

Oracle® Agile Product Lifecycle Management for Process
Extended Attribute Denormalization Guide

Extensibility Pack 2.7
E37242-01

September 2012

ORACLE®

Copyrights and Trademarks

Agile Product Lifecycle Management for Process

Copyright © 1995, 2012, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

OVERVIEW.....	4
Denormalization Process Overview	4
Available Extended Attributes for Denormalization	4
EXTENDED ATTRIBUTE DENORMALIZATION TABLES.....	5
Denormalization Metadata	5
Denormalization Data Tables.....	5
Numeric and Calculated Numeric EA Types.....	6
Free Text, Qualitative, and Qualitative Lookup EA Types.....	6
Date EA Types	8
Boolean EA Types.....	8
Quantitative Range EA Types.....	9
Quantitative Tolerance EA Types.....	9
Denormalization Logging	10
DENORMALIZED DATA FORMATS	10
Numeric and Date Nulls	10
Text Based Values	11
Base Unit of Measure Values	11
Qualitative Lookup Limitation.....	11
INSTALLATION.....	11
Installing the Scripts.....	11
EXECUTION	12
Execution Script	12
Performance Considerations	13

Overview

Extended Attribute Denormalization (EA Denorm) is a feature that provides the ability to convert the internal data storage of Extended Attributes into data structures that are easier to understand and report against while providing improved query performance.

The EA Denorm process pulls data for all activated (Active, Archive, and Inactive) Extended Attributes from Specifications (or other business objects, such as Sourcing Approvals, NPD Projects, etc), and populates that data into a new set of denormalization tables. These denormalization tables include additional information such as attribute IDs, custom section IDs, etc, that make the data easier to query against for reporting purposes.

Denormalization Process Overview

The EA Denorm process consists of running a provided stored procedure on a recurring basis, which processes each Extended Attribute set up for Denormalization, extracts the relevant data from it (where it is saved on a Specification, Sourcing Approval, etc) and populates that data into the relevant Extended Attribute denormalization table(s). This stored procedure may be run as an automated process on a predetermined interval (usually nightly) to extract data from Extended Attributes and Custom Sections, and populate it into the new denormalization tables. A log entry may optionally be written to a new database table to record the execution results of the Extended Attribute denormalization process.

Available Extended Attributes for Denormalization

Extended Attributes (EAs) are available for Denormalization if the Extended Attribute Template has been activated (Status is Active, Archived, or Inactive) and is marked as:

1. **Simple** - meaning it is contained within the Extended Attribute listing on the object (Spec/Facility/etc.), or
2. **Custom Section** – EA Templates within a Custom Section must be marked as *Distinct*. Non-distinct EAs in a Custom Section will not be included.

Attribute Configuration

Attribute Name: Adhesives 
Attribute ID: Adhesives
Distinct: ☒
Available in: Printed Packaging Specification
Class: Custom Sections, Simple
Tags:
Group(s): Printed Packaging
Status: Active
Security Classification:
Type: Qualitative

Extended Attribute Denormalization Tables

A new metadata table, `Denorm_EA_Templates`, will be used to control which extended attributes will be denormalized and drive the denormalization process. This table will be populated automatically by the main stored procedure included in the release package.

Individual Denormalization tables will be created for the various data types supported by the existing Extended Attribute types.

Denormalization Metadata

Extended Attribute denormalization metadata will be stored in a new table:

Table 1. `Denorm_EA_Templates`

ID	varchar(36)	Unique GUID
fkEATemplate	varchar(40)	containing a foreign key to the EA Template(CommonExtendedAttributeType table)
Status	int	Denorm status -1, 0, or 1
LastDenormTimestamp	datetime	Time of last Denormalization of this EA Template

Extended Attribute Types that are to be denormalized will be automatically added to this table using the provided SQL scripts. The `LastDenormTimestamp` value will be set for each EA when the denormalization process runs.

Denormalization Data Tables

New Denormalization tables are created for the various data types supported by the existing Extended Attribute types. Different EA Types with a common base data type, such as Text, will be stored in the same table. EA Types that contain a Unit of Measure field (UOM) will be denormalized to also include the Base UOM and base numeric value(s).

