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

Feature Pack 4.3

E79161-01

May 2017

ORACLE®

Copyrights and Trademarks

Agile Product Lifecycle Management for Process

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

Preface

Audience

This guide is intended for client programmers involved with integrating Oracle Agile Product Lifecycle Management for Process. Information about using Oracle Agile PLM for Process resides in application-specific user guides. Information about administering Oracle Agile PLM for Process resides in the *Oracle Agile Product Lifecycle Management for Process Administrator User Guide*.

Variability of Installations

Descriptions and illustrations of the Agile PLM for Process user interface included in this manual may not match your installation. The user interface of Agile PLM for Process applications and the features included can vary greatly depending on such variables as:

- Which applications your organization has purchased and installed
- Configuration settings that may turn features off or on
- Customization specific to your organization
- Security settings as they apply to the system and your user account

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Software Availability

Oracle Software Delivery Cloud (OSDC) provides the latest copy of the core software. Note the core software does not include all patches and hot fixes. Access OSDC at:

http://edelivery.oracle.com.

Chapter 1—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 is required when using Oracle Product Lifecycle Analytics (OPLA).

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

Near Real-Time Denormalization

When a business object (such as a GSM Specification) is saved, the Near Real-Time Denormalization (NRTD) process will determine if any of the extended attributes in the simple EA listing, or any distinct extended attributes within custom sections on that item have been added, deleted, or modified. If so, a extended attribute denormalization request is added to a queue, which is then processed by a Service on the Remoting Container application. This Service will trigger denormalization for each request in the queue, which processes only that specific business object. Since the denormalization is only occurring for small sets of data, the NRTD Service can run on an ongoing basis, keeping the denormalized data very closely in synch with the live data.

The frequency of the NRTD Service can be configured, along with options for logging, email notifications if any errors occur, and more.

Batch Denormalization

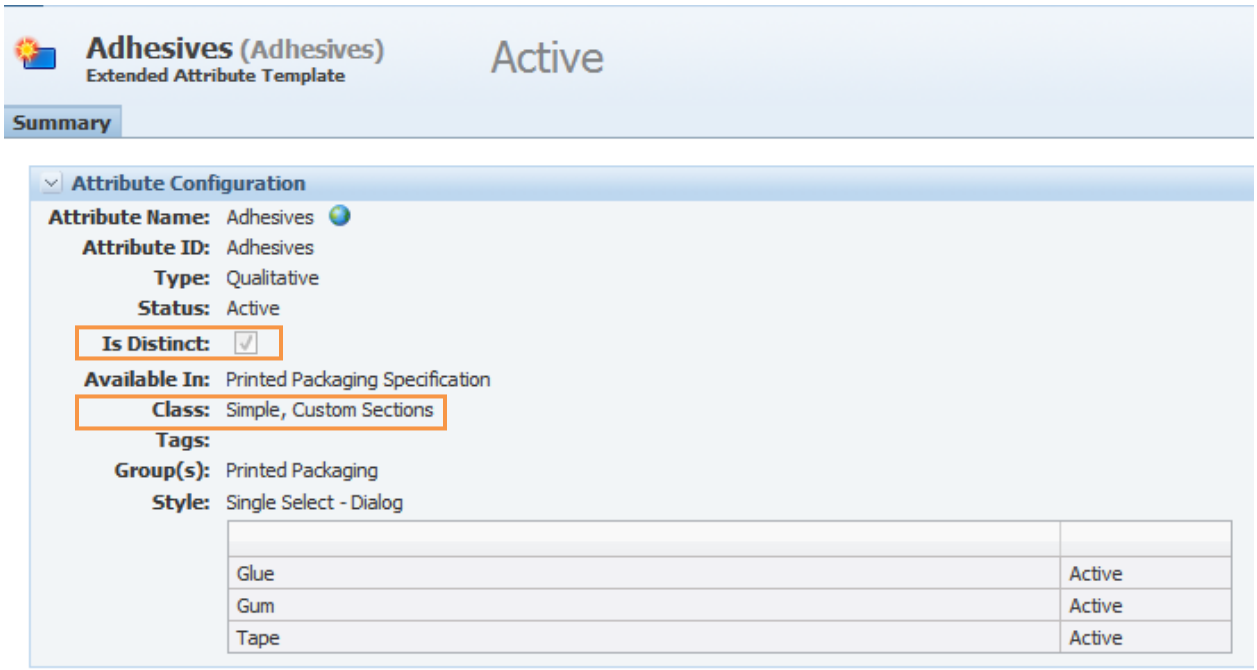
Prior to the Near Real Time Denormalization, denormalization was processed in a scheduled batch manner; all denormalized extended attributes for all business objects that have been updated since the last batch denormalization were processed together. While *this process should no longer be required if using NRTD*, there may be one-off instances where clients still wish to use this. As such, it is still available for use.

