Accelerators. . . . .	42
Data Model Metadata. . . . .	43
Data Archive Metadata (Custom/Standard Entities). . . . .	43
Component (Standard Data). . . . .	43
Exporting a Custom Product Family Version. . . . .	43
Exporting an Entity. . . . .	44
Importing Accelerators. . . . .	44
Importing an Entity. . . . .	45
Import Job Parameters. . . . .	46
Importing as a Background Job. . . . .	47
Importing through the Enterprise Data Manager. . . . .	47
Virtual Views. . . . .	48
Virtual Views Taskflow. . . . .	48
Creating a Virtual View. . . . .	49
Enabling Search Data Vault. . . . .	49
Selecting Columns for the Search Index. . . . .	49
Specifying Large Number of Columns at a Time. . . . .	50
Troubleshooting Enterprise Data Manager. . . . .	50
<b>Chapter 3: ILM Repository Constraints. . . . .</b>	<b>52</b>
ILM Repository Constraints Overview. . . . .	52
Constraint Definition. . . . .	53
<b>Chapter 4: Partition Exchange Purging. . . . .</b>	<b>54</b>
Partition Exchange Purging Overview. . . . .	54

Working with Partition Exchange. . . . .	54
Partition Exchange Procedure. . . . .	56
Database Requirements. . . . .	56
Configuring Partition Exchange. . . . .	57
Step 1. Create and Configure the Interim Tables. . . . .	57
Step 2. Configure the Entities. . . . .	61
Step 3. Verify the Primary Key Constraints. . . . .	62
Managing Partition Exchange. . . . .	63
Disabling or Removing Partition Exchange. . . . .	63
<b>Chapter 5: APIs. . . . .</b>	<b>64</b>
TPT_EXTRACT_SCRIPT_TEMPLATE. . . . .	64
Script Template Parameters. . . . .	64
<b>Chapter 6: Salesforce Accelerator. . . . .</b>	<b>66</b>
Salesforce Accelerator Overview. . . . .	66
Connecting to Salesforce and Importing Metadata. . . . .	66
Importing Metadata from Child Tables. . . . .	67
Salesforce Accelerator Entities . . . . .	69
Task Entity Tables. . . . .	70
Task Entity Parameters. . . . .	70
Task Entity Business Rules. . . . .	71
Excluded Child Tables. . . . .	71
Event Entity Tables. . . . .	72
Event Entity Parameters. . . . .	73
Event Business Rules. . . . .	73
Excluded Child Tables. . . . .	73
Partial Recurrence Activities. . . . .	74
<b>Appendix A: SAP Application Retirement Entities. . . . .</b>	<b>78</b>
SAP Application Retirement Entities Overview. . . . .	78
Entities for Attachments. . . . .	78
Entities for Transparent HR Cluster Tables and Text Tables. . . . .	80
<b>Appendix B: Import Formats for Constraints. . . . .</b>	<b>81</b>
Import Formats for Constraints Overview. . . . .	81
Importing Referential Integrity Constraints. . . . .	81
Importing Primary Integrity Constraints. . . . .	82
Importing Column Translation Data. . . . .	82
<b>Appendix C: Glossary. . . . .</b>	<b>83</b>
<b>Index. . . . .</b>	<b>86</b>

# Preface

Read the *Data Archive Enterprise Data Manager Guide* to learn how to use the Enterprise Data Manager tool. You can use the tool to manage application archiving, manage records, perform sensitive data masking, and application retirement.

## Informatica Resources

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

### Informatica Network

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

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

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

### Informatica Knowledge Base

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

To search the Knowledge Base, visit <https://search.informatica.com>. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at [KB\\_Feedback@informatica.com](mailto:KB_Feedback@informatica.com).

### Informatica Documentation

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

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at [infa\\_documentation@informatica.com](mailto:infa_documentation@informatica.com).

## Informatica Product Availability Matrices

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

## Informatica Velocity

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

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

## Informatica Marketplace

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

## Informatica Global Customer Support

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

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

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

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



# CHAPTER 1

## Introduction to Enterprise Data Manager

This chapter includes the following topics:

- [Enterprise Data Manager Overview, 9](#)
- [Data Archive-Specific Functions, 9](#)
- [Third-Party Application Server Support, 10](#)
- [Database Support, 10](#)
- [Installing Enterprise Data Manager, 10](#)
- [Logging in to Enterprise Data Manager, 11](#)
- [Enterprise Data Manager Application Interface, 11](#)

### Enterprise Data Manager Overview

Enterprise Data Manager is an integrated tool that provides centralized management of all application data throughout an organization. You can manage application archiving, records management, database subsectioning, sensitive data masking, and application retirement from a unified, integrated platform.

You use Enterprise Data Manager to build metadata for Data Archive. Enterprise Data Manager adapts dynamically and extends related functionality for metadata definitions.

### Data Archive-Specific Functions

Enterprise Data Manager allows the following metadata definitions for installation:

- Defining accelerators for transaction data in the form of nodes in the hierarchical representation for an application such as the application version, application modules, entities, and tables or interim tables.
- Defining reporting statements, entity parameters, list of values, and business rules for entities.
- Defining indexes and constraints on tables.

# Third-Party Application Server Support

The Information Lifecycle Management Platform develops accelerators for the following third-party application servers:

- Amdocs
- JD Edwards
- Oracle E-Business
- PeopleSoft
- Siebel
- Custom applications

# Database Support

Data Archive is customized for applications on the following databases:

- IBM Db2 AS400
- IBM Db2 LUW
- IBM Netezza
- IBM Db2 zOS
- Informix
- Microsoft SQL Server
- MongoDB
- Oracle
- PowerExchange adapter
- Sybase
- Salesforce
- Teradata

# Installing Enterprise Data Manager

To use Enterprise Data Manager, you download and extract `EDM.zip`. Azul OpenJDK is included in the `EDM.zip` file.

1. From the Data Archive UI, click **Accelerators > Enterprise Data Manager**.  
A new tab opens and the `EDM.zip` file downloads.  
You can continue to work in Data Archive as the file downloads.
2. Extract the files to a folder.
3. Navigate to the folder where you extracted the files.

4. Double-click `edm.bat`.  
The **Enterprise Data Manager Login** window appears.

## Logging in to Enterprise Data Manager

When you open Enterprise Data Manager, specify the URL and the login credentials for the Data Archive instance. Enterprise Data Manager uses the information to connect to Data Archive.

1. On the **Enterprise Data Manager Login** window, enter the URL of the Data Archive instance with port number in the **URL** field.

Use the following format:

```
http://<DA instance>:<DA port>
```

```
or https://<DA instance>:<DA port>
```

For https URLs, <DA instance> must match the Common Name (CN) on the keystore certificate.

To check the CN in the correct keystore file, follow the path specified in the `informia.sslKeystoreFile` property in the `conf.properties` file.

For example: `https://<Server.Domain.com>:8080`

2. Enter your user name and password.
3. Click **OK**.

Enterprise Data Manager authorizes the user with a valid login profile for accessing the Enterprise Data Manager Server according to the roles granted by the administrator from the Data Archive user interface.

**Note:** You must use the Data Archive Web user interface to create user accounts.

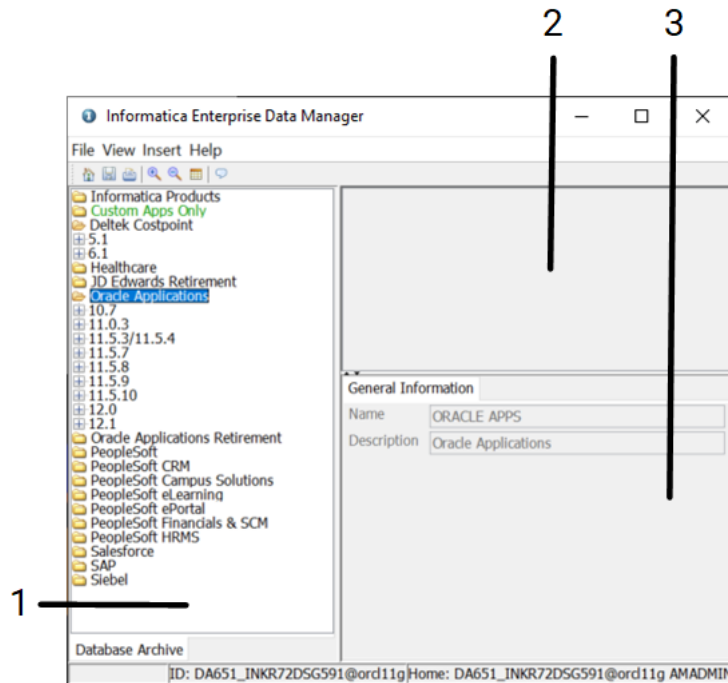
The following table shows user authorization information for each user role:

User Role	Privileges
Server Viewer	View metadata in the Enterprise Data Manager.
Server Developer	Develop and view metadata for Enterprise Data Manager.

## Enterprise Data Manager Application Interface

In the Enterprise Data Manager Application interface you can perform most metadata development tasks in a single window.

The interface consists of three panes as shown in the following image.



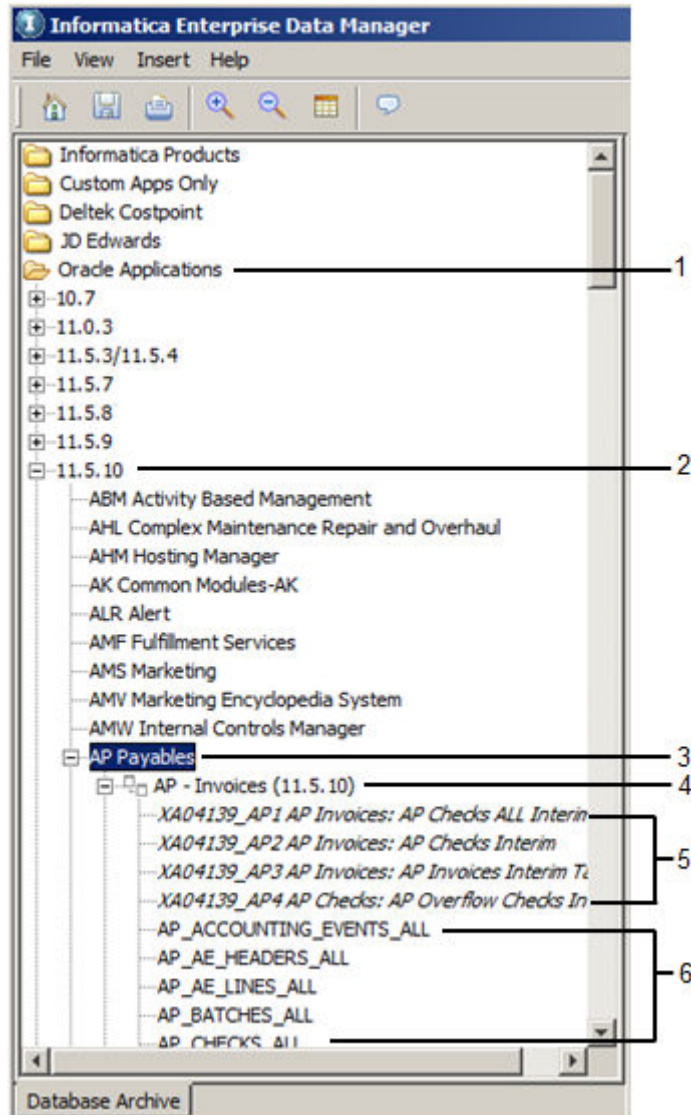
1. Explorer pane
2. Canvas pane
3. Details pane

Each component of the application interface is explained in the following sections.

## Explorer Pane

Data Archive internally employs a hierarchical structure for its components, and carries out all operations based on it. The Explorer pane shows the tree view of the components.

The following image shows the different levels in the hierarchical structure:



1. Application
2. Application Version
3. Application Module
4. Entity
5. Interim Tables
6. Business Tables

## Nodes

The Explorer pane displays the following nodes:

### Application

Custom or brand name ERP applications, such as Oracle E-Business Suite and PeopleSoft.

### Application Version

Supported versions for a particular application.

### Application Module

Schemas for an application version.

### Entity

Group of tables whose data collectively comprise a set of business transactions. Each table in the entity connects to one or more other tables through a primary and foreign key relationship.

### Interim Tables

Temporary tables generated for an entity during an archive.

### Business Tables

All related tables created to define an entity. Business tables appear at the same level of the hierarchy as the interim table.

## Canvas Pane

The Canvas pane shows the diagram view of the relationships between tables within an entity.

The following list explains what the colors on the Canvas pane indicate:

- A blue background for a table name indicates a business table.
- A gray arrow indicates a relationship between tables.
- An orange arrow within connecting tables indicates a referential integrity constraint.
- A green background for a table name indicates a interim table.
- A green line indicates a relationship that includes interim tables.

## Details Pane

The Details pane contains detailed information related to a node or component.

You can select a node, such as an application version, entity, or interim table from the Explorer pane and view the details in the Details pane.

You can select a component such as a table or relationship from the Canvas pane and view the details in the Details pane.

## Menu

The following table describes the menu structure.

Menu Sequence	Function
File > Connect to Repository	Create a connection profile or use an existing one with information such as database Driver, Host, Port, SID, User name, and Password (for Schema). You can use this option to connect to a different Data Archive repository. When you connect to Enterprise Data Manager from the <b>Accelerators</b> menu, this option is disabled.
File > Login	Provide the user name and password for logging in to Enterprise Data Manager.
File > Connect to ID Repository	Specify a connection profile for a database that stores sequences for unique IDs generated while creating nodes and subnodes in the application tree.

Menu Sequence	Function
File > Save	Save changes to the Data Archive repository.
File > Import Metadata from Database	Integrate one or more tables from a schema into the metadata. Before you import metadata from a database, verify the following requirements: <ul style="list-style-type: none"> <li>- The user has the Import Metadata system-defined role.</li> <li>- The <b>informia.enable.mining</b> property in the <code>conf.properties</code> file is set to Y.</li> </ul> For more information, see the <i>Data Archive Administrator Guide</i> .
File > Import	Import Data Model accelerators or Data Archive accelerators for standard or custom applications. You can also use this option for bulk import of (primary and referential) integrity constraints and translation columns.
File > Export	Export Data Model accelerators or Data Archive accelerators for standard or custom applications.
File > Print	Print Diagram View (Canvas pane) of the selected entity.
File > Exit	Exit from Enterprise Data Manager.
View > Refresh	Refresh the information on the Enterprise Data Manager pane.
View > Zoom In	Magnify Entity illustration in the Canvas pane.
View > Zoom Out	Shrink Entity illustration in the Canvas pane.
View > Columns > All	View Entity tables with all columns and relationships.
View > Columns > Related	View entity tables with respective columns that participate in a relationship.
View > Columns > None	View table names, without columns.
View > Default Layout	Return to the default layout.
Help > About	Retrieve information about the current Enterprise Data Manager version and build.

## Toolbar

A toolbar appears at the upper left corner of the application, with common functions. The following table describes the functions.

Toolbar Button (and tool tips)	Function
Connection to Repository	Connect to a Connection Profile or specify a new one. This is synonymous with <b>File &gt; Connection to Repository</b> . You can also use this option to connect to a different Data Archive repository. When you connect to Enterprise Data Manager from the <b>Accelerators</b> menu, this option is disabled.
(Save)	Commit metadata changes to the Data Archive repository. This is synonymous with <b>File &gt; Save</b> .

Toolbar Button (and tool tips)	Function
(Print)	Print Diagram View for currently selected entity. This is synonymous with <b>File &gt; Print</b> .
(Zoom In)	Enlarge Entity illustration in the Canvas pane by a zoom factor. This is synonymous with <b>View &gt; Zoom In</b> .
(Zoom Out)	Minimize Entity illustration in the Canvas pane by a zoom factor. This is synonymous with <b>View &gt; Zoom Out</b> .
(Columns toggle)	Toggle between different views for displaying columns in entity tables in the Canvas pane. View all columns, related columns, or no columns. This is synonymous with <b>View &gt; Columns &gt; All</b> , <b>View &gt; Columns &gt; Related</b> , and <b>View &gt; Columns &gt; None</b> .
(Annotate)	Create an annotation for a node in the application tree.

## Tabbed Navigation

When you select a view from the **View** menu, a tab appears at the bottom left corner of the window.

## Error Messages

An error message appears as a popup, whenever an operation initiated by the user fails to complete. A notification is also displayed in the status bar. To ensure metadata consistency, the application usually does not allow the user to proceed further unless the error is resolved.

## Status Bar

A status bar at the bottom of the window displays the following information:

- Status Messages

Status messages display the status of an operation initiated by a user. They also display in response to an error that occurs as a result of a user operation. Status messages appear as a notification to the extreme left in the status bar.

- Connection to ID Repository

The schema appears next to the status messages. The schema stores information on ID sequences for nodes that you create in the hierarchical structure of Enterprise Data Manager metadata. This is generally negative for custom components (a user-defined application or entity) and positive for standard components (offered by ERP).

- Home Schema

The Home Schema which stores the metadata that you can alter using Enterprise Data Manager, displays on the extreme right.

## Temporary Save Mechanism

Changes made to the metadata are not directly committed to the Home Schema, unless you explicitly save it by clicking **Save** on the Enterprise Data Manager Toolbar or use **File > Save**. When you add a node to an application hierarchy, it is initially saved with an asterisk (\*) before its name in the Explorer pane.



## CHAPTER 2

# Enterprise Data Manager

This chapter includes the following topics:

- [Customizing Data Archive Metadata, 17](#)
- [Basic Operations, 20](#)
- [Business Tables, 30](#)
- [Interim Tables, 33](#)
- [Entities, 35](#)
- [Exporting Accelerators, 42](#)
- [Exporting a Custom Product Family Version, 43](#)
- [Exporting an Entity, 44](#)
- [Importing Accelerators, 44](#)
- [Importing an Entity, 45](#)
- [Import Job Parameters , 46](#)
- [Virtual Views, 48](#)
- [Enabling Search Data Vault, 49](#)
- [Troubleshooting Enterprise Data Manager, 50](#)

## Customizing Data Archive Metadata

Before you can archive or retire data, you must customize the Data Archive metadata.

To customize Data Archive metadata, perform the following steps:

1. Import metadata from the source database.
2. Add an application module for the application.
3. Add entities to the application module.
4. Define parameters for each entity.
5. Insert default steps for each entity.
6. Add interim tables.
7. Add tables to the interim tables.
8. Define default columns for the interim tables.
9. Define business rules for the interim tables.