All Denorm tables will contain the following columns:

ID – unique GUID

fkOwner – foreign key to the application object that owns this Extended Attribute instance (e.g., GSM Specifications, Sourcing Approval, etc.)

fkExtendedAttributeTemplateID – foreign key to the EA TemplateID - same as the `Denorm_EA_Templates`

fkExtendedAttributeInstanceID – foreign key to the Individual Extended Attribute source table entry,

AttributeID – the attributeID value from the `CommonExtendedAttributeType` table

fkSectionTemplateID – foreign key to the Custom Section Template (commonEASectionTemplate), if populated from a Custom Section

fkSectionInstanceID – foreign key to the Custom Section instance (commonEASectionInstance), if populated from a Custom Section

SectionID – the Custom Section’s ID value from the commonEASectionTemplate table, if populated from a Custom Section

IsDistinct – Boolean identifier indicating if this EA has the IsDistinct tag

The following denormalization tables will be used based on the extended attribute type:

Numeric and Calculated Numeric EA Types

Table 2. Denorm_EA_Numeric

ID	varchar(36)
fkOwner	varchar(40)
fkExtendedAttributeTemplateID	varchar(40)
fkExtendedAttributeInstanceID	varchar(40)
AttributeID	varchar(24)
fkSectionTemplateID	varchar(40)
fkSectionInstanceID	varchar(40)
SectionID	varchar(24)
IsDistinct	bit
Value	float
UOM	varchar(10)
ValueBase	float
UOMBase	varchar(10)
IsCalculated	bit

Free Text, Qualitative, and Qualitative Lookup EA Types

Free Text, Qualitative, and Qualitative Lookup EAs will be denormalized into the following table. Note that multi-select items will be denormalized in both of the following ways:

1. As a comma-delimited list into the Value column
2. As individual rows into the Denorm_EA_Text_Multi table

Table 3. Denorm_EA_Text

ID	varchar(36)	
fkOwner	varchar(40)	
fkExtendedAttributeTemplateID	varchar(40)	
fkExtendedAttributeInstanceID	varchar(40)	
AttributeID	varchar(24)	
fkSectionTemplateID	varchar(40)	
fkSectionInstanceID	varchar(40)	
SectionID	varchar(24)	
IsDistinct	bit	
AttributeType	varchar(40)	Free Text, Qualitative, or Qualitative Lookup
IsMulti	bit	
Value	varchar(500)	

Table 4. Denorm_EA_Text_Multi

ID	varchar(36)
fkDenorm_EA_Text_ID	varchar(36)
Value	varchar(500)
ExternalID	varchar(80)
Sort Order	int

Date EA Types**Table 5. Denorm_EA_Date**

ID	varchar(36)
fkOwner	varchar(40)
fkExtendedAttributeTemplateID	varchar(40)
fkExtendedAttributeInstanceID	varchar(40)
AttributeID	varchar(24)
fkSectionTemplateID	varchar(40)
fkSectionInstanceID	varchar(40)
SectionID	varchar(24)
IsDistinct	bit
Value	datetime

Boolean EA Types**Table 6. Denorm_EA_Boolean**

ID	varchar(36)
fkOwner	varchar(40)
fkExtendedAttributeTemplateID	varchar(40)
fkExtendedAttributeInstanceID	varchar(40)
AttributeID	varchar(24)
fkSectionTemplateID	varchar(40)
fkSectionInstanceID	varchar(40)
SectionID	varchar(24)
IsDistinct	bit
Value	bit

1=true, 0 = false, NULL = not set

Quantitative Range EA Types**Table 7. Denorm_EA_QuantitativeRange**

ID	varchar(36)
fkOwner	varchar(40)
fkExtendedAttributeTemplateID	varchar(40)
fkExtendedAttributeInstanceID	varchar(40)
AttributeID	varchar(24)
fkSectionTemplateID	varchar(40)
fkSectionInstanceID	varchar(40)
SectionID	varchar(24)
IsDistinct	bit
Target	float
Min	float
Max	float
UOM	varchar(10)
TargetBase	float
MinBase	float
MaxBase	float
UOMBase	varchar(10)

Quantitative Tolerance EA Types**Table 8. Denorm_EA_QuantitativeTolerance**

ID	varchar(36)
fkOwner	varchar(40)
fkExtendedAttributeTemplateID	varchar(40)
fkExtendedAttributeInstanceID	varchar(40)

AttributeID	varchar(24)
fkSectionTemplateID	varchar(40)
fkSectionInstanceID	varchar(40)
SectionID	varchar(24)
IsDistinct	bit
Value	float
Tolerance	float
UOM	varchar(10)
ValueBase	float
ToleranceBase	float
UOMBase	varchar(10)

Denormalization Logging

Extended Attribute denormalization results may be logged to the DENORM_EA_LOG table. Each Extended Attribute Type is denormalized separately, and an entry indicating the number of records updated and inserted will be stored in this log table, along with a timestamp.

Any errors that occur will be logged with a value of “EA Denorm Error” in the MODULE column.

