Oracle® Endeca Server

Migration Guide

Version 7.6.1 • December 2013

ORACLE

Copyright and disclaimer

Table of Contents

Preface

Oracle® Endeca Server is a hybrid search-analytical engine that organizes complex and varied data from disparate sources. At the core of Endeca Information Discovery, the unique NoSQL-like data model and in-memory architecture of the Endeca Server create an extremely agile framework for handling complex data combinations, eliminating the need for complex up-front modeling and offering extreme performance at scale. Endeca Server also supports 35 distinct languages.

About this guide

This guide helps you upgrade your Oracle Endeca Server implementation by describing the major changes between versions 7.5.x and 7.6.x.

Who should use this guide

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

Conventions used in this guide

The following conventions are used in this document.

Typographic conventions

This table describes the typographic conventions used when formatting text in this document.

Typeface	Meaning
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
`Code Sample`	This formatting is used for sample code phrases within a paragraph.
Variable	This formatting is used for variable values. For variables within a code sample, the formatting is *`Variable`*.
`File Path`	This formatting is used for file names and paths.

Symbol conventions

This table describes the symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

Path variable conventions

This table describes the path variable conventions used in this document.

Path variable	Meaning
`$MW_HOME`	Indicates the absolute path to your Oracle Middleware home directory, which is the root directory for your WebLogic installation.
`$DOMAIN_HOME`	Indicates the absolute path to your WebLogic domain home directory. For example, if `endeca_server_domain` is the name of your WebLogic domain, then the `$DOMAIN_HOME` value would be the `$MW_HOME/user_projects/domains/endeca_server_domain` directory.
`$ENDECA_HOME`	Indicates the absolute path to your Oracle Endeca Server home directory, which is the root directory for your Endeca Server installation.

Contacting Oracle Customer Support

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

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

Chapter 1

Upgrading to Oracle Endeca Server Version 7.6.1

This section provides instructions for upgrading to Oracle Endeca Server 7.6.1. 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

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 Endeca Server version 7.6.x.

Release Notes

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

Installation Guide

The *Oracle Endeca Server Installation Guide* contains installation instructions, and information on how to verify your installation.

Upgrading to 7.6.1 from 7.5.x

This topic describes how to upgrade to the 7.6.1 version of Oracle Endeca Server.

This upgrade procedure assumes that you have both WebLogic 10.3.6 and Oracle ADF (Application Development Framework) Runtime package 11.1.1.6 installed, as well as Endeca Server 7.5.x. The upgrade procedure will not remove the WebLogic and ADF software, but will remove the WebLogic domain you created for the 7.5.x Endeca Server. Because you will manually install Endeca Server 7.6.1, this procedure does not use the 7.6.1 orchestration script. Please keep in mind that 7.5.x Endeca data domains cannot run under Endeca Server 7.6.1.

Note: If you are upgrading an Endeca Server cluster, follow the steps for uninstalling the Endeca Server instances on each machine, remove the Endeca Server WebLogic domain, and then remove the Endeca Server data domains.

To upgrade to Oracle Endeca Server 7.6.1 from Oracle Endeca Server 7.5.x:

1. Uninstall Endeca Server version 7.5.x.
   Use the uninstall instructions from the *Oracle Endeca Server Installation Guide* (version 7.5.x.).
2. Manually remove the WebLogic domain for Endeca Server.
   Use the uninstall instructions from the *Oracle Endeca Server Installation Guide* (version 7.5.x.).
3. Install Oracle Endeca Server 7.6.1. As part of the installation, make sure you create a new WebLogic domain for the 7.6.1 version of Endeca Server.
   Use the instructions from the chapter titled "Manually Installing Endeca Server on a Single Machine", in the *Oracle Endeca Server Installation Guide* (version 7.6.x.). If you are installing in a cluster, use the "Installing and Deploying an Endeca Server Cluster", in the *Oracle Endeca Server Installation Guide* (version 7.6.x).
   After you are finished, you should have a running Oracle Endeca Server.
4. Install the ADF Interim Patch for Bug 17509664.
   Use the instructions from the chapter titled "Applying the ADF patch", in the *Oracle Endeca Server Installation Guide* (version 7.6.x.).

To verify that the upgrade was successful, create an Endeca data domain (with no source data loaded).

Upgrading to 7.6.1 from 7.6.0

This topic describes how to upgrade from version 7.6.0 of Oracle Endeca Server.

If you are running version 7.6.0 of Endeca Server, you have two options in upgrading to version 7.6.1:

- Use the upgrade procedure described in *Upgrading to 7.6.1 from 7.5.x on page 6*. The disadvantage is that you have to re-create your WebLogic domain. The advantage is that you can specify a new name for the Endeca Server home directory (such as "EndecaServer7.6.1").
- Use the procedure described in this topic. The disadvantage is that you must specify the same name for the Endeca Server home directory as was used in 7.6.0 (because of pathnames specified in the Endeca Server configuration file). The advantage is that you can re-use the 7.6.0 WebLogic domain (including any 7.6.0 Endeca data domains).

This upgrade procedure assumes that you have both WebLogic 10.3.6 and Oracle ADF (Application Development Framework) Runtime package 11.1.1.6 installed, as well as Endeca Server 7.5.x. The upgrade procedure will not remove the WebLogic and ADF software, nor will it remove the WebLogic domain you created for the 7.6.0 Endeca Server. Because you will manually install Endeca Server 7.6.1, this procedure does not use the 7.6.1 orchestration script.

To upgrade to Oracle Endeca Server 7.6.1 from Oracle Endeca Server 7.6.0:

1. Shut down WebLogic Server.
2. Write down the name of the Endeca Server installation directory (for example, "EndecaServer7.6.0"). You will be using this same name for the 7.6.1 installer.

