Oracle Endeca Platform Services

Log Server and Report Generator Guide

Version 6.1.4 • June 2013

ORACLE®

ENDECA

Contents

Copyright and disclaimer

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.

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

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.

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

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

Preface

The Oracle Endeca Commerce solution enables your company to deliver a personalized, consistent customer buying experience across all channels — online, in-store, mobile, or social. Whenever and wherever customers engage with your business, the Oracle Endeca Commerce solution delivers, analyzes, and targets just the right content to just the right customer to encourage clicks and drive business results.

Oracle Endeca Commerce is the most effective way for your customers to dynamically explore your storefront and find relevant and desired items quickly. An industry-leading faceted search and Guided Navigation solution, Oracle Endeca Commerce enables businesses to help guide and influence customers in each step of their search experience. At the core of Oracle Endeca Commerce is the MDEX Engine™, a hybrid search-analytical database specifically designed for high-performance exploration and discovery. The Endeca Content Acquisition System provides a set of extensible mechanisms to bring both structured data and unstructured content into the MDEX Engine from a variety of source systems. Endeca Assembler dynamically assembles content from any resource and seamlessly combines it with results from the MDEX Engine.

Oracle Endeca Experience Manager is a single, flexible solution that enables you to create, deliver, and manage content-rich, cross-channel customer experiences. It also enables non-technical business users to deliver targeted, user-centric online experiences in a scalable way — creating always-relevant customer interactions that increase conversion rates and accelerate cross-channel sales. Non-technical users can control how, where, when, and what type of content is presented in response to any search, category selection, or facet refinement.

These components — along with additional modules for SEO, Social, and Mobile channel support — make up the core of Oracle Endeca Experience Manager, a customer experience management platform focused on delivering the most relevant, targeted, and optimized experience for every customer, at every step, across all customer touch points.

About this guide

This guide describes how to use the Endeca Logging API, implement the Endeca logging and reporting system in Oracle Endeca Workbench, and generate customized reports from logs of activity on your site.

Who should use this guide

This guide is intended for programmers using the Endeca Logging API to implement logging in a Web application with Java or .NET, and for merchandisers customizing reports based on the generated logs.

Conventions used in this guide

This guide uses the following typographical conventions:

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

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

Contacting Oracle Support

Oracle 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 Support through Oracle's Support portal, My Oracle Support at *https://support.oracle.com*.

Chapter 1

Introduction to Logging and Reporting

This section introduces the components of the Endeca logging and reporting system, and describes the system's architecture.

About logging and reporting

The Endeca Log Server is a stand-alone server that translates application logging requests into flat log files. These files are later streamed through the Report Generator and transformed into configurable reports.

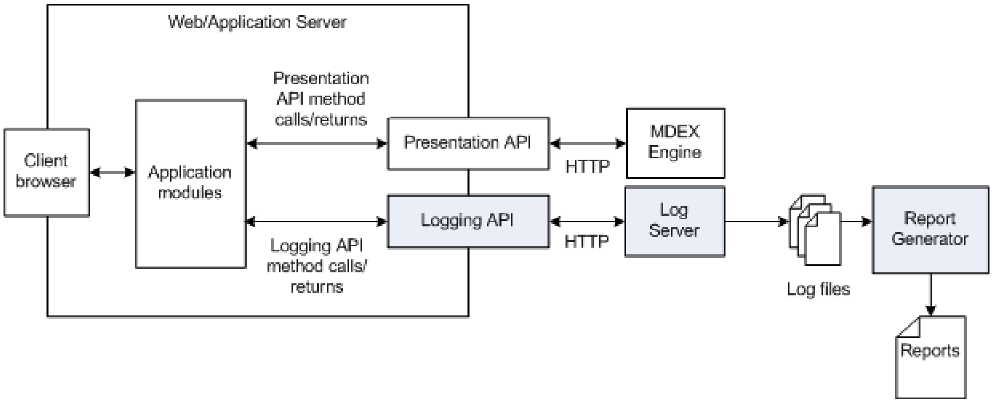
The Report Generator reads the log files created by the Log Server and creates XML-based or HTML-based reports that can be displayed on the View Reports tool in Workbench. These reports allow you to look at what has happened on your Web site on a daily or weekly basis. The information in these reports can help answer the following questions:

- How much search and navigation traffic is my site getting?
- How are visitors searching and browsing the site?
- What conversion rates are occurring as a result of searching, navigating, and reacting to merchandising or content spotlighting?
- What are the most popular search terms and navigation requests?
- How effective are their searching and browsing techniques?

Logging and Reporting System architecture

The Endeca Logging and Reporting System provides an application-level logging solution.

1. The Logging API receives requests from the application modules, and passes them to the Log Server via HTTP.
2. The Log Server translates the log requests into log files.
3. The Report Generator uses the log files created by the Log Server to generate reports.



Logging API

The application is responsible for constructing and submitting all log requests to the Log Server via the Logging API. Log requests are completely independent of MDEX Engine requests. This architecture allows maximum flexibility when designing your application's logging functionality.