Denormalized Data Formats

This section describes how the Extended Attribute data is denormalized for different Extended Attribute types and values

Numeric and Date Nulls

There are two different ways that non-existent data is denormalized for numeric values and date-based extended attribute values:

1. **NULL:** A database NULL is stored when a custom section cell (row and column combination) has not been added to a Custom Section.
2. A PLM4P internal representation for a null value is added when an extended attribute has been added to a business object or a custom section cell has been added to a Custom section, but no data is entered.
 - a. The null representation for numeric values is **-1234567890**
 - b. The null representation for Date fields is **‘9999-12-31 00:00:00.000’**

Text Based Values

Text, Qualitative, and Qualitative Lookup Extended Attributes are denormalized using the English only values.

Multi-select values are denormalized as:

One record, in comma delimited format.
Individual rows into the Denorm_EA_Text_Multi table

Base Unit of Measure Values

When executing the EA Denorm process, the numeric values entered in the UI are denormalized, as are the conversion to the base Unit of Measure values. Likewise, Quantitative Range and Quantitative Tolerance Extended Attributes also can include Base UOM information for their Min, Max, and Target values.

Qualitative Lookup Limitation

When configuring a Qualitative Lookup in Data Admin, an *internal* PLM4P category, such as Allergens, can be selected. However, the Extended Attribute Denorm process currently only supports denormalization of the following *internal* lookup categories:

Countries
Additives
Allergens
Intolerances

Alternatively, external lookup categories, as available via Custom Lookups are fully supported.

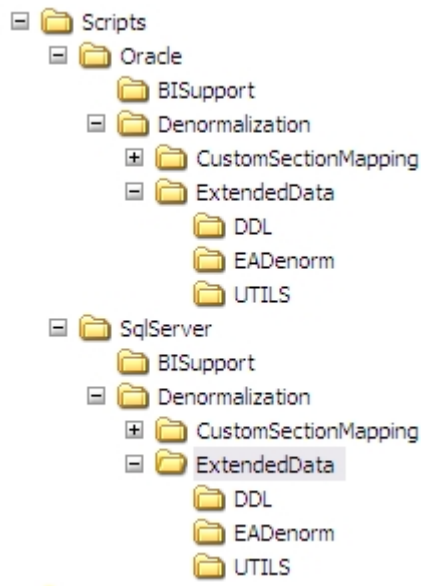
Installation

Several database scripts are provided as part of the Extensibility Pack release. The files include a script to generate the new tables used in the denormalization process, and individual stored procedures that handle the denormalization process.

All of these scripts must be executed in order to add the various stored procedures and functions to the database.

Installing the Scripts

Locate the Scripts directory in the release package. There are two different folders: Oracle and SqlServer. Open the folder that corresponds to the database provider you are using, then open the Denormalization folder, and the ExtendedData folder within.



All of the scripts in the ExtendedData folder must be compiled.

Note that an extension point has been added which allows clients to add custom tasks to the denorm process. This EA Denorm process calls the stored procedure named `sp_After_Internal_Denorm_EA` after it processes the core EA Denormalization. Clients may modify this stored procedure to add their own functionality if needed.

See the [Execution](#) section for details of the denormalization process.

Execution

Once the database scripts have been added to the database, the EA Denorm process is available for use.

Execution Script

The stored procedure that is used to execute the EA Denorm process is called **`sp_Denorm_EA`**.

This stored procedure can be run manually, or it can be scheduled to run on a recurring basis using the Database server tools (such as SQL Server Agent for SQL Server).

An optional parameter, `@log_level`, is used to indicate if logging to the `DENORM_EA_LOG` table should be enabled; a value of 1 will enable logging, 0 will disable logging.

The `sp_Denorm_EA` stored procedure does the following:

- Populates the `denorm_ea_templates` table with any Active, Archived, or Inactive extended attribute types
- Denormalizes each extended attribute type into its corresponding denormalization table and logs the results into the `DENORM_EA_LOG` table (if enabled).
 - Booleans
 - Dates

- Numerics and Calculated Numerics
- Quantitative Ranges
- Quantitative Tolerances
- Texts
- Qualitative Lookups
- Qualitatives

Deletes any denormalized records which no longer exist on the business object (Spec, Sourcing Approval, etc.)

Calls the `sp_After_Internal_Denorm_EA` stored procedure, which can be used by clients for any additional processing.

Performance Considerations

Before setting up denormalization, DBAs must understand the runtime characteristics of their routine. At a minimum, they need to understand how long the routine will run and what impact it will have on users. Database server hardware makes a *very significant* impact on runtime performance of the EA Denorm process.

The first run of denormalization for the Extended Attributes will take the longest time, but subsequent denormalization runs only pull in the changes since the last denormalization run.