3. Deinstall your current 7.6.0 Endeca Server, using the instructions in Chapter 8 of the 7.6.1 *Oracle Endeca Server Installation Guide*.
   Use the appropriate deinstaller in the `$MW_HOME/EndecaServer7.6.0/oui/bin` directory: `runInstaller` for Linux or `setup.exe` for Windows.

   

   Note: Do not delete the WebLogic domain you created for your 7.6.0 installation, located in the `$MW_HOME/user_projects/domains` directory. You will be re-using this domain for the 7.6.1 installation.

4. Go to the WebLogic `$DOMAIN_HOME/servers/AdminServer/tmp` directory and delete all of its files and sub-directories. This will remove the cached versions of Endeca Server artifacts (such as the Endeca Server EAR file).
5. Install the Endeca Server 7.6.1 software by using the instructions in Chapter 5 ("Manually Installing Endeca Server on a Single Machine") of the 7.6.1 *Oracle Endeca Server Installation Guide*.
   Note these two important requirements:
   (a) If you were previously using SSL, make sure that you also select "Yes" at the **Select Deploy Mode** screen. Otherwise uncheck the box.
   (b) At the **Specify Installation Location** screen, use the same location (from Step 2) in the Oracle Home Directory field.
6. When the Endeca Server installation finished, do NOT create a WebLogic domain. You will re-use the pre-existing WebLogic domain.
7. Start WebLogic Server.

Because you are re-using the pre-existing WebLogic domain, you do not have to regenerate the SSL certificates (if you are using secure mode) or re-register the Data Enrichment plugins. You can also re-use the Endeca data domains you created with the 7.6.0 Endeca Server.

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 3.0, where 3 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.

To obtain documentation on Web service topics:

- 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 Server 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*.

Chapter 2

Required Changes

This section describes required changes in the Oracle Endeca Server.

Terminology changes

The following table lists the concepts and terms that have changed in this Endeca Server release compared with the previous release.

New term	Description
collection	Starting with the Endeca Server 7.6.0 and Studio 3.1.0, Endeca Server introduces collections (known in Studio as data sets), while keeping the ability to create entities (views). ***Collections*** let you logically group records into buckets in the data domain index. The Entity and Configuration Web Service (sConfig) is used for the purpose of creating schema records for collections and entities. In the previous release, views offered a way of grouping records. Endeca Server had a representation for them as entities, through the Entity Configuration Web Service (sConfig). In the Endeca Server 7.6.x and Studio 3.1.x, both entities (views) and collections (data sets) are provided—views let you group records in Studio at the user interface level, while collections are a way of grouping records in the index of the Endeca data domain hosted by the Endeca Server. Collections are then represented in Studio as data sets within an Endeca Server connection that you configure. When initially loading records, you can load them directly into collections, within a single data domain. You can then filter records from records that belong to different collections, using Conversation Web Service queries and its filter rules. Note that in Studio, both views (which represent entities in the Endeca Server) and data sets (which represent collections in the Endeca Server) can be used, depending on the user scenarios and data. For information on how to create collections, see the *Oracle Endeca Server Developer's Guide*.
filter rule	In the Endeca Server, ***filter rules*** let you apply a filter across attributes from many collections at once. You create filter rules after creating collections, using the Entity and Configuration Web Service (sConfig). When you filter a collection by an attribute from a specific filter rule, other collections are refined by other attributes from the same filter rule. For example, when refining a Sales collection by the sales date, the Twitter collection is refined by the post date. You can apply filter rules to different types of filters, such as selection filters (used in refinements) and EQL filters (used for range filters). In Studio, a filter rule is known as a refinement rule. It lets you apply a filter across many data sets at once, thus allowing users to connect attributes from different data sets in an Endeca Server connection. For information on how to create filter rules, see the *Oracle Endeca Server Developer's Guide*.

New term	Description
EQL set (or set)	In EQL (Endeca Query Language), multi-assign attributes from the corpus are represented as sets. A ***set*** consists of a group of elements, typically derived from the values of a multi-assign attribute. For a summary of changes in EQL for sets, see *Endeca Query Language changes on page 18* in this guide. For information on working with sets in EQL, see the *Oracle Endeca Server EQL Guide*.

Security-related changes

In this release, the following security-related changes took place.

- Endeca Server supports SSL v.3.0 and TLS V1.0, as the WebLogic Server.
- The `--sslcipher` Dgraph flag is removed. Instead of using this flag, when installed securely over SSL, Endeca Server by default supports a list of approved SSL/TLS ciphers for encryption of its messages between the Endeca Server and the Dgraph, and does not support any others. In particular, the following ciphers and cipher methods are disabled: EXPORT ciphers, ciphers with no authentication (aNULL), and no encryption (eNULL), MD5 hash functions and RC4.

For detailed information about security in the Endeca Server, see the *Oracle Endeca Server Security Guide*.

Changes to endeca-cmd

The `endeca-cmd` command-line script has been enhanced in this release.

The following table highlights changes to the command's options:

Options in release 7.5.x	Changes in release 7.6.x
`endeca-cmd update-spelling-dictionaries <data-domain> [global-options]`	This option is removed (along with the equivalent operation of the Manage Web Service, which is also removed). It is replaced with the `updateSpellingDictionaries` operation of the Data Ingest Web Service. Use soapUI or Integrator to issue this operation to the Endeca Server. For information, see the *Oracle Endeca Server Data Loading Guide*. See *Changes to the Data Ingest Web Service on page 17*.
Does not exist in 7.5.x. Was added for 7.6.x.	New: `endeca-cmd put-dd-profile <new-profile-name> [global-options] --search-char-limit`. The `--search-char-limit` is a new, optional parameter that you can specify for the data domain profile. It sets the maximum length (in characters) of a search term for record and value searches. The default is 132 characters. Any term exceeding this length will not be indexed, and thus will not be found in record and value searches. See *Data domain changes on page 23*.

