



Relevance Ranking Evaluator Usage Guide

The Oracle Endeca Relevance Ranking Evaluator provides business users with an interactive tool for experimenting with and comparing the results of different relevance ranking strategies. It can be used as a testing tool to aid with application development, as well as search tuning and "scorecard" review processes.

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.

Contents

Introduction	5
System Requirements and Installation	5
MDEX Engine.....	Error! Bookmark not defined.
Configuring Your MDEX Engine.....	5
Optional MDEX Engine Configuration.....	6
Connecting to Your MDEX Engine.....	7
Configuring Searches	7
Naming Searches	8
Search Criteria	8
Rollup Key	9
Search Filters.....	10
Record Filters	10
Range Filter.....	10
Endeca Query Language (EQL) Filter.....	10
Relevance Ranking Strategies.....	11
Static Module.....	12
Stratify Module	12
Ordering Relevance Ranking Modules	12
Pre-defined Strategies.....	Error! Bookmark not defined.
Working with Multiple Searches.....	13
Viewing Results	14
Setting the Number of Records Displayed.....	14
Search Summary Information	15
Base Information	15
Information Provided for MDEX Engine Version 6.2 and Higher	15
Record Information	16
Comparing Search Results	17
Starting Over.....	18
Optional Relevance Ranking Evaluator Configuration	18
MDEX Engine_VERSION.....	Error! Bookmark not defined.
DISPLAYED_ATTRIBUTES.....	Error! Bookmark not defined.
MAX_DISPLAYED_ATTRIBUTES.....	Error! Bookmark not defined.

STATIC_MODULE_PROPERTIES.....	Error! Bookmark not defined.
RECORDS_TO_DISPLAY_OPTIONS.....	Error! Bookmark not defined.
DEFAULT_ROLLUP_KEY	Error! Bookmark not defined.
SAVED_MODULE_LIST	Error! Bookmark not defined.
ORDER_ATTRIBUTES.....	Error! Bookmark not defined.
IMAGE_DISPLAY_TEMPLATE.....	Error! Bookmark not defined.
IMAGE_PROPERTY_NAME.....	Error! Bookmark not defined.
Deploying a Relevance Ranking Strategy	Error! Bookmark not defined.
Appendix A: Relevance Ranking Modules	23
Exact	23
Field	23
Freq (Frequency)	24
Glom.....	25
Interp (Interpreted)	25
Maxfield (Maximum field)	25
Nterms (Number of terms)	26
Numfields (Number of fields)	26
Phrase.....	26
Using Approximate Matching	26
Spell	27
Static	27
Stem.....	28
Stratify.....	28
Thesaurus	28
Wfreq (Weighted frequency).....	28

Introduction

The Oracle Endeca Relevance Ranking Evaluator is a web application utilized by business users to test and experiment with relevance ranking strategies. Like the Oracle Endeca Workbench, it uses a business's own data and MDEX Engine to contextualize how different strategies will behave.

The Relevance Ranking Evaluator can run multiple searches simultaneously, using different relevance ranking modules and other search parameters (such as match modes), allowing for users to compare the order and quality of returned results. The application provides comparison of search results and their relevance ranking strategies side-by-side, highlighting the location of records in each strategy. Additionally, the Relevance Ranking Evaluator provides performance metrics for each query, which can help users evaluate the tradeoffs between run-time performance and the detailed result ordering of some of the more complex relevance ranking modules.

For complete documentation on relevance ranking, please refer to the documentation for the Oracle Endeca MDEX Engine, which is available on the Oracle Technology Network.

System Requirements and Installation

The Oracle Endeca Relevance Ranking Evaluator is a Java web application, and, as such, requires installation into an existing Java web application server. The Endeca Tools Service, the servlet container that hosts Oracle Endeca Workbench, can be used as this web application server, or a separate application server may be used if desired. For detailed installation instructions and system requirements, please refer to the "InstallationGuide.pdf" located in the same directory as this Usage Guide.

Additionally, it is recommended to use the Relevance Ranking Evaluator with a recent version of Mozilla Firefox with JavaScript enabled. While other JavaScript-compatible web browsers may work with the Relevance Ranking Evaluator, such as Internet Explorer, Safari and Chrome, they have not been fully tested.

Finally, it is crucial that Relevance Ranking Evaluator users be familiar with their Oracle Endeca application's configuration. Specifically, users should be aware of search interface configuration, such as the fields contained, the field order, partial matching strategy and cross-field matching strategy. This information is important to know when evaluating relevance ranking strategies, and is not generally available from within the Relevance Ranking Evaluator.

MDEX Engine

Configuring Your MDEX Engine

In order to provide results that are specific to your application and dataset, the Relevance Ranking Evaluator requires the use of an MDEX Engine to power its results. It is not necessary to dedicate a single MDEX Engine solely for use by this application; the MDEX Engine used may be shared by another web application or instance of your website.

However, it is recommended that the MDEX Engine used be a part of a staging or QA environment, where the configuration is stable, but not competing for resources with production.

Optional MDEX Engine Configuration

While the Relevance Ranking Evaluator can be used with any MDEX Engine, there are certain configuration options that enhance the information and user experience provided. These configuration options are optional, and are not required in order to use the application.

- **"whymatch"**: For MDEX Engine versions before 6.2, the Relevance Ranking Evaluator can take advantage of the information returned by either the `--whymatch` (recommended) or `--whymatchConcise` command-line MDEX Engine option. If one of these options is enabled, the Relevance Ranking Evaluator will display specific information about how each record was matched by the entered search terms. However, these options should be used with care in a production environment, as they may cause degradation of run-time performance.
- **"wordinterp"**: The Relevance Ranking Evaluator can take advantage of the information returned by the `--wordinterp` command-line MDEX Engine option. If so enabled, the application will display word interpretations considered for the search query, including thesaurus and stemming expansions. Again, care should be used in a production environment, as this option may cause degradation of run-time performance.
- **Snippeting**: For MDEX Engine versions before 6.2, in order to provide information about cross-field matching (where a multi-term search matches records that contain one entered search term in one property, and another entered term in another property), it is necessary to enable the snippeting feature for all fields in the search interface(s) that will be used in the application. Developer Studio should be used to enable the snippeting feature for each field contained within a search interface. Like "whymatch," snippeting should be used with care in production environments, as the overhead required to calculate the snippet information, as well as the extra information returned by the MDEX Engine, can contribute to run-time performance degradation.
- **"DGraph.BinRelevanceRank"**: When the MDEX Engine performs a relevance ranked search, it determines a relevance ranking "score" for each record, and uses that score internally to sort the results. This score, along with an indication of records that tie with the same the score, can be displayed by the Relevance Ranking Evaluator if the `--stat-brel` command-line MDEX Engine option is used. Like the other options, this should be used with care in a production environment, as it may cause degradation of run-time performance.

