

Oracle® Endeca Information Discovery

Migration Guide

Version 2.3.0 • June 2012 • Revision A

Copyright and disclaimer

Copyright © 2003, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

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 is software or related documentation that 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 END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware 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.

Rosette® Linguistics Platform Copyright © 2000-2011 Basis Technology Corp. All rights reserved.

Teragram Language Identification Software Copyright © 1997-2005 Teragram Corporation. All rights reserved.

Table of Contents

| | |
|--|-----------|
| Copyright and disclaimer | ii |
| Preface | v |
| About this guide | v |
| Who should use this guide | v |
| Conventions used in this guide | v |
| Contacting Oracle Customer Support | v |
| Chapter 1: Upgrading to Oracle Endeca Information Discovery Version 2.3.x | 1 |
| Required reading | 1 |
| High-level upgrade procedure | 1 |
| Upgrading to the Oracle Endeca Server | 2 |
| About upgrading client stubs | 2 |
| Upgrading Integrator | 3 |
| Running the upgraded graphs | 4 |
| Upgrading Studio | 4 |
| Requirement to use the same context root | 4 |
| Stopping and uninstalling the Windows service | 4 |
| Backing up your Studio 2.2 files | 4 |
| Installing Studio 2.3.x and restoring backups | 5 |
| Reconfiguring existing components after the upgrade | 5 |
| Chapter 2: Required Changes | 7 |
| General required changes | 7 |
| Changes to product names and terminology | 7 |
| Required changes in the Oracle Endeca Server | 9 |
| Changed syntax for accessing the WSDL documents | 9 |
| Upgrading clients to use versioned Web services | 10 |
| Handling secure Web service requests | 11 |
| Required changes in Integrator | 11 |
| Required changes for existing Integrator graphs | 11 |
| Required changes to the Endeca Query Language | 12 |
| Query Language name change | 12 |
| Analytics support removed | 12 |
| Required changes in Studio | 13 |
| Studio data sources moved to the database (as of 2.3) | 13 |
| About this change | 13 |
| Importing your existing data source files into the database | 13 |
| New datastoreName parameter for Studio data sources (as of 2.3) | 14 |
| Chart component replaced (as of 2.3) | 14 |
| Changes to JSON serialization and parsing (as of 2.3) | 14 |
| New web service for parsing EQL (as of 2.3) | 16 |

| | |
|--|-----------|
| Chapter 3: Behavioral Changes | 17 |
| Behavioral changes in the Oracle Endeca Server and its interfaces | 17 |
| Obtaining the version number of the Oracle Endeca Server | 17 |
| Changed syntax for accessing the Dgraph process flags | 17 |
| Changes to the administrative and configuration URL commands | 18 |
| Web services compliance with the wrapped-document/literal pattern of binding | 19 |
| Syntax for specifying outer transaction ID | 19 |
| Improvements in the behavior of outer transactions | 19 |
| Changes to the Transaction Web Service | 20 |
| Changes to the Administration Web Service | 21 |
| Changes to the Configuration Web Service | 21 |
| Changes to the Conversation Web Service | 22 |
| Behavioral changes in Studio | 23 |
| Component edit view is now full screen (as of 2.3) | 23 |
| Change to the Studio default user (as of 2.3) | 23 |
| Behavioral changes in EQL | 24 |
| Change to syntax of returned time values | 24 |
| Trailing semi-colon now allowed | 24 |
| Changes to EXTRACT, ROUND, and TRUNC behavior | 24 |
| Change to BETWEEN/AND behavior | 24 |
| Sorting of managed attributes in ORDER clauses | 24 |
| Managed attribute aliasing | 24 |
| Conversion between integer and Boolean values | 24 |
| Controlling sort order | 25 |
| Filtering behavior change | 25 |
| Using filters in WHERE statements | 25 |
| Using COALESCE statements with mixed types | 25 |
| Quoting of literals | 25 |
| Behavior of queries without RETURN statements | 26 |
| Behavior of multi-assign attributes in SELECT statements | 26 |
| Accessing an ancestor at a particular level of hierarchy | 26 |
| Behavior of hierarchy level 0 | 26 |

Preface

Oracle® Endeca Information Discovery is an enterprise data discovery platform for advanced, yet intuitive, exploration and analysis of complex and varied data.

Information is loaded from disparate source systems and stored in a faceted data model that dynamically supports changing data. This integrated and enriched data is made available for search, discovery, and analysis via interactive and configurable applications.

Oracle Endeca Information Discovery enables an iterative “model-as-you-go” approach that simultaneously frees IT from the burdens of traditional data modeling and supports the broad exploration and analysis needs of business users.

About this guide

This guide helps you upgrade your Oracle Information Discovery implementation by describing the major changes between versions 2.2.x and 2.3.x.

Who should use this guide

This guide is intended for system administrators and developers who are upgrading Oracle Endeca Information Discovery on Windows or Linux.

For Studio, this guide is intended for users whose Studio applications are built using standard Studio components. It does not necessarily apply to Studio applications built with custom components.

Conventions used in this guide

This guide uses the following typographical conventions:

Code examples, inline references to code elements, file names, and user input are set in `monospace` font. In the case of long lines of code, or when inline monospace text occurs at the end of a line, the following symbol is used to show that the content continues on to the next line: `¬`

When copying and pasting such examples, ensure that any occurrences of the symbol and the corresponding line break are deleted and any remaining space is closed up.

Contacting Oracle Customer Support

Oracle Customer Support provides registered users with important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at <https://support.oracle.com>.



Chapter 1

Upgrading to Oracle Endeca Information Discovery Version 2.3.x

This section provides instructions for upgrading to Oracle Endeca Information Discovery 2.3.x. Before you start the upgrade processes, check the remaining sections of this guide to learn about the changes that will affect you during or after an upgrade.

Required reading

High-level upgrade procedure

Required reading

In addition to reading this document, it is recommended that you read the following documents for important information about the release.

Release Announcement

The Release Announcement outlines the new features that were added in Version 2.3.x.

Release Notes

The Release Notes for each package provide information about known issues and bug fixes for this release.

Installation Guide

The *Oracle Endeca Information Discovery Installation Guide* provides an overview of the components and includes information about configuration scenarios. It contains installation instructions for each component, and information on how to verify your installation.