Options in release 7.5.x	Changes in release 7.6.x
Does not exist in 7.5.x. Was added for 7.6.x.	New: When you create a new data domain profile with `endeca-cmd put-dd-profile <new-profile-name> [global-options] [create-options]`, the list of `create-options` includes the following new options: • `--auto-idle <bool>`, and `--idle-timeout <int>`. These options support the configuration of data domains that are configured to automatically idle if during the idle timeout they do not receive queries. • A list of options for the Dgraph process, which in the previous release could only be specified in the `--args`, can now be specified as part of `endeca-cmd put-dd-profile [create-options]`. For a list of Dgraph flags that can now be specified as part of `[create-options]`, see *Changes to the Dgraph flags on page 20*.
Does not exist in 7.5.x. Was added for 7.6.x.	New: `endeca-cmd update-dd <data-domain> --dd-profile-name [global-options]`. The `update-dd` option is added to `endeca-cmd`. It lets you update an existing data domain with the new data domain profile. For this option to work, the new data domain profile must exist, the data domain must be disabled, and the Endeca Server must have sufficient resources to accommodate the data domain updated with the new profile. See *Changes to the Manage Web Service on page 15*.
Does not exist in 7.5.x. Was added for 7.6.x.	New: `endeca-cmd warm-cache-dd <data-domain> --time-limit-sec <sec> [global-options]`. The `warm-cache-dd` command is added to `endeca-cmd`. This command causes the Dgraph process to warm up its cache without the need to create a custom warm-up script. The command takes into account current Dgraph process usage pattern by selecting a set of queries for replay for a specified period of time. It allows the Dgraph to reuse its computation results across queries, and thus helps reduce the user-observable latencies in query processing and improves performance.

For detailed information about the commands for Endeca Server version 7.6.x, and other newly added commands in `endeca-cmd`, see the *Oracle Endeca Server Administrator's Guide*.

Changes to the EndecaServer.properties file

The `EndecaServer.properties` file sets global-wide parameters for Endeca Server, such as the default locations of mandatory files and directories. The file is located in the `$DOMAIN_HOME/config` directory.

The `EndecaServer.properties` file is automatically created when you create an Endeca Server domain in the WebLogic Server. If you are upgrading from the previous release of the Endeca Server, do not save or use this file from the previous release. Instead, use the new file that is available after the creation of the new WebLogic domain. For the most part, you will not be modifying this file. The file parameters are explained in the *Oracle Endeca Server Administrator's Guide*. If you change any parameters in this file, the Endeca Server should be previously stopped and then restarted.

Note: In the Endeca Server cluster, the configuration of this file for a specific data domain must be identical on all Endeca Server nodes that support this data domain.

In this release, this file has the following new parameters:

Endeca Server parameter	Description
`endeca-ds-pin-timeout-min`	The default minimum value is 60000 milliseconds (ms). This is the lowest timeout value the Dgraph process can use when `PinDataVersion` request is issued. This parameter, as well as `endeca-ds-pin-timeout-max` and `endeca-ds-pin-timeout-default` are used in the Endeca Server to enable index version pinning requests. See *Data version pinning on page 27*.
`endeca-ds-pin-timeout-max`	The default maximum value is 300000 ms. This is the highest timeout value the Dgraph process can use when `PinDataVersion` request is issued.
`endeca-ds-pin-timeout-default`	The default value for the timeout is 120000 ms. This is the default timeout value used by the Dgraph if you do not specify the value in the `PinDataVersion` request. (You typically use this request of the Conversation Web Service to pin a data version).
`endeca-memory-to-index-size-ratio`	Specifies the ratio of virtual memory allocated for a data domain to the index size. The default is 2.0. This setting is used by the Endeca Server for data domain allocation on its nodes. It affects how the Endeca Server calculates whether it has sufficient amount of memory to allocate memory on its nodes to the newly-created data domains. For example, if the index size is 40MB and the default ratio of 2 is used, the Endeca Server will attempt to allocate 80 MB for the data domain.  **Note:** This setting should only be modified by the system administrator of the overall Endeca Server deployment (such as a cluster), and should not be modified by the system administrators of specific data domains hosted in the Endeca Server. Before modifying this setting, learn more about how this setting is used and assess how it applies to your own deployment. For more information on how this setting is used, see the topic on data domain allocation in the *Oracle Endeca Server Cluster Guide*.

Endeca Server parameter	Description
`endeca-threads-allowed-per-core`	Specifies how many threads are actually allowed on a single core, on each Endeca Server node. The default is 1. This setting is used by the Endeca Server for data domain allocation. It affects how the Endeca Server calculates whether it has sufficient number of processing threads to allocate on its nodes to the newly-created data domains.  **Note:** This setting should only be modified by the system administrator of the overall Endeca Server deployment (such as a cluster), and should not be modified by the system administrators of specific data domains hosted in the Endeca Server. For more information on this setting, see the topic on data domain allocation in the *Oracle Endeca Server Cluster Guide*.
`endeca-cgroups-enabled`, `endeca-cgroups-specified-by-percentage`, `endeca-cgroups-reserved-ram-mb`, `endeca-cgroups-reserved-swap-mb`, `endeca-cgroups-reserved-ram-percentage`, `endeca-cgroups-reserved-swap-percentage`	These settings are only applicable to the Endeca Server deployment on Linux 6 (Red Hat Enterprise Linux 6). These settings control how the Endeca Server uses the cgroups feature in Linux 6. For information on configuring the use of cgroups for the Endeca Server in Linux 6, see the *Oracle Endeca Server Administrator's Guide*.
`endeca-data-enrichment-puginsDir`, `endeca-data-enrichment-resourceDir`, `endeca-data-enrichment-collectionBatchSize`, `endeca-data-enrichment-collectionMatchRetries`, `endeca-data-enrichment-taskHistorySize`	These parameters are added for the Data Enrichment module, which may be used for enrichments in Studio.