For more information on the "whymatch", "wordinterp", "DGraph.BinRelevanceRank", and snippeting configuration options, please refer to the MDEX Engine documentation.

Connecting to Your MDEX Engine

During installation, the Oracle Endeca Relevance Ranking Evaluator should be configured to connect by default to the appropriate MDEX Engine for your environment. Please refer to the Installation Guide for specific details on setting up the default MDEX Engine. However, if required, you may change the MDEX Engine that the Oracle Endeca Relevance Ranking Evaluator connects to from within the application itself.

To connect to an MDEX Engine other than the default, click on the button labeled "Change MDEX Engine" in the upper-right of the screen. A new window will open, in which you may enter the host and port of the desired MDEX Engine. To save the connection information, click the "Submit" button.



The image shows a dialog box titled "Specify MDEX Engine". It has a light gray header with the title in white. Below the header, there are two input fields: "Host" with the value "localhost" and "Port" with the value "15000". At the bottom of the dialog is a "Submit" button.

Note that changes to which MDEX Engine is used by the Relevance Ranking Evaluator will only be in effect for the particular user and browser session in which it was changed, and will not affect other users of the application.

Configuring Searches

To access the Relevance Ranking Evaluator, navigate to the appropriate URL for your installation using a JavaScript-enabled browser such as Internet Explorer or Mozilla Firefox. The URL used to access the application may differ depending on your installation, so please refer to the Installation Guide in order to determine the appropriate location. However, if the application been deployed using the Endeca Tools Service, meaning it is hosted along with the Oracle Endeca Workbench, the URL is likely to be: <http://<hostname>:8006/relevanceRankingEvaluator>, where <hostname> is the hostname of the machine where the application is hosted.

When first accessed, the Relevance Ranking Evaluator application displays a single, empty search configuration, which is ready to be customized with your first test search strategy.

Search 1 +		Clear All	Change MDEX Engine	Show 10 ▾
Search Name: Search 1		Go		
Search Criteria				
Search Interface	Match Mode	Search Term(s)	Rollup Key	
All ▾	All > Partial ▾		None ▾	
Search Filters +				
Relevance Ranking				
Strategies	Available Modules		Selected Modules	
<input type="radio"/> Retail Strategy <input type="radio"/> Document Strategy <input type="radio"/> Custom Strategy	Exact Field First Freq Glom Interp Maxfield	<input type="button" value="→"/> <input type="button" value="←"/>		<input type="button" value="↑"/> <input type="button" value="↓"/>
Search Results Comparison				
To see results, please configure your relevance ranking strategy.				
Copyright © 2003, 2012 Oracle and/or its affiliates. All rights reserved.				

The Relevance Ranking Evaluator can be used by more than one user simultaneously; each user will see his or her own searches, and changes made by one user will not be visible to others. Be aware that testing performed in the application is not stored outside of the user's particular browser session; if the browser window is closed or if the session is allowed to time out due to inactivity, the tests will be cleared and the application will be reset to its initial state. The amount of inactive time that must pass before a browser session times out is dependent on the web application server's configuration settings; note that most application servers default to an idle time of 15-20 minutes. This can be changed by your application server administrator.

Naming Searches

The Relevance Ranking Evaluator is able to display multiple search results side by side in its results area, so that users can compare different test configurations. In order to differentiate between different search result sets, it is recommended provide descriptive names for each test strategy. A test can be given a unique identifying name by entering the desired name in the "Search Name" text box.

Search Criteria

In order to test a particular relevance ranking strategy, a text search must be performed against a search interface. In the "Search Criteria" area, enter a search term (or multiple terms) on which

to search, and choose the desired search interface and match mode from the appropriate drop-downs.

Search Criteria 		
Search Interface	Match Mode	Search Term(s)
All ▼	All > Partial ▼	

Each defined search allows for individual specification of the search interface, match mode, and search terms. Therefore, while the most common use of the Relevance Ranking Evaluator is to compare different relevance ranking strategies for the same search criteria, it can also be used to compare different combinations of search configuration options. For instance, the application can be used to:

- Ensure that spell-correction is behaving as expected, by comparing the results of a search done for a correctly-entered search term (such as "chardonnay") with the results of a search for a misspelled term (such as "chardonney")
- Compare the behavior of different match modes, such as "All > Partial" vs. "All > Any"
- Compare results from searches performed against slightly different search interfaces, such as one search interface that contains only short text properties (like "Name" and "Keywords"), and one interface that contains the same short text properties plus a longer "Description" field

It is recommended that a variety of common search terms be used for tests, whether the tests are designed to determine the optimal relevance ranking module for your dataset, or for another test case. A relevance ranking strategy that seems to be perfect for one particular search term may not be appropriate for others, so it is important to test several different types of searches.

Rollup Key

In addition, as part of Search Criteria, it is possible to submit searches as aggregated record navigation queries. The interface presents a drop-down with the available rollup keys defined in the index. Select one of these keys to submit the search as an aggregated record navigation query.

Search Interface	Match Mode	Search Term(s)	Rollup Key
All ▼	All > Partial ▼		None ▼ None product.code

Aggregated queries and records are further described in detail in MDEX Engine documentation.

Search Filters

Though hidden by default, the Relevance Ranking Evaluator allows for the configuration of three types of query-based filters: record filters, range filters and Endeca Query Language (EQL) filters.