10. Generate default indexes.
11. Specify reporting statements.

## Customizing Data Archive Metadata Example

A practical example of Data Archive Metadata definition is as follows:

1. Import the metadata from the source database. Create a node for the respective application under the intended Application Version. For example, add “BOM Bills of Material” as an Application to Oracle Version 11.5.10. The node you create is called the Application Module.
2. Add an entity to the application module. For example: “BOM - Standard Cost Update (11.5.10)” to the application module “BOM Bills of Material.”
3. Define one or more parameters for the entity. The following table shows examples of parameters:

Order	Column	Name	Data type	Required	List of Values	Security group
1	p_cost_org_id	Cost Organization	Number	-	INV Organization	-
2	p_update_date	Update Date	Time	true	INV Accounting Period (PF-8)	Organization Unit

The final SQL query for List of Values “INV Organization” and “INV Accounting Period (PF-8)” reads as follows:

List of Value Name	SQL Test Query
INV Organization	<pre>SELECT A.organization_name "Organization", A.organization_code "Code", TO_CHAR(A.organization_id) FROM ORG_ORGANIZATION_DEFINITIONS A WHERE A.inventory_enabled_flag = 'Y' ORDER BY A.organization_name, A.organization_name</pre>
INV Accounting Period (PF-8)	<pre>SELECT A.period_name "Period Name ", T.organization_name "Organization Name", T.organization_code "Organization Code", A.period_name "Period Name ", T.organization_code "Organization Code", T.organization_name "Organization Name", TO_CHAR(A.schedule_close_date, 'DD-MON-YYYY') FROM ORG_ACCT_PERIODS A, ORG_ORGANIZATION_DEFINITIONS T WHERE A.open_flag = 'N' AND A.organization_id = T.organization_id AND T.organization_id = NVL(:params.param1,T.organization_id)</pre>

**Note:** It is assumed that these List of Values have been created and available under the **List of Values** tab of the Explorer pane.

4. Insert default steps for Archive Definition under the Entity “BOM - Standard Cost Update (11.5.10).”

5. Add required interim tables. For example, for the Entity “BOM - Standard Cost Update (11.5.10)”, add an interim table with the following FROM and WHERE clauses:

Clause Type	Statement
FROM	bom.Cst_Cost_Updates C, inv.Mtl_Parameters Mp
WHERE	C.Organization_Id =decode(:p_cost_org_id,null,c.organization_id,:p_cost_org_id) And C.Organization_Id = Mp.Organization_Id And trunc(Update_Date) <= :P_Update_Date and Mp.Primary_Cost_Method=1

6. Add tables to the **Tables** tab of this interim table that will participate in the archive job execution, and define their respective INSERT, DELETE, and UPDATE statements. For this example, specify the following information:

Order	Name	Insert Statement	Delete Statement
1	CST_STD_COST_ADJ_VALUES	a.cost_update_id in (select cost_update_id from XA04156_BOM1 x where x.master_org_flag = 'Y' and x.purgeable_flag = 'Y')	a.cost_update_id in (select cost_update_id from XA04156_BOM1 x where x.master_org_flag = 'Y' and x.purgeable_flag = 'Y')
2	CST_STANDARD_COSTS	A.cost_update_id IN (Select cost_update_id From XA04156_BOM1 x where x.purgeable_flag = 'Y')	A.cost_update_id IN (Select cost_update_id From XA04156_BOM1 x where x.purgeable_flag = 'Y')
3	CST_ELEMENTAL_COSTS	A.cost_update_id IN (Select cost_update_id From XA04156_BOM1 x where x.purgeable_flag = 'Y')	A.cost_update_id IN (Select cost_update_id From XA04156_BOM1 x where x.purgeable_flag = 'Y')
4	CST_COST_UPDATES	A.cost_update_id IN (Select cost_update_id From XA04156_BOM1 x where x.purgeable_flag = 'Y')	A.cost_update_id IN (Select cost_update_id From XA04156_BOM1 x where x.purgeable_flag = 'Y')

7. Add the default columns. For example, for the interim table created above, define the following default columns:

Order	Name	Type	Length	PK	Select Clause
1	cost_update_id	Number	-	true	C.Cost_Update_Id
2	cost_update_date	Date	-	false	C.Update_Date
3	org_id	Number	-	false	mp.Organization_Id
4	master_org_flag	Char	1	false	decode(Mp.Organization_Id, mp.cost_organization_id,'Y','N')

Order	Name	Type	Length	PK	Select Clause
5	purgeable_flag	Char	1	false	'Y'
6	stats_date	Date	-	false	-

8. Define business rules. For example, on the interim table generated for “BOM - Standard Cost Update,” define the business rule “HAS\_NONPURGEABLE\_STD\_COSTS” with the following information:
  - Description. Cost Update has standard costs that are not being purged in this cycle.
  - Business rule order. 1
  - Condition. `A.cost_update_id in (select cost_update_id From bom.cst_standard_costs b where not exists (select 1 from XA04156_BOM1 x where x.purgeable_flag = 'Y' and b.cost_update_id = x.cost_update_id and x.org_id=b.organization_id))`
9. Generate default indexes.
10. Specify reporting statements. For example:

Order	Statement Label	Literals	Group
1	COST_UPDT_DT	A	Primary

A parameter **cost\_update\_date** is specified here for the statement which relies on INTERIM\_VALUE.

## Basic Operations

Basic operations for Data Archive Metadata definition are included in the subsections below.

### Metadata Import for SAP Applications

The Import Metadata job runs steps that are specific to SAP applications when you indicate that the source is an SAP application. The Import Metadata job reads metadata for all SAP tables in the database, imports the data to the ILM repository, and adds SAP-specific table types.

When you import metadata from SAP applications, the Import Metadata job uses the parameters that you configured for the job to connect to the database. The job reads metadata from the database and the SAP catalog tables for all pool, cluster, and transparent tables, except for the tables that are included in the pre-packaged entities for SAP. Metadata for the transparent HR PCL1-PCL5 cluster tables, the transparent STXL table, and attachment tables are included in the pre-packaged entities. The job also reads metadata for ADK files.

The job adds an SAP-specific table type to the imported metadata in the ILM repository. The table type is required to differentiate SAP tables. When you run the Retirement Auto Entity Creation job, the job uses the table types to determine the tables to add to entities and the entity step configuration.

## Table Types

When you import metadata from SAP applications, the Import Metadata job adds table types to the corresponding metadata tables in the ILM repository. The Retirement Auto Entity Creation job uses the table types to determine the tables to add to entities and the procedures the job adds to the entity steps.

Table types are required to determine the data source of the tables in the entity. Entities can include one source for the Insert into Archive Tables step. Some SAP tables store data in the database and in an external file system. A table is included in two separate entities if the table has data in the database and archived data in ADK files in an external file system. One entity includes the database as the source. The other entity includes the external file system as the source for the archived data.

The Import Metadata job adds one of the following table types to the imported metadata in the ILM repository:

Table Type	Description
Converted Transparent Table with Archived Data	Logical tables that store archived data in ADK files in an external file system. This table type is for archived data from transparent HR PCL1-PCL5 cluster tables and the STXL text table.
Pool Cluster	Pool and cluster logical tables that store data in the database in an SAP proprietary format.
Pool Cluster with Archived Data	Pool and cluster logical tables that store archived data in ADK files in an external file system.
Transparent Table	Physical tables that store data in the database.
Transparent Table with Archived Data	Physical tables that store archived data in ADK files in an external file system.

The prepackaged entities for SAP include tables with the Converted Transparent Tables and Attachments table types. The job does not add these table types because the prepackaged entities include these table types.

## Importing Metadata from the Database

Before you can run an archive or retirement job, you must import metadata from the source database to the ILM repository.

To import metadata from a database, ensure the following prerequisites are in place:

- You have the Import Metadata system-defined role.
- You set the **informia.enable.mining** property file to Y in the `conf.properties` file.

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

For SAP application retirement, you must initiate the metadata import from a customer defined application. Run the Copy Application Version for Retirement job to copy the prepackaged SAP application to a customer defined application.

To import metadata from the source database to the ILM repository, perform the following steps:

1. Select an application version and click **File > Import Metadata from Database**.

The **Connect to Import Metadata from Database** dialog box appears.

2. Enter the connection details.

To create a connection for Informix sources, use the following syntax for the service name:

```
<server_name>;databasename=<dbname>
```

If the connection type is a Microsoft SQL Server, provide the following attributes:

```
Database_Port=xxxx;encryptionMethod=SSL;ValidateServerCertificate=false/true;CryptoProtocolVersion=TLsv1.2
```

Where xxxx is the port number. For example:

```
Database_Port=1433;encryptionMethod=SSL;ValidateServerCertificate=false/true;CryptoProtocolVersion=TLsv1.2
```

CryptoProtocolVersion is the name of the protocol supported by the database server.

If the database type is Oracle 11g, 12c, or 18c, the following options are available:

Oracle Database Type	Options
Oracle 11g, 12c, or 18c	<p><b>SSL Enabled.</b> Establish an SSL connection to the Oracle database.</p> <p><b>Note:</b> To use SSL, you must add the SSL certificate of the Oracle server to the respective Java cacerts file where Enterprise Data Manager is running.</p> <p><b>Advanced Security Options (ASO)</b></p> <ul style="list-style-type: none"> <li>- <b>ASO Encryption Level</b></li> <li>- <b>ASO Encryption Type</b></li> <li>- <b>ASO Checksum Level</b></li> <li>- <b>ASO Checksum Type</b></li> </ul> <p><b>Note:</b> You cannot use SSL and ASO options together. If you enable SSL, ASO options are not available. For more information, see the <i>Data Archive Administrator Guide</i>.</p>
Oracle 11g or 12c	<p><b>OID Enabled.</b> Establish a connection through an OID database, to the internal Oracle database. Enter the OID database host name in the <b>Host</b> field. Enter the service name in the <b>Service Name</b> field. The service name specifies the distinguished name of the database, which is configured in OID.</p> <p>For example:</p> <pre>ServiceName = ORA12C,cn=OracleContext,dc=informatica,dc=com</pre> <p><b>Note:</b> You cannot use OID and ASO options together. When you enable OID, ASO options are not available.</p>

3. Click **OK**.

The **Import Metadata from Database Wizard** appears.

4. Select the schemas to import metadata from.

To insert the tables into the **Selected** box, click the down arrow. To select multiple tables or all tables, use the control keyboard options.

**Note:** Tables that have special characters in the table name, do not appear in the list. Special characters include: ~, !, @, #, \$, %, ^, &, \*, \_, +, -, and =.

5. Click **Next**.

6. Choose one of the following options to run the job:

- **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.

Required 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 other tasks. Additionally, if you run the job in the foreground and you have a large volume of metadata to import, you might receive memory errors. Consider running the job in the foreground if you have a low volume of metadata to import.

7. Click **Next**.

8. Enter the mining parameters.

To import metadata from databases for SAP applications, select **SAP Application Metadata**. The Import Metadata job runs steps that are specific to SAP applications. If you import from SAP applications and do not select the parameter, the job will not import metadata for pool tables, cluster tables, and tables that store data in ADK files.

9. Click **Finish**.

10. Click **OK**.

## Adding or Deleting an Application

You can add an application from **Insert > New Application**.

To delete an application, right-click **Delete Application**.

## Adding or Deleting an Application Version

You can add an application version from **Insert > Application Version**. You can also right-click the corresponding application and select **New Application Version**.

To delete an application version, right-click **Delete Application Version**.

## Adding or Deleting an Application Module

You can add an application module from **Insert > New Application Module**. A menu option for the corresponding application version, called **Add Application Module** is also available for similar purposes.

To delete an application module, right-click **Delete an Application Module**.

## Adding or Deleting an Entity

You can add an entity from **Insert > New Entity**. A menu option for the corresponding application module, called **Add New Entity** is also available for the same.

To delete an entity from the menu option, select **Delete Entity**.

## Copying and Pasting Modules

To copy all the modules for a particular application version, right-click the corresponding application version and select **Copy Archive Modules**.

To paste the modules into another application version, select **Paste Archive Modules**.

## Creating or Deleting Reporting Statements

Create reporting statements for interim tables in an entity definition to generate Detail reports when the same is requested during an Data Archive Definition specified from the Data Archive user interface.

To create a reporting statement, perform the following steps:

1. Select the parent application version.
2. Click **Insert > Statement**. The **New Statement** dialog box appears.
3. Specify a label for the statement.
4. Specify whether values in a report must be generated using an SQL query or obtained from the interim table.
5. Specify SQL query if SQL is selected.
6. Click **Finish** to save the statement. The statement appears under the **Statements** tab in the Explorer pane.
7. Click **Save** to commit changes to Home Schema.

To delete a statement, navigate to the **Statements** tab in the Explorer pane. Select **Delete Statement** from the menu.

## Creating a List of Values

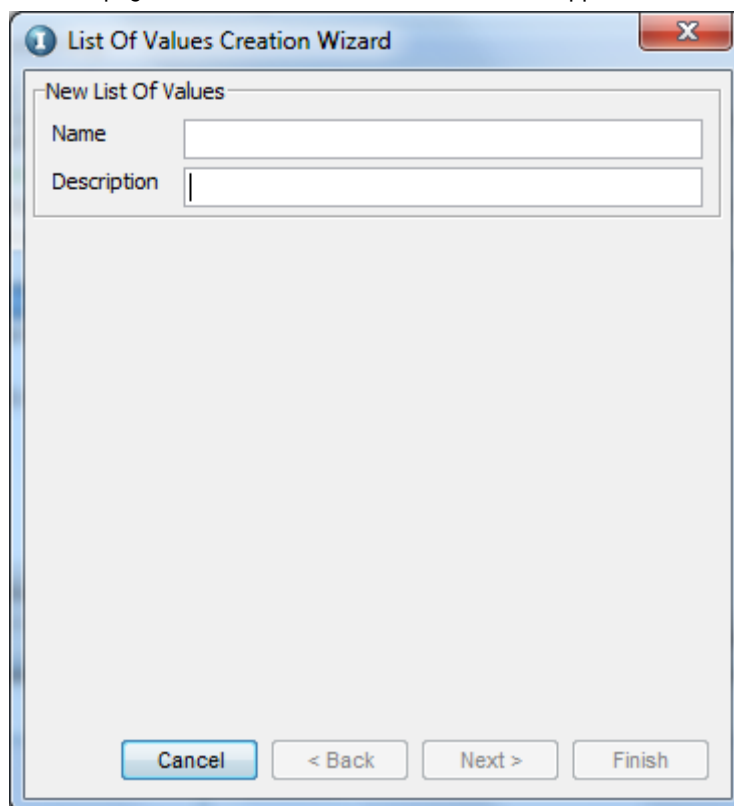
Create a List of Values to associate values with a parameter for an entity.

During Archive Definition, you supply values for these parameters from a List of Values. The list is populated internally by building an SQL query according to a List of Values definition.

To create a List of Values, perform the following steps:

1. Select the required Application Module and click **Insert > List of Values**.

The first page of the **List of Values Creation Wizard** appears as shown in the following image:

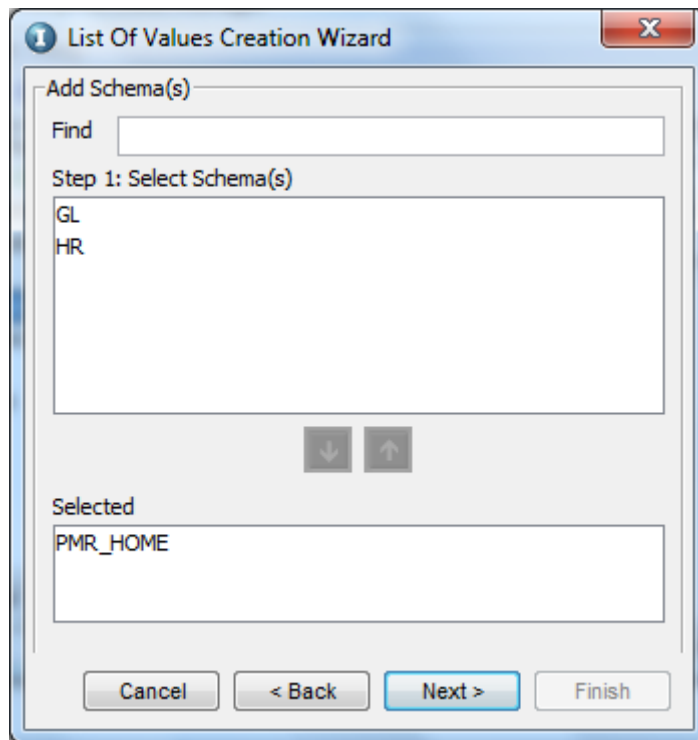


The image shows a dialog box titled "List Of Values Creation Wizard". The main content area is titled "New List Of Values" and contains two text input fields: "Name" and "Description". At the bottom of the dialog, there are four buttons: "Cancel", "< Back", "Next >", and "Finish".