High-level upgrade procedure

This topic provides the recommended order for upgrading to version 2.3.x.

The upgrade process assumes you need to install the components on dedicated servers. Because of this, the upgrade process references installation instructions for each component, and does not utilize the All-in-One Installer. The All-in-One installer is intended for single-server development environments only, and should be used with the Quick Start project.

The upgrade process includes separate upgrade procedures for the Oracle Endeca Server, Integrator, and Studio. In addition, as part of the upgrade, you need to make changes to the existing pre-upgrade graphs so that you can run these graphs in Integrator after the upgrade, and load data and configuration to the Oracle Endeca Server data store.

The following procedure provides high-level steps for the entire upgrade process. See the individual topics in this section for detailed instructions in each step.

To upgrade to Oracle Endeca Information Discovery release version 2.3.x:

1. Upgrade to the Oracle Endeca Server.

As a result of this step, you should have the Oracle Endeca Server installed and started. You may also have an empty Endeca data store that you created to verify the installation.

2. Upgrade Integrator.

As a result of this step, you should have Integrator installed and pointing to the workspace that you used for your version 2.2.x Integrator graphs.



Note: Do not run any existing graphs in Integrator before you make changes to them.

3. In Integrator, make changes to the graphs you have created in the previous release.
4. Run the upgraded graphs.
5. Upgrade Studio.

[Upgrading to the Oracle Endeca Server](#)

[Upgrading Integrator](#)

[Upgrading Studio](#)

Upgrading to the Oracle Endeca Server

This topic describes how to upgrade to the 2.3.x version of Oracle Endeca Server.

To upgrade to the Oracle Endeca Server:

1. Uninstall the previous version (in the previous release, the version is known as the MDEX Engine 2.2.x), using instructions from the Installation Guide for the release you have installed. If required, first uninstall the Cluster Coordinator.
2. Install the Oracle Endeca Server, using instructions from the *Oracle Endeca Information Discovery Installation Guide* version 2.3.x. If required, install the Cluster Coordinator.

After you are finished, you should have a running Oracle Endeca Server. Optionally, you may have created an Endeca data store (with no source data loaded) to verify that the installation was successful.

[About upgrading client stubs](#)

[Upgrading clients to use versioned Web services](#)

About upgrading client stubs

In this release, each Web service packaged with the Oracle Endeca Server includes a version consisting of major and minor numbers, for example 1.0, where 1 is a major version number and 0 is a minor version

number. Therefore, if you are planning to issue requests via stubs that utilize any of the Web services, you must generate new client stubs from the currently supported versions of each Web service.



Note: You only should generate new client stubs if you are planning to use direct requests to the Oracle Endeca Server utilizing any of its packaged Web services. Studio and Integrator use the supported versions of each Web service, therefore no special procedures apply in this case.

For information on changes that took place for each of the Web services, see the chapters in this guide.

For information on versions of Web services supported in this release, see the topic in the *Oracle Endeca Information Discovery Installation Guide*.

For information on how Web service versions are assigned and how to avoid version incompatibility, see the section on Web service versions in the *Oracle Endeca Server Developer's Guide*.

Upgrading Integrator

To upgrade Integrator, you uninstall the existing version and install the new one. You can still use your existing workspace after you install Integrator version 2.3.x.

If you have custom plug-ins deployed to Integrator 2.2, you should back up the plug-ins before beginning the Integrator upgrade. You can reload them after the upgrade.

To upgrade Integrator:

1. Uninstall version 2.2.x. For information, see the *Latitude Installation Guide*, version 2.2.x.
After you uninstall, the workspace you used for creating your Latitude graphs should remain in place on your machine.
2. Install Integrator version 2.3.x. For information, see the *Oracle Endeca Information Discovery Installation Guide*, version 2.3.x.
3. Start Integrator and point it to use the workspace you used for your version 2.2.x graphs.

Now you have upgraded to the new version of Integrator. However, do not run your existing graphs until you have made changes to them that are required for Oracle Endeca Information Discovery, version 2.3.x.

After the upgrade, you can proceed to making changes to your existing Integrator graphs, as described in the topic [Required changes for existing Integrator graphs on page 11](#).

If you are reloading a custom plug-in, keep in mind that the installation path has changed. For example, on Windows the 2.2.x default installation path has changed from:

```
C:\Endeca\Latitude\2.2.2\DataIntegrator
```

to this 2.3.x default path:

```
C:\Oracle\Endeca\Discovery\Integrator\2.3.0
```

Because Integrator would still be referencing the `customcomponents.xml` at the previous location, it will have to be updated to the new 2.3.x location.

[Running the upgraded graphs](#)

Running the upgraded graphs

After you have made changes to the existing graphs, you can run them in Integrator.

After running the graphs and troubleshooting any errors, you can upgrade Studio, and proceed to working with your records using Studio.

Upgrading Studio

After completing the upgrades to the Endeca Server and Integrator, you then upgrade Studio. These instructions are for upgrading to Studio 2.3.x.

Requirement to use the same context root

[Stopping and uninstalling the Windows service](#)

[Backing up your Studio 2.2 files](#)

[Installing Studio 2.3.x and restoring backups](#)

[Reconfiguring existing components after the upgrade](#)

Requirement to use the same context root

The upgrade process assumes that you are using the same installation path and URL. You cannot change the context root during the upgrade process.

Stopping and uninstalling the Windows service

If your current instance of Studio is running as a Windows service, then before you can start the upgrade, you must first stop and uninstall the service.

To stop and uninstall the Studio Windows service:

1. To stop the service, from the **Services** list, right click the service, then click **Stop**.
2. To uninstall the service:
 - (a) From the command line, navigate to the Studio Tomcat bin directory.
`endeca_portal\tomcat-6.0.29\bin\`
 - (b) Run the following command:
`service.bat uninstall`

Backing up your Studio 2.2 files

Before installing Studio 2.3.x, you must back up your database and other customized files from Studio 2.2.

To back up your Studio 2.2 files:

1. Stop your Studio server. If you were running Studio as a Windows service, you should already have stopped and uninstalled the service.
2. Back up your database as follows:
 - If you are using HSQL, skip this step. Your database will be backed up as part of step 4 below.

- If you are using a different database, follow the backup procedures from your vendor.