Search Filters		
Record Filter (Nr)	Range Filter (Nr)	EQL Filter (Nrs)
<input type="text"/>	Select Attribute -- <input type="text"/> Select Function -- <input type="text"/>	<input type="text"/>

Record Filters

If desired, the user can enter a record filter that limits the records returned and restrict the scope of the search results. This is useful when testing relevance ranking strategies for a particular navigation state, for example, restricting searches to just those records that are (e.g.) in stock or available, or when using a dataset that contains multiple types of records (e.g. both products and reviews).

Examples of possible record filters that can be entered are:

- **AND(Wine Type:Red)** – To show/search only those records that have a Wine Type of red
- **AND(8032)** – To show/search only those records that are tagged with the dimension value ID 8032 (Price: under \$10)
- **AND(8032,Wine Type:Red)** – To show/search only those records that are tagged with the dimension value ID 8032 and which have a Wine Type of red

Record filters are described in detail, along with examples, in the MDEX Engine documentation.

Range Filter

A range filter can be applied to numeric attributes to restrict the records returned. Though there is no limit in the Presentation API to the number of range filters that can be applied, in the Oracle Relevance Ranking Evaluator, only one range filter can be applied.

Endeca Query Language (EQL) Filter

An EQL filter can be applied to limit the records returned. Note that Record Relationship Navigation (RRN) can only be used if a proper license is acquired. As an EQL filter can make use of multiple criteria, this can be used in lieu of multiple range filters. Note that properties used in EQL must be enabled for record filters. In addition, the attribute names must be NC_NAME compliant. That is, they must not contain spaces. The power of EQL is the ability to include complex Boolean logic across various filter types. Here is an example of an EQL filter that returns usual restaurants located in the Fenway or Back Bay regions:

```
collection()/record[
  Type = "Casual" and
```

```
PriceRating < 10 and (
  Region = "Fenway" or
  Region = "Back Bay"
)]
```

Relevance Ranking Strategies

The "Relevance Ranking" area of the page is used to select the particular relevance ranking modules to be applied for the search. A set of relevance ranking modules, combined with the order in which they are evaluated, is referred to as a relevance ranking "strategy." For instance, the module "Field" followed by the module "Interp" comprises a different strategy than the module "Interp" followed by the module "Field."



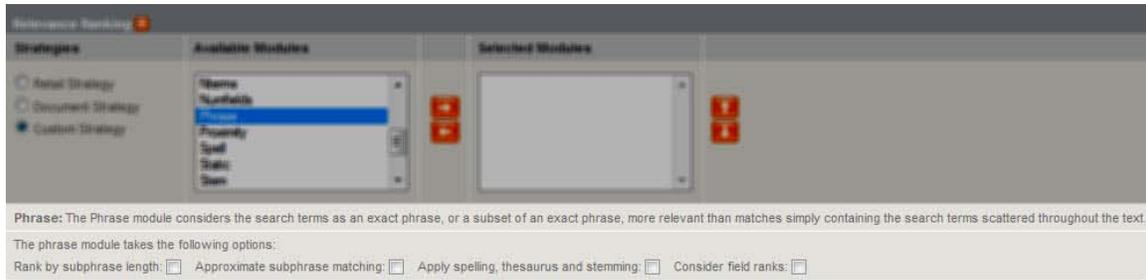
To configure a relevance ranking strategy, begin by single-clicking on a module from the "Available Modules" listing on the mid-left side of the Relevance Ranking area. When a module is selected, a brief description of that module's behavior will appear below the module listing box. You may view a description of any module by selecting it in this way:



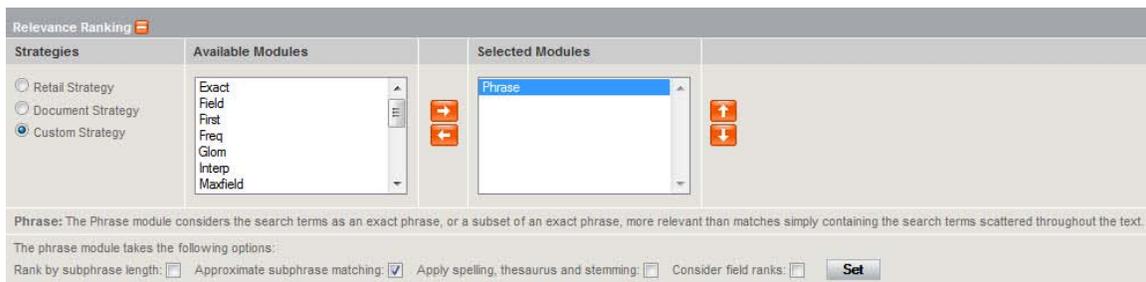
After identifying the first module you wish to use, click the right-pointing arrow or double-click the module name to move the module to the Selected Modules list. Continue to select modules in this way until you have defined your desired relevance ranking strategy. (Note that strategies containing more than 4 or 5 modules are unlikely to produce results that are significantly different from strategies with fewer modules; see below for an explanation of why this is true, and further discussion of the "tie-breaking" behavior of relevance ranking modules in a strategy.)

Note that certain relevance ranking modules, such as Exact, First, Freq, Nterms, Numfields, Phrase, Proximity, Static, and Stratify, may require or take additional configuration options depending on the MDEX Engine version used. These configuration options should be specified

after single-clicking the module from the left-hand list of Available Modules, but before moving the module to the right-hand list of Selected Modules.



You may also modify the selected configuration options for these modules after they have been moved into the Selected Modules listing, by single-clicking on the module in the Selected Modules list, modifying the configuration appropriately, and then clicking the "Set" button.



Static Module

For the Static module, the Relevance Ranking Evaluator will, by default, allow you to select from a drop-down list containing all of your data's properties. To modify the list of properties presented for selection by the Static module, please see the [Optional Relevance Ranking Evaluator Configuration](#) section below. For further information regarding the available configuration options for these modules, please refer to the Developer Studio Help or Appendix A of this guide.