2. Enter a name and description for the List of Values and click **Next**.

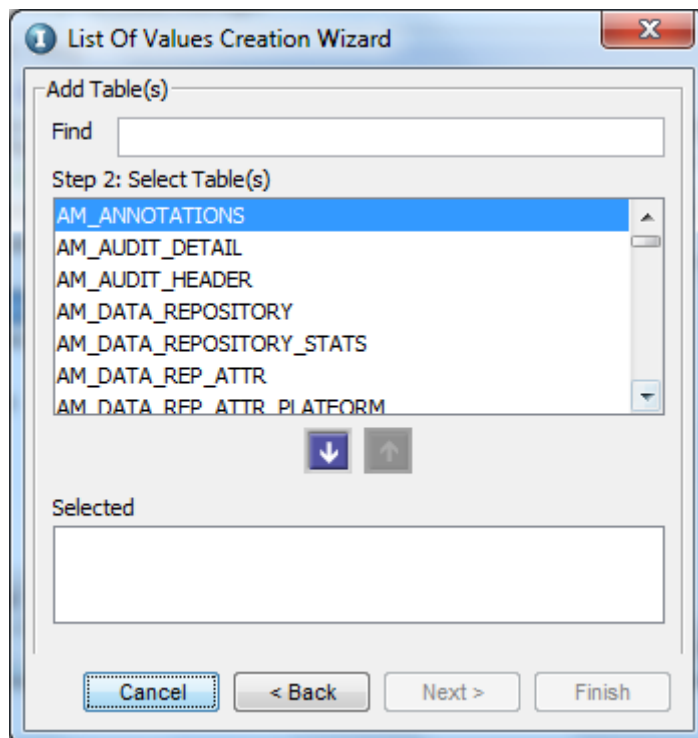


The following page appears:



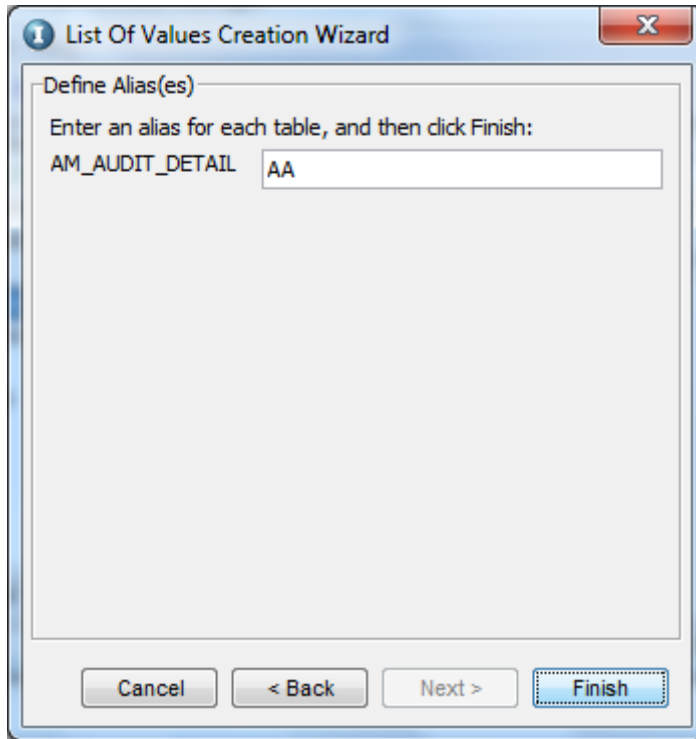
3. To add a schema to the **Selected** pane, select a schema from the **Select Schema(s)** pane and click the down arrow. To clear the selection, select the selected schema and click the up arrow (Step 2). When you are finished, click **Next**.

The following page appears:



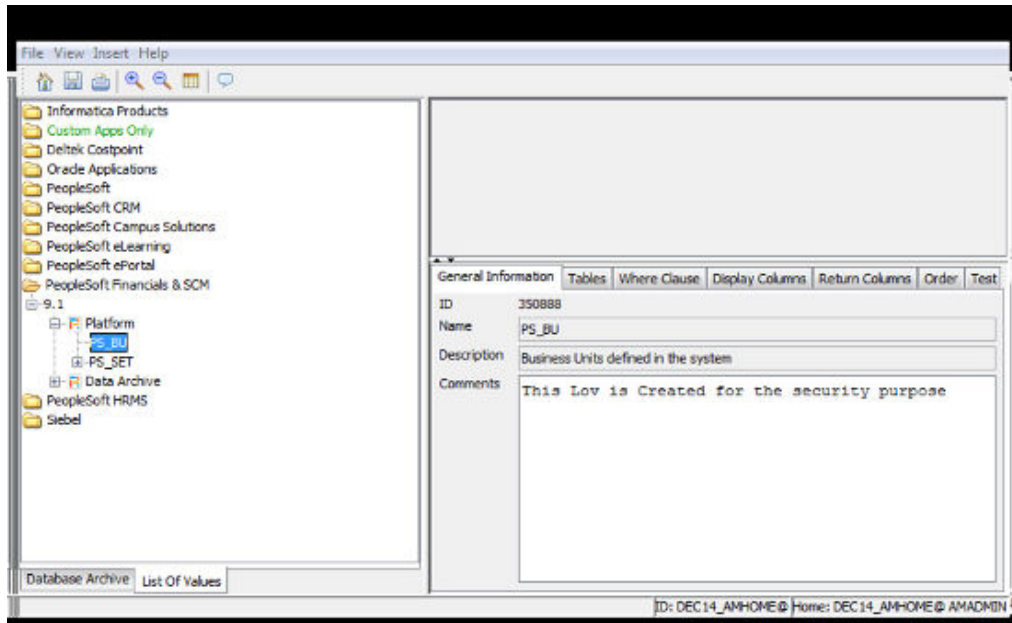
- To select Business Tables (Step 3) click the down arrow (or click the up arrow to clear the selection). Click **Next**.

The following page appears:



- Specify aliases for each of the selected tables (Step 4) and click **Finish**.

When you complete the wizard, the List of Values is created and appears in the **List of Values** tab of the Explorer pane. Each List of Values appears with the following tabs in the Details pane:



List of Values Detail Tab	Function
General Information	<p>Basic information about the List of Values such as:</p> <ul style="list-style-type: none"> <li>- Auto generated ID</li> <li>- Name (specified in <b>List of Values Creation Wizard</b>)</li> <li>- Description (specified in <b>List of Values Creation Wizard</b>)</li> <li>- Comments</li> </ul>
Tables	<p>Selected tables with the following information:</p> <ul style="list-style-type: none"> <li>- Order of Tables according to selection from <b>List of Values Creation Wizard</b>. To change the order, click the up or down arrows.</li> <li>- Table Name</li> <li>- Table Alias (as specified in <b>List of Values Creation Wizard</b>).</li> <li>- Check boxes to determine whether to include the table in the FROM clause of the SQL query.</li> </ul> <p>To add or delete tables, click <b>Add</b> or <b>Delete</b>.</p>
Where Clause	<p>The WHERE clause for filtering data.</p>
Display Columns	<p>Columns to select in the SQL query. Each column appears with the following information:</p> <ul style="list-style-type: none"> <li>- Order of selection. To change the order, click the up arrow and down arrow.</li> <li>- Expression to execute on the column. For example, TOCHAR(). Generally, you specify the column name here.</li> <li>- Data Type for the column.</li> <li>- Label / Alias for the column (column heading)</li> </ul> <p>To add or delete columns, click <b>Add</b> or <b>Delete</b>.</p>

List of Values Detail Tab	Function
Return Columns	Return value for the SQL query (not displayed in the List of Values).
Order	Columns for the ORDER BY clause for sorting data.
Test	The SQL query, built from the previously specified information. If values are not valid and the query does not comply with SQL syntax, error messages appear in this area.

- Click **Save** to save changes to the List of Values from the Details pane.

The Detail pane for a List of Values appears. The **Test** tab shows the SQL query that fetches values for this List of Values.

The selected tables appear in the Canvas pane.

## Creating a List of Value Variant

To create a list of value variant, select **Create List of Value Variant** from the menu of the respective List of Value.

## Updating Entity Information

You can update information for an entity from the **Details** pane.

To view details about an entity, click the entity from the **Explorer** pane. The details for the entity appear on the **Details** pane within the **General Information**, **Parameters**, and **Steps** tabs.

The following table describes the information in each tab:

Entity Detail Tab	Function
General Information	<p>The <b>General Information</b> tab lists the following pre-specified information:</p> <ul style="list-style-type: none"> <li>- Auto-generated ID</li> <li>- Entity name</li> <li>- Entity description</li> <li>- Driving table name</li> </ul>
Parameters	<p>The <b>Parameters</b> tab contains the archive definition parameters. You can set the following parameters:</p> <ul style="list-style-type: none"> <li>- Order of Parameters. Change the order of parameters by using the up and down arrows on the right.</li> <li>- Parameter Name. Enter a name for the parameter.</li> <li>- Parameter Title. Enter a title to display in the user interface.</li> <li>- Parameter Data Type. Specify the SQL data type for the parameter.</li> <li>- Required. Click the box if the entity specification step for an archive definition requires the parameter.</li> <li>- The List of Values. Specify a name of the previously created list of values for this parameter.</li> <li>- Security Group. Enforce constraints on a particular list of value.</li> </ul>
Steps	<p>The <b>Steps</b> tab lists steps that the job engine uses to run the archive job. Click <b>Insert Default Steps</b> to insert the standard set and the order of steps. You can configure the following columns on the <b>Steps</b> tab:</p> <ul style="list-style-type: none"> <li>- Order. To change the order of execution for SQL statements, use the up arrow and down arrow buttons on the right.</li> <li>- Action. Select an action from the menu.</li> <li>- Interim Table. Select an interim table.</li> <li>- Entity Table. If the action relates to operations on a business table, select a business table.</li> <li>- Procedure. If the action relates to operations on a procedure, enter a procedure.</li> <li>- Business Rule Order. This order matches the order specified for business rules for a particular interim table.</li> <li>- Enabled. Select boxes in this column to include the step in the archive job.</li> </ul>

To delete entities, right-click the entity and select **Delete Entity**.

## Copying Archived Entities to Customize

To customize an entity that you previously archived to the Data Vault or a database, create a copy of the entity and then add or remove tables.

1. Go to **View > Database Archive**.
2. Navigate to the entity.
3. Right-click the entity and select **Copy Entity**.
4. Right-click the same application module folder you copied the entity from and select **Paste Entity**.
5. Rename the copied entity.
6. Add or delete tables to the entity.

## Copying Standard Entity Metadata for Data Vault Search

After you retire an entity, you can use the prebuilt application-accelerator metadata to search and examine data in the entity through Data Vault search.

To search for retired entities in Data Vault and to keep the table relationships that existed at the time of retirement, you must copy the application module from a prebuilt application version to the custom application version folder. This triggers the Copy Entity from PFV job to run. When the job completes, assign the appropriate Data Vault access role to the entities to access these entities in Data Vault search.

**Note:** You can copy entities between versions in an application but not between applications.

1. Go to **View > Database Archive**.
2. Navigate to the application version folder.
3. Right-click the application version folder and select **Copy Entities from Application Version**.  
The **Copy Entities from Application Version** window appears.
4. Select the application version that contains the application module you want to copy.
5. Enter a descriptive prefix to identify the application module in Data Vault search.
6. Click **OK**.  
The Copy Entity from PFV job begins.
7. From the Data Archive interface, go to **Jobs > Monitor Jobs**.
8. After the Copy Entity from PFV job completes, expand the row and view the job log.  
The job log shows details such as mismatched tables between the accelerator and source database.
9. Assign Data Vault access roles to the entities.  
You can search and examine the data in these entities through Data Vault search.

## Adding and Deleting Interim Tables

To add interim tables, click **New Interim** from the menu of the respective entity.

To delete an interim table, click **Delete Interim** from the menu.

Enter the interim table name in uppercase letters.

## Inserting and Removing Business Tables

To add a business table to an entity, click **Insert Table** from the entity menu.

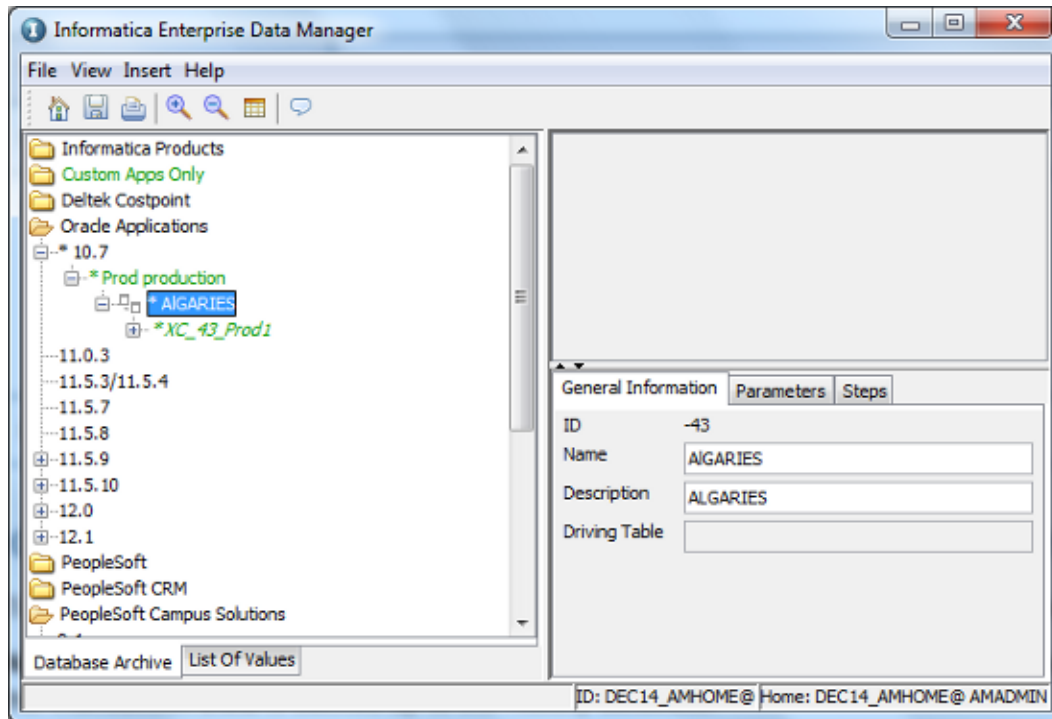
Select a schema and target tables.

To view a list of available tables that were mined for the respective application, select the **Show only tables that have been mined for the application** check box.

To disassociate a table from an entity, select **Remove Table** from the former's menu.

# Business Tables

Each business table that pertains to an entity is accompanied by the following information, as displayed in its Details pane.



Each tab is described in the following table:

Table Detail Tab	Function
General Information	Usually contains the table name.
Columns	Columns available in the table, displayed with the following information: <ul style="list-style-type: none"> <li>- Name</li> <li>- Data type</li> <li>- Check box to identify participation in the Primary key constraint</li> <li>- Check box to identify whether it is Nullable</li> <li>- Check box to identify whether it is a Translation Column</li> </ul>
Constraints	Available Integrity Constraints for the table.

## Specifying Translation Columns (Independent Archive)

Translation Columns, specified here are meant for Data Archive job executions containing Independent Archive.

**Note:** In the case of Independent Archive, constraints and data are exported. For Database archive, only the transaction data is backed up to the History database.

A column in a table is a candidate for translation when data must be extracted from a second table based on a referential integrity constraint.

There might be circumstances where data is extracted from the first and a third table based on similar foreign key constraints, linking the first with second, and second with third table. In such a case, the column in the second table is de-referenced to make way for displaying a column in the third table.

**Note:** Column Translation occurs when two tables participate in a JOIN. Columns are de-referenced to include constraints when more than two tables are involved in data extraction through a JOIN query.

You can perform de-referencing from the **Columns** tab of the Details pane for a table.

The following steps are involved in specifying a Translation Column:

1. Select the check box under **Translation** for that column.
  - Application Version: "Oracle Apps 11.5.10"
  - Application Module: "AP Payables"
  - Entity: "AP\_INV\_APRVL\_HIST\_ALL"
  - Table: "AP\_ENCUMBRANCE\_LINES\_ALL"

Name	Type	Private	Nullable	Translation
ENCUMBRANCE_TYPE_ID	NUMBER	False	False	True

2. When the column is selected for translation, the following information is displayed:

Table Column	Constraint	Where Clause	Parent Table Alias	Ref Table Alias	Display Column	De-Reference
ENCUMBRANCE_TYPE_ID	AP_ENCUMBRANCE_LINE_S_FK3 (GL_ENCUMBRANCE_TYPES)		A	B	ENCUMBRANCE_TYPE	False

3. If a third table is included in the JOIN, the **De-Reference** check box beside the column ("ENCUMBRANCE\_TYPE" in this example) must be checked and used to link it to a third table. The column for a **De-Reference** must also participate in a referential integrity constraint.



## Specifying Integrity Constraints

The following information is available from the Details pane when you select a table from the Explorer pane:

Constraint Detail Tab	Function
General Information	Displays the name and description for the constraint.
Columns	Displays columns in the selected table with the name, data type, and check box to indicate if it serves as a primary key for the table.
Constraints	<p>The constraint, with the following information:</p> <ul style="list-style-type: none"> <li>- Name: A unique name for the constraint.</li> <li>- Type: One of the values: CHECK, PRIMARY, REFERENTIAL and UNIQUE.</li> <li>- Constraint: Specify a condition for the constraint. For example, CHECK (NumParts &gt; 5).</li> <li>- Enabled: There is also an option to disable a constraint, where it will not participate in defining data integrity of the respective table.</li> </ul>

To view constraints on tables and specify new constraints, go to **View > Constraints**. The **Constraints** tab in the Explorer pane appears with the schema and its related tables in a hierarchy.

You can also create constraints between tables by using a drag-drop mechanism for target columns from the Canvas pane.

## Interim Tables

Adding an interim table to an entity is discussed in [“Adding and Deleting Interim Tables” on page 30](#).

You can change the following information for the metadata as described in the following table:

Interim Detail Tab	Function
General Information	A generated name and pre-specified description of the interim table.
Tables	<p>A list of all tables involved in the Data Archive job executions. You can add or delete tables using <b>Add</b> and <b>Delete</b>. The following information is specified for each table involved:</p> <ul style="list-style-type: none"> <li>- Order: INSERT, UPDATE, and DELETE statements are executed on the business tables in the order specified. You can change the order using the up arrow and down arrow.</li> <li>- Table Name.</li> <li>- Insert Statement: Generally executed while copying ROWS into Data Destination during a job execution.</li> <li>- Delete Statement: Generally executed while purging ROWS from the Data Source in a job execution.</li> <li>- Update Statements, if any.</li> </ul>

Interim Detail Tab	Function
Default Columns	<p>You can specify default columns for the interim tables, each accompanied with the representation of a column from a business table. Use the following values:</p> <ul style="list-style-type: none"> <li>- Order: Order of Columns in the interim table. To promote or demote a column, use the up arrow and down arrow.</li> <li>- Name: Column name.</li> <li>- Type: Data type for records to be inserted. Generally, the data type of the business table column as specified in the SELECT clause.</li> <li>- Length: Maximum number of characters allowed, according to the specified data type.</li> <li>- PK: Use the check boxes to assign a column as a Primary Key of the interim table.</li> <li>- Select Clause: Column names from one or more business tables, whose values are fetched to populate the respective default column for the interim table.</li> </ul> <p><b>Note:</b> When an interim table is first created the columns stats_date, org_id and purgeable_flag are created by default. While stats_date and org_id can be assigned business table columns in the SELECT clause, purgeable_flag consists of a flag to determine whether the record is archivable.</p> <p>From Clause: FROM clause in the SQL query, specifying target columns to be selected while building the interim table.</p> <p>Where Clause: WHERE clause to filter ROWS for columns selected while building the interim table.</p>
Business Rules	<p>You can specify business rules, which occupy one column each in the interim table (as a flag which indicates its validation status).</p> <p><b>Note:</b> The value inserted in the purgeable_flag column is a binary summation of the values in the columns complying to each of the business rules.</p> <p>Define business rules with the following information:</p> <ul style="list-style-type: none"> <li>- Order: Order of columns in the interim table.</li> <li>- Name: The column name in interim table which stores the validation flag for the business rule.</li> <li>- Description: A description of the business rule.</li> <li>- Business Rule Order: Order of execution of UPDATE statements on the interim table intended for applying the business rule to the data.</li> <li>- Type: Type of clause used to construct the business rule. Select from FULL, INLINE, LOOP, SET, and WHERE clauses.</li> <li>- Purge Qualification: Statement for the clause type.</li> <li>- Exclude from Evaluation: When selected, the corresponding business rule is excluded from the binary summation that determines the value in the purgeable_flag column.</li> <li>- Condition: Conditional statement that identifies the business rule.</li> </ul>

Interim Detail Tab	Function
Indexes	To specify an index, click <b>Add</b> . To delete an index, click <b>Delete</b> . To change the order of execution of indexes, use the up arrow and the down arrow to the right of the Details pane. Generally, default indexes are generated by clicking on <b>Insert Default Index</b> , at the bottom of the Details pane.
Report Statements	Specify statements for generating Detail reports. To associate a report statement with an interim table, specify the following information: <ul style="list-style-type: none"> <li>- Order: Order of display for columns.</li> <li>- Statement Label: A list of predefined (Reporting) statements.</li> <li>- Literals (comma separated): Specify labels for display columns.</li> <li>- Group: A column can be part of a Primary or Secondary group. This specifies the placement of columns in the Detail report of an Archive Definition.</li> </ul> The following information is displayed on the right, when you select a statement. <ul style="list-style-type: none"> <li>- Parameters: You can specify parameters in the same order as they occur in the SQL query.</li> <li>- Statement: Displays the SQL query or INTERIM_VALUE, as specified during statement creation.</li> </ul>

## Entities

An entity is a group of business tables that might or might not be related, and an interim table. An entity with related tables can also contain information on table relationships, a master table, and reference tables.