Endeca Server interfaces

This section describes required changes that apply to the Web services and other interfaces of the Endeca Server.

Changes to the Cluster Web Service

Changes to the Manage Web Service

Changes to the Conversation Web Service

Changes to the Data Ingest Web Service

Changes to the Configuration Web Service

Changes to the Entity and Collection Configuration Web Service

Changes to the Cluster Web Service

This topic discusses the changes in the Cluster Web Service in the Endeca Server 7.6.x.

The `putDataDomainProfile` operation of the Cluster Web Service includes the following new options:

- `autoIdle` indicates whether to automatically turn this data domain idle if, during the optionally specified timeout the data domain does not receive queries. If set to `false`, the data domain is never turned idle. If set to `true`, the Endeca Server makes this data domain idle after the `idleTimeoutMinutes` expires. The default is `false`.
- `idleTimeoutMinutes` is the time, in minutes, after which a data domain that is set to automatically idle is turned idle if it does not receive queries during this time. The default is 10 minutes. If not specified, the default is used. This parameter only affects the data domain if the `autoIdle` is set to `true`, for the data domain's profile; it does not affect data domains that are not allowed to auto-idle.

Changes to the Manage Web Service

This topic discusses the changes in the Manage Web Service in the Endeca Server 7.6.x.

The Manage Web Service includes the following new operations:

- An operation for updating the data domain, `updateDataDomain`. This operation lets you change the data domain's profile, for an already created data domain. The data domain must be disabled to run this operation. This operation has an equivalent Endeca Server command: `endeca-cmd update-dd <name> --dd-profile-name <profile>`.
- An operation for warming up the underlying Dgraph cache, `warmCache`. This operation lets you launch a Dgraph warming script that runs in the background, for the optionally specified time limit. The default time limit is 1800 seconds (30 minutes). This operation has an equivalent Endeca Server command: `endeca-cmd warm-cache-dd <name> --time-limit-sec <value>`.

Changes to the Conversation Web Service

This topic discusses the changes in the Conversation Web Service for the Endeca Server 7.6.x.

The Conversation Web Service has undergone enough refactoring that you must re-write your 7.5.x queries.

Request state

As in Endeca Server 7.5.x, each request has a state (defined by a `<State>` element). In Endeca Server 7.6.x, the request may have multiple (two or more) states. In this case, each state must have a name (defined by a `<Name>` element) that is unique among the other state names in the request. Names states, therefore, are new in Endeca Server 7.6.x.

The state can optionally refer to one (and only one) collection name via the `<CollectionName>` element.

Note that the Conversation Service response no longer includes the full request that spawned it. Only the state from the request is returned.

ContentElementConfigs replaced with specific subclasses

In Endeca Server 7.5.x, there was one `ContentElementConfig` that used `xsi:type` to declare the type. In Endeca Server 7.6.x, the `xsi:type` has been removed and the config now uses the name of its sub-class

(such as `RecordListConfig`). In addition, the `HandlerFunction` and `HandlerNamespace` elements have also been removed.

Therefore, this example from the Endeca Server 7.5.x:

```
<ContentElementConfig
    xsi:type="RecordListConfig"
    HandlerFunction="RecordListHandler"
    HandlerNamespace="http://www.endeca.com/MDEX/conversation/2/0"
    Id="RecordList" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <RecordsPerPage>25</RecordsPerPage>
    <Sort Key="Spec" Direction="Ascending"/>
</ContentElementConfig>
```

looks like this in Endeca Server 7.6.x:

```
<RecordListConfig Id="RecordList">
    <RecordsPerPage>25</RecordsPerPage>
    <Sort Key="Spec" Direction="Ascending"/>
</RecordListConfig>
```

Note that the 7.5.x `LQLConfig` config has been renamed to `EQLConfig`.

Removal of operators

In 7.5.x, filter components were specified within the context of an operator. For example, a record search filter would use a `<SearchFilter>` element within an `<Operator>` of type `<SearchOperator>` as in this example:

```
<Request>
   <State/>
   <Operator xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:type="SearchOperator" Within="false">
      <SearchFilter Mode="AllPartial" Key="Prod_Category" Language="en">
       electronics
      </SearchFilter>
   </Operator>
...
```

In 7.6.x, the operators have been removed from queries and the filter names are specified directly in the state. So the above 7.5.x example would look like this in 7.6.x:

```
<Request>
   <State>
      <TextSearchFilter Key="Prod_Category" RelevanceRankingStrategy="numfields"
         Mode="AllPartial" Language="en">
         electronics
      </TextSearchFilter>
   </State>
...
```

EQL filters

The 7.5.x `DataSourceFilterString` has been renamed to `DataSourceFilter`, and the `SelectionFilterString` has been renamed to `SelectionFilter`. Both filters have a new (and optional) `Id` attribute that provides a name for the filter.

Each filter can be used multiple times in a state, instead of only once as in 7.5.x. When using multiple filters, each filter must have a name that is unique among other filter names within the state.

As in 7.5.x, each filter uses the EQL `WHERE` clause syntax. In 7.6.x, you can also use the new set functions in the filter strings when dealing with multi-assign attributes.

Record search filter

The 7.5.x `SearchFilter` (used for record search) has been renamed to `TextSearchFilter` (as shown in the example above).

Note that the `SelectedRefinementFilter` and `RecordKind` filters have the same name as in 7.5.x.

Changes to the Data Ingest Web Service

This topic discusses the changes in the Data Ingest Web Service in the Endeca Server 7.6.x.

Addition of updateSpellingDictionaries

The `updateSpellingDictionaries` operation has been added to the Data Ingest Web Service. In the previous release, it was part of the Manage Web Service.