3. Back up any Studio customizations you have made, such as database connectivity options in `portal-ext.properties`.
4. Back up the entire `endeaca-portal\data` directory to a safe backup location.
If you have changed the location of anything normally kept in `endeaca-portal\data`, such as your JCR repository or Lucene search indexes, back up your custom location(s) as well.
5. After backing up the files, rename the `endeaca-portal` directory.
This ensures that the Studio 2.3.x installation will not overwrite any of the existing files.
You will remove this directory when you complete the Studio 2.3.x installation.

Installing Studio 2.3.x and restoring backups

You next install Studio 2.3.x. After installing, you use your backed-up files to restore your data and custom settings. You delete the directory containing your previous version, then start Studio 2.3.x.

To complete the upgrade to Studio 2.3.x:

1. Install Studio 2.3.x, following the steps in the *Oracle Endeca Information Discovery Installation Guide*. Do not start the server.
2. Restore the entire `endeaca-portal\data` directory from your safe backup location. Overwrite the `endeaca-portal\data` files that were installed with Studio 2.3.x.
If you have changed the location of anything normally kept in `endeaca-portal\data`, such as your JCR repository or Lucene search indexes, restore and re-verify your custom locations as well.
3. Restore your database, as follows:
 - If you are using HSQL, skip this step, since your database was restored as part of step 2.
 - If you are using MYSQL, DB2, or some other RDBMS, your database should still be intact. If it is not, follow the restore procedures from your vendor.
4. Configure Studio to connect to your database.
If you are not using HSQL, you should have created a backup of your database connection strings earlier.
5. Delete the renamed directory containing your earlier version of Studio.
6. Place the Studio license file in the `endeaca-portal\deploy` directory.
7. Start Studio 2.3.x.

Before you start to use the application, make sure that you wait until all of the components have been deployed and registered. To verify that the upgrade is truly complete, you can monitor the log files to see when all activity has stopped.

Reconfiguring existing components after the upgrade

Upgrading to Studio 2.3.x may cause your Studio components to lose some or all of their entire configuration. The components will still display on your pages, but may for example revert to default settings.

After you complete the upgrade, review all of your components and reconfigure them as needed to restore your preferred settings.



Chapter 2

Required Changes

You must make the changes listed here if they apply to your application.

[*General required changes*](#)

[*Required changes in the Oracle Endeca Server*](#)

[*Required changes in Integrator*](#)

[*Required changes to the Endeca Query Language*](#)

[*Required changes in Studio*](#)

General required changes

This section describes required changes that apply to the entire Oracle Endeca Information Discovery software product.

Changes to product names and terminology

In this release, the names of the product and components have changed. In addition, several terminology changes took place to reflect changes in the product capabilities. This topic provides a mapping between the names and terms from the previous release and the names and terms introduced in this release.

Product names

| New name | Previously used name | Notes |
|-------------------------------------|----------------------|--|
| Oracle Endeca Information Discovery | Endeca Latitude | The Oracle Endeca Information Discovery package includes Studio, Integrator, and the Oracle Endeca Server package. |

| New name | Previously used name | Notes |
|----------------------|----------------------|---|
| Oracle Endeca Server | Endeca MDEX Engine | <p>The Oracle Endeca Server package is included into the Oracle Endeca Information Discovery package. The Oracle Endeca Server package can also be used on its own.</p> <p>The term "Oracle Endeca Server" refers to the following items:</p> <ul style="list-style-type: none"> • The name of the product package • The Java-based process that enables you to control data stores |

Product component names

| New name | Previously used name | Notes (if applicable) |
|----------------------------------|---|--|
| Studio | Endeca Latitude Studio | |
| Integrator | Latitude Data Integrator (LDI) | <p>The term "Integrator" refers to the following items:</p> <ul style="list-style-type: none"> • The name of the entire Integrator package • The client-side portion of the Integrator package (previously known as the LDI Designer) <p> Note: The server-side portion of Integrator is referred to as "Integrator Server".</p> |
| Endeca Query Language (EQL) | Latitude Query Language | |
| Content Acquisition System (CAS) | Endeca Content Acquisition System (CAS) | |

Operational model terminology

| New name | Previously used name | Notes (if applicable) |
|----------------------|----------------------|---|
| Oracle Endeca Server | Did not exist | <p>When used in the context of a process running inside installed software, Oracle Endeca Server is a Java-based utility for creating and administering the Endeca data stores.</p> <p>Only one Oracle Endeca Server can be installed on a machine.</p> |

| New name | Previously used name | Notes (if applicable) |
|----------------------------------|----------------------|--|
| Endeca data store, or data store | MDEX Engine instance | <p>A discrete set of data, representing particular data records. Includes system records and indexed data records for the particular data store.</p> <p>To create an Endeca data store, run the <code>create-ds</code> command of the Oracle Endeca Server. This command creates the data store, attaches it to the server, and starts the Dgraph process for that data store.</p> <p>You can later attach and detach the same data store, as well as start and stop it.</p> |
| Dgraph process | MDEX Engine | <p>Each Endeca data store has a Dgraph process that handles requests made to that store.</p> <p>For example, if you have used the Oracle Endeca Server to create three Endeca data stores and they have all been started, then three Dgraph processes are running on that machine.</p> <p> Note: The term "MDEX Engine" is no longer used for any component or entity in the product set.</p> |

Required changes in the Oracle Endeca Server

This section describes required changes in the Oracle Endeca Server.

Changed syntax for accessing the WSDL documents

The syntax for accessing the WSDL documents for each of the Web services should specify the host and port of the Oracle Endeca Server and the name of the data store.

You can access a WSDL for a specific Web service using the following syntax in your URL request:

```
http://host:port/ws/WebServiceName/DataStore?wsdl
```

where:

- host and port point to the location of the Oracle Endeca Server
- WebServiceName is the name of the specific Web service, such as `conversation`, `admin`, `config`, `ingest`, or others.
- DataStore is the name of your Endeca data store.



Note: In the previous release, the host and port were pointing to the location of the Dgraph process. In this release, the Dgraph process must be running on the same machine as the Oracle Endeca Server, but they must have different ports.

Upgrading clients to use versioned Web services

Starting with the Oracle Endeca Server version 2.3.0, each packaged Web service is assigned its own version number consisting of a major and minor version. After you install the Oracle Endeca Server, you must update your client-side code to the supported versions of each interface.