You create entities to perform the following tasks in Data Archive:

- Archive data to a database.
- Archive data to the Data Vault.
- Retire an application to the Data Vault.
- Use the Data Discovery portal in Data Archive.
- Create and run data visualization reports.
- Partition the source database.
- Archive attachments.

### Before You Create an Entity

Before you create an entity, understand and verify the table relationships of the tables you want to archive.

When you create an entity, the entity creation process creates the data model based on the constraints you enable. If the constraints are not correctly enabled, then the data in the entity might not be archived correctly. For example, you archive an entity with a table that contains multiple paths to the driving table. When you archive the entity, you might encounter the following issues:

- The archive job archives records more than once.
- The Delete from Source step fails.

To avoid these issues, verify that the constraints for each table you plan to include in the entity are correct. If you have a child table related to more than one parent table, you must define the constraint conditions to

specify the rows that belong to that relationship. After you verify that the constraints are correct, create the entity.

## Verifying Constraints Before You Create an Entity

Before you create an entity, verify that the constraints for each table in the entity are correct.

1. Click **View > Constraints**.
2. On the Explorer pane, expand the application node, application version, and application module to view the tables.
3. On the Details pane, select the **Constraints** tab.
4. Verify if the constraints are correct.
5. If necessary, disable or modify a constraint.

## Multiple Entity Generation

You can use wizards to generate multiple entities at a time. Generate multiple entities to reduce the amount of time required to create entities manually. Manual creation of entities can be time-consuming. You can generate entities for live archiving or for retirement.

The method that you select to generate entities depends on the archive scenario. Use one of the following methods to generate multiple entities:

### Automatic entity generation

Use the **Multiple Entity Creation** wizard to generate entities for live archiving and for data discovery. The wizard uses the constraints in the ILM repository to generate the entities. You can use the entities for archive projects and for Data Discovery portal searches.

### Retirement entity generation

Use the **Generate Retirement Entity** wizard to generate entities for retirement. The wizard uses the imported schema and table metadata to generate the entities. You can use the entities for retirement projects. If you want to discover data from the Data Discovery portal, you must generate entities based on constraints.

## Entity Naming Convention

Data Archive names an entity based on the name of the driving table in the entity and the number of entities in the application version.

When you create an entity, Data Archive assigns a unique name to the entity. The entity name contains two parts. The first part is the name of the driving table for that entity. The second part is a number. The number indicates the entity count at the time the entity was created.

For example, you have application modules A, B, and C in an application version. The application modules have the following entity history:

- Module A has one entity named `Orders_001`.
- Module B has one entity named `Customers_003`. Module B had an entity named `CustomerAddress_002` that you deleted.
- Module C does not have entities.

You create an entity in Module C so you can archive a group of tables. You select the table `Supplies` to include in the entity. You specify the `Supplies` table to be the driving table in the entity. Data Archive assigns the name `Supplies_004` to the entity. The `Supplies` part of the entity name indicates the driving table in the entity. The `004` part of the entity name indicates that this is the fourth entity in the application version.

## Verifying Constraints Before You Create an Entity

Before you create an entity, verify that the constraints for each table in the entity are correct.

1. Click **View > Constraints**.
2. On the Explorer pane, expand the application node, application version, and application module to view the tables.
3. On the Details pane, select the **Constraints** tab.
4. Verify if the constraints are correct.
5. If necessary, disable or modify a constraint.

## Automatic Entity Generation

Use the **Multiple Entity Creation** wizard to generate entities for live archiving and for data discovery. The wizard uses the constraints in the ILM repository to generate the entities. You can use the entities for archive projects and for Data Discovery portal searches. You might want to generate entities for custom applications.

Entity generation for custom applications includes the following high-level steps:

1. Import metadata from the source database.
2. Discover table relationships.
3. Review and import the table relationships that you want to create constraints from.
4. Use the **Multiple Entity Creation** wizard to create entities based on the constraints that you imported from the table relationship discovery process.

## Generating Entities Automatically

Use the **Multiple Entity Creation** wizard to generate entities. You might want to generate entities after you use one of the table relationship discovery methods to create constraints.

The ILM repository must contain constraints for the imported source metadata.

1. Click **View > Constraints**.
2. In the **Explorer** pane, expand the application node to view the application versions.
3. Right-click the application version that you want to create entities in, and select **Auto-create new entity(s)**.  
The **Multiple Entity Creation** wizard appears.
4. Select the schemas, and click the down arrow.
5. Click **Next**.
6. Select the table types, and click the down arrow.
7. Click **Next**.
8. Select the tables and click the down arrow.  
**Note:** If the source database is Oracle, you can archive tables that contain up to 998 columns.
9. Click **Next**.
10. Select the child tables that you want to include, and click the down arrow.
11. Click **Next**.
12. Select the table types for the child tables that you selected, and click the down arrow.

13. Click **Finish**.

The **Entity Creation Process** page displays the status of the creation process. A message appears that prompts you to save the data.

## Entities for Informix Databases

When you create an archive entity for a Informix source database, adhere to the following guidelines:

- The interim table name must be in uppercase characters.
- In the default columns of the entity interim table, enter a double quotation mark in the SELECT clause field for the STATS\_DATE and ORG\_ID columns. Do not leave these fields blank.
- In the table insert and delete statements, the interim table name and column names must be surrounded by double quotation marks. The entity table name and column names should not be in double quotation marks.
- Delete statements should not contain a table alias name with the primary key. For example: Change (A. "integer") to (integer).

## Retirement Entity Generation

Use the **Generate Retirement Entity** wizard to automatically generate entities for retirement projects. When you use the wizard, you specify parameters for the Retirement Auto Entity Creation job. The job simultaneously creates entities for all tables in the application that you want to retire. The job replaces the time-intensive task of creating entities one at a time.

The entities do not include constraints and are only available for data extract. If you want to use the Data Discovery portal to search the retired data, you must generate entities based on constraints. To generate entities based on constraints, use the **Multiple Entity Creation** wizard.

### Retirement Auto Entity Creation Job

The Retirement Auto Entity Creation job creates entities for all tables in the application that you want to retire. You use the entities in a retirement project. The job uses the imported schema and table metadata in the ILM repository and the parameters that you configure in the **Generate Retirement Entity** wizard to automatically generate entities.

The job uses the following naming convention to create entities:

```
<prefix>_<entity number>_<suffix>
```

The job generates the entity number in sequential order. For example, if you use `Finance` for the prefix, the job generates `Finance_1` for the first entity, `Finance_2` as the second entity, and so on. The job uses the meta schema ID and the meta table ID to determine the order in which it adds tables to the entity.

The number of entities the job creates depends on the parameters that you provide. You specify the number of tables in each entity, the prefix and suffix for the entity name, and whether you want to add tables from more than one schema.

The best practice is to create one entity for every 200 tables in the application. The job can create entities with up to 999 tables in each entity. By default, if the retirement application contains multiple schemas, the job creates an entity with tables from more than one schema. An entity can have tables from multiple schemas if the maximum number of tables in an entity exceeds the amount of tables in the first schema. For example, the application has two schemas with 150 tables in each schema. You configure the job to create 200 tables in each entity. The job uses 150 tables from the first schema and then 50 tables from the second schema to create the first entity.

If you do not want to combine tables from multiple schemas in one entity, you can disable this feature. In the example above, the job creates two entities. The first entity includes 150 tables from the first schema. The second entity includes 150 tables from the second schema.

## Retirement Auto Entity Creation Job Parameters

Enter the parameters when you run the **Generate Retirement Entity** wizard to automatically generate entities for application retirement. The wizard runs the Retirement Auto Entity Creation job.

The Retirement Auto Entity Creation job includes the following parameters:

### Number of Tables in each Entity

Required. Maximum number of tables to include in each entity. You can create up to 999 tables in each entity. The recommended number of tables in each entity is 200.

Use up to three digits to specify the number of tables to create. For example, enter 1, 01, or 001 to specify one table.

### Prefix for Entity

Required. Text to append before the entity name. The prefix helps identify what application the entity corresponds to. The full entity name, including the prefix and suffix, cannot exceed 45 characters.

### Suffix for Entity

Optional. Text to append after the entity name. The full entity name, including the prefix and suffix, cannot exceed 45 characters.

### Group Tables beyond the Schemas

Optional. Determines whether the entity can include tables from multiple schemas. If enabled, the wizard creates the maximum number of tables that you specify from across schemas. If disabled, the wizard creates the entity with the maximum number of tables within one schema. Default is enabled.

## Retirement Auto Entity Creation for SAP Application Retirement

The Retirement Auto Entity Creation job runs steps that are specific to SAP applications. When you run the Import Metadata job for SAP sources, the job adds a table type to imported metadata tables in the ILM repository. The Retirement Entity Creation job uses the table types to determine the tables the job adds to entities and the configuration of the default entity steps.

By default, the job inserts default entity steps. However, only the Insert into Archive Tables step is enabled. The Insert into Archive Tables step determines from where the retirement job reads data for tables in the entity. By default, the step reads data directly from the database. SAP applications includes special tables that the retirement job cannot read directly from the database. The job cannot read data directly from the database for pool and cluster tables and any tables that have archived data in ADK files. The tables require a additional procedure for the Insert into Archive Tables step.

In addition, the retirement job may need to read data from the same table from multiples sources. SAP tables can store data in both the database and in archived ADK files in an external file system. For a table that has data in the database and in ADK files, the table is included in two separate entities. Separate entities are required because entities can only use one source to read data from. One entity reads data from the database. The other entity reads data from the ADK files in the external file system. Although the retirement job uses two entities to read data for the same table, the table data is consolidated when the Data Vault Loader job loads the data in to the Data Vault. You only see one table in the Data Vault.

The job uses the following naming convention to create the entities for SAP applications:

```
<Prefix>_<Entity Number>_<Suffix>_<SAP-Specific Suffix>
```

The following table describes the parameters in the naming convention:

Parameter	Description
Prefix	User-defined entry in the job parameters.
Entity Number	Job-defined entry.
Suffix	User-defined entry in the job parameters.
SAP-Specific Suffix	Job-defined entry. The job appends the following SAP-specific suffixes: <ul style="list-style-type: none"> <li>- <code>_SAP</code>. For entities that include pool and cluster tables.</li> <li>- <code>_SAP_ADK</code>. For entities that include tables with archived data.</li> </ul>

## Table Types for SAP Application Retirement

The Retirement Entity Creation job adds tables to entities based table types in the ILM repository. The Import Metadata job adds SAP-specific table types when you import metadata from SAP applications.

The Retirement Entity Creation job uses the following table types to determine how to create entities:

### Converted transparent table with archived data

Converted transparent tables are logical tables that are not stored in the database. This table type is specifically for transparent HR PCL1-PCL5 cluster tables and the STXL text table that store data in ADK files, in addition to data in the database. The job inserts tables with this table type into a separate entity and inserts the `ArchiveSAPArchivedTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the external storage that stores the ADK files.

Note that tables with the converted transparent table types are already included in the pre-packaged SAP entities. The Retirement Auto Entity Creation job does not add those table types to entities because the tables are already included in pre-packaged SAP entities. However, the job needs to add tables to new entities if the tables store data in ADK files.

### Pool cluster

Pool and cluster tables are logical tables that store data in the database in an SAP proprietary format. The job adds the pool and cluster tables to entities and inserts the `ArchiveSAPLogicalTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the SAP application layer.

### Pool cluster with archived data

Pool and cluster tables with archived data are pool and cluster tables that store data in ADK files, in addition to data in the database. The job inserts tables with this table type into a separate entity and inserts the `ArchiveSAPArchivedTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the external storage that stores the ADK files.

### Transparent table

Transparent tables are physical tables that are stored in the database. The job adds transparent tables to entities and inserts the standard Insert into Table entity step. The step reads data directly from the database.



### Transparent table with archived data

Transparent tables with archived data are transparent tables that store data in ADK files, in addition to data in the database. The job inserts tables with this table type into a separate entity and inserts the `ArchiveSAPArchivedTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the external storage that stores the ADK files.

## Generating Retirement Entities

Use the **Generate Retirement Entity** wizard to automatically generate entities for application retirement.

1. Click **View > Constraints** or **View > Data Vault**.
2. Navigate to the application version.
3. Right-click and select the application version and choose **Generate Retirement Entities**.  
The **Generate Retirement Entity** wizard appears.
4. Enter the retirement entity generation parameters.
5. Click **OK**.

You receive a message that the wizard started the Retirement Entity Creation job in the background. You can use the job number in the message to monitor the job status in Data Archive.

After the job completes, exit and log back in to view the generated entities.

## Troubleshooting Entities

### After I create an entity, a message informs me that there are multiple paths for tables in the entity.

First, verify the constraints for each table listed in the message. If the constraints are incorrect, delete the entity. Then, disable or modify the constraints. Then, create the entity.

To verify the constraints, perform the following steps:

1. Click **View > Constraints**.
2. On the **Explorer** pane, select a table listed in the message.
3. On the **Details** pane, select the **Constraints** tab.
4. Verify if the constraints are correct.
5. If the constraints are correct, you can ignore the message.
6. If the constraints are incorrect, you must perform the following steps:
  - a. Delete the entity.
  - b. Disable or modify the constraints for the tables you want in the entity.
  - c. Create the entity.
7. Repeat steps 4 to 6 for each table listed in the message.

To delete the entity, perform the following steps:

1. Click **View > Database Archive**.
2. Right-click the entity.
3. Select **Delete Entity**.
4. Click **Delete** to confirm the delete action.

To disable a constraint, perform the following steps:

1. Click **View > Constraints**.
2. On the **Explorer** pane, select a table listed in the message.
3. On the **Details** pane, select the **Constraints** tab.
4. Click the **Enabled** check box.

**Note:** If the constraint you are deleting is a self-referential physical constraint, note that this constraint still exists in the source database. After you archive the entity, you must delete this data from the source database.

5. Click **View > Refresh**.

To modify a constraint, perform the following steps:

1. Click **View > Constraints**.
2. On the **Explorer** pane, select a table listed in the message.
3. On the **Details** pane, select the **Constraints** tab.
4. Click **Constraint Condition**.
5. In the text box that appears, specify the rows that apply to the constraint.
6. Click **View > Refresh**.

To create the entity, perform the following steps:

1. Click **View > Refresh**.
2. Click **View > Constraints**.
3. On the **Explorer** pane, select the driving table.
4. Right-click the driving table and select **New Entity**.
5. Specify the child tables and table types in the **Multiple Entity Creation Wizard**.

## Exporting Accelerators

Enterprise Data Manager allows you to export metadata from existing ERP applications, commonly referred to as Accelerators. This is achieved through one of the **File > Export** submenu items.

When you export accelerators, run the export task as a background job if the volume of metadata is high to avoid memory-related issues.

The following metadata objects can be exported from Archive for the selected Application Version:

## Data Model Metadata

The following table describes Data Model metadata.

Type	Description
Both	Exports Custom and Standard Data Model metadata.
Custom Only	Export Data Model Metadata for user-defined objects.
Standard Only	Imports Data Model Metadata for predefined standard objects in the ERP Application.

## Data Archive Metadata (Custom/Standard Entities)

The following table describes the Data Archive Metadata Custom and Standard entities.

Type	Description
Both	Export Data Archive specific data for user-defined (Custom) and predefined (Standard) objects.
Custom Only	Exports Data Archive specific objects that were created by a user.
Standard Only	Exports Standard objects specific to Data Archive.

## Component (Standard Data)

The following table describe the Component Only type.

Type	Description
Component Only	Exports Data Archive specific data for a selected Component (Entity / Application).

# Exporting a Custom Product Family Version

When you export a product family version, export both the data model metadata and the application metadata to a location of your choice. You can export a custom product family version (PFV) to an XML file. You might want to export a custom product family version so that you can import the product family version to a different environment.

When you export a product family version, you export both the data model metadata and the application metadata. When you export the data model metadata, you choose to submit the export job as a background job or continue the export through the Enterprise Data Manager.

You also have the option of selecting **Enhanced Export**. Select the enhanced export option if you need to import the product family version to an existing environment. If you do not choose enhanced export, and you try to import the exported data model metadata to an environment where a product family version exists, the data becomes corrupted.

To export a custom PFV, perform the following steps:

1. In the Enterprise Data Manager, select the PFV that you want to export and click **File > Export > Data Model Metadata**.

The **Export Database Metadata Options** window appears.

2. Select **Both**.
3. Click **OK**.

The **Export Database Metadata Options** window appears.

4. Run the export job as a background job or through the Enterprise Data Manager.
5. If you plan to import the PFV to an existing environment, select **Enhanced Export**.
6. Click **OK**.
7. Select a location to export the XML file to and click **Save**.