In the Data Ingest Web Service, the `updateSpellingDictionaries` behaves in the same way as in previous release when it was part of the Manage Web Service, allowing you to enable spelling correction and Did You Mean (DYM) features for the data domain, and also to update the spelling dictionaries. This operation now takes the outer transaction ID as an optional argument and handles it in the same way as other Data Ingest Web Service operations. If successful, `updateSpellingDictionaries` returns an empty response.

Deprecation of the ingestManagedAttributeValues operation

The `ingestManagedAttributeValues` operation is supported in this release for backward compatibility. However, Oracle recommends that you migrate away from using this operation. Instead of using this operation, use the `putManagedAttributeValues` operation of the Configuration Web Service. This is the preferred way to add managed attribute values. For information on `putManagedAttributeValues`, see *Changes to managed attribute values on page 26*.

For detailed information on these changes, see the *Oracle Endeca Server Data Loading Guide*.

Using EQL set functions to identify records

When modifying or deleting records with the `ingestChanges` operation, you can use EQL set functions to identify the records, as in this example:

```
<recordSpecifier>SOME i IN NumRevs SATISFIES (i > 45)</recordSpecifier>
```

For more information on using multi-assign attributes with EQL, see the *Oracle Endeca Server EQL Guide*.

Changes to the Configuration Web Service

In Endeca Server version 7.6.x, two new operations for managed attribute values have been added to the Configuration Web Service.

The `putManagedAttributeValues` operation lets you add one or more managed attribute values to the data domain before loading any data records. It assumes that the associated managed attribute has already been created.

The `listManagedAttributeValues` operation lists the managed attribute values for a specified managed attribute or for all managed attributes in the data domain.

For detailed information on how to add and list managed attribute values using the Configuration Web Service, see the *Oracle Endeca Server Developer's Guide*.

Changes to the Entity and Collection Configuration Web Service

This topic discusses the changes in the Entity and Collection Configuration Web Service in the Endeca Server 7.6.x.

In version 7.5.x, this service (sConfig) was called the Entity Configuration Web Service. In version 7.6.x, it has been renamed to the Entity and Collection Configuration Web Service.

In addition to operations on entities, in this release this web service now has operations that:

- Create, update, and remove collections.
- Create, update, and remove filter rules.

For information on these operations, see the *Oracle Endeca Server Developer's Guide*.

Endeca Query Language changes

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

Sets and full support for multi-assign attributes

Endeca Server Version 7.5.x had limited support for multi-assign attributes; for example, 7.5.x EQL did not support multi-assign attributes in statement results. 7.5.x EQL also performed implicit de-multi-assigning on multi-assign corpus attributes in most contexts. De-multi-assigning operated on a single record, selecting a particular assignment value within that record. In addition, in statements that referenced multi-assign attributes, EQL would select an arbitrary value from a multi-assign attribute for each input record, instead of selecting all of the attribute values.

In the Endeca Server 7.6.x, EQL contains full support for multi-assign attributes. For example, multi-assign attributes are used in statement results. All the values (and not just one) from a multi-assign attribute assignments on records are returned as a set in EQL. Note that the empty set is returned from a record that has no assignments from a selected multi-assign attribute.

You can perform operations on the sets with a variety of new set functions in EQL, such as intersecting or unioning two sets. This also means that implicit de-multi-assigning of multi-assign corpus attributes is no longer performed, and EQL now selects all of the multi-assign values from a record instead of only one arbitrary value.

It is important to know that sets have different data types than atomic values. For example, a set from a multi-assign attribute of type integer will have a data type of `mdex:long-set`, while an atomic value from a single-assign attribute of type integer will have a data type of `mdex:long`.

If you have 7.5.x EQL statements that select or use multi-assign attributes, you may have to re-write them to work with the set implementation.

For details on working with sets and multi-assign data in EQL 7.6.x, see the *Oracle Endeca Server EQL Guide*.

Changes to previous EQL functions

For EQL 7.6.x, the ARB, COUNT, and COUNTDISTINCT functions have been updated to work with multi-assign attributes. The = (equal), < (less than), and > (greater than) operators have also been updated to allow you to test equality between sets.

All other 7.5.x functions and operators will not work with sets, and will generate data type mismatch errors if used. For example, if the attribute named Score is a multi-assign attribute of type integer, then this statement using ARB works fine:

```
RETURN results AS
SELECT ARB(Score) AS scores
GROUP BY WineType
```

However, if you use the SUM function:

```
RETURN results AS
SELECT SUM(Score) AS scores
GROUP BY WineType
```

the statement will fail with this error:

```
In computing attribute "scores": "SUM" is not a valid
aggregating function over the types (mdex:long-set)
```

Note that the IS NULL and IS NOT NULL operations are not supported on sets. Instead, use the new IS_EMPTY and IS_NOT_EMPTY functions (or their alternative syntax versions, IS EMPTY and IS NOT EMPTY).

For documentation on the new set functions, see Chapter 5 ("Sets and Multi-assign Data") of the *Oracle Endeca Server EQL Guide*.

Removal of implicit ARB

EQL no longer inserts an implicit ARB around an implicit ARB attribute reference that requires aggregation but instead signals an error to the user. For example, in 7.5 this statement was allowed even though Price (which is a single-assign double attribute) was not aggregated:

```
RETURN results AS
SELECT Price AS totals
GROUP
```

The reason the statement was parsed correctly is that an implicit ARB was added around the Price attribute.

In 7.6, an implicit ARB is no longer added. That is, a SELECT of non-aggregate attributes is no longer allowed. In a grouped statement, all selected attributes that are not grouping attributes must be explicitly aggregated.