Stratify Module

For the Stratify module, it is required to specify a proper Endeca Query Language (EQL) clause, such as "collection()/record[P_Score>90],*,collection()/record[P_Score<50]", which boosts records with P_Score values higher than 90 and buries records with P_Score values less than 50. Note: the application makes no attempt to validate the EQL clause. It is assumed to be syntactically valid.

Ordering Relevance Ranking Modules

To change the order of already-selected relevance ranking modules, single-click on a module in the right-hand Selected Modules listing, and then use the up and down buttons on the far right to move the module up or down in the strategy listing.

Relevance Ranking 			
Strategies	Available Modules		Selected Modules
<input type="radio"/> Retail Strategy <input type="radio"/> Document Strategy <input checked="" type="radio"/> Custom Strategy <input type="radio"/> Stored Strategy	<div style="border: 1px solid gray; padding: 2px;"> Exact Field First Freq Glom Interp Maxfield </div>	 	<div style="border: 1px solid gray; height: 80px; width: 100%;"></div>

The order in which relevance ranking modules are specified determines their precedence when ordering search results. Relevance ranking modules act in combination by functioning as a series of tie-breakers. The first module specified groups results into a number of different strata based on its rules. Then, within each stratum produced by the first module, the second module further subdivides results into follow-on strata based on its rules, and so on, until there are no further ties or there are no more modules to apply.

For some searches, there may appear to be no difference between two strategies that share the same first module but have different second modules; the most likely cause of this is that the first module grouped each resultant record individually, and resulted in no "ties" to be broken. Because each additional module in a strategy results in smaller and smaller strata, the modules listed at the bottom of a strategy containing more than 4 or 5 modules are unlikely to have much an effect on the order of the search results. To fully understand which modules are evaluated, it is recommended to use MDEX Engine version 6.2 or higher as with this version additional information on relevance is ranking not previously available is returned per query and per record.

Pre-defined and Stored Strategies

The Ranking Evaluator provides pre-defined relevance ranking strategies that have proven to be useful starting points for specific application types. To select a pre-defined strategy, such as Retail Strategy or Document Strategy, click on the radio button to the left of the strategy name. If you make modifications to a strategy or create your own strategy, the "Custom" strategy choice will be automatically selected. If desired, you can then reset your choice to either of the pre-defined strategies by clicking on the appropriate radio button.

The Retail Strategy is a specific set of relevance ranking modules that provide a result order that is favored by retail sites that contain only short text fields, while the Document Strategy is useful for sites that contain larger text documents. Note that it is unrealistic to use these strategies as your own in production; these are considered starting points from which you can customize to your application's needs.

Both the Retail and Document strategies are examples of stored strategies, which are set and configured in the application's properties file, evaluator.properties. The administrator has the ability to set an arbitrary number of stored strategies and can modify or delete the default Retail and Document strategies as desired. Once defined, these stored strategies are made available to all application users. Typically, stored strategies are defined when the application is deployed, or after an initial amount of testing is done to warrant their inclusion.

Working with Multiple Searches

After creating one search strategy, you may wish to compare your results with a slightly different search strategy. To this end, click on the tab with the plus sign (“+”) at the top of the page. A mouseover description informs the user that clicking here adds a new search tab.



Once clicked, a new search tab is created, pre-populated with the values from the search viewed when clicking the link. You may then modify this new search as desired, using the same controls and options described above. When you have completed configuring the new search, click the Go button to run the new search and display the results, or click the tab with the plus sign (“+”) button again to create and configure another new search.

To return to the configuration page for an existing search, click on the tab at the top of the page which contains the name of the search you wish to modify. Once you have made the desired changes, click the Go button to save the changes and re-run the modified search.

Though it is required to have at least one search tab, if there are multiple search tabs, the user is able to remove an existing search by clicking the X next to the search name in its tab.



If you wish to delete all configured searches and start over, click the "Clear All" button. This button will prompt for confirmation before removing all previously-configured search tabs.

Again, note that the search configurations set in the Relevance Ranking Evaluator are not stored outside of the user's browser session; if the browser window is closed or if the session is allowed to time out due to inactivity, the tests will be cleared and the application will be reset to its initial state.

Viewing Results

After one or more searches have been configured, press the Go button towards the top of the screen to run the search(es) and display the results.

Setting the Number of Records Displayed

The number of records displayed can be set with using the Show dropdown control at the top of the page.

Search Summary Information

Each set of search results is displayed in its own column in the "Search Results Comparison" section at the bottom of the screen, with its header listing the name of the search and the number of records matched. To help the user better understand the results, each search results header provides detailed information about the search performed. To see this information, simply click on the orange plus sign icon to the right of the search name and results count. Note that due to feature enhancements in MDEX Engine version 6.2, this information is significantly more detailed when the application works with and MDEX Engine of this version or later. In addition, to foster comparison, clicking any top-level orange plus sign shows all search results' detailed information

Search Results Comparison			
Show All Properties and Dimensions +	Default (10 records) - X		Custom 1 (10 records) - X
	Search Terms: cherry bright zinfandel Search Interface: All + Match Mode: matchallpartial Approximate Search Time: 1,292 ms Word Interpretations: <ul style="list-style-type: none"> cherry = cherries zinfandel = zin, zinfandels Relevance Ranking: All's defined strategy + <ol style="list-style-type: none"> Nterms - 10 results evaluated Maxfield - 10 results evaluated Glom - 10 results evaluated Exact - 10 results evaluated Static - 10 results evaluated 		Search Terms: cherry bright zinfandel Search Interface: All + Match Mode: matchallpartial Approximate Search Time: 1,349 ms Word Interpretations: <ul style="list-style-type: none"> cherry = cherries zinfandel = zin, zinfandels Relevance Ranking: Custom strategy + <ol style="list-style-type: none"> Nterms - 10 results evaluated Maxfield - 10 results evaluated Glom - 10 results evaluated Exact (considerFieldRanks) - 10 results evaluated Stratify (<ul style="list-style-type: none"> collection()/record[P_Score>90], * collection()/record[P_Score<50]) - 10 results evaluated Static (P_SalesRank, descending) - 9 results evaluated

Base Information

Regardless of MDEX Engine version, the following information is returned per search:

- Search name
- Number of records matched
- Search terms
- Search Interface
- Match mode
- Approximate Search Time
- Word Interpretations (thesaurus, spelling corrections, stemming)
- Relevance ranking modules applied, including configured options

Information Provided for MDEX Engine Version 6.2 and Higher

Utilizing MDEX Engine version 6.2 and later provides a more detailed set of information. This includes:

- Search interface field ranks, when these are pertinent for relevance ranking
- Number of results evaluated per module
- Number of results evaluated per stratum created by applied relevance ranking module
- Evaluation time of each relevance ranking module

Record Information

The number of records displayed for the search is based upon the Show <N> control at the top of the application.