A popup appears to indicate that the export has completed successfully.

After you export a custom product family version, you can import it into a different environment. The process is the same as importing an accelerator. When you import the product family version, you again have the option of choosing enhanced import. You also have the option to update the metadata of the entities in the environment you are importing to.

## Exporting an Entity

You can export a single entity using the enhanced export option. You might want to export a new entity or changes to an existing entity so that you can import the entity to a different environment, without importing the entire product family version. When you import, ensure the metadata of the exported entity tables are present in environment that you plan to import to.

When you export an entity, you export both the data model metadata and the application metadata. To export at the entity level, select **Export Application Metadata** and select the **Enhanced Export** check box. When you export the metadata, you choose to submit the export job as a background job or continue the export through the Enterprise Data Manager.

After you export an entity, you can import it into a different environment. When you import the entity, you again have the option of choosing enhanced import.

## Importing Accelerators

After you export to a set of XML files, you can import them to a different environment. While you run the import job, you specify the job execution mode, the location of the XML metadata files, and the job parameters. By default, when you import an accelerator Data Archive uses the enhanced import option. You can change the import method using the properties provided in the configuration file.

You can import accelerators as a background job or in the foreground. The background job runs on the Data Archive server instead of on the Enterprise Data Manager client machine. You might want to run the job on the Data Archive server to improve the job performance and to reduce the amount of resources consumed on the Enterprise Data Manager client machine. To import accelerators as a background job, you must start Enterprise Data Manager from Data Archive. Standalone Enterprise Data Manager does not support importing as a background job.

When you configure the import job, you can select whether to continue processing on error. For custom PFV's, can also select the enhanced import option and choose whether to update entities in the custom PFV.

If you selected enhanced export, select enhanced import when you import the custom PFV. Selecting enhanced import allows you to import to the environment without corrupting data. In addition to enhanced import, you can choose the option to update the entities in the custom PFV, if the entity already exists in the target environment.

When you use enhanced import and choose to update the entities, Data Archive first verifies that each entity in the custom PFV has not been implemented in an archive or retirement project, or in a Data Discovery operation. If an entity has been implemented in a project or a Data Discovery operation, you receive a message that the entity will not be updated. You can continue with the import of remaining entities that have not been implemented in a project or a Data Discovery operation. If the entities are not implemented, they are deleted and reinserted.

If you do not select the **Update Existing Entities** check box, then the job updates the entities based on the collision check.

By default, the export job uses enhanced import. To change the behavior, configure the following properties in the `conf.properties` file:

- `#informia.enable.enhanced.import=Y`  
Enables enhanced import for entities and product family versions. Default is Y. Valid values are Y and N.
- `#informia.enable.traditional.import=N`  
Enables traditional import. Default is N. Valid values are Y and N.

To perform any import operations, a user must have the Import Metadata role assigned in Data Archive. The Administrator user is assigned this role by default, but other users must be assigned this role. For more information on the Import Metadata role, see the chapter "Security" in the *Data Archive Administrator Guide*.

## Importing an Entity

After you export an entity to an XML file, you can import that entity to a different environment. When you import an entity, you specify the job execution mode, the location of the XML metadata files, and the job parameters.

You can import entities as a background job or in the foreground. The background job runs on the Data Archive server instead of on the Enterprise Data Manager client machine. You can select whether or not to continue processing on error. You can also select the enhanced import option and choose whether or not to update the existing entity.

If you selected enhanced export when you exported the entity, select enhanced import when you import the entity. Selecting enhanced import allows you to import the entity to the environment without corrupting data. In addition to importing the entity, you can choose the option to update the entity if the entity already exists in the target environment. When you update the existing entity, the import job updates metadata changes to that entity. The job deletes all of the entity steps and tables and then reinserts them. If you do not choose to update an existing entity, metadata changes to the entity are ignored.

When you use enhanced import and choose to update the existing entity, Data Archive first verifies that the entity has not been implemented in an archive or retirement project, or in a Data Discovery operation. If the entity has been implemented in a project or a Data Discovery operation, you receive a message that the entity will not be updated.

By default, the import job uses enhanced import. To change the behavior, configure the following properties in the `conf.properties` file:

- `#informia.enable.enhanced.import=Y`  
Enables enhanced import for entities and custom product family versions. Default is Y. Valid values are Y and N.  
When this property is set to Y, you are unable to disable the enhanced import option when you import the entity.
- `#informia.enable.traditional.import=N`  
Enables traditional import for entities and custom product family versions. Default is N. Valid values are Y and N.  
When these properties are set to N in the `conf.properties` file, you are unable to select the corresponding options when you import the entity.

To perform import operations, a user must have the Import Metadata role assigned in Data Archive. The Administrator user is assigned this role by default, but other users must be assigned this role. For more information on the Import Metadata role, see the chapter "Security" in the *Data Archive Administrator Guide*.

## Import Job Parameters

Enter the job parameters when you import accelerators, custom product family versions, or entities.

The import job includes the following parameters:

### **Commit after Each File**

Determines when the database transaction is committed if you import multiple files.

Enable to commit after each file is imported. The commit occurs if the file is imported successfully and is independent of other files. For example, if you import ten files and there was an error with one of the files, then nine files are imported.

Disable to commit after all of the files are imported. The commit occurs if all of the files are imported successfully. If an error occurs with one of the files, no files are committed. For example, if you import ten files and there was an error with one of the files, then no files are imported.

**Note:** You cannot select this option if you use the enhanced import option.

### **Continue Processing on Error**

Determines whether the system continues to process multiple files if there is an error with one of the files.

Enable to process files subsequent to the file that has the error. For example, you import ten files. An error occurred for the sixth file. The job ignores the sixth file and continues with the rest of the files.

Disable to stop processing the files subsequent to the file that has the error. For example, you import ten files. An error occurred for the sixth file. The job does not process the rest of the files.

### **Enhanced Import**

Enables enhanced import of entities and custom PFV's. If you used enhanced export to export the custom PFV or entity, select this option to enable enhanced import. Use enhanced import when you import to an existing environment that contains any PFV's/entities. You do not need to use enhanced export and import if you plan to import the PFV or entity to a new environment without any existing PFV's.

### Update Existing Entities

Imports the existing entities in the custom PFV or entity. The import job updates any metadata changes to all of the entities in the custom PFV or the single entity you are importing. If entity is not implemented in any project definition, the job deletes all of the entity steps and tables in each entity and then reinserts them. If the entity is implemented in a project definition, the job ignores the entity.

## Importing as a Background Job

You can import accelerators, custom PFV's, and entities as a background job. The background job runs on the Data Archive server instead of on the Enterprise Data Manager client machine.

To import accelerators, custom PFV's, or entities as a background job, you must start the Enterprise Data Manager from Data Archive. The standalone version of the Enterprise Data Manager does not support offline import.

You might want to run the job on the Data Archive server to improve the job performance and to reduce the amount of resources consumed on the Enterprise Data Manager client machine. Run the job on the Data Archive server to avoid memory-related issues if you need to import a large volume of metadata.

**Note:** You can import XML files only when the export and import environments are on same version of Data Archive. If you used enhanced export to export an XML file through a previous version of Data Archive, you cannot import the same XML file in version 6.4.4. Only XML files exported through version 6.4.4 can be imported in version 6.4.4.

If you used traditional export to export the metadata files, you cannot import them using the enhanced import option. Use the same type of import as export.

1. Click **File > Import > Accelerators**

The **Import Metadata Options** screen appears.

2. Choose **Submit Import Metadata as a Background Job**.

3. Select the machine and location of the .xml metadata files.

- **Location on Server.** Select if the .xml metadata files exist on the Data Archive server. Enter the full path to the files. For example, `/home/oracle/ILM/export`
- **Location on Client.** Select if the .xml metadata files exist on the Enterprise Data Manager client machine. Click **Browse** to navigate to the directory.

**Note:** If you specify the directory that you exported to during the export job, the import job fails because the export job saves the exported metadata in subdirectories within the directory that you specified during export. The subdirectories have a timestamp as the name. When you select the directory to import, give the full file path including the subdirectory.

4. Configure the job parameters. If you used the enhanced export option to export the data, select **Enhanced Import**.

5. Click **OK**.

The system immediately runs the job in the background and provides a message that the job was submitted. The message includes the job ID. You can monitor the job status in Data Archive.

## Importing through the Enterprise Data Manager

You can import accelerators, custom PFV's, and entities through the Enterprise Data Manager. The job runs on the Enterprise Data Manager client machine.

**Note:** You can import XML files only when the export and import environments are on same version of Data Archive. If you used enhanced export to export an XML file through a previous version of Data Archive, you

cannot import the same XML file in version 6.4.4. Only XML files exported through version 6.4.4 can be imported in version 6.4.4.

1. Click **File > Import > Accelerators**.

The **Import Metadata Options** screen appears.

2. Choose **Continue Import through EDM**.
3. Configure the job parameters. If you used the enhanced export option to export the data, select **Enhanced Import**.
4. Click **OK**.

The Windows Explorer dialog box appears.

5. Navigate to the directory that contains the metadata files that you want to import.

**Note:** If you specify the directory that you exported to during the export job, the import job fails because the export job saves the exported metadata in subdirectories within the directory that you specified during export. The subdirectories have a timestamp as the name. When you select the directory to import, specify the full file path including the subdirectory.

6. Click **OK**.

## Virtual Views

A virtual view is a type of table that uses SQL logic to convert the format of retired data. You can create virtual views for any task that you can specify in the SQL.

Create virtual views to store retired data in a different format from data on the production database. For example, the date-of-birth values in a table are in integer format such as 01021991. When you retire this table to the Data Vault, the date-of-birth values remain in the same integer format. You can create a virtual view table to convert these date values to a more readable, alphanumeric format, such as 01-Feb-1991, on the retirement database.

To display the converted data, Data Archive uses the SQL created in the virtual view instead of the dynamically generated SQL used during an archive or a retirement job. You can create virtual views for data that you plan to retire. You can also create virtual views when you cannot create views on a read-only source database.

Create a virtual view to accomplish the following goals:

- Hide the complexity of the underlying data to create a simpler user experience.
- Save space and cost because the table is virtual.
- Simplify the process of capturing metadata from a retirement database.

## Virtual Views Taskflow

Add virtual views to the entities you want to retire.

1. Identify data that you want to retire.
2. Create virtual views of source databases in the ILM repository.
3. Add virtual views to entities.
4. Include entities in retirement projects.



## Creating a Virtual View

To create a virtual view, use the **Virtual View Wizard**.

1. Go to **View > Constraints**.
2. Navigate to the application version.
3. Right-click the application version and select **Create Virtual View**.  
The **Virtual View Wizard** appears.
4. Enter a name and a description for the virtual view that you want to create.
5. Click **Next**.  
A list of the available schemas for the home database appears.
6. Select the schemas for the virtual view, and click the down arrow.
7. Click **Next**.
8. Click **Add** to insert a blank line. Enter a name for the column, select the data type, and specify the length.
9. Click **Next**.  
The **Virtual View's SQL Text** page appears.
10. Enter the SQL code that the ILM Engine uses to archive the virtual view. Prefix the SQL code with the schema name so that the row count report generates correctly.  
You must provide the correct number of columns in the SQL code for the job to run successfully.
11. Click **Finish**.  
The virtual view appears under the application module of the application version that you selected.

## Enabling Search Data Vault

Use keywords to search for records across entities and applications in Data Vault.

To enable Search Data Vault, you must create a search index for each table you want to include in the search. Each search index contains a list of columns you specify in Enterprise Data Manager. When you search Data Vault with a keyword, the search engine looks for the keyword in indexed columns. If the keyword is in an indexed column, the corresponding record appears in the search results. If the keyword is not in an indexed column, the record does not appear in the results even if the record contains the keyword in another column.

### Selecting Columns for the Search Index

Specify columns to include in the search index. Optionally, specify the columns for the record header in the search results.

1. Select an application version and a table.
2. Click **View > Constraints**.
3. Click the **Columns** tab.  
A list of all the columns in the table appears.
4. Enable the **Index for Search** field for each column you want to add to the search index.
5. Click the **Constraints** tab.
6. Click the add icon to add a constraint.

A row appears with the name `NEW CONSTRAINT`.

7. Double-click the **Name** field and enter a name.
8. Optionally, to specify a column name as the record heading on the results page, click the **Type** field and select **Search Header**.

If you do not specify a column for the record heading, Data Archive uses the column name of the primary key. In the absence of a primary key, Data Archive uses the row ID for the record heading.

9. Enable the **Enabled** field.
10. Click the add icon in between the Child Table Columns and Parent Table Columns sections of the **Constraints** tab.

A row appears in the Child Table Columns section.

11. Click the **Type** field and select **Column**.
12. Click the **Name** field and select a column.
13. To add more columns to the record header, repeat steps [10](#) to [12](#).
14. Click **File > Save**.

## Specifying Large Number of Columns at a Time

To add many columns to the search indexes, update the index metadata file in Enterprise Data Manager.

1. Click **View > Constraints**.
2. Right-click an application version and select **Export Index Metadata**.  
A csv file appears with a list of columns for all the tables in the application version.
3. Enter **Y** in the **Quick\_Search\_Index** column for the columns you want to include in the search index.
4. Save the csv file.
5. Right-click the application version and select **Import Index Metadata**.

## Troubleshooting Enterprise Data Manager

### **When I mine data, the mining job fails with JDBC driver errors.**

You might receive an error if the source database includes data types that are not supported by the JDBC driver.

To resolve the error, perform one of the following tasks:

- Convert the table to a view. Use SQL functions, such as `CONVERT`, to convert the table to a view. Translate the column data type to a generic data type such as `CHAR`, `VARCHAR`, or `NUMBER`. Then, mine the view.
- Use a custom JDBC driver to mine the data. When you mine the database, use the `CUSTOM_JDBC` option. If you use a custom JDBC driver, you must use the standalone version of the Enterprise Data Manager. You cannot assign custom JAR files for the web-based Enterprise Data Manager. Copy the JAR files to the local machine from where you run the mining job.

### **When I mine data, export metadata, or import metadata, the job stops responding. I get insufficient memory or heap errors.**

The Enterprise Data Manager is a client application that is launched using JNLP protocol. Every client application has memory limitations. The Enterprise Data Manager might run out of memory when you use the client to mine data, export metadata, or import metadata.

You might receive a `java.lang.OutOfMemoryException` error when you mine large numbers of tables. Or, when you import or export a metadata for an application version that includes a large number of tables and constraints. When the Enterprise Data Manager is out of memory, the job does not progress in the status window. Java can use up to 2 gigabytes of RAM, which is usually not enough for large applications.

To resolve this error, run the jobs in the background. When you run the job in the background, the job runs on the Data Archive server instead of on the Enterprise Data Manager client machine.

**When I import metadata, I cannot see my table listed in the Import Metadata from Database Wizard.**

The reason you cannot see a table in the **Import Metadata from Database Wizard** might be because the table contains a special character in the table name.

You cannot import metadata if the table name contains one of the following special characters: ~, !, @, #, \$, %, ^, &, \*, \_, +, -, and =.

## CHAPTER 3

# ILM Repository Constraints

This chapter includes the following topics:

- [ILM Repository Constraints Overview, 52](#)
- [Constraint Definition, 53](#)

## ILM Repository Constraints Overview

To search the Data Vault from the entity level and to automatically create entities for data discovery, you must add constraints to the ILM repository. You can add constraints to the repository when you import metadata from a source database or you can add them manually.

When you search the Data Vault from the entity level, the entity that you search on must have the relationships required to the associated tables to show related information at the transaction level.

To create entities for data discovery, you use the **Multiple Entity Creation** wizard. The wizard uses the constraints in the ILM repository to generate the entities. The wizard adds all related child tables for the parent table you select and adds all related tables to the entity. For more information, see [“Automatic Entity Generation” on page 37](#).

You do not need constraints to archive data. When you run an archive or retirement job, the ILM engine creates a WHERE clause that determines the data to archive.

The ILM repository stores constraints for source metadata. Data Archive stores primary, referential, and unique constraint types. When you import metadata from a source database to the ILM repository, all constraints that are defined on the database level are imported. If the source database does not include constraint information at the database level, such as in the application level, then you must add the constraints to the ILM repository. You can add constraints manually. Or, you can use tools to help you discover table relationships and create constraints based on the suggested relationships.

# Constraint Definition

Constraints might be included in the ILM repository when you import metadata for the source application. If the constraints are not included in the ILM repository, you can use the manual definition option to define constraints.

## Manual Definition

If you are familiar with the data model of the source and if the data model is not too large, you can manually create the constraints. Consider manually creating constraints if you have a small amount of tables to maintain constraints for.

After you import or define the constraints for source metadata in the ILM repository, you can maintain the constraints if you need to change them. For example, you can create, delete, or change constraints. The constraints are changed for the source metadata in the ILM repository, not in the source database.

After you import or define the constraints, you can use the constraints to automatically create entities.

## Pre-Packaged Applications

For pre-packaged applications, constraint definition is required if you have custom tables or if you modified standard application tables. Constraints for standard application tables are included in the application accelerators. When you install the accelerators, the metadata and the constraints are imported to the ILM repository.

## Custom Applications

For custom applications, constraint definition depends on how the source database stores constraints. If the source database maintains constraints at the database level, the constraints are automatically created in the ILM repository when you import metadata from the source database. If the source database maintains constraints other than at the database level, such as in the application logic, you must define the constraints. You must define constraints if constraint information is not included when you import metadata from the source application.

## SAP Applications

For SAP applications, constraint definition is required. When you import metadata from SAP applications, you import constraints that are defined at the database level. Constraints that are defined in the SAP ABAP Dictionary are not imported. Constraints are not required to retire the SAP application. Constraints are required to create entities for Data Discovery portal searches. You might want to define the constraints after you retire the application.

## CHAPTER 4

# Partition Exchange Purging

This chapter includes the following topics:

- [Partition Exchange Purging Overview, 54](#)
- [Working with Partition Exchange, 54](#)
- [Partition Exchange Procedure, 56](#)
- [Database Requirements, 56](#)
- [Configuring Partition Exchange, 57](#)
- [Managing Partition Exchange, 63](#)

## Partition Exchange Purging Overview

You can use Oracle partition exchange to delete data from sources on Oracle databases, version 10g or later. Configure partition exchange when you want to delete a significant percentage of the source data. For example, you want to delete twenty to thirty percent or more records from the source table.