The batch 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 the Class is marked as:

1. **Simple** , *or*
2. **Custom Section** –EA templates within a custom section must be marked as *Distinct* to be included in the EA Denorm. Non-distinct EAs in a custom section will not be included.



Chapter 2—Extended Attribute Denormalization Tables

The Denorm_EA_Templates metadata table 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 extended attribute denorm 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 is stored in the DENORM_EA_TEMPLATES table. This table is used by the denorm process and is not needed when querying against the detail tables.

Column	DataType	Comments
ID	varchar(36)	Unique GUID
FKEATEMPLATE	char(40)	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 (used by the Batch Denorm process)
AttributeType	varchar(40)	extended attribute type
IsDistinct	bit	
UOMCategory	char(2)	
AttributeID	varchar(24)	
DexVersion	float	

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:

Column	Comments
ID	Unique GUID
fkOwner	Foreign key to the application object that owns this extended attribute instance (e.g., GSM Specifications, Sourcing Approval, etc.)

Column	Comments
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

DENORM_EA_NUMERIC

Column	DataType	Length
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	1
VALUE	float	
UOM	varchar	10
VALUEBASE	float	
UOMBASE	varchar	10
ISCALCULATED	bit	
PRECISION	int	

Free Text, Qualitative, and Qualitative Lookup EA Types

Free Text, Qualitative, Qualitative Lookup, and Calculated Text 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

DENORM_EA_TEXT

Column	DataType	Length
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	1
ATTRIBUTETYPE	varchar	40
ISMULTI	bit	1
VALUE	varchar	8000

DENORM_EA_TEXT_MULTI

Column	DataType	Length
ID	varchar	36
FKDENORM_EA_TEXT_ID	varchar	36
VALUE	varchar	500
EXTERNALID	varchar	80
SORTORDER	int	

Date EA Types

Denorm_EA_Date

Column	DataType	Length
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	1
VALUE	datetime	

Boolean EA Types

Boolean and Calculated Boolean EA types will be denormalized into the following tables:

DENORM_EA_BOOLEAN

Column	DataType	Length	
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	1	
VALUE	bit	1	1=true, 0=false, NULL = not set

Quantitative Range EA Types

DENORM_EA_QUANTITATIVERANGE

Column	DataType	Length
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	1
TARGET	float	
MIN	float	
MAX	float	
UOM	varchar	32
TARGETBASE	float	
MINBASE	float	
MAXBASE	float	
UOMBASE	varchar	32
TARGETPRECISION	int	
MINPRECISION	int	
MAXPRECISION	int	

Quantitative Tolerance EA Types

Denorm_EA_QuantitativeTolerance

Column	DataType	Length
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	1
VALUE	float	
TOLERANCE	float	
UOM	varchar	32
VALUEBASE	float	
UOMBASE	varchar	32
VALUEPRECISION	int	
TOLERANCEPRECISION	int	

Chapter 3—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.

Near Real Time Denorm Status

The Real Time Denorm process will log basic informational messages, as well as error messages, to a rolling file log in the Denorm_ExtendedAttributes directory within the default Logs directory. A new log file will be generated each day.