Thus, in 7.6 the above query will fail with this error message:

```
In computing non-grouping attribute "totals":
Source attribute "NavStateRecords"."Price" must be aggregated
```

You, therefore, have to add an explicit ARB to the statement:

```
RETURN results AS
SELECT ARB(Price) AS totals
GROUP
```

Changes to the Dgraph flags

A subset of Dgraph flags have been modified, allowing you to specify them in a simpler way, with changed names, with `endeca-cmd put-dd-profile [create-options]`, and without using the `--args` parameter of the `endeca-cmd put-dd-profile` command.

In the previous release, a number of flags could be specified to the Dgraph processes of the data domains via `endeca-cmd put-dd-profile --args` command, for example:

```
endeca-cmd put-dd-profile MyProfile --description
"test profile" --oversubscribe false --args --net-timeout 60
```

In this release, some of the lower-level flags are still specified this way. However, the following flags that you could formerly specify at the level of `--args` can now be also specified at the higher level, similar to other `[create options]` flags. For example, in this release, the command analogous to the previous example looks like this:

```
endeca-cmd put-dd-profile MyProfile --description
"test profile" --oversubscribe false --net-timeout 60
```

Note also that some of the flag names have changed, compared with the previous release. The following table lists those Dgraph flags that can now be specified differently (without the `--args`), and also provides their new names:

Name of the Dgraph flag in Endeca Server version 7.5.x	**Name of the flag in Endeca Server 7.6.x (if specified with `endeca-cmd put-dd-profile --<Dgraph_flag>`)**
`--ancestor_counts`	`--ancestor-counts`
`--backlog-timeout`	`--backlog-timeout`
`--esampmin`	`--refinement-sampling-min`
`--implicit_exact`	`--implicit-exact`
`--implicit_sample`	`--implicit-sample`
`--net-timeout`	`--net-timeout`
`--search_max`	`--search-max`
`--snip_cutoff`	`--snippet-cutoff`
`--snip_disable`	`--snippet-disable`
`--stat-all`	`--dynamic-category-enable`
`--unctrct`	`--contraction-disable`
`--wildcard_max`	`--wildcard-max`

For a detailed description of each of these flags, see the *Oracle Endeca Server Administrator's Guide*.

Note: If you continue to specify the arguments using the same method as in the Endeca Server 7.5.x (with `--args`), and continue using the names as in version 7.5.x, the Endeca Server 7.6.x will accept these commands as valid (to ensure backwards-compatibility). Additionally, arguments specified this way supercede arguments specified in the new way (that is, without the `--args`). Even though this backwards-compatibility exists, the preferred way to specify these arguments is the one adopted in the Endeca Server 7.6.x release. Similarly, the representation of these arguments in the Cluster Web Service requests is treated in the same way — specifying them as in version 7.5.x remains valid (thus your Java code generated from this web service can be largely reused in version 7.6.x). However, going forward, it is recommended to use the new way of specifying these flags. For information on Cluster Web Service options, see the *Oracle Endeca Server Cluster Guide*.

Additionally, the `--sslcipher <cipher-list>` flag for the Dgraph has been removed. Instead of using this flag, the Endeca Server supports a list of approved ciphers and does not support any others; for a summary, see the *Oracle Endeca Server Security Guide*. The `--stat-brel` Dgraph flag has been deprecated and should not be used.

Chapter 3

Behavioral Changes

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

Cluster changes

In this release, additional features are added to the management of data domains in the Endeca Server cluster.

The following statements provide a brief listing of the changes. For detailed information, see the *Oracle Endeca Server Administrator's Guide*.

- You can configure a data domain profile that allows you to automatically idle a data domain after a specified timeout, if the data domain does not receive queries during this timeout.
- You can update a disabled data domain with a new data domain profile.
- You can issue a request to warm up the Dgraph process for the data domain, based on the most recent representative list of queries and the usage pattern for the data domain.

Improved data domain allocation

In this release, the Endeca Server better utilizes the underlying system resources and uses heuristic methods to more efficiently allocate data domains to the nodes in the Endeca Server deployment, based on calculations of total available resources—memory and the number of threads.

These improvements are implemented to support the creation of a large number of data domains for scenarios with a large number of applications, (such as, self-service applications in the Oracle Endeca Information Discovery product). Such scenarios typically involve a large number of relatively small-scale applications powered by the Endeca Server, with a small number of users per application.

The improved data domain allocation allows to host new or additional data domains if the system resources are sufficient to host them, and, if the underlying Endeca Server does not have sufficient resources, limits the proliferation of a large number of such data domains.

Specifically, to allocate data domains, Endeca Server chooses nodes which have sufficient free virtual memory and processing cores for the new data domains. It denies the creation of additional data domains when it does not have enough processing cores to host them (this behavior is better guaranteed on Linux 6 with cgroups enabled, if the Endeca Server is configured to utilize cgroups). To conserve resources, Endeca Server turns to idle those inactive data domains that are configured to auto-idle, and automatically activates such data domains when a user request arrives. To help the cluster administrators of the Endeca Server tune the allocation of data domains to Endeca Server instances, Endeca Server allows specifying the number of threads per core, and the ratio of allocated virtual memory to the index size, for the nodes in the Endeca Server deployment. Additionally, if the Endeca Server is deployed on Oracle Linux 6 or RHEL 6, cgroups can be used for data domain allocation, providing additional guarantees.

Important: If you are planning to deploy a large number of self-service applications in the Endeca Server, to increase the allocation guarantees and ensure that Endeca Server nodes continue to operate even when many data domains are provisioned, you are strongly encouraged to deploy Endeca Servers on Oracle Linux 6 or RHEL 6, which both allow Endeca Server to utilize cgroups.

For detailed information on these features, see the following documentation:

- Information about Endeca Server memory consumption in the *Oracle Endeca Server Administrator's Guide*.
- Information about auto-idling of the data domains in the *Oracle Endeca Server Cluster Guide*.
- Information about allocation of data domains in the *Oracle Endeca Server Cluster Guide*.
- Information about using cgroups feature of the Red Hat Enterprise Linux 6 in the Endeca Server (for those Endeca Server deployments that are installed on Oracle Linux 6 or RHEL 6), in the *Oracle Endeca Server Administrator's Guide*.