For MDEX Engine versions 6.1.4 and lower, by default only the MDEX Engine's defined Record Identifier, the determined Relevance Ranking Score and the Why Did It Match? properties are initially displayed for each record, assuming they are available. For MDEX Engine versions 6.2 and higher, by default, the Why Did it Rank Here? is returned by default as well.

Note that the Relevance Ranking Score property does not represent a percentage score; it is simply an internal calculation that the MDEX Engine uses to determine ranking. A higher number equates to greater relevancy based on the applied strategy. See the [Optional MDEX Engine Configuration](#) section above for more information about configuring your MDEX Engine to return relevance ranking score and Why Did It Match? information. Note that Why Did it Match? and Why Did It Rank Here? properties are returned by default for MDEX Engine version 6.2 and higher without any additional configuration

To display additional attributes for each record, click on the Show All Properties And Dimensions link, which expands the results display area and exposes all available properties and dimensions for the records. To modify the list of exposed attributes, including their order, see the [Optional Relevance Ranking Evaluator Configuration](#) section. When all attributes are being displayed, you may temporarily hide individual attributes by clicking on the "X" icon to the right of each attribute name. Note that this doesn't actually delete the attribute from display permanently.

Search Results Comparison			
Hide All Properties and Dimensions 	Default (10 records) 	X Custom 1 (10 records) 	X
Record 1			
Record Identifier	71984	62392	
Relevance Ranking Score	X 4281614250 (10 records with this score)	4282812975 (1 record with this score)	
Why Did It Match?	X 1 Match Field Matched: P_Description Terms Matched: "bright", "cherry", "zinfandel"	1 Match Field Matched: P_Description Terms Matched: "bright", "cherry", "zinfandel" (thesaurus)	
Why Did It Rank Here?	X 5 Modules Evaluated  1. Nterms: matched 3 of 3 terms 2. Maxfield: field match (P_Description) 3. Glom: single-field match 4. Exact: no exact match and no phrase match 5. Static: sorted by P_Price (descending)	5 Modules Evaluated  1. Nterms: matched 3 of 3 terms 2. Maxfield: field match (P_Description) 3. Glom: single-field match 4. Exact: no exact match and no phrase match 5. Stratify: EQL match (collection()/record[P_Score>90])	
Body	X Bright Firm Firm Tannins Rich Ripe Tannins	Bright Elegant Ripe Smooth Supple	
Designation	X	Spectator Selection	
Drinkability	X Drink now through 2002		
Flavors	X Anise Berry Cedar Cherry	Berry Cherry Currant Juicy	

To hide all properties, click on the "Hide All Properties and Dimensions" link, which replaces the "Show All Properties and Dimensions" link when additional properties are being shown.

Comparing Search Results

With the Relevance Ranking Evaluator, you can compare search results, including where a given record is ranked in each set. This is done simply by mousing over the displayed properties for the record in which you are interested. When your mouse is placed over the record's properties, the record is highlighted in any search in which it is displayed. Additionally, a tooltip appears that provides the specific location of the record in each set of search results

Record 2		
Record Identifier	72091	39670
Relevance Ranking Score	4281614250 (10 records with this score)	4282783510 (9 records with this score)
Why Did It Match?	1 Match Field Matched: P_Description Terms Matched: "bright", "cherry", "zinfandel" (thesaurus)	1 Match Field Matched: P_Description Terms Matched: "bright", "cherry", "zinfandel" (thesaurus)
Why Did It Rank Here?	5 Modules Evaluated 1. Nterms: matched 3 of 3 terms 2. Maxfield: field match (P_Descri 3. Glom: single-field match 4. Exact: no exact match and no phrase match 5. Static: sorted by P_Price (descending)	6 Modules Evaluated 1. Nterms: matched 3 of 3 terms 2. Maxfield: field match (P_Description) 3. Glom: single-field match 4. Exact: no exact match and no phrase match 5. Stratify: no match 6. Static: sorted by P_SalesRank (descending)
Record 3		
Record Identifier	62392	72091
Relevance Ranking Score	4281614250 (10 records with this score)	4282783510 (9 records with this score)
Why Did It Match?	1 Match Field Matched: P_Description Terms Matched: "bright", "cherry", "zinfandel" (thesaurus)	1 Match Field Matched: P_Description Terms Matched: "bright", "cherry", "zinfandel" (thesaurus)
Why Did It Rank Here?	5 Modules Evaluated 1. Nterms: matched 3 of 3 terms 2. Maxfield: field match (P_Description) 3. Glom: single-field match 4. Exact: no exact match and no phrase match 5. Static: sorted by P_Price (descending)	6 Modules Evaluated 1. Nterms: matched 3 of 3 terms 2. Maxfield: field match (P_Description) 3. Glom: single-field match 4. Exact: no exact match and no phrase match 5. Stratify: no match 6. Static: sorted by P_SalesRank (descending)

This functionality can be used, for example, to determine where the top-ranked record from one search falls when the search's relevance ranking strategy is modified.

Starting Over

To clear all search tabs to start anew, simply click the Clear All button at the top of the application. Also note that closing your browser window or allowing your session to time out due to inactivity will delete all search strategies.

Optional Relevance Ranking Evaluator Configuration

The Relevance Ranking Evaluator offers additional configuration options through the use of an application properties file. These are geared towards overall application behavior changes and tend to be for specific use cases, such as deployments with very large data sets, very large number of properties, or the desire to display record images. Some of the optional configuration described in this section may be necessary if you experience performance issues when using the Relevance Ranking Evaluator, or if you want more control over certain behaviors.