Extended Attribute Request Status

Near real time de-normalization requests in the DENORM_EA_REQUEST table are defined as follows:

Denorm Status	*Notes*
-1: Error	Errors notifications are sent via email and are logged to the event log
0: Pending	Items in this status will get processed during the next NRTD Service execution on the Remoting Container
1: Processing	Items are currently being processed
2: Completed	Denormalization completed. These records will be removed on the next execution of the CS and EA Cleanup Service.

The **MaxLogFilesToKeep_EA** configuration setting controls the number of daily log files that will be kept. To disable this file logging you must set the MaxLogFilesToKeep_EA value to "0" (zero). Note that errors will still be emailed and an entry in the ProdikaCommon Event Log will also be added. See the readme.txt file in Utilities\DenormServices for more details.

Chapter 4—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 stored 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**'

The EA Denorm process treats both scenarios as a null value, and therefore **denormalizes the data as NULL values** in the DENORM_EA_* tables.

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
- Complies With

- NPD Brands

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

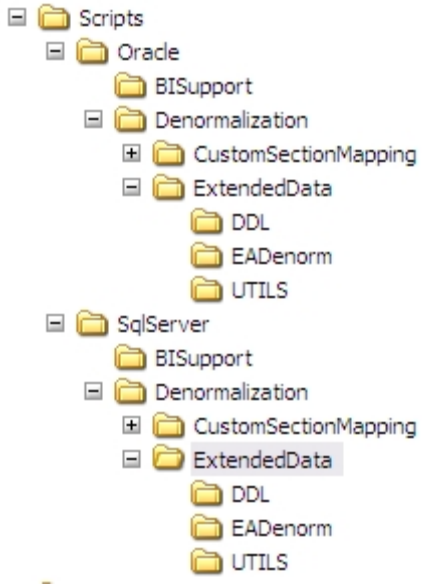
Chapter 5—Installation

Several database scripts are provided as part of the Feature 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.

Near Real Time Denorm Configuration and Installation

The Near Real Time Denormalization process is service that runs on the Remoting Container to process Extended attribute denormalization requests in near real time. Follow the instructions below for installation:

1. Stop IIS.

2. Stop the Remoting Container.
3. Enable Real Time Denorm by enabling the feature configuration entries:
   Add the following entries to the config\Custom\CustomerSettings.config file, in the FeatureConfig node:

```
<add key="Common.CustomData.NearRealTimeDenormRequest.ExtendedAttributes.Enabled"
value="true" configDescription="Evaluates if Extended Attribute data (including
Distinct EAs in a Custom Section) has been modified since load"/>
```

4. Add the following entry to the environmentvariables.config file, in the # Port Numbers section, replacing 8112 with a valid open/unused port:

   PLM4P.CSDenormService.Port = 8112

5. Customize the Real Time Denorm process.
   The following configuration entries allow for customization of the Real Time Denormalization process:

   <add key="PollingIntervalInSeconds_ExtendedAttributeRequests" value="90" />

   The frequency (in seconds) that extended attribute denorm requests are processed.

   <add key="AllowCustomSectionEAsInEADenorm" value="true" />

   Extended attribute denorm includes *distinct* extended attributes that are in custom sections. To exclude these EAs, change this value to "false"

   <add key="ErrorNotifyFromAddress" value="@@VAR:Prodika.From.EmailAddress@@" />

   <add key="ErrorNotifyToAddress" value="@@VAR:Prodika.To.EmailAddress@@"/>

   Errors in Real Time Denorm process will generate an email containing error details, along with additional useful information. These configuration settings are used to indicate the sender and the recipient of these emails. The default setting will use the values entered in environmentvariables.config, but can be modified here to use different values.