For a list of packaged Web services and their supported versions, see the Oracle Endeca Server portion of the *Oracle Endeca Information Discovery Installation Guide*.

To obtain a supported version of a Web service, issue a request to the service to see its WSDL document, as shown in this example for the Configuration Web Service:

```
http://host:port/ws/config/DataStore?wsdl
```

where the host and port represent the Oracle Endeca Server, config is the name of the Configuration Web Service, and the DataStore is the name of the data store created on the server.

The WSDL document for each Web service includes a namespace that lists the Web service's version, as shown in this example for the Administration Web Service:

```
xmlns:v1_0="http://www.endeca.com/MDEX/admin/1/0"
```

In this example, the string 1/0 indicates the version as 1.0, where 1 is the major version, and 0 is the minor version.

If more than one minor version is supported, all of them are listed, as in this example for the Transaction Web Service, which supports versions 1.0 and 1.1:

```
xmlns:v1_0="http://www.endeca.com/MDEX/transaction/1/0"
xmlns:v1_1="http://www.endeca.com/MDEX/transaction/1/1"
```

Note that, while minor versions are backward-compatible, changes to major versions are not backward-compatible.

In the Endeca Latitude version 2.2.x, all packaged Web services did not contain version numbers in the format shown above. As a result, if you issue client requests originating from the Web services in the previous release, these requests are rejected by the Oracle Endeca Server due to a version mismatch.

Therefore, after installing the Oracle Endeca Server version 2.3.0, you must regenerate your client stubs from the supported versions of the Web services and also update your client code to utilize the most recent versions of each Web service. (If you do not use any client code generated from stubs and only use Integrator or Studio, no action is necessary since Studio and Integrator components designed to work with the Oracle Endeca Server already utilize the most recent supported versions of the Oracle Endeca Server Web service interfaces.



Note: If you are upgrading from an interface without a version to an interface with a version (this is the case when upgrading from the Endeca MDEX Engine version 2.2.x to the Oracle Endeca Server version 2.3.0), the initial upgrade to the client stubs requires changing all import statements in your client code, similar to the following example. If, before the upgrade, the import statement in the client code looked similar to this example:

```
import com.endeca.www.mdex.transaction._2011.startOuterTransactionDocument;
```

change the import statement to indicate the major and minor versions for the service's operation:

```
import com.endeca.www.mdex.transaction._1._0.startOuterTransactionDocument;
```

The namespaces for each operation indicate which versions are supported for this operation. In this case, the startOuterTransaction operation is supported as of version 1.0.

For more information on Web service versions, see the *Oracle Endeca Server Developer's Guide*.

Updating your client code includes but is not limited to using client stubs generated from the most recent versions of a particular Web service. Read this guide to be aware of other changes to the Web services that took place in this release, such as changed or new operations, handling of HTTPS requests, handling of an outer transaction ID, and compliance with the wrapped-document/literal pattern of binding for a web service. You may need to update your client code based on these changes.

Handling secure Web service requests

Depending on the protocol in the web service request, the Oracle Endeca Server issues `http` or `https` response.

In previous releases, if the Dgraph process of the Oracle Endeca Server was running over HTTPS, the `?wsdl` operation returned an incorrect URL. The work-around was to manually specify the service endpoint URL in your client.

In this release, if the Web service request uses HTTPS, the Oracle Endeca Server recognizes it properly and issues the response using HTTPS. For those requests that use HTTP, an HTTP response is issued.



Note: The front-end clients of the Oracle Endeca Server that do not wish this behavior, (for example, because they have used a proxy between the front-end application server and the Oracle Endeca Server), may override it by adding `?https=true`, or `?https=false` to their WSDL query requests, as in this example:

```
http://localhost:<port>/ws/config/DataStore?wsdl?https=true
```

If you use the override as shown in this example, then regardless of the type of request sent to the server, it is always returned as `https`.

Required changes in Integrator

This section describes required changes in Integrator.

Required changes for existing Integrator graphs

This topic lists changes you should make in existing graphs before attempting to run them after you upgrade to Integrator version 2.3.x.

Configuring components for the Endeca Server

In version 2.2.x, the Endeca-developed ingest components (such as the **Bulk Add/Replace Records** component) were configured with the hostname and port of the MDEX Engine.

In version 2.3.x, you configure these components with the hostname and port of the Endeca Server. In addition, you also specify the name of the Endeca data store. In general, it is a good idea to replace the 2.2.x component with the 2.3.x versions, rather than updating the components in place.

Changes in the namespace of the Configuration Web Service

Change the namespace in the **Request Structure** of those components that use calls to the Configuration Web Service, such as the **WebServiceClient** component.

The namespace of the Configuration Web Service has changed to the following:

```
<config-service:configTransaction
  xmlns:config-service="http://www.endeca.com/MDEX/config/services/types/1/0"
  xmlns:mdex="http://www.endeca.com/MDEX/XQuery/2009/09">
```

This means that if you used the **WebServiceClient** for sending requests through the Configuration Web Service, you need to change the way the **Request Structure** is specified. Use the following example:

```
<config-service:configTransaction
  xmlns:config-service="http://www.endeca.com/MDEX/config/services/types/1/0">
<config-service:putConfigDocuments
  xmlns:mdex="http://www.endeca.com/MDEX/XQuery/2009/09">
<mdex:configDocument name="recsearch_config">
<RESEARCH_CONFIG>
  $xmlString
</RESEARCH_CONFIG>
</mdex:configDocument>
</config-service:putConfigDocuments>
</config-service:configTransaction>
```

In this example, the line:

```
xmlns:config-service="http://www.endeca.com/MDEX/config/services/types/1/0"
```

correctly identifies the namespace.

Required changes to the Endeca Query Language

This section describes required changes to the Endeca Query Language (or EQL).

Query Language name change

The Endeca Query Language (or EQL) was known as the Latitude Query Language (or LQL) in version 2.2.x. EQL is part of the Oracle Endeca Server package.

Query Language documentation change

For general information about using EQL, see the *Oracle Endeca Server Query Language Reference*, which is new with this release.

Analytics support removed

With the introduction of EQL, the former Analytics language is no longer supported.

Requests containing `Analytics` and `AnalyticsConfig` complex types will return a SOAP fault from the Oracle Endeca Server, whether they are sent through Studio components or by the Conversation Web Service directly.