Partition exchange increases the performance of the archive job when the job deletes data from the source. When you use partition exchange, the archive job calls a procedure that creates a partition for the records to keep in the source. The procedure swaps the original source segment with the new partition. It is more efficient to swap partitions than to individually delete a large amount of records from the source.

You can configure partition exchange for applications on Oracle databases that support partitioning. You can use partition exchange to delete data from partitioned and non-partitioned source tables. You can use partition exchange for archive and restore jobs.

Configure the entity to use partition exchange. When the archive job deletes data from the source, the job uses partition exchange for the tables that you configured in the entity. The job uses individual record deletion for the remainder of the tables in the entity that are not configured for partition exchange.

You can also disable or delete the configuration if you do not want jobs to use partition exchange.

## Working with Partition Exchange

When to configure partition exchange depends on several factors. The main factor is the volume of data that you want to delete from the source. Other factors include the archive or restore use case, the processing time, and the database down time.

Configure partition exchange based on the following factors:

### **Amount of data to delete from the source**

The biggest factor to determine if you should configure partition exchange is the amount of data that you want to delete from the source. Configure partition exchange when you want to delete a significant percentage of data from the source. For example, you want to delete twenty percent or more records from the source table.

### **Archive use case**

You can enable or disable partition exchange based on the archive job requirements, such as initial archiving or frequency of archiving.

You may want to configure partition exchange for the initial archive run if you have a lot of data to archive and purge. Then, disable the partition exchange for next regularly scheduled archive jobs. For example, you have 10 years worth of data in your source. For the first archive run on the source, you want to archive and purge the last 7 years worth of data. Then, you schedule a job that runs on a monthly schedule. The monthly jobs will not archive a large percentage of data. You configure partition exchange for the first archive run because there is a significant amount of data to delete. You disable the partition exchange configuration for the remainder of the jobs as the volume of the data to archive and purge will be significantly less.

Use the frequency of how often you plan to run an archive job to determine if you should configure partition exchange or not. For example, you may want to configure partition exchange for archive jobs that are scheduled on a less frequent basis, such as annual archiving. You may not want to configure partition exchange for jobs that run on a more frequent basis, such as monthly archiving. However, you should still consider the volume of the data even for archive jobs that run on a frequent basis.

### **Restore use case**

You can enable or disable partition exchange to restore and purge data from the history database to the source database. The type of restore job and the amount of data that you need to restore determines if you should configure partition exchange.

For transaction restore jobs, partition exchange is not relevant because the restore job does not involve a large amount of data.

For cycle restore jobs, you can configure partition exchange if the restore cycle restores and purges a large percentage of the data in the history table.

### **Processing time**

The main steps in the partition exchange procedure include creating a keep table, swapping the partitions, and rebuilding the source indexes. The time it takes to swap the partitions is minimal. However, the processing time to create the keep table and to rebuild the source indexes depends on the size of the data that remains in the table and the amount of indexes on the source. The processing time increases with the size of the data and the number of indexes on the source.

### **Database down time**

When the archive job deletes data from the source using partition exchange, the database must be off line to perform the partition exchange. To reduce the impact of the database down time, you can configure the job to pause before the delete from source step. Then, take the database off line.

# Partition Exchange Procedure

The `AA_PARTITION_PKG.exchange_partition` procedure runs the Oracle partition exchange logic for the archive job when the job deletes from the source. When you configure the entity steps, you add the procedure for the delete from source step. A single call to the procedure processes the partition exchange for all tables.

When the archive job calls the procedure, the procedure performs the following steps for each table that is configured for partition exchange:

1. The procedure creates a keep table in the staging user. The keep table contains all records from the source table that are not flagged for archiving. The run procedure uses the following naming convention to create the keep table:  
`<table name>_KEEP`
2. The procedure adds records to the keep table. To determine which records to add, the procedure subtracts the records in the staging table from the source table. The remaining records are not included in the archive job and should remain in the source.
3. The procedure invokes the Oracle partition exchange feature to swap the segments from the source table with the segments from the keep table.
4. The procedure rebuilds the source table indexes.
5. The procedure drops the keep table.

The procedure performs the steps for one source table at a time. After the procedure processes the first source table, the procedure performs the cycle again for the next source table. If the source table includes multiple partitions, then the procedure performs the steps for one partition at a time.

## Database Requirements

Additional staging space in the source staging tablespace is required to create the keep table and to store data in the keep table. The staging space amount depends on the amount of data that is going to be kept in the source table.

When the archive job runs the partition exchange procedure, the procedure only creates one keep table. When the database completes the partition exchange for the source table, the procedure drops the keep table. This process is repeated for every table that is configured for partition exchange. The size of the keep table depends on each source table. Verify that you have enough staging space for the maximum difference between the source and staging tables.

To estimate the amount of required staging space, configure the archive job to pause after the copy to staging step. Review the log to calculate the sizes of the staging tables and the source tables.



# Configuring Partition Exchange

To configure partition exchange, create and configure the interim tables, configure the entity, and verify primary key constraints.

1. Create and configure interim tables.  
Create an interim table and add the source tables. The archive job uses partition exchange to delete records from the tables that you add to the interim table. Then, configure the default columns in the interim table.
2. Configure the entity steps.  
Add steps to process the interim table you created, to disable the individual record deletion, and to enable the partition exchange run procedure.
3. Verify that the primary key constraints are defined for the source tables.

## Step 1. Create and Configure the Interim Tables

Create an interim table and add the source tables. The archive job uses partition exchange to delete records from the tables that you add to the interim table. Then, configure the default columns in the interim table.

Add columns for the table owner, table name, and partition name and define the corresponding select clauses. The partition name select clause depends on the partitioned status of the source tables in the interim table. The configuration depends on if all of the source tables in the interim table are partitioned, not partitioned, or both.

### Select Clause for Partitioned Source Tables

If all tables in the interim table are partitioned in the source, the archive job uses the ALL\_TAB\_PARTITIONS Oracle view to populate the interim table values.

The select clause you configure depends on if the tables have the same table owner or multiple table owners. You can add multiple tables for each table owner if the tables have the same partition name. The procedure can process one partition for each table owner. If the table includes multiple partitions, then you must list the table owner, table name, and each partition name.

### Single Table Owner Select Clause

Use the following syntax to form a select clause for the partition name when the interim tables have the same table owner:

```
SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE table_owner='<table owner>' AND table_name IN ('<table name>') AND
partition_name='<partition name>'
```

For example, you want to add the following values in the interim table:

Table Owner	Table Name	Table Partition
XLA	XLA_AE_HEADERS	CST

Use the following syntax:

```
SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE table_owner='XLA' AND table_name IN ('XLA_AE_HEADERS') AND partition_name='CST'
```

## Single Table Owner with Multiple Partitions within a Table Select Clause

Use the following syntax to form a select clause for the partition name when the interim tables have the same table owner and include multiple partitions for a table:

```
SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE table_owner='<table owner>' AND table_name IN ('<table name>') AND partition_name
IN ('<partition name>', '<partition name>')
```

For example, you want to add the following values in the interim table:

Table Owner	Table Name	Table Partition
XLA	XLA_AE_HEADERS	CST
XLA	XLA_AE_HEADERS	AR

Use the following syntax:

```
SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE table_owner='XLA' AND table_name IN ('XLA_AE_HEADERS') AND partition_name IN
('CST', 'AR')
```

## Multiple Table Owners Select Clause

Use one select statement for each table owner. Use a UNION statement to join multiple select statements.

Use the following syntax to form multiple select statements when the interim tables have different table owners:

```
SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE A.table_owner='<table owner 1>' AND A.table_name IN ('<table name>') AND
A.partition_name='<partition name>'
UNION
SELECT B.table_owner, B.table_name, B.partition_name, 'Y', null, null
FROM all_tab_partitions B
WHERE B.table_owner='<table owner 2>' AND B.table_name IN ('<table name>') AND
B.partition_name='<partition name>'
UNION
SELECT C.table_owner, C.table_name, C.partition_name, 'Y', null, null
FROM all_tab_partitions C
WHERE C.table_owner='<table owner 3>' AND C.table_name IN ('<table name>') and
C.partition_name='<partition name>'
```

For example, you want to add the following values in the interim table:

Table Owner	Table Name	Table Partition
XLA	XLA_AE_LINES	CST
APPLSYS	WF_ITEM_ACTIVITY_STATUSES	WF_ITEM45
PA	PA_SUMM_BALANCES	PA_SUMM_BALANCES_PAR11

Use the following syntax:

```
SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE A.table_owner='XLA' AND A.table_name IN ('XLA_AE_LINES') AND
A.partition_name='CST'
UNION
SELECT B.table_owner, B.table_name, B.partition_name, 'Y', null, null
FROM all_tab_partitions B
WHERE B.table_owner='APPLSYS' AND B.table_name IN ('WF_ITEM_ACTIVITY_STATUSES') AND
B.partition_name='WF_ITEM45'
UNION
SELECT C.table_owner, C.table_name, C.partition_name, 'Y', null, null
FROM all_tab_partitions C
WHERE C.table_owner='PA' AND C.table_name IN ('PA_SUMM_BALANCES') AND
C.partition_name='PA_SUMM_BALANCES_PAR11'
```

## Select Clause for Non-Partitioned Source Tables

If all tables in the interim table are not partitioned in the source, the archive job uses the ALL\_TABLES Oracle view to populate the interim table values. Note that the partition name column will not include any values.

The select clause you configure depends on if the tables have the same table owner or multiple table owners. You can add multiple tables for each table owner.

### Single Table Owner Select Clause

Use the following syntax to form the select clause for the table name when the interim tables have the same table owner:

```
SELECT A.owner, A.table_name
FROM all_tables A
WHERE A.Owner='<table owner>' AND A.table_name IN ('<table name>')
```

For example, you want to add the following values in the interim table:

Table Owner	Table Name
GL	GL_IMPORT_REFERENCES

Use the following syntax:

```
SELECT A.owner, A.table_name
FROM all_tables A
WHERE A.Owner='GL' AND A.table_name IN ('GL_IMPORT_REFERENCES')
```

### Multiple Table Owners Select Clause

Use one select statement for each table owner. Use a UNION statement to join multiple select statements.

Use the following syntax to form the select clause for multiple select statements when the interim tables have different table owners:

```
SELECT A.owner, A.table_name, null, 'Y', null, null
FROM all_tables A
WHERE A.Owner='<table owner 1>' AND A.table_name IN ('<table 1 name>', '<table 2 name>')
UNION
SELECT B.owner, B.table_name, null, 'Y', null, null
FROM all_tables B
B.owner='<table owner 2>' AND B.table_name IN ('<table 1 name>', '<table 2 name>')
UNION
SELECT C.owner, C.table_name, null, 'Y', null, null
FROM all_tables C
C.owner='<table owner 3>' AND C.table_name IN ('<table 1 name>', '<table 2 name>')
```

For example, you want to add the following values in the interim table:

Table Owner	Table Name
GL	GL_IMPORT_REFERENCES
	GL_JE_LINES
XLA	XLA_AE_HEADERS
	XLA_AE_LINES
APPLSYS	WF_ITEMS
	WF_ITEM_ATTRIBUTE_VALUES

Use the following syntax:

```
SELECT A.owner, A.table_name, null, 'Y', null, null
FROM all_tables A
WHERE A.Owner='GL' AND A.table_name IN ('GL_IMPORT_REFERENCES', 'GL_JE_LINES')
UNION
SELECT B.owner, B.table_name, null, 'Y', null, null
FROM all_tables B
B.owner='XLA' AND B.table_name IN ('XLA_AE_HEADERS', 'XLA_AE_LINES')
UNION
```

```

SELECT C.owner, C.table_name, null, 'Y', null, null
FROM all_tables C
C.owner='APPLSYS' AND C.table_name IN ('WF_ITEMS', 'WF_ITEM_ATTRIBUTE_VALUES')

```

## Select Clause for Partitioned and Non-Partitioned Source Tables

If the interim tables include a combination of partitioned tables and non-partitioned tables, the archive job uses a union of the ALL\_TAB\_PARTITIONS and ALL\_TABLES Oracle views to populate the interim table values.

Use the following syntax to form the select clause for the partition name:

```

SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE table_owner='<table owner>' AND table_name IN ('<table name>') AND
partition_name='<partition name>'
UNION
SELECT B.owner, B.table_name, null, 'Y', null, null
FROM all_tables B
WHERE B.Owner='<table owner>' AND B.table_name IN ('<table name>')

```

For example, you want to add the following values in the interim table:

Table Owner	Table Name	Partition Name
XLA	XLA_AE_HEADERS	CST
	XLA_AE_LINES	
GL	GL_IMPORT_REFERENCES	(None)
	GL_JE_LINES	

Use the following syntax:

```

SELECT A.table_owner, A.table_name, A.partition_name
FROM all_tab_partitions A
WHERE table_owner='XLA' AND table_name IN ('XLA_AE_HEADERS', 'XLA_AE_LINES') AND
partition_name='CST'
UNION
SELECT B.owner, B.table_name, null, 'Y', null, null
FROM all_tables B
WHERE B.Owner='GL' AND B.table_name IN ('GL_IMPORT_REFERENCES', 'GL_JE_LINES')

```

## Creating and Configuring the Interim Tables

Before you begin, identify the tables that you want to delete data from by using partition exchange. In the corresponding entities, create interim tables and add the tables. Then, configure the interim table.

1. Access the entity that includes the tables.
2. Create an interim table.  
You may want to indicate in the description that the interim table is for partition exchange.
3. In the **Tables** tab, add the tables.
4. In the **Default Columns** tab, add the following entries for the table owner, table name, and partition name:

Order	Name	Type	Length	Select Clause
1	TABLE_OWNER	VARCHAR2	50	A.TABLE_OWNER
2	TABLE_NAME	VARCHAR2	50	A.TABLE_NAME
3	PARTITION_NAME	VARCHAR2	50	A.PARTITION_NAME

5. Configure the select clause for the partition name.

The configuration depends on if all of the source tables in the interim table are partitioned, not partitioned, or both. If the source table is partitioned and you archive data from multiple partitions in the table, then list each partition from which you archive data.

6. Save the interim table.

## Step 2. Configure the Entities

After you add an interim table to the entity, configure the entity steps. Configure steps to process the interim table you created, to disable the individual record deletion, and to enable the partition exchange run procedure.

1. Access the entity **Steps** tab.
2. Add the Drop Interim Table, Create Interim Table, and Insert Interim Table steps for the partition exchange interim table.  
The order can be at any point in candidate generation.
3. Add a Run Procedure in Generate Candidates step for the partition exchange interim table and add the `am_plsqlutil_pkg.compile_self` procedure.  
The procedure is required for the `AA_PARTITION_PKG.exchange_partition` procedure to run the partition exchange logic.
4. Add a Run Procedure in Generate Candidates step for the partition exchange interim table and add the `aa_util_pkg.compile_self` procedure.  
The procedure is required for the `AA_PARTITION_PKG.exchange_partition` procedure to run the partition exchange logic.
5. Add a Run Procedure in Generate Candidates step for the partition exchange interim table and add the `aa_partition_pkg.compile_self` procedure.  
The procedure allows the `AA_PARTITION_PKG` package to compile objects in the staging user.
6. Delete or disable the corresponding Delete Apps Tables step for each table that you added to the interim table.
7. After the Insert into Archive Tables steps, add a Run Procedure in Delete from Source step for the partition exchange interim table and add the `AA_PARTITION_PKG.exchange_partition` procedure.
8. Save the entity.

### Example

You created the XC\_384\_XLA2 Exchange Partition interim table for the XLA Subledger Accounting entity. You added the XLA\_AE\_HEADERS and XLA\_AE\_LINES tables to the interim.

The following table lists the steps that you configure in the entity Steps tab:

Configuration	Action	Interim Table	Entity Table	Procedure	Enabled
Add	Drop Interim Table	XC_384_XLA2 Exchange Partition	(None)	(None)	Yes
Add	Create Interim Table	XC_384_XLA2 Exchange Partition	(None)	(None)	Yes

Configuration	Action	Interim Table	Entity Table	Procedure	Enabled
Add	Insert Interim Table	XC_384_XLA2 Exchange Partition	(None)	(None)	Yes
Add	Run Procedure in Generate Candidates	XC_384_XLA2 Exchange Partition	(None)	am_plsqlutil_pkg.compile_self	Yes
Add	Run Procedure in Generate Candidates	XC_384_XLA2 Exchange Partition	(None)	aa_util_pkg.compile_self	Yes
Add	Run Procedure in Generate Candidates	XC_384_XLA2 Exchange Partition	(None)	aa_partition_pkg.compile_self	Yes
Delete or Disable	Delete Apps Tables	XC_384_XLA1 Subledger Accounting	XLA_AE_HEADERS	(None)	No
Delete or Disable	Delete Apps Tables	XC_384_XLA1 Subledger Accounting	XLA_AE_LINES	(None)	No
Add	Run Procedure in Delete From Source	XC_384_XLA2 Exchange Partition	(None)	AA_PARTITION_PKG.exchange_partition	Yes

After you configure the entities, configure the source connection to use staging. Staging is required because the procedure creates the keep table in the staging database user.

### Step 3. Verify the Primary Key Constraints

Verify that the primary key constraints are defined for the source tables that you added to the partition exchange interim tables. By default, pre-packaged accelerators include key constraints. If you modified or added metadata within the pre-packaged accelerators or if you have custom applications, verify that the primary key constraints exist.

The archive job uses a combination of the row ID and the primary key to delete from the source. If the primary key constraint is not defined, then you receive an error in the job log. The job cannot populate the keep table without the primary key.

# Managing Partition Exchange

You can enable or disable partition exchange configuration based on the archive job requirements. After you configure partition exchange and run an archive job, you may want to disable or remove the partition exchange configuration. Whether you disable or remove the configuration depends on if you want to enable the configuration again in the future.

You may want to enable the configuration for some jobs and disable it for other jobs. For example, you configured partition exchange for the first time you ran an archive job due to the volume of the data to archive. You want to disable or remove the partition exchange because the subsequent archive job volume is considerably less than the initial archive. However, if the volume changes in the future, you can enable partition exchange again.

You may want to disable or remove partition exchange for restore jobs as partition exchange may not be the optimized method of deletion for some restore jobs. For example, partition exchange is not recommended to restore transactions or to restore a cycle that has a small percentage of data in the history table.

## Disabling or Removing Partition Exchange

To remove or disable partition exchange, configure the entity steps.

1. In the entity steps, disable or remove the Run Procedure in Delete from Source step that calls the `AA_PARTITION_PKG.exchange_partition` procedure.
2. Enable or add a Delete Apps Tables step for each table that you included in the partition exchange interim table.