   <add key="MaxLogFilesToKeep_EA" value="30"/> <!-- set to 0 to disable file logging -->

   The Real Time Denorm process will log basic informational messages, as well as error messages, to a rolling file log in the Denorm_ExtendedAttributes directory within the default Logs directory. A new log file will be generated each day. These configuration settings control the number of daily log files that will be kept.

   To completely disable file logging, set the MaxLogFilesToKeep_EA value to "0" (zero). Note that errors will still be emailed (as discussed above) and an entry in the ProdikaCommon Event Log will also be added.

   <add key="DaysForProcessedEARequestExpiration" value="15"/>

These settings are used to clean up (remove) old, successfully processed denormalization requests (from the denorm_ea_request table) after the configured number of days. Set to zero to never remove the denorm request entries.

CUSTOMIZATIONS:

To make any customizations to the above Real Time Denorm configurations, copy the following and add it to the CustomerSettings.config file, in the CustomerSettings/Core node.

```
<DenormRequestProcessingServices>
    <AppSettings configChildKey="key">
    </AppSettings>
</DenormRequestProcessingServices>
```

Then copy the desired entries from above into the AppSettings node and modify the settings as needed.

6. Enable Real Time Denorm processing for each object type.

Update the \config\Extensions\CustomPluginExtensions config file to enable Real Time Denorm processing for each object type, inserting the following entries into the ValidatePlugins node.

```
<Plugin name="NearRealTimeDenormRequestPlugin"
FactoryURL="Class:Xeno.Prodika.DenormServices.DenormChangePluginFactory,DenormServices" />

<Plugin name="PostSaveSpecPlugin" inheritFromPluginName="NearRealTimeDenormRequestPlugin"
/> <!-- Enable Real Time Denorm Request processing for GSM Specs -->

<Plugin name="PostSavePQMItemPlugin"
inheritFromPluginName="NearRealTimeDenormRequestPlugin" /> <!-- Enable Real Time Denorm
Request processing for PQM Items -->

<Plugin name="PostSaveSCRMPlugin" inheritFromPluginName="NearRealTimeDenormRequestPlugin"
/> <!-- Enable Real Time Denorm Request processing for SCRM items -->

<Plugin name="PostSaveNPDProjectPlugin"
inheritFromPluginName="NearRealTimeDenormRequestPlugin" /> <!-- Enable Real Time Denorm
Request processing for NPD Projects -->

<Plugin name="PostSaveSmartIssuePlugin"
inheritFromPluginName="NearRealTimeDenormRequestPlugin" /> <!-- Enable Real Time Denorm
Request processing for Smart Issue Requests -->
```

7. Copy the DenormConfig.xml file to the RemotingContainer\bin directory.
8. Add the following entry to EnvironmentSettings.config in the EnvironmentSettings/RemotingContainer/RemoteServices node, making sure to set the isActive attribute to true.

```
<Service name="Custom Section Near Real Time Denorm Service"
port="@@VAR:Prodika.CSDenormService.Port@@" isActive="true" />
```

9. Make a backup of the existing RemotingContainer.exe.config file in Apps\RemotingContainer\bin and then replace the current version with the one in this directory.

10. Copy the DenormServices.dll file from the SharedLibs directory and place it in the following folders:
    - RemotingContainer\dependentAssemblies
    - web\gsm\bin
    - web\pqm\bin
    - web\scrm\bin
    - web\npd\bin

11. Restart IIS.

12. Restart the Remoting Container.

To verify the service is running, you can check the Logs\(Denorm_CustomSections and Logs\Denorm_ExtendedAttributes directories, or the ProdikaCommon Event log. Once the Remoting Container is started, the denorm requests get processed after the configured number of seconds specified in the config.

Chapter 6—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 Batch 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).

Executing the sp_Denorm_EA stored procedure is not required when using the Near Real Time Denorm process. However, it should be called for an initial run, to start the denormalization for existing data.

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 and Calculated Booleans
    - Dates
    - Numerics and Calculated Numerics
    - Quantitative Ranges
    - Quantitative Tolerances
    - Texts and Calculated 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 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.