Data domain changes

This topic lists various changes to what you can configure for data domains.

The changes are:

- The default number of threads for the default data domain profile has changed from 2 to 4.
- In the previous release, on Windows, only a disabled data domain could be exported. In this release, this limitation is removed — both on Windows and Linux, you can take a snapshot (or export) a data domain that is either enabled or disabled.

- In this release, when returning the information about the data domain health (`endeca-cmd get-dd-health`), the Endeca Server provides details for each Dgraph node in the data domain that is down. In the previous release, it only listed that some Dgraph nodes are not running.
- In this release, updating spelling dictionaries for the data domain is handled by the Data Ingest Web Service, and the option for this in `endeca-cmd` has been removed. In the previous release, it was handled by the Manage Web Service and an option in `endeca-cmd`.
- You can optionally specify for the data domain profile the maximum length (in characters) of a search term for record and value searches. You can do using the new option, `--search-char-limit <int>`. The default is 132 characters. Any term exceeding this length will not be indexed, and thus will not be found in record and value searches. An example of a command with this option is:

  ```
  endeca-cmd put-dd-profile <new-profile-name> [global-options] --search-char-limit
  ```

- You can update the data domain by changing its profile, with the `endeca-cmd update-dd <name> --dd-profile-name <name>` command. You can run this command only on a previously disabled data domain. The command reallocates Endeca Server resources to this data domain, using the specified data domain profile, which must already exist.
- You can warm up the Dgraph cache for the data domain, with the `endeca-cmd warm-cache-dd <name> --time-limit-sec <value>` command. The command replies the cache warming queries to the Dgraph processes of the data domain. The time limit is optional, and if not specified defaults to 1800 seconds (30 minutes). For more information on cache warming operation, see the *Oracle Endeca Server Administrator's Guide*.
- You can specify the auto-idling setting for the data domain profile. If a data domain profile specifies that a data domain should automatically idle, the data domain that uses this profile behaves as follows: Once it is enabled, its timer for the idling timeout is started. If during the optionally-specified timeout period (the default is 10 minutes), the data domain does not receive any queries, the Endeca Server turns this data domain to idle, by stopping its Dgraph process. The data domain is restarted automatically whenever it receives a query. The idling timer is also restarted each time a query arrives that restarts an idle data domain. The data domain that is set to auto-idle does not shut down if it is loading data or performing any other query. Having the auto-idle setting helps control the proliferation of data domains in self-service applications of the Endeca Server, thus improving the utilization of the Endeca Server processing resources. For more information on the behavior of idling data domains, see the *Oracle Endeca Server Cluster Guide*.

DDR changes

The Dimension Description Records (DDR) have changed this release and include two new properties, both of which are required. They define the structure for the managed attribute values.

Note: Only the Dimension Description Records have changed in this release. No changes to other system records were made in this release.

DDR attributes

The following new attributes are added to the DDR definition. These attributes together define the managed attribute values associated with any managed attribute.

Schema attribute	Type	Description
`mdex-dimension-value_Spec`	string	The primary key (spec) of the managed attribute value. It cannot be modified after it has been added. If, when adding a managed attribute, you do not specify the value of this attribute for the DDR, the default value is used, in this format: `mdex-dimension_<key>_Spec`, where <*key*> is the key of the managed attribute you are adding.
`mdex-dimension-value_Parent`	string	The key of the parent managed attribute value. It cannot be modified after it has been added. If, when adding a managed attribute, you do not specify the value of this attribute for the DDR, the default value is used, in this format: `mdex-dimension_<key>_Parent`, where <*key*> is the key of the managed attribute you are adding.

Additional supported languages

You can now configure a specific language for standard attributes and for queries.

Version 7.5.x supported twenty-two languages, plus the default "unknown" language code. Version 7.6.x supports those twenty-two languages (plus "unknown) and includes support for these additional thirteen languages:

- Arabic — `ar`
- Croatian — `hr`
- Danish — `da`
- English, British — `en_GB`
- Finnish — `fi`
- Norwegian Bokmal — `nb`
- Norwegian Nynorsk — `nn`
- Persian — `fa`
- Portuguese, Brazilian — `pt_BR`
- Serbian, Cyrillic — `sr_Cyrl`
- Serbian, Latin — `sr_Latn`
- Slovak — `sk`
- Slovenian — `sl`

The new language codes, like the previous ones, are case insensitive.

See the *Oracle Endeca Server Developer's Guide* for information on specifying per-query language codes for record searches and value searches.

Changes to managed attribute values

In the Endeca Server 7.5.x, you could add managed attribute values using the `IngestManagedAttributeValues` operation in the Data Ingest Web Service. While this operation is supported in this release for backward compatibility, Oracle recommends that you migrate away from using this operation. Instead of using this operation, use the `putManagedAttributeValues` operation of the Configuration Web Service. This is the preferred way to add managed attribute values.

The following statements describe all changes related to managed attribute values in the Endeca Server version 7.6.x:

- The preferred way to add managed attribute values is by using the new operation `putManagedAttributeValues` of the Configuration Web Service.
- Additionally, you can use the `addRecords` sub-operation of the `ingestChanges` operation in the Data Ingest Web Service, to add managed attribute values. This is because, starting with the Endeca Server version 7.6.0, managed attribute values are represented in the index as records (once you add them).
- If you use the `putManagedAttributeValues` operation of the Configuration Web Service to add them, you can also define static ranks and synonyms for managed attribute values, and the Endeca Server will re-assign the ranking of existing managed attribute values accordingly (if any ties are created). If you use the `addRecords` from the Data Ingest Web Service, you can only add values that have synonyms and ranks, but ranks are not re-assigned, and duplicate ranks are not prevented.
- The deprecated operation, `IngestManagedAttributeValues`, allows to add additional custom properties to managed attribute values. The operation from the Configuration Web Service, which replaces `IngestManagedAttributeValues`, does not let you add properties. However, you can optionally add metadata, as you would for regular records, when adding managed attribute values with `addRecords` that is part of `ingestChanges`.
- Managed attribute values should be added before any data records are loaded. Adding managed attribute values requires having a defined managed attribute with which the values will be associated (for example, they can form a hierarchy).