# CHAPTER 5

## APIs

This chapter includes the following topic:

- [TPT\\_EXTRACT\\_SCRIPT\\_TEMPLATE, 64](#)

## TPT\_EXTRACT\_SCRIPT\_TEMPLATE

TPT\_EXTRACT\_SCRIPT\_TEMPLATE is the Teradata Parallel Transporter default script template. Archive or retirement jobs can use Teradata Parallel Transporter to extract data from Teradata to the Data Vault. You enable Teradata Parallel Transporter in the Teradata source connection.

The script template includes supported Teradata scripting language and parameters. The job uses the script template that is configured in the Teradata source connection to create one script for each table the job exports data from. During the script creation, the job replaces the parameters with values that are specific to the table. The job uses the script to pass the query request to Teradata Parallel Transporter. Teradata Parallel Transporter rewrites the query request to optimize the query for maximum performance and throughput. Then, Teradata Parallel Transporter creates one BCP file for each table.

No additional configuration is required for the script template. TPT\_EXTRACT\_SCRIPT\_TEMPLATE is the default script template name in the Teradata source connection.

Optionally, you can copy the script template and modify the copied script template. You may want to modify the script template if you want to use additional parameters or to change the scripting language. For example, you may modify the script to optimize performance. If you copy the script template, you must update the script template name in the Teradata source connection.

### Script Template Parameters

The Teradata Parallel Transporter script template includes default parameters. The archive or retirement job replaces the parameters at runtime when the job creates scripts for tables.

The default parameters are required. No additional configuration is needed. Optionally, you can add parameters to the script. For example, you may want to use a different login name. You can use any property from the Teradata source connection as a script parameter. To add a parameter, use the technical name from the AM\_DATA\_REP\_ATTR table in the ILM repository.

The script template includes the following required parameters:

**&TABLE\_NAME**

Table name that the job needs to export data from.



**&JOB\_ID**

Job ID of the archive or retirement job. Teradata Parallel Transporter uses the job ID as part of the BCP file name.

**&DIR\_PATH**

BCP file staging directory in the Teradata source connection.

**&TDPID**

Host name in the Teradata source connection.

**&SESSION**

Maximum number of parallel sessions in the Teradata source connection.

**&USER\_NAME**

Application login name in the Teradata source connection.

**&PASSWORD**

Password for the application login name in the Teradata source connection.

**&SELECT\_STMT**

SQL select statement that the job creates to extract the data from Teradata.

**&FILE\_NAME**

Name of the BCP file that the Teradata Parallel Transporter creates. By default, the job uses the following convention to create BCP files:

`<Job ID>_<Table Name>.bcp`

## CHAPTER 6

# Salesforce Accelerator

This chapter includes the following topics:

- [Salesforce Accelerator Overview, 66](#)
- [Connecting to Salesforce and Importing Metadata, 66](#)
- [Salesforce Accelerator Entities , 69](#)

## Salesforce Accelerator Overview

If you install the Data Archive Salesforce accelerator, you can connect to Salesforce to import metadata from Salesforce the same way you import metadata for any ERP or custom application. By default, the Salesforce application and an application version called "Sales" are visible in the Enterprise Data Manager application hierarchy. When you install the accelerator, the Task and Event entities are installed within the Sales application version. The Task and Event entities contain standard Salesforce tables that you might want to archive. You can also create custom Salesforce entities to archive.

## Connecting to Salesforce and Importing Metadata

Connect to Salesforce to import metadata.

1. In the **Canvas** pane, select the **Sales** application version, or a custom application version you created, under the **Salesforce** application folder. Note that you must import the Salesforce table metadata inside the Salesforce application folder in the application hierarchy.
2. Click **File > Import Metadata from Database**.
3. From the menu, select the database type **Salesforce**.
4. For the database host parameter, enter one of the following values, for example:
  - "login.salesforce.com" for production environments.
  - "test.salesforce.com" for sandbox environments.
  - A Salesforce customized URL without the preceding "https://" For example, if the customized URL is "http://my-domain.salesforce.com," enter the value as "my-domain.salesforce.com."
5. When the Salesforce JDBC driver connects to Salesforce, it creates an embedded database in the local machine for internal use. The driver database parameter specifies the file name prefix or file absolute

path plus the file name prefix that the driver uses to create or locate the set of files that define the internal database. Enter one of the following values:

- You can leave the parameter blank, if the system user has permission to create files in the location where the Enterprise Data Manager runs. By default a blank value is taken as "<host>\_<username>." For example, if the host is "login.salesforce.com," and the username is "abc," then the default value is "login.salesforce.com\_abc." Files will be created with the same prefix. For example, "login.salesforce.com\_abc.config."
- Enter any name for the database. The driver creates a local database with the specified name. For example, if you enter the name "Test," the driver creates a local database with file names such as `Test.config` and `Test.properties` in the location where the Enterprise Data Manager runs.
- Enter an absolute file path plus a database name. The driver creates a local database with the database name in the specified path. For example, if you enter the value as "C:\ILM\_DA\Test," the driver creates a local database with file names such as `Test.config` and `Test.properties` under the path `C:\ILM_DA`

For offline metadata import, use only the database name.

For online metadata import, verify that the path exists on the Enterprise Data Manager client machine.

6. For the username parameter, enter the Salesforce username.
7. For the password parameter, enter the Salesforce password.
8. If you have a Salesforce security token, enter it as the security token parameter. This field is optional.
9. Click **OK**.
10. In the schema selection window, click the **SFORCE** schema and then the **Down** arrow button to select it.
11. Click **Next** and proceed through the Import wizard.

## Importing Metadata from Child Tables

As you configure the metadata import for the Salesforce schema, you have the option to automatically extract metadata from the child tables in the SFORCE schema. Some Salesforce objects contain circular relationships that make automatic entity creation impossible when you automatically extract metadata from the child tables.

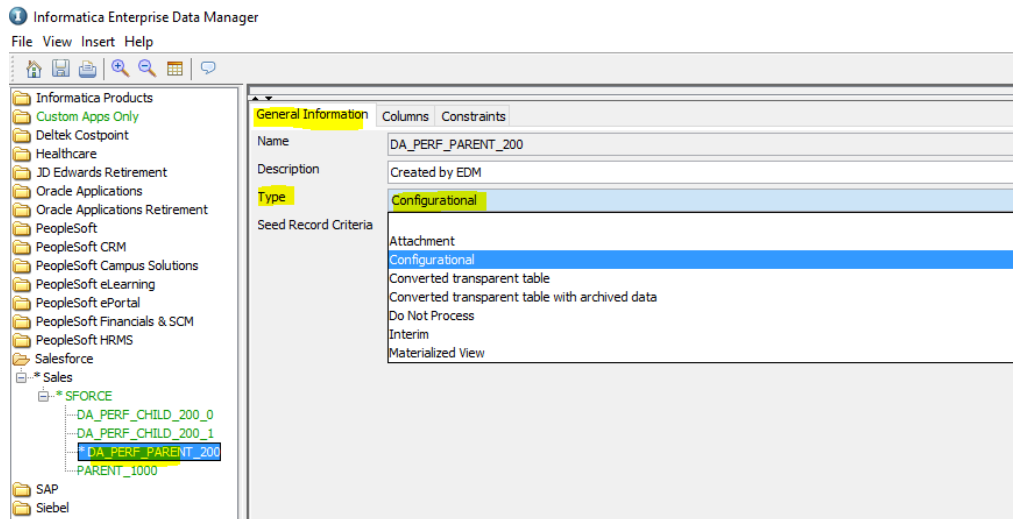
When you import metadata, adhere to the following guidelines:

- If you submit the import metadata job as a background job, to avoid mining circular relationships within the objects, give a proper regular expression value for objects that need to be included in the entity and select **No** for the "Mine child tables" option. If you select **Yes** for the "Mine child tables" option, you receive a warning message that automatic entity creation will not be possible.
- If you import the metadata through the Enterprise Data Manager, select the objects that need to be included in the entity and then click **No** when you are asked if you want to extract metadata from the child tables of these tables.

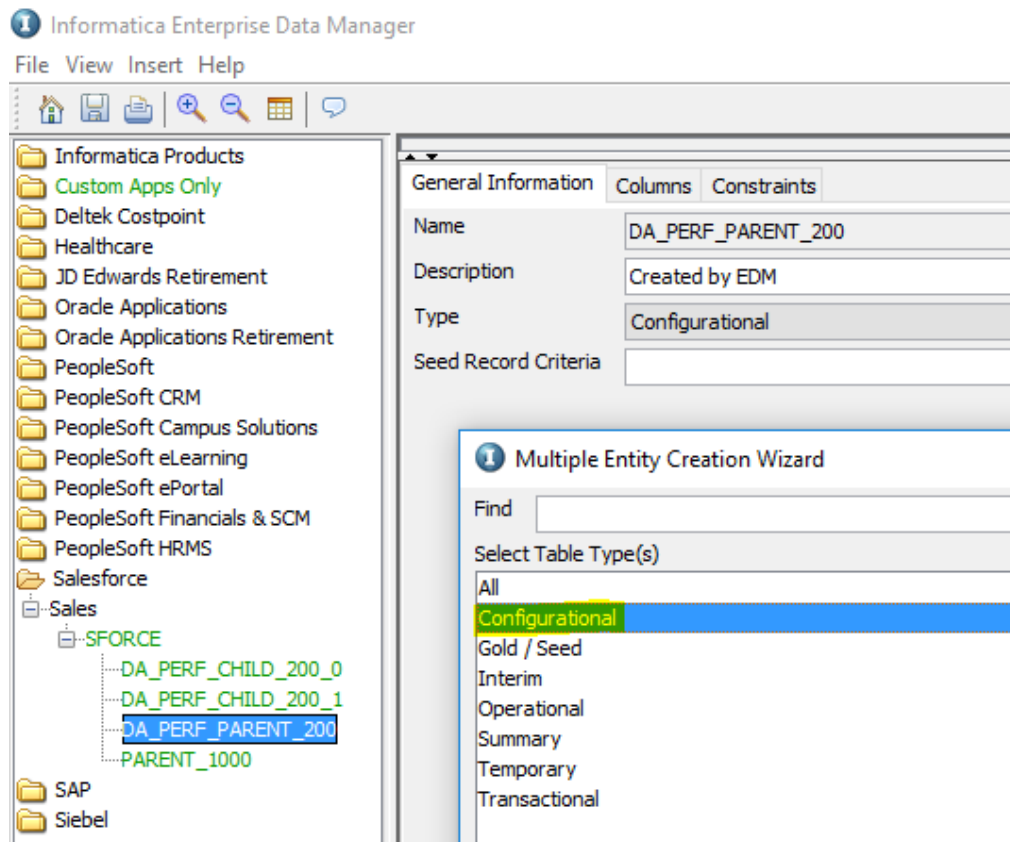
## Automatic Entity Creation for Mined Child Objects

If you already extracted metadata from child objects that have circular relationships, but you want to use automatic entity generation, perform the following steps:

1. In the **Canvas** pane, select the related tables that you want in the entity and change their type on the **General Information** tab. For example, change all of the related tables to the Configurational or Temporary type.



2. When you run the **Multiple Entity Generation Wizard**, create the entity using the same type of table that you configured above.



Salesforce has cascade delete functionality. When you create the entity, make sure the entity includes both parent and child tables if the child tables have cascade delete set to true. Otherwise, in the Delete from Source step of the purge job or archive and purge job, Salesforce deletes the child records automatically while deleting the parent records.

When you select the **Include Salesforce Archived Records** check box in the source connection, the driver gets the records that are archived by Salesforce in addition to the soft-deleted records. If you use the same connection to archive data from any custom entities, it is your responsibility to filter out the soft-deleted records by appending "IsDeleted=false" in the entity's select, insert and delete query.

## Salesforce Accelerator Entities

The Salesforce accelerator contains two standard entities, Task and Event. These entities contain commonly-used Salesforce objects that you might want to archive.

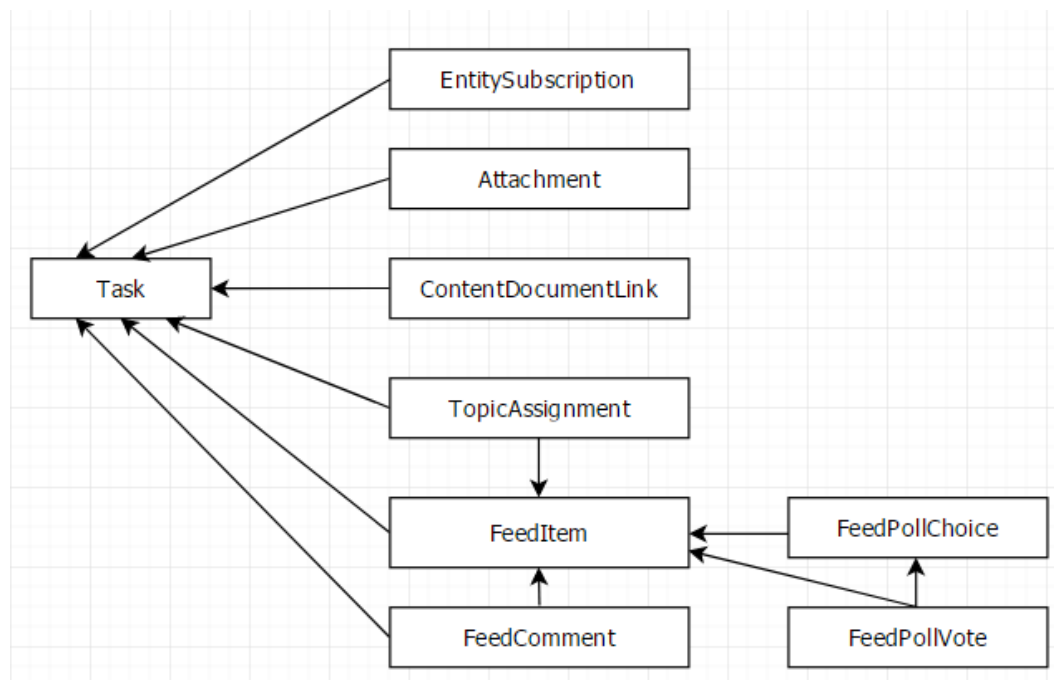
You cannot customize the Task and Event entities. You can edit the entities by copying them and pasting them within the Sales version. The copied entities will appear as "Copy of <entity name>."

## Task Entity Tables

The Task entity contains the following tables:

- Task
- EntitySubscription
- Attachment
- ContentDocumentLink
- TopicAssignment
- FeedItem
- FeedComment
- FeedPollChoice
- FeedPollVote

The following graphic displays the relationships between the tables:



## Task Entity Parameters

The standard Task entity has the following parameters:

Parameter Name	Value	WHERE Clause Operator	Default Value	Mandatory
Created By	Salesforce login username	=	Empty	No
Created Date	Task 'CreatedDate' column value	<=	Empty	Yes

Parameter Name	Value	WHERE Clause Operator	Default Value	Mandatory
Due Date	Task 'ActivityDate' column value	<=	Empty	No
Last Modified Date	Task 'LastModifiedDate' column value	<=	Empty	No
Status	Task 'Status' column value	=	Completed (will not be selected by default in UI but hardcoded in EDM default columns)	No

The list of values for the "Status" parameter pop-up is populated with all of the Task 'Status' column values. If any new value is added after you created the source connection, those values are not present in the pop-up list of values. To populate the list with new values, run the Refresh Schema for Salesforce standalone job and re-open the list of values to select the new value.

## Task Entity Business Rules

The Task entity contains a business rule for partial recurrence records.

Data Archive archives a recurring Salesforce task only when the full transaction (the entire recurring task) is selected with reference to the particular filter criteria. You must give the values so that the entire recurring transaction will be selected. This business rule updates the interim table PURGEABLE\_FLAG column to 'N' if any partial recurrence records are found in it. For more information on partial recurrence tasks, see the topic "Partial Recurrence Activities" in this chapter.

## Excluded Child Tables

The following child tables are not included in the Task entity:

Table Name	Reason for Exclusion	Cascade Delete
AttachedContentDocument	Salesforce limitation - not queryable	True
CombinedAttachment	Salesforce limitation - not queryable	True
ContentVersion	-	False
EmailMessage	-	False
EmailStatus	-	False
PartnerNetworkRecordConnection	-	True
TaskFeed	Subset of FeedItem	True

When the archive job purges task records, records from the excluded child tables that have cascade delete set to true will also be purged by Salesforce. Some tables are excluded from the entity because they are a

subset of another table. Even though data from these tables is purged by Salesforce, there is no data loss on the subset tables because the same data is available in the superset table, which is part of the Task entity.

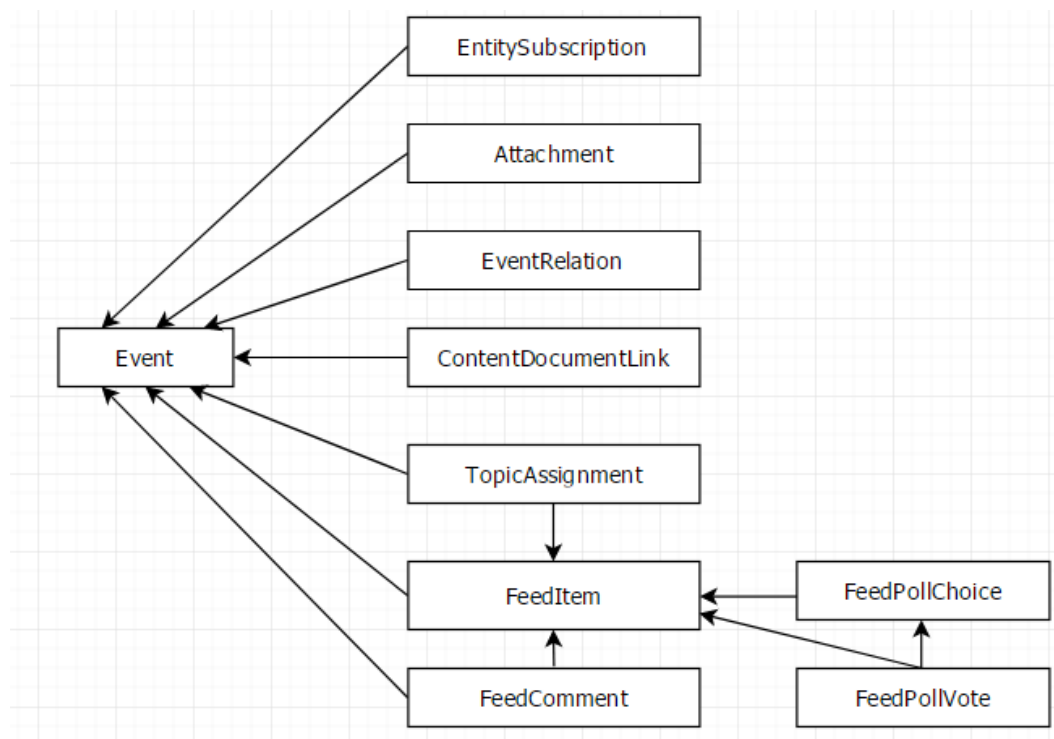
Note that some tables are excluded from the entity because they are not available unless features like tag, shared activities, and network communities are enabled.