Required changes in Studio

For Studio, the following changes are required.

Studio data sources moved to the database (as of 2.3)

As of version 2.3, Studio has changed how data sources are managed and stored.

About this change

Previously, the Studio data sources were created as JSON files, which were stored in the `endeca-data-sources` directory.

As of version 2.3, the data sources are stored as JSON strings in the portal database.

The **Data Sources** component in Studio is now used to create, edit, and delete data sources. The `endeca-data-sources` directory is only used to store the Java Keystore files for connecting to a secured Endeca Server.

Importing your existing data source files into the database

After you install Studio, you must migrate your existing data source files into the database.

To migrate your data sources:

1. Stop Studio if it is running.
2. Place the data source files in the `endeca-data-sources` directory.
3. Start Studio.

The data sources are loaded into the database.

When Studio finishes loading a data source, it appends ".processed" to the data source file name.

If the data source included an `id` parameter, the parameter is removed from the file and used as the data source ID.

If the data source did not include an `id` parameter, then the file name is used as the data source ID.

4. To verify that the data sources have been loaded, display the **Data Sources** component.

The data sources should be included in the list.

5. Click the **Edit** button for each data source, and verify that the ID and definition are correct.

After you migrate your data sources, you will also need to edit them to add the new `datastoreName` parameter. For details, see [New `datastoreName` parameter for Studio data sources \(as of 2.3\) on page 14](#).

New datastoreName parameter for Studio data sources (as of 2.3)

In the data source definition, the `server` and `port` settings now connect to the Endeca Server. Because an Endeca Server can connect to multiple data stores, the data source definitions must now specify the data store to use.

After you migrate your data sources using the steps outlined in [Importing your existing data source files into the database on page 13](#), you need to edit each data source to add the `datastoreName` parameter.

For example:

```
{  
  "server": "server01.lab.acme.com",  
  "port": "15000",  
  "datastoreName": "acmeDB"  
}
```

Chart component replaced (as of 2.3)

The **Chart** component has been completely redesigned and rewritten. See the release announcement for more details.

When you upgrade, any existing charts in your Studio application will no longer work. They will display an error message indicating that the component has been undeployed.

You must remove the existing components from your pages, and then recreate them by adding and configuring the new **Chart** component.

For details on using and configuring the new **Chart** component, see the *Oracle Endeca Information Discovery Studio User's Guide*.

Changes to JSON serialization and parsing (as of 2.3)

As of 2.3.x, we have made several changes to how JSON is serialized and parsed.

If you created custom extensions using the `MDEXStateManager`, `MDEXSecurityManager`, or `QueryFunctions`, or have built custom components, you will need to update them.

The changes are:

- Previously, the `QueryFunction` serialization and parsing framework required a string version of setters for any non-string properties. This is no longer required. If any setters are left with both string and other data type overloads, JSON deserialization will break.
- Jackson annotations can now be used to customize JSON output for custom `QueryFunctions`.
- The following interfaces have been removed:
 - `com.endeca.portal.data.functions.QueryFunction.getInstance`
 - `com.endeca.portal.data.functions.QueryFunction.toJSON`
- Built-in `QueryFunctions` no longer set their content in the `PassThroughElements` in the Conversation Service.
- The default `MDEXSecurityManager` has been updated to adjust how it serializes and parses `QueryFunctions`.

- Previously, the JSON produced by serialization converted all values to Strings regardless of their type.

So an object with a boolean getter or an int getter would return "prop": "true" or "prop": "42", instead of "prop": true or "prop": 42.

The new `QueryFunctionUtils` can parse JSON of the form "prop": "true" into an object with a boolean setter. This could potentially affect the following `QueryFilter`-generated JSON:

- `ExposeRefinement` - `dimExposed`, `exposeAll`
- `RefinementFilter` - `isNavigable`
- `SearchFilter` - `isSearchWithin`, `enableSnippeting`, `snippetLength`



Note: This does not mean that bookmarks, deep links, or data source definitions will become invalid.

- In JSON serializations, `RecordDetailsConfig` must have both `attributeKey` and `attributeValue` properties.

To reflect these changes, you will need to update your custom `QueryFunctions`. As a reminder, make sure you load all custom `QueryFunctions` into the global classloader.

The required changes are:

1. Remove the string setters for non-string properties. To do this, either:

- Delete the string overload, or
- Add `@org.codehaus.jackson.annotate.JsonIgnore` to any methods that include both String and the actual datatype.

For example:

```
void setIsShown(boolean shown) { ... }
void @JsonIgnore setIsShown(String shown) { ... }
```

2. Remove all `getGetters` and `getSetters`.

3. For `com.endeca.portal.data.functions.QueryFunction.getInstanceMove`, move customizations to:

```
com.endeca.portal.data.functions.utils.QueryFunctionUtils.fromString/fromJSONObject
```

For `com.endeca.portal.data.functions.QueryFunction.toJSON`, move customizations to:

```
com.endeca.portal.data.functions.utils.QueryFunctionUtils.toString/toJSONObject
```

Update any custom `MDEXStateManagers` that relied on these removed `QueryFunction` interfaces.

4. Update any customized versions of the `MDEXSecurityManager` to reflect the change to the serialization and parsing of `QueryFunctions`.

5. Instead of scanning the `com.endeca.mdex.conversation.Results` object's passthrough elements for filters, look in `MDEXState/DataSource.getQueryState()`.

New web service for parsing EQL (as of 2.3)

The existing Presentation API, including the navigation .jar file, has been removed and replaced with a parsing web service.

If you have created custom EQL-based components that depend on `AnalyticsQuery` or that use `ENEQuery` code paths, they may no longer work.



Chapter 3

Behavioral Changes

The changes listed here do not require any action on your part, but will affect how your application behaves after you upgrade.

[Behavioral changes in the Oracle Endeca Server and its interfaces](#)

[Behavioral changes in Studio](#)

[Behavioral changes in EQL](#)

Behavioral changes in the Oracle Endeca Server and its interfaces

This section describes behavioral changes to the Oracle Endeca Server and its interfaces.

Obtaining the version number of the Oracle Endeca Server

Use the `endeca-cmd version` command of the Oracle Endeca Server to obtain the version of the Oracle Endeca Server and the versions of the Dgraph processes powering each of the data stores (if the Dgraph processes for these data stores are currently running).