For information on the structure of managed attribute values, and how to add them using `putManagedAttributeValues`, see the *Oracle Endeca Server Developer's Guide*.

For information on how to add managed attribute values using `addRecords` of the Data Ingest Web Service, and to learn how managed attribute values are assigned onto to source data records, see the *Oracle Endeca Server Data Loading Guide*.

Retrieving explicitly-selected and implicit refinements

In this release, the Conversation Web Service has been updated to allow you to retrieve explicitly-selected and implicit refinements (in addition to retrieving the suggested refinements).

In the previous release, the Conversation Web Service only returned explicitly-selected refinements as part of the `BreadcrumbConfig` content element configuration. However, implicit refinements could not be retrieved.

In this release, you can issue a request that should retrieve both kinds of applied refinements, if needed, by requesting them either globally for all attributes in the `NavigationMenuConfig`, or by requesting to retrieve

them per each attribute, in `RefinementConfig`. The values for each attribute override the values you specify globally.

To request applied refinements globally, you can optionally specify values for the following two new attributes in the `NavigationMenuConfig`:

- `IncludeAllExplicitSelections` specifies whether Endeca Server should retrieve explicitly-selected refinements. The default is `false`.
- `IncludeAllImplicitSelections` specifies whether Endeca Server should retrieve implicit refinements. The default is `false`.

Once the refinements are retrieved, `ExposeAllRefinements="true"` is the setting that controls whether they are also returned in the web service response.

To request applied refinements per attribute, you can optionally specify values for the following two new attributes in the `RefinementConfig`:

- `IncludeExplicitSelections` specifies whether Endeca Server should retrieve explicitly-selected refinements, for this attribute. The default is `false`.
- `IncludeImplicitSelections` specifies whether Endeca Server should retrieve implicit refinement, for this attribute. The default is `false`.

Once the refinements are retrieved, `Expose="true"` is the setting that controls whether they are also returned in the web service response.

The settings made per attribute (in the `RefinementConfig` for each attribute) override the analogous settings made for all attributes (in the `NavigationMenuConfig`).

If present and requested to be exposed, explicitly-selected and implicit refinements are returned inside the `NavigationMenuItem`, for each refinement for which they are present, in the following format:

- Explicitly-selected:
```
<cs:SelectedRefinement Name="Winery" Spec="A.R. Lenoble" Label="A.R. Lenoble" Count="3"/>
```
- Implicit:
```
<cs:ImplicitRefinement Name="WineType" Spec="Sparkling" Label="Sparkling"/>
```

For detailed information on retrieving a full list of refinements, see the *Oracle Endeca Server Developer's Guide*.

Data version pinning

Two new elements are added to the Conversation Web Service — the `PinDataVersion` element, for holding on to a specific data version, and the `DataVersionRequested` element, for requesting this version in a subsequent request that is issued within a previously specified timeout.

To pin a version, add the following to a Conversation Web Service request:

```
<PinDataVersion>optionalPinTimeout<\PinDataVersion>
```

This timeout must be between `endeca-ds-pin-timeout-min` and `endeca-ds-pin-timeout-max`, specified in the `EndecaServer.properties` file. If the timeout is not specified, the default timeout, `endeca-ds-pin-timeout-default`, also specified in this file, is used.

If the request is successful, it returns an empty response, with a header that includes two versions:

- `X-Endeca-Served-Data-Version`, this is the data version used for the query, or, the "pinned version".
- `X-Endeca-Data-Version`, this is the most up-to-date version.

To request the Endeca Server to use a previously pinned version in a query, add the following to any of your Conversation Web Service requests:

```
<DataVersionRequested>versionNum<\DataVersionRequested>
```

where `versionNum` is the version that was previously pinned, and whose timeout has not yet expired. If the version cannot be served, an error is returned. Issuing this request within an existing timeout renews the timeout.

Additionally, starting with the Endeca Server 7.6.x., each Conversation Web Service request has two headers returned with it:

```
X-Endeca-Served-Data-Version - the data version used for the query
X-Endeca-Data-Version - the most up-to-date data version
```

For more information on how to pin a data version and then request this version in a subsequent Conversation Web Service request, see the *Endeca Server Developer's Guide*.

Performance improvements

Endeca Server 7.6.x improves query performance by generating query-specific code at run-time, which is then compiled into machine code. The cost of code-generation is mitigated using code caching.

Also, Endeca Server now better analyzes complex conditionals, resulting in improved performance for some complex queries.

The Dgraph stats page

Starting from this release, the Dgraph stats page is intended for use by the Oracle Endeca Support only. If you use this page, you should be aware that this page is not guaranteed to return results from a specific Dgraph node (in the Endeca Server cluster). Additionally, this page is not guaranteed to be supported in the future releases of the Endeca Server.

Documentation changes

The following high-level changes took place in the Endeca Server documentation in this release.

- The information about managing data domains (adding, updating, exporting, importing and other tasks, such as cache warming), is moved from the *Endeca Server Cluster Guide* to the *Endeca Server Administrator's Guide*.
- The examples of using the Conversation Web Service have been updated in the Endeca Server Developer's Guide, to reflect the simplification and other changes in the Conversation Web Service in this release.