## Event Entity Tables

The Event entity contains the following tables:

- Event
- EntitySubscription
- Attachment
- EventRelation
- ContentDocumentLink
- TopicAssignment
- FeedItem
- FeedComment
- FeedPollChoice
- FeedPollVote

The following graphic displays the relationships between the tables:





## Event Entity Parameters

The standard Event entity has the following parameters:

Parameter Name	Value	WHERE Clause Operator	Default Value	Mandatory
Created By	Salesforce login username	=	Empty	No
Created Date	Event 'CreatedDate' column value	<=	Empty	No
Last Modified Date	Event 'LastModifiedDate' column value	<=	Empty	No
Start Date	Event 'StartDateTime' column value	<=	Empty	No
End Date	Event 'EndDateTime' column value	<=	Empty	Yes

## Event Business Rules

The Event entity contains a business rule for partial recurrence records.

Data Archive archives a recurring Salesforce event only when the full transaction (the entire recurring event) is selected with reference to the particular filter criteria. You must give the values so that the entire recurring transaction will be selected. This business rule updates the interim table PURGEABLE\_FLAG column to 'N' if any partial recurrence records are found in it. For more information on partial recurrence events, see the topic "Partial Recurrence Activities" in this chapter.

## Excluded Child Tables

The following child tables are not included in the Event entity:

Table Name	Reason for Exclusion	Cascade Delete
AcceptedEventRelation	Subset of EventRelation table	False
AttachedContentDocument	Salesforce limitation - not queryable	True
CombinedAttachment	Salesforce limitation - not queryable	True
ContentVersion	-	False
DeclinedEventRelation	Subset of EventRelation table	False
EventFeed	Subset of FeedItem table	True
UndecidedEventRelation	Subset of EventRelation table	False

When the archive job purges event records, records from the excluded child tables that have cascade delete set to true will also be purged by Salesforce. Some tables are excluded from the entity because they are a subset of another table. Even though data from these tables is purged by Salesforce, there is no data loss on the subset tables because the same data is available in the superset table, which is part of the Event entity.

Note that some tables are excluded from the entity because they are not available unless features like tag, shared activities, and network communities are enabled.

In the Salesforce Event page, there is an option to create a meeting request using the **New Meeting Request** button. Once the meeting is created, by default it will exist in an "Unconfirmed" state. In an unconfirmed state, some of the event columns like "EndDateTime" and "StartDateTime" are empty. Because of this unconfirmed state, records will not be included in an archive or purge job.

## Partial Recurrence Activities

In Salesforce, you can create normal activities or recurring activities (both event and task). When you select records to archive, if the particular criteria selects only part of the recurring activities, then those activities are not archived.

For example, a recurrence activity was created on 30-Dec-2015 to recur 30-Dec-2015 to 1-Jan-2016. The activity has four records, where one record is the parent record and three are child records, as shown in the table below.

ID	Recurrence ID	Subject	Start Date	End Date	Recurrence End Date Only
XXXXXXX1	XXXXXXX1	Recurrence Test	30-Dec-2015	30-Dec-2015	01-Jan-2016
XXXXXXX2	XXXXXXX1	Recurrence Test	30-Dec-2015	30-Dec-2015	
XXXXXXX3	XXXXXXX1	Recurrence Test	31-Dec-2015	31-Dec-2015	
XXXXXXX4	XXXXXXX1	Recurrence Test	01-Jan-2016	01-Jan-2016	

If you want to archive all of the 2015 data and give the "End Date" parameter in the Event entity of the archive job definition as 31-Dec-2015, then the first three records in the table are filtered. Data Archive will not archive these records, however, because the particular date selected partial records, or a partial selection of the transaction. These records are eliminated by the business rule, so none of the records will be archived or purged with the particular end date criteria.

To further understand how partial recurrence activities are archived, see the following example of an archive project that contains both the Task and Event entities.

The Task entity parameters are configured with the values in the following table:

Task Parameter	Value
CreatedDate*	31-Dec-2015
DueDate	null
LastModifiedDate	null
Created By	null
Status	Completed

The Event entity parameters are configured with the values in the following table:

Event Parameter	Value
EndDate*	31-Dec-2015
StartDate	null
CreatedDate	null
LastModifiedDate	null
Created By	null

Given these start and end dates in the entity parameters, see the table below for a description of how Data Archive would archive different types of records in different scenarios.

Activity	Description	StartDateTime/ ActivityDate	EndDateTime	RecurrenceEnd DateOnly	No. of Records in Transaction	Filtered Records	Ready for Archive/ Purge
Past normal record	Normal activity that is older than the particular date 31- Dec-2015	8-Sep-14	8-Sep-14	-	1	1	Yes
Today normal record	Normal activity that is happening on the particular date 31- Dec-2015	31-Dec-15	31-Dec-15	-	1	1	Yes
Future normal record	Normal activity in the future with the particular date 31- Dec-2015	5-Jan-16	5-Jan-16	-	1	0	No

Activity	Description	StartDateTime/ ActivityDate	EndDateTime	RecurrenceEnd DateOnly	No. of Records in Transaction	Filtered Records	Ready for Archive/ Purge
Normal Event record which starts today and ends tomorrow (one normal event can span for 14 days)	Normal activity happening on the particular date 31-Dec-2015, but ending on a future date	31-Dec-15	1-Jan-16	-	1	0	No
Past recurrence parent record	Recurrence activity older than the particular date 31-Dec-2015	8-Sep-14	8-Sep-14	10-Sep-14	4	4	Yes
Future recurrence parent record	Recurrence activity in the future with the particular date 31-Dec-2015	8-Feb-16	8-Feb-16	10-Feb-16	4	0	No
Recurrence parent record that ends in future	Recurrence activity ending in the future with the particular date 31-Dec-2015	30-Dec-15	30-Dec-15	1-Jan-16	4	3	No

Activity	Description	StartDateTime/ ActivityDate	EndDateTime	RecurrenceEnd DateOnly	No. of Records in Transaction	Filtered Records	Ready for Archive/ Purge
Recurrence parent record that ends today	Recurrence activity ending on the particular date 31-Dec-2015	29-Dec-15	29-Dec-15	31-Dec-15	4	4	Yes
Edited recurrence parent record (After creating, one of the child dates has been moved to a future date, for example 05-Jan-2016)	Edited recurrence activity which ends in future with the particular date 31-Dec-2015	27-Dec-15	27-Dec-15	29-Dec-15	4	3	No

# APPENDIX A

## SAP Application Retirement Entities

This appendix includes the following topics:

- [SAP Application Retirement Entities Overview, 78](#)
- [Entities for Attachments, 78](#)
- [Entities for Transparent HR Cluster Tables and Text Tables, 80](#)

### SAP Application Retirement Entities Overview

Data Archive includes pre-packaged entities for SAP applications. No installation is required. Entities are available for transparent HR tables, transparent text tables, and attachments.

The Enterprise Data Manager includes an application for SAP. The application includes application modules for CRM, ERP, SCM, and SRM. Each application module has attachment entities. In addition, the ERP application module includes entities for the transparent HR PCL1-PCL5 cluster tables and the STXL text table.

The pre-packaged entities are for retirement only. The entities do not include constraints. If you need to use the Data Discovery portal to search the retired data, you must create entities that include constraints.

The pre-packaged entities are standard and are read-only. No installation is required. To modify the entities, use the Copy Application job to copy the SAP application to a customer-defined application. Copy the pre-packaged entities to the customer defined application, and then modify the copied entities. You may want to modify entities to change the run procedure statement. For example, you may want to change a procedure parameter to keep external attachments that are stored in an external file system in the original file system.

### Entities for Attachments

Attachment entities are included for each SAP application module. The entities include tables that store attachments for each application module. There is an entity that includes the attachment link table, an entity

for tables that store attachments in the database, and an entity for tables that store attachments in ADK files in an external file system.

Each entity includes default entity steps. However, only the Insert into Archive Tables step is enabled. The rest of the steps are not relevant for retirement. The Insert into Archive Tables step determines from where the retirement job reads data for tables in the entity.

Each application module includes the following attachment entities:

#### <Application Module> Attachments Links

Includes the attachment link table, ZINFA\_ATTCH\_LINK. The entity includes the default Insert Into Archive Table step that reads directly from the database.

The ZINFA\_ATTCH\_LINK table includes information on the downloaded attachments so you can view the attachments after you retire them. If you archive attachments to a file system or external storage, then you can use the link table to build the path and file name for Data Discovery style sheets. If you archive attachments to the Data Vault, the table includes information to link attachments with the AM\_ATTACHMENTS table.

#### <Application Module> Attachments Part 1

Extracts the attachments that are stored in the SAP database. The entity includes the default Insert Into Archive Table step that reads directly from the database.

The entity includes a `ArchiveSAPAttachmentTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the file system or external storage.

The full path for the procedure is:

```
java://
com.application.archive.dao.impl.ArchiveSAPAttachmentTable(TABNAME=<attachment table
name>,ABAP_PRG_NAME=ZINFA_DOWNLOAD_ATTACHMENTS,EXT_STORAGE=N,DTFORMAT=yyyy-MM-dd-
HH.mm.ss)
```

Set the `EXT_STORAGE=` parameter in the procedure to N.

Default is N.

#### <Application Module> Attachments Part 2

Extracts the attachments that are located in a file system or external storage.

The entity includes a `ArchiveSAPAttachmentTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the file system or external storage.

The full path for the procedure is:

```
java://
com.application.archive.dao.impl.ArchiveSAPAttachmentTable(TABNAME=<attachment table
name>,ABAP_PRG_NAME=ZINFA_DOWNLOAD_ATTACHMENTS,EXT_STORAGE=Y,DTFORMAT=yyyy-MM-dd-
HH.mm.ss)
```

Set the `EXT_STORAGE=` parameter in the procedure to Y.

Default is N.

#### <Application Module> Attachments Part 3

Extracts the attachments that are stored in the SAP database. The entity includes the default Insert Into Archive Table step that reads directly from the database.

The entity includes a `ArchiveSAPAttachmentTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the file system or external storage.

The full path for the procedure is:

```
java://  
com.application.archive.dao.impl.ArchiveSAPAttachmentTable (TABNAME=<attachment table  
name>, ABAP_PRG_NAME=ZINFA_DOWNLOAD_ATTACHMENTS, EXT_STORAGE=N, DTFORMAT=yyyy-MM-dd-  
HH.mm.ss)
```

Set the EXT\_STORAGE= parameter in the procedure to N.

Default is N.

#### <Application Module> Attachments Part 4

Extracts the attachments that are located in a file system or external storage.

The entity includes a `ArchiveSAPAttachmentTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the file system or external storage.

The full path for the procedure is:

```
java://  
com.application.archive.dao.impl.ArchiveSAPAttachmentTable (TABNAME=<attachment table  
name>, ABAP_PRG_NAME=ZINFA_DOWNLOAD_ATTACHMENTS, EXT_STORAGE=Y, DTFORMAT=yyyy-MM-dd-  
HH.mm.ss)
```

Set the EXT\_STORAGE= parameter in the procedure to Y.

Default is N.

Copy the entities from the "SAP 4.X and Above" product family version based on the SAP application.

## Entities for Transparent HR Cluster Tables and Text Tables

The ERP application module includes entities for the transparent HR PCL1-PCL5 cluster tables and the STXL text table.

Each entity includes default entity steps. However, only the Insert into Archive Tables step is enabled. The rest of the steps are disabled as the steps are not relevant for retirement. The Insert into Archive Tables step determines from where the retirement job reads data for tables in the entity.

The default entities are only for tables that store data in the database. The entities include the `ArchiveSAPLogicalTable` procedure for the Insert into Archive Table entity step. The procedure overrides the default entity step to connect to the database. The procedure calls an ABAP function module to read data from the SAP application layer.

If you archived data from the HR cluster tables or the STXL text table, the Retirement Auto Entity Creation job creates separate entities for the tables that store data in ADK files.

The ERP application module includes the following entities:

#### **HR Cluster Tables**

Includes tables for the PCL1-PCL5 cluster tables.

#### **Text Tables**

Includes the STXL text table.



# APPENDIX B

## Import Formats for Constraints

This appendix includes the following topics:

- [Import Formats for Constraints Overview, 81](#)
- [Importing Referential Integrity Constraints, 81](#)
- [Importing Primary Integrity Constraints, 82](#)
- [Importing Column Translation Data, 82](#)

### Import Formats for Constraints Overview

This section enlists sample formats for bulk import of Constraints specified in an Excel Sheet.

### Importing Referential Integrity Constraints

The following table explains each column of a sample Referential Integrity Constraints XLS file:

Column Name	Value
Table Name	Name of the Primary Table.
Constraint Name	Name of the Constraint.
Constraint Column	Column names where Constraint is imposed in the Primary and Secondary Tables.
Primary Key Name	Column name in the Primary Table which functions as its Primary key.
Ref Table	Secondary Table name.
Owner	Owner of the Primary Table (that is, its Parent Application Module).
Ref Owner	Owner of the Secondary Table (that is, its Parent Application Module).
Constraint Type	Type of Constraint ("R" for Referential Integrity).
Status	Specifies whether the Constraint is to be imposed currently.

# Importing Primary Integrity Constraints

The following table explains each column of a sample Primary Integrity Constraints' XLS file.

Column Name	Value
Table Name	Name of the Table.
Constraint Name	Name of the Constraint.
Constraint Column	Column(s) which will act as Primary key.
Owner	Owner of the Table (that is, its Parent Application Module).
Constraint Type	Type of Constraint ("P" for Primary Integrity).

# Importing Column Translation Data

Column Translation data can also be imported in bulk by specifying a format for XLS. A brief explanation of fields is included in the following table:

Column Name	Value
Table Name	Name of the "first" Table.
Field Name	Column in the "first" Table that needs to be translated.
Related Table Name	Name of the "second" Table.
Related Field Name	Column in the "second" Table that will be either displayed or De-referenced for connecting to a "third" Table.
Constraint	Constraint that defines a relation between the "first" and "second" Tables.
Parent Table Alias	Alias for the "first" Table.
Ref Table alias	Alias for the "second" Table.
Where Clause	WHERE clause for the JOIN between the two Tables.

# APPENDIX C

## Glossary

### **application**

A list of brand names that identify a range of ERP applications, such as Oracle and PeopleSoft.

### **application module**

A list of supported Application Modules for a particular Application Version.

### **application version**

A list of versions for a particular application.

### **archive**

Informatica refers to a backed up database as an Archive. From the process perspective, it also means archiving data from an ERP/CRM instance to an online database using business rules.

### **archive definition**

An Archive Definition defines what data is archived, where it is archived from, and where it is archived to.

### **archive engine**

A set of software components that work together to archive data.

### **business rule**

A business rule is criteria that determines if a transaction is eligible to be archived or moved from the default segment.

### **business table**

Any Table that (with other Tables and Interim Tables) contributes to define an Entity.

### **custom object**

An Object (Application / Entity) defined by an Enterprise Data Manager Developer.

### **Data Archive**

A product of the Information Lifecycle Management Suite, which provides flexible archive / purge functionality that quickly, reduces the overall size of production databases. Archiving for performance, compliance, and retirement are the three main use cases for Data Archive.

### **Data Archive metadata**

Metadata that is specific to a particular ILM product.

**data destination**

The database containing the archived data.

**Data Model Metadata**

Metadata that is common to Data Archive.

**data source**

The database containing the data to be archived.

**de-referencing**

The process of specifying inter-Column Constraints for JOIN queries when more than two Tables are involved. Enterprise Data Manager dictates that two Tables are specified first and then a Column in the Second Table is De-Referenced to specify a Third Table (and so on) for building the final JOIN query.

**entity**

An entity is a hierarchy of ERP/CRM tables whose data collectively comprise a set of business transactions. Each table in the hierarchy is connected to one or more other tables via primary key/foreign key relationships. Entity table relationships are defined in a set of metadata tables.

**ERP application**

ERP applications are software suites used to create business transaction documents such as purchase orders and sales orders.

**home database**

The database containing metadata and other tables used to persist application data. This is contained in the Schema, usually known as "AMHOME".

**interim**

Temporary tables generated for an Entity for data archive.

**metadata**

Metadata is data about data. It not only contains details about the structure of database tables and objects, but also information on how data is extracted, transformed, and loaded from source to target. It can also contain information about the origin of the data.

Metadata contains Business Rules that determines whether a given transaction is archivable.

**restore (Cycle)**

A restore operation based on a pre-created Archive Project.

**security group**

A Security Group is used to limit a User (during an Archive Project) to archive only data that conforms to certain scope restrictions for an Entity.

**staging schema**

A staging schema is created at the Data Source for validations during data backup.

**standard object**

An Object (Application / Entity) which is pre-defined by the ERP Application and is a candidate for Import into Enterprise Data Manager.

**transaction data**

Transaction data contain the information within the business documents created using the master data, such as purchase orders, sales orders etc. Transactional Data can change very often and is not constant. Transaction data is created using ERP applications.

Transaction data is located in relational database table hierarchies. These hierarchies enforce top-down data dependencies, in which a parent table has one or more child tables whose data is linked together using primary key/foreign key relationships.

**translation column**

A column which is a candidate for a JOIN query on two or more Tables. This is applicable to Independent Archive, as Referential Data is backed up exclusively for XML based Archives and not Database archive, which contain only Transaction Data.

Note that such a scenario occurs when a Data Archive job execution extracts data from two Tables that are indirectly related to each other through Constrains on an intermediary Table.

**user**

Data Archive is accessible through an authentication mechanism, only to individuals who have an account created by the Administrator. The level of access is governed by the assigned system-defined roles.

**virtual view**

A virtual tables that includes SQL logic to convert the format of retired data.

# INDEX

## D

database  
support [10](#)

## E

Explorer pane [12](#)

## N

nodes [12](#)