In this release, the Oracle Endeca Server utility is used to manage the data stores and the Dgraph processes for each of the data stores.

You do not need to specify the name of the data stores when using the `endeca-cmd version` command.

The following example illustrates an output of the `endeca-cmd version` command that is run on the Oracle Endeca Server in which `datastoreA` is running and `datastoreB` has been created but is stopped:

```
Oracle Endeca Server version 2.3.0, build 403
Data store information:
  datastoreA: version 2.3.0, build 632019
  datastoreB: -- No version available --
```

Notice that if a data store is not running on the Oracle Endeca Server, the version of its Dgraph process is not available and not returned in this command.

For more information on the commands used with the Oracle Endeca Server, see the *Oracle Endeca Server Administrator's Guide*.

Changed syntax for accessing the Dgraph process flags

In the previous release, you started the Dgraph process directly and thus access to the Dgraph process arguments (flags) was provided on the command line of the machine on which the process was started.

In this release, you interact with the Oracle Endeca Server, which in turn manages any of the data stores you need to create and, correspondingly, their Dgraph processes. Therefore, the way to access the Dgraph process arguments for a particular data store has changed.

To obtain a list of the Dgraph process flags, use either of the following commands:

- `endeca-cmd create-ds --args --usage`
- `endeca-cmd attach-ds --args --usage`



Note: After you have obtained a list of possible Dgraph process flags, you can specify any of them for the data store with the `--args` option of `create-ds` or `attach-ds` commands. The `--args` flag must be the last flag on the command line as all of its arguments are passed on to the Dgraph process, as in the following example:

```
endeca-cmd create-ds books --args --threads 6 --net-timeout 60
```

For more information on the Oracle Endeca Server and its commands, see the *Oracle Endeca Server Administrator's Guide*.

Changes to the administrative and configuration URL commands

In this release, two administrative URL commands have been removed. In addition, all administrative and configuration URL requests should be sent to the host and port of the Oracle Endeca Server. The requests should also specify the name of the running data store. This topic provides the details.

The syntax of administrative URL commands has changed to the following:

```
http://<host>:<port>/admin/<datastore>?op=<supported-operation>
```

The syntax of configuration URL commands has changed to the following:

```
http://<host>:<port>/config/<datastore>?op=<supported-operation>
```

where:

- `<host>` refers to the hostname or IP address of the Oracle Endeca Server.
- `<port>` refers to the port on which the Oracle Endeca Server is listening.
- `<datastore>` refers to the name of the Endeca data store on which the command will operate.



Note: If you are using HTTPS mode, use `https` in the URL.

In addition, the following administrative URL commands have been removed:

- `/admin/?op=exit`. This command is removed. It is replaced by the Oracle Endeca Server command that lets you stop the data store, `stop-ds`. For more information on the `stop-ds` command, see the *Oracle Endeca Server Administrator's Guide*.
- `/admin/?op=rollback`. This command is removed. It is replaced by the `rollBackOuterTransaction` operation of the Transaction Web Service. For more information on this operation, see the *Oracle Endeca Server Developer's Guide*.

Web services compliance with the wrapped-document/literal pattern of binding

All Web services packaged with the Oracle Endeca Server are compliant with the wrapped-document/literal pattern of binding.

Syntax for specifying outer transaction ID

For all web service requests, the syntax for specifying the ID of the outer transaction, if one is in progress, has changed. Instead of an attribute, a new element, `OuterTransactionId`, must be specified as the first element in the request.

The `outerTransactionID` attribute is removed and is replaced with the `OuterTransactionId` element.

In the previous release, to run within an outer transaction, any Web service request had to specify an outer transaction ID as an attribute in the web service's operation. In this release, the outer transaction ID must be specified as the first element in the request, as follows:

```
<ingest:clearMdex>
<OuterTransactionId>MyID</OuterTransactionId>
</ingest>
```

Improvements in the behavior of outer transactions

This topic discusses behavioral changes related to how the Oracle Endeca Server processes queries during outer transactions.

Summary of changes

- **Continuous query is supported while an outer transaction is in progress.** In previous releases, while the node processed an outer transaction, it was locked for all incoming queries for its duration. In this release, you can group various updating requests and run them within an outer transaction on a single Oracle Endeca Server. While updates are running within the outer transaction, non-updating queries continue to be served by the same server against the pre-transaction version of the data files. Once the outer transaction is committed (it can be also rolled back if any of its inner transactions fail), the queries start being processed against the published version of the data files.

This behavior is also supported in a clustered environment. In previous releases, when an outer transaction was in progress on the leader node, all nodes except the leader node continued to serve query requests, while the leader node was locked processing updates within an outer transaction. In this release, while an outer transaction is in progress on the leader node, continuous query is supported on the leader node as well.

- **Updating queries without an outer transaction ID are not rejected but wait for the completion of the outer transaction.** In the previous release, for any updating query to be processed if an outer transaction was running, it had to specify the transaction ID. Those updating queries that didn't specify the ID of the currently running outer transaction were rejected by the node that was processing the outer transaction.

In this release, if you send an updating query without the outer transaction ID while the outer transaction is in progress, the query waits until the transaction closes (is committed or rolled back), and is processed against the version of the data files that is published when the transaction closes.

If a transaction is still open when the Dgraph process tries to exit, it is rolled back automatically. This allows any pending updates that were waiting for the transaction to close to be processed before exit — those updates are not applied; they are returned with a 503 HTTP code `service unavailable`.



Note: The `admin?op=updateASpell` operation is rejected if it is issued during an outer transaction and does not specify its ID. In addition, updating requests from the Bulk Ingest interface are rejected if an outer transaction is in progress and these requests do not specify the correct outer transaction ID.

- **Changes to the configuration documents are supported in the context of outer transactions.** In the previous release, you could not send any configuration documents to the Oracle Endeca Server while an outer transaction was in progress (requests to change the configuration are updating requests from the Configuration Web Service). In this release, while an outer transaction is being processed, you can update the configuration, such as the text search configuration. For example, any changes to the configuration of search interfaces, as well as to `dimsearch_config`, `en_word_forms_collection`, `recsearch_config`, `rerank_strategies`, `stop_words`, and `thesaurus` configuration documents are possible while an outer transaction is in progress. If the correct `OuterTransactionId` is specified in these requests, the requests run within the transaction; otherwise, they wait for the duration of the outer transaction and then are processed against the version that is published once the transaction is committed or rolled back.