Each option is described below, and can be set or changed by modifying the file "evaluator.properties", located in the "WEB-INF\conf" subdirectory of the Oracle Endeca Relevance Ranking Evaluator's final installation directory. If you have installed the Oracle Endeca Relevance Ranking Evaluator so that it uses the Endeca Tools Service's application server, this file can generally be found in the following location:

```
C:/Endeca/Solutions/relevance Ranking Evaluator-  
[VERSION]/relevance Ranking Evaluator/WEB-INF/conf/evaluator.properties
```

If you have copied the "relrankRelevance Ranking Evaluator" directory to a different location, such as into the "webapps" directory of an existing application server, the "evaluator.properties" file will be located in the "WEB-INF/conf" subdirectory of the "relrankRelevance Ranking Evaluator" directory.

The "evaluator.properties" file can be opened and edited with any text editor (for example, Notepad). Changes made to the "evaluator.properties" file do not require a restart of the Endeca HTTP Service or other application server in order to take effect. The "evaluator.properties" file contains the following configuration options:

MDEX_VERSION

As the MDEX Engine's relevance ranking capabilities have changed with releases, it may be helpful for application users to not have options presented to them that are invalid for their version number. For example, the Stem, Stratify and Thesaurus modules work only for MDEX version 6.1.4 and higher. If you are using an earlier release of the product and want to hide these modules as options, simply enter your version number here as a three digit number, such as "510", "483", "612", etc. The default behavior of the Evaluator shows all available options.

Example: To display only the application modules relevant for your version, set this to your three-digit MDEX version number, without dots. For example, if you are using MDEX 6.1.2, add the following line to the evaluator.properties file:

```
MDEX_VERSION=612
```

DISPLAYED_ATTRIBUTES

If your data has an extremely large number of properties and dimensions on each Endeca record, or if you wish to limit the properties that are displayed within the Relevance Ranking Evaluator's Results area, the "DISPLAYED_ATTRIBUTES" configuration option can be set. This option should be listed on a single line within the evaluator.properties file, and should consist of a comma-delimited list of property and dimension names. The dimension and property names listed are displayed within the Search Results Comparison area when a user clicks on "Show All Dimensions and Properties." If you receive an error or experience performance issues when using the Relevance Ranking Evaluator, and your data contains a large number of properties, it may be necessary to set this option.

Example: To display only the "P_DateReviewed" and "P_Description" properties in the Relevance Ranking Evaluator's Results area, add the following line to the evaluator.properties file:

```
DISPLAYED_ATTRIBUTES=P_DateReviewed,P_Description
```

MAX_DISPLAYED_ATTRIBUTES

If you have a large number of dimensions and properties, MAX_DISPLAYED_ATTRIBUTES should be set to restrict the number returned for display. If this is not set, the default is set to 100, meaning that only the first 100 dimensions and properties will be viewable in the application.

Example: To show only the first 50 properties and dimensions, add the following line to the evaluator.properties file

```
MAX_DISPLAYED_ATTRIBUTES=50
```

STATIC_MODULE_PROPERTIES

If your data has an extremely large number of properties, or if you wish to limit the properties that are available for selection when choosing a property for the "Static" relevance ranking module, the "STATIC_MODULE_PROPERTIES" option can be set. This option should be listed on a single line within the evaluator.properties file, and should consist of a comma-delimited list of property names. The property names listed will be the properties that are displayed for selection when a user configures the "Static" relevance ranking module. If you receive an error or experience performance issues when attempting to configure the "Static" relevance ranking module, it may be necessary to set this option.

Example: To allow only the "P_Name" and "P_Description" properties to be selected for use in the "Static" relevance ranking module, add the following line to the evaluator.properties file:

```
STATIC_MODULE_PROPERTIES=P_Name,P_Description
```

RECORDS_TO_DISPLAY_OPTIONS

If you would like to display a different number of records than the default set of options provides (10, 25 and 50), the "RECORDS_TO_DISPLAY_OPTIONS" property can be set. This option should be listed on a single line within the evaluator.properties file, and should consist of a s.

Example: To give the user the ability to show 10, 25, 50 and 100 records, add the following line to the evaluator.properties file:

```
RECORDS_TO_DISPLAY_OPTIONS=10|25|50|100
```

DEFAULT_ROLLUP_KEY

Though the application allows users to specify the aggregated record rollup key, a default setting is able to be set as well.

Example: To have the default rollup key for your records be p_id, simply add this line to the evaluator.properties file:

```
DEFAULT_ROLLUP_KEY=p_id
```

STORED_STRATEGY_<N>

You can include an arbitrary number of pre-defined search strategies in the Relevance Ranking Evaluator. By default, the application ships with example Retail and Document strategies, which can be modified, deleted or extended.

In order to function properly, each stored strategy must be distinct in both module order and module options enabled. Each strategy must be identified by "STORED_STRATEGY_<N>",

ending with its numeric identifier, which must start at 1 and increment by 1 for each additional strategy.

Example: To have two strategies added to the application, add the following pipe-delimited lines to the evaluator.properties file. Note that these are example strategies and may not be appropriate for your application :

```
STORED_STRATEGY_1=stem|thesaurus|stratify(collection()/record[P_Score>90],*,collection()/record[P_Score<50])|nterms(considerFieldRanks)|maxfield|glom|phrase(s
ubphrase,query_expansion,considerFieldRanks)|static(P_Price,descending)
STORED_STRATEGY_2=nterms(considerFieldRanks)|maxfield|glom|exact(considerFieldR
anks)
```

STORED_STRATEGY_<N>_NAME

If using stored strategies as described in the previous section, include these properties to label them. Each strategy defined as a STORED_STRATEGY must have a corresponding name defined via this property.

Example: To specify names for the STORED_STRATEGY_1 and STORED_STRATEGY_2 examples above, include the following in evaluator.properties:

```
STORED_STRATEGY_1_NAME=Mike's Stratify Strategy
STORED_STRATEGY_2_NAME=Document Strategy
```

ORDER_ATTRIBUTES

This option allows the application administrator to set the order for the properties to display to be the value set for DISPLAYED_ATTRIBUTES. By default, the application shows properties in alphabetical order.