For more information on outer transactions and query processing, see the section about the Transaction Web Service in the *Oracle Endeca Server Developer's Guide*.

Changes to the Transaction Web Service

Names of operations in the Transaction Web Service have changed, two new operations have been added, and an attribute specifying the outer transaction ID was replaced by an element. This topic discusses the details.

Changes to how the outer transaction ID is specified

The `outerTransactionId` attribute is removed and is replaced with the `OuterTransactionId` element.

This change in the Transaction Web Service affects all other Web service requests — to run within an outer transaction, any Web service request has to specify the correct outer transaction ID as the first element in the request.

Changed and new operations

In this release, the following operations changed their names:

| Name in the previous release | New name |
|---|-------------------------------------|
| <code>StartTransactionOperation</code> | <code>startOuterTransaction</code> |
| <code>CommitTransactionOperation</code> | <code>commitOuterTransaction</code> |

The following new operations have been added:

| Operation | Description |
|--------------------------|--|
| ListOuterTransaction | An operation to request an ID of a running outer transaction. If an outer transaction is in progress, this operation returns its ID. |
| RollBackOuterTransaction | An operation to roll back an outer transaction with the ID specified in the OuterTransactionIdToRollBack element. |

Changes to the Administration Web Service

Using operations from the Administration Web Service, you can list and cancel jobs. The names of these operations have changed in this release. The updated version of the Administration Web Service contains the new operations. They are also described in this topic.

To make a job monitoring request for a list of jobs that are currently running in the Oracle Endeca Server or pending in the queue, use the `listJobsOperation` of the Administration Web Service.

When you use the `listJobsOperation` in your request, the response contains a list of currently running or pending jobs, and includes the following information:

- Job ID
- Job type
- Job queued time
- Whether it is running
- Whether it is canceled

If the job is not running, it can be canceled. To cancel a job that has been scheduled but have not yet run, obtain a list of jobs that are either running or waiting in the queue, and then issue the `cancelJobOperation` of the Administration Web Service.

For more information on job monitoring, see the *Oracle Endeca Server Administrator's Guide*.

Changes to the Configuration Web Service

In this release, the read-only version of this service has been removed. A new operation, `updateGroupConfigs`, has been added. In addition, one namespace in this service has been changed.

In this release, the read-only version of the Configuration Web Service has been removed. The full-featured version of the service is available. In addition to updating configuration operations, it includes all read-only operations.

In the previous release, the read-only version was used by Latitude Studio only. In this release, Studio does not use it. Any other front-end clients of the Oracle Endeca Server should use the full-featured version of the service, since this version includes both updating and non-updating operations.

A new operation, `updateGroupConfigs` has been added. It allows you to update assignments on the group description record. In particular, this operation lets you change the display name of a group, including letting

you change the display name of the `system-navigation_InternalGroup` (this is the group that contains all attributes that do not belong to any user-configured groups).

One of the namespaces used by the Configuration Web Service has been changed. In the previous release, the following namespace was used:

```
xmlns:mdex="http://www.endeca.com/MDEX/XQuery/2009/09"
```

In this release, the namespace changed as follows (the string `config` has been added):

```
xmlns:mdex="http://www.endeca.com/MDEX/config/XQuery/2009/09"
```

For more information on the Configuration Web Service, see the *Oracle Endeca Server Developer's Guide*.

Changes to the Conversation Web Service

The Conversation Web Service has been changed in this release. The notable changes are listed in this topic.

- The `PropertyGroupList` content element is replaced by the new `AttributeGroupList` content element. The `PropertyGroupListConfig` element is replaced by the new `AttributeGroupListConfig` element. Similar to `PropertyGroupListConfig` in the previous release, the new element `AttributeGroupListConfig` is an extension of `ContentElementConfig` and provides information about attribute groups.
- The `RefinementConfig` complex type that controls the behavior of groups of attributes in guided navigation has an attribute `Expose`. Its default value has changed to `false`, meaning that by default, no refinements are exposed under the currently chosen attribute, and only the root value for the attribute is exposed. To expose all possible refinement values, change the value to `true`.
- In this release, the Conversation Web Service returns hierarchy information (managed attribute values) for `RecordList` and `RecordDetails`.
- The complex type `Property` has been updated and now returns `PropertyRecord` and `DimensionRecord`. `PropertyList` content element has been updated, and now returns records that are PDRs and DDRs.
- A new operator pair (`RecordKindOperator`,`PopRecordKindOperator`) has been added to the Conversation Web Service. The `RecordKind` simple type has been added. Its enumeration values are restricted to `data`, `nodata`, `properties`. The `RecordKindOperator` operator takes the string value `data` for all records that represent user-added source records, `properties` for all records that define PDRs, or `nodata` for all system records, such as PDRs, DDRs, GCRs, groups, precedence rules, and semantic entity definitions. Record kinds are not "visible" in the front-end application, and thus do not appear in breadcrumbs, and are not cleared by any of the general guided navigation mechanisms for clearing the state. The mechanism for filtering the kinds of records is used by Studio.
- Several changes have been made to extend the Conversation Web Service to support the Endeca Query Language (EQL): a new schema `lql_parser_types.xsd` has been added to provide support for `lql_parser` Web Service types, such as joins, "group by" operations and other EQL expressions. EQL filters in XML format have been added (`DataSourceFilter` and `SelectionFilter`). Also, new types, `LQL` and `LQLConfig`, have been added for handling the evaluation of arbitrary EQL statements. These statements can be submitted either as string or as XML.

For more information on how to use the Conversation Web Service to issue queries, see the *Oracle Endeca Server Developer's Guide*.

For more information on the Conversation Web Service WSDL, see the *Oracle Endeca Server API Reference*.

For more information on the Endeca Query Language, see the *Oracle Endeca Server Query Language Reference*.

Behavioral changes in Studio

For Studio, you should be aware of the following behavioral changes.

Component edit view is now full screen (as of 2.3)

Previously, when power users selected the **Preferences** option to display the component edit view, the edit view took up the same space as the end user view.

To allow more space for the component configuration options, the edit view now takes up the entire Studio work area. When you are configuring a component, you cannot see other components.