Example: To display properties in the order specified by the setting for DISPLAYED_ATTRIBUTES, add the following line to the evaluator.properties file:

```
ORDER_ATTRIBUTES=true
```

IMAGE_DISPLAY_TEMPLATE

Used in conjunction with IMAGE_PROPERTY_NAME, this option allows administrators to specify a URL format by which to retrieve images to display for the records.

Example: If the default image path URL is [http://www.acme.com/image/products/123456\\$thumbnail\\$](http://www.acme.com/image/products/123456$thumbnail$), where "123456" is the image id as per the value of IMAGE_PROPERTY_NAME, then the following line should be added to the evaluator.properties file:

```
IMAGE_DISPLAY_TEMPLATE =
http://www.acme.com/image/products/###IMAGE###?$thumbnail$
```

IMAGE_PROPERTY_NAME

Used in conjunction with IMAGE_DISPLAY_TEMPLATE, this option allows administrators to specify a URL format by which to retrieve images to display for the records.

Example: If the default image path URL is [http://www.acme.com/image/products/123456\\$thumbnail\\$](http://www.acme.com/image/products/123456$thumbnail$), where "http://www.acme.com/image/products/###IMAGE###?\$thumbnail\$" is the value of IMAGE_DISPLAY_TEMPLATE, then the following line should be added to the evaluator.properties file:

```
IMAGE_PROPERTY_NAME = P_Product_Image
```

Deploying a Relevance Ranking Strategy

The Relevance Ranking Evaluator does not provide an automatic method for deploying a selected relevance ranking strategy within another application. Therefore, once the tool has been used to determine an appropriate relevance ranking strategy, it is necessary to manually set that strategy as the default for your search interface, in order for it to be used by your MDEX Engine.

To deploy a relevance ranking strategy, manually record which modules were selected and in which order, being sure to include any optional configuration information for modules such as Stratify, Phrase and Static.

Note that modules such as Stratify, Stem and Thesaurus as well as the considerFieldRanks option are only available to be configured in the application tier and are not supported in Developer Studio.

Appendix A: Relevance Ranking Modules

The following are descriptions of the different relevance ranking modules. Please refer to the MDEX Engine documentation for further details about relevance ranking behavior in the Endeca MDEX Engine.

Exact

The Exact module provides a finer grained (but more computationally expensive) alternative to the Phrase module. The Exact module groups results into three strata based on how well they match the query string:

- The highest stratum contains results whose complete text matches the user's query exactly.
- The middle stratum contains results that contain the user's query as an exact substring.
- The lowest stratum contains other hits (such as normal conjunctive matches). Any match that would not be a match without query expansion lands in the lowest stratum.

Note: The Exact module is computationally expensive, especially on large text fields. It is intended for use only on small text fields (such as dimension values or small property values like part IDs). This module should not be used with large or offline documents (such as FILE or ENCODED_FILE properties). Use of this module in these cases will result in very poor performance and/or application failures due to request timeouts. The Phrase module, with and without approximation turned on, does similar but less sophisticated ranking that can be used as a higher performance substitute.

Field

The Field module ranks documents based on the search interface field with the highest priority in which it matched. Only the best field in which a match occurs is considered. The Field module is often used in relevance ranking strategies for catalog applications, because the category or product name is typically a good match.

Field assigns a score to each result based on the static rank of the dimension or property member or members of the search interface that caused the document to match the query.

In Developer Studio, static field ranks are assigned based on the order in which members of a search interface are listed in the Search Interfaces view. The first (left-most) member has the highest rank.

By default, matches caused by cross-field matching are assigned a score of zero. The score for cross-field matches can be set explicitly in Developer Studio by moving the <<CROSS_FIELD>> indicator up or down in the Selected Members list of the Search Interface editor. The <<CROSS_FIELD>> indicator is available only for search interfaces that have the Field module and are configured to support cross-field matches.

All non-zero ranks must be non-equal and only their order matters. For example, a search interface might contain both Title and DocumentContent properties, where hits on Title are considered more important than hits on DocumentContent (which in turn are considered more important than <<CROSS_FIELD>> matches). Such a ranking is implemented by assigning the highest rank to Title, the next highest rank to DocumentContent, and setting the <<CROSS_FIELD>> indicator at the bottom of the Selected Members list in the Search Interface editor.

Notes:

- The Field module is only valid for record search operations. This module assigns a score of zero to all results for other types of search requests.
- Field treats all matches the same, whether or not they are due to query expansion.

Freq (Frequency)

The Frequency (Freq) module provides result scoring based on the frequency (number of occurrences) of the user's query terms in the result text. Results with more occurrences of the user search terms are considered more relevant.

The score produced by the Freq module for a result record is the sum of the frequencies of all user search terms in all fields (properties or dimensions in the search interface in question) that match a sufficient number of terms. The number of terms depends on the match mode—all terms in a "matchall" query, a sufficient number of terms in a "matchpartial" query, and so on. Cross-field match records are assigned a score of zero. Total scores are capped at 1024—in other words, if the sum of frequencies of the user search terms in all matching fields is greater than or equal to 1024, the record gets a score of 1024 from the Freq module.

For example, suppose we have the following record:

```
{ Title="test record", Abstract="this is a test", Text="one test this is" }
```

A "matchall" search for "test this" would cause Freq to assign a score of 4, since "this" and "test" occur a total of 4 times in the fields that match all search terms (Abstract and Text, in this case). The number of phrase occurrences (just one in the Text field) doesn't matter—only the sum of the individual word occurrences. Also note, the occurrence of "test" in the Title field doesn't contribute to the score, since that field didn't match all of the terms.

A "matchall" search for "one record" would hit this record, assuming that cross field matching was enabled. But the record would get a score of zero from Freq, since no single field matches all of the terms.

Note: Freq ignores matches due to query expansion (that is, such matches are given a rank of 0).

Glom

The Glom module ranks single-field matches ahead of cross-field matches. This module serves as a useful tie-breaker function in combination with the Maximum Field module. It is only useful in conjunction with record search operations.

Note: Glom treats all matches the same, whether or not they are due to query expansion.

Interp (Interpreted)

Interpreted (Interp) is a general-purpose module that assigns a score to each result record based on the query processing techniques used to obtain the match. Matching techniques considered include partial matching, cross-attribute matching, spelling correction, thesaurus, and stemming matching.

Specifically, the Interpreted module ranks results as follows:

- All non-partial matches are ranked ahead of all partial matches.
- Within the above strata, all single-field matches are ranked ahead of all cross-field matches.
- Within the above strata, all non-spelling-corrected matches are ranked above all spelling-corrected matches.
- Within the above strata, all thesaurus matches are ranked below all non-thesaurus matches.
- Within the above strata, all stemming matches are ranked below all non-stemming matches.

Maxfield (Maximum field)

The Maximum Field (Maxfield) module behaves identically to the Field module, except in how it scores cross-field matches. Unlike Field, which assigns a static score to cross-field matches, Maximum Field selects the score of the highest-ranked field that contributed to the match.

Notes:

- Because Maximum Field defines the score for cross-field matches dynamically, it does not make use of the <<CROSS_FIELD>> indicator set in the Search Interface editor.
- The Maxfield module is only valid for record search operations. This module assigns a score of zero to all results for other types of search requests.
- Maxfield treats all matches the same, whether or not they are due to query expansion.

Nterms (Number of terms)

The Number of Terms (or Nterms) module ranks matches according to how many query terms they match. For example, in a three-word query, results that match all three words will be ranked above results that match only two, which will be ranked above results that match only one.

Notes:

- The Nterms module is only applicable to search modes where results can vary in how many query terms they match. These include MatchAny, MatchPartial, MatchAllAny, and MatchAllPartial.
- Nterms treats all matches the same, whether or not they are due to query expansion.

Numfields (Number of fields)

The Number of Fields (Numfields) module ranks results based on the number of fields in the associated search interface in which a match occurs. Note that we are counting whole-field rather than cross-field matches. Therefore, a result that matches two fields matches each field completely, while a cross-field match typically does not match any field completely.

Notes:

- Numfields treats all matches the same, whether or not they are due to query expansion.
- The Numfields module is only useful in conjunction with record search operations.

Phrase

The Phrase module states that results containing the user's query as an exact phrase, or a subset of the exact phrase, should be considered more relevant than matches simply containing the user's search terms scattered throughout the text.

The Phrase module has three options that can be used to customize its behavior:

- Rank by subphrase length
- Use approximate subphrase/phrase matching
- Apply expansions, such as stemming, thesaurus and spelling correction
- Consider field ranks

When you add the Phrase module in the Relevance Ranking Modules editor, you are presented with an editor that allows you to set this option.

Using Approximate Matching

Approximate matching provides higher-performance matching, as compared to the standard Phrase module, with somewhat less exact results. With approximate matching enabled, the

Phrase module looks at a limited number of positions in each result that a phrase match could possibly exist, rather than all the positions. Only this limited number of possible occurrences is considered, regardless of whether there are later occurrences that are better, more relevant matches. The approximate setting is appropriate in cases where the runtime performance of the standard Phrase module is inadequate because of large result contents and/or high site load.

Spell

The Spell module ranks true matches ahead of spelling-corrected matches.

Note: Spell assigns a rank of 0 to matches from spelling correction, and a rank of 1 from all other sources. That is, it ignores all other sorts of query expansion.

Static

The Static module assigns a static or constant data-specific value to each search result, depending on the type of search operation performed and depending on optional parameters that can be passed to the module:

For record search operations, the first parameter to the module specifies a property, which will define the sort order assigned by the module. The second parameter can be specified as ascending or descending to indicate the sort order to use for the specified property.

For example, using the module **Static(Availability,descending)** would sort result records in descending order with respect to their assignments from the Availability property. Using the module **Static(Title,ascending)** would sort result records in ascending order by their Title property assignments. Note: In a catalog application, setting the static module by Price, descending leads to more expensive products being displayed first.

For dimension search, the first parameter can be specified as nbins, depth, or rank:

- Specifying nbins causes the static module to sort result dimension values by the number of associated records in the full data set.
- Depth causes the static module to sort result dimension values by their depth in the dimension hierarchy.
- Rank causes dimension values to be sorted by the ranks assigned to them for the application.

Notes:

- The Static module is not compatible with the Agraph.
- Static treats all matches the same, whether or not they are due to query expansion.

Stem

Available with MDEX Engine version 6.1.4, the Stem module ranks matches due to stemming below other kinds of matches.

Stem assigns a rank of 0 to matches from stemming, and a rank of 1 from all other sources. That is, it ignores all other sorts of query expansion

Stratify

Available with MDEX Engine 6.1.4, the Stratify module is used to boost or bury records in the result set.

The Stratify module takes one or more EQL (Endeca Query Language) expressions and groups results into various strata. Records are placed in the stratum associated with the first EQL expression they match. The first stratum is the highest ranked, the next stratum is next-highest ranked, and so forth. If an asterisk is specified instead of an EQL expression, unmatched records are placed in the corresponding stratum.

The Stratify module is the basic component of the record boost and bury feature, which is described in MDEX Engine documentation.

Thesaurus

Available with MDEX Engine version 6.1.4, the Thesaurus module ranks matches due to thesaurus entries below other sorts of matches.

Thesaurus assigns a rank of 0 to matches from the thesaurus, and a rank of 1 from all other sources. That is, it ignores all other sorts of query expansion.

Wfreq (Weighted frequency)

Like the Frequency module, the Weighted Frequency (Wfreq) module scores results based on the frequency of user query terms in the result. Additionally, the Weighted Frequency module weights the individual query term frequencies for each result by the information content (overall frequency in the complete data set) of each query term. Less frequent query terms (that is, terms that would result in fewer search results) are weighted more heavily than more frequently occurring terms.

Notes:

- Weighted Frequency values are capped at 1024.
- Wfreq ignores matches due to query expansion (that is, such matches are given a rank of 0)