Change to the Studio default user (as of 2.3)

Previously, when a new instance of Studio was first installed, the default user name and password were test@endeca.com/test.

As of 2.3, for a new Studio instance, the name and password for the default user are admin@oracle.com/admin.

Note that if you are upgrading an existing instance, Studio does not make any changes to your security. It does not add a new user or update any existing users.

After you install a new instance of Studio, for additional security, you should either:

- Change the password for the default user
- Create a new administrative user, then remove the default user

Behavioral changes in EQL

This section describes behavioral changes in the Endeca Query Language (or EQL).

Change to syntax of returned time values

In EQL, time values that are returned now contain a trailing Z.

Trailing semi-colon now allowed

A trailing semi-colon on the last statement in an EQL query is no longer an error.

Changes to EXTRACT, ROUND, and TRUNC behavior

The behavior of the EXTRACT, ROUND, and TRUNC functions has changed in EQL 2.3.

- BROUND and BTRUNC are no longer supported.
- Both one and two argument ROUND and TRUNC functions with integer and double (as well as the DATETIME version of TRUNC) arguments are supported.

In LQL 2.2, some EXTRACT keywords (QUARTER, DAY_OF_WEEK, DAY_OF_YEAR, WEEK) returned 0-based values while others (MONTH, DAY_OF_MONTH) returned 1-based values. In EQL 2.3, all of these EXTRACT keywords return 1-based values. January is now month 1 rather than month 0.

Change to BETWEEN/AND behavior

BETWEEN/AND are now inclusive of both endpoints.

Sorting of managed attributes in ORDER clauses

In EQL 2.3, managed attributes in ORDER clauses are sorted case insensitively by primary key (also called record spec), with case sensitive sort breaking any resulting ties.

In LQL 2.2, managed attributes in ORDER clauses to be ordered case insensitively by name.

To approximate 2.2 behavior during ingest, create records for each managed attribute value, join them, and then sort by the joined display name.

Managed attribute aliasing

In EQL version 2.3, the ability to alias managed attributes has been added.

Conversion between integer and Boolean values

In EQL 2.3, integer and Boolean values require an explicit conversion.

In earlier versions, the conversion between the two types was automatic.

To approximate the old behavior, use the function `TO_INTEGER(Boolean)`, which casts 1 for TRUE and 0 for FALSE.

Controlling sort order

The way you control sort order has changed in EQL 2.3.

In LQL 2.2, using a managed attribute in an `ORDER` clause caused records to be ordered case-insensitively by attribute name.

In EQL 2.3, ordering of strings is case-sensitive. To impose a sort order, first create a column specifically for ordering. In that column, do any necessary work such as case folding or diacritic folding, and then order by that column.

Filtering behavior change

In version 2.3, EQL supports filtering on arbitrary, Boolean expressions, as long as the final value is Boolean. This includes filtering on Boolean columns.

You can place any function or Boolean value after a `WHERE` or `HAVING` clause, including a per-aggregate `WHERE` clause.

In earlier versions, filtering had a separate, more limited syntax.

Using filters in WHERE statements

In EQL version 2.3, you must enclose filters used in `WHERE` statements in parentheses.

In earlier versions of LQL, parentheses around filters were optional.

Using COALESCE statements with mixed types

In LQL 2.2, you could apply `COALESCE` to a mixture of integers and doubles. Integers would be promoted to doubles as needed.

In EQL 2.3, applying `COALESCE` to a mixture of integers and doubles results in a type error.

Quoting of literals

In earlier versions of LQL, all literals, whether string or numeric, could be placed in single quotation marks.

In EQL 2.3, literals are treated differently, depending on type:

- String literals may be placed within single quotation marks.
- Numeric and Boolean literals must be unquoted.
- Literals of structured types (such as Date, Time, and Geocode) must use appropriate conversions, as shown in the following example:

```
RETURN Result AS
SELECT  TO_GEOCODE(45.0, 37.0) AS Geocode,
        TO_DATETIME('2011-11-21T08:22:00Z') AS Timestamp
```

Behavior of queries without RETURN statements

In LQL 2.2, a query consisting only of `DEFINE` statements produced an empty result.

In EQL version 2.3, such a query results in an error during EQL checking.

Behavior of multi-assign attributes in SELECT statements

In LQL 2.2, using a multi-assign attribute in a `SELECT` statement would implicitly remove the multi-assign aspect of its values. This was particularly apparent when grouping by that column.

For example, if the `Multi` attribute contained the values `first`, `second`, `third` on a given record, then the following expression:

```
RETURN Results AS
SELECT Multi AS Multi
GROUP BY Multi
```

would arbitrarily select one of `first`, `second`, `third` and place the record only into that group. In contrast, the following would map the record into the groups for all three values:

```
RETURN Results AS
SELECT 1 as dummy
GROUP BY Multi
```

In EQL 2.3, `SELECT` preserves multi-assign for grouping, so the record would appear in the groups for all three values in both examples above.

Accessing an ancestor at a particular level of hierarchy

The way you access an ancestor at a particular level of hierarchy has changed.

In EQL 2.3, you can use a new set of row functions, including `ANCESTOR`, `HIERARCHY_LEVEL`, `IS_DESCENDANT`, `IS_ANCESTOR`, and `GET_LCA`.

In earlier versions, you would use the syntax `managedAttr:level` to access an ancestor at a specified level of hierarchy.



Note: `GROUP BY` statements still need to use the syntax `managedAttr:level`, because you cannot group by an expression. Level 0 now refers to the root.

Behavior of hierarchy level 0

The behavior of hierarchy level 0 with both managed and unmanaged attributes has changed in EQL 2.3.

Using hierarchy level 0 with managed attributes

When working with a managed attribute, the level of hierarchy of interest can be specified using a colon-integer suffix on the hierarchy name. For example, `WineType:1` indicates those values immediately below the root of the `WineType` hierarchy.

In earlier versions of LQL, `:0` behaved the same as not specifying a suffix. As of EQL 2.3, `:0` specifies the root of the hierarchy. To replicate the old behavior, do not specify a hierarchy level.

Using hierarchy level 0 with unmanaged attributes

In LQL 2.2, specifying a :0 level for an unmanaged attribute was ignored. As of EQL 2.3, this is a type error.