Log Server

The Log Server translates application logging requests into log files that the Report Generator can process. You can use Workbench to:

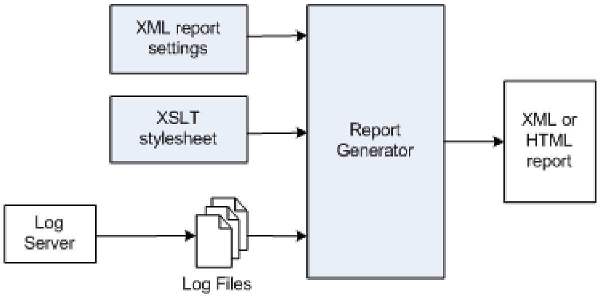
- Specify the port on which the Log Server listens.
- Start and stop the Log Server.
- Determine if the Log Server is running.

Report Generator

The Report Generator uses the log files created by the Log Server to construct HTML and XML-based reports. The HTML reports can be viewed directly in a Web browser. The XML reports can be displayed on the View Reports page in Workbench.

The Report Generator uses two files to do its job:

- Report settings file—Determines which reporting data, if any, is excluded from a report. Also sets chart display increments and session queue size.
- XSLT stylesheet—Determines whether the Report Generator creates an HTML or XML report.



About the Log Server

The Log Server logs information about your Endeca application to a log file. This file is later consumed by the Report Generator.

As soon as the Log Server starts, it attempts to open a log file and write a header and timestamp. If this fails, it exits immediately, without accepting any requests. The Log Server begins a new file when you issue the roll command, or automatically if the current file becomes larger than 1 GB.

The log file name is a combination of the current date and time and the log file prefix that you specify, in the format `prefix.timestamp`. You specify this when provisioning your Oracle Endeca Commerce implementation. The timestamp indicates when the particular file was started and makes it possible to distinguish among multiple log files. When you provision a Log Server, you specify a path and prefix for its output.

About the Report Generator

The Report Generator transforms the log files created by the Log Server into easy-to-read reports. These can be viewed in Oracle Endeca Workbench or elsewhere.

Report details

Reports help you make informed decisions about how your application is being used. For example, you can obtain information about searches that do not return desired results. By analyzing these searches, you can determine what aspects of your Oracle Endeca Commerce implementation may require changes.

Report generation has the following characteristics:

- You can enable both daily and weekly reports.
- If you schedule reports via Oracle Endeca Workbench, daily reports start at 12 am and finish at 11:59:59 pm. Weekly reports start at 12 am on the day that you specify and finish at 11:59:59 pm on the day that ends a week. For example, if you select Monday as the day to start your weekly report, your report includes logs from 12 am on Monday until 11:59:59 pm on the following Sunday.
- By default, the Endeca software saves generated reports to the Workbench directory `/workspace/reports/<application_name>` on UNIX and `\workspace\reports\<application_name>` on Windows. It is possible to specify an alternate reports directory. You can also customize the contents of your reports.

Requests and sessions

Requests and sessions are defined in part by your Endeca implementation.

An Endeca application creates log entries using the Endeca Logging API. The meaning of a query request in an Endeca report is partially defined by the implementation of your Endeca application. Endeca recommends that each log entry represent a distinct query request from an end-user.

Note: Request metrics in reports do not correspond directly to query load metrics for the MDEX Engine. Differences in request metrics can arise from pages that issue multiple queries and from caching. You can analyze query load metrics for the MDEX engine by using the Request Log

Analyzer to process the engine request logs. The Request Log Analyzer is installed with MDEX Engine 6.2.2 and later.

Sessions group together requests. Like requests, the specific meaning of a session is defined by the implementation of your Endeca application. Typically the Web server hosting your search application assigns a session ID to an end-user when he or she first accesses the Web site. (This session ID is independent of any search requests the user makes.) In most cases, it makes sense for a search application to use the session ID generated by the Web server. You can pass this session ID to the Endeca Logging API as a way to identify distinct search sessions.

Note: The num_requests_by_hour, num_requests_by_date, num_unique_sessions_by_hour, and num_unique_sessions_by_date report items are only produced if the Report Generator is run using the appropriate`--time-series` flag (“hourly” or “daily”). See the *Oracle Endeca Workbench Help* for information about setting the `--time-series flag` in Workbench.

Prerequisites

Before you implement logging and reporting, note the following requirements.

- In addition to running on the Oracle Endeca Workbench server, an EAC Agent must be installed on any host where you will be running the Log Server and Report Generator components.
- Report generation depends on the information collected in the log files. To enable logging, your development team should implement logging API calls to your application modules.

Chapter 2

Using the Endeca Logging API

The Endeca logging and reporting functionality requires that your Web application builds log requests and sends them to the Log Server. You incorporate this functionality into your Web application using the Logging API.

Logging API core classes

The Logging API uses a similar paradigm to the Presentation API, and is based on two core classes: the LogEntry class and the LogConnection class.

You use these two classes together to build a log request and send it to the Log Server:

- The LogEntry class serves as a container for key/value pairs that are sent to an Endeca Log Server in support of a log request.
- The LogConnection class enables connections with an Endeca Log Server.

About the LogEntry class

You use the `LogEntry` class to create a log entry consisting of one or more key/value pairs.

Most report items that the Report Generator can output require a `LogEntry` object with at least one key/value pair (although, some report elements do not require any key/value pairs). In general, each `LogEntry` corresponds to a single request (such as the number of search-only requests or the number of search requests returning no results).

For example, to output the Request and Session Summary report item, you must ensure that your log entries have a session ID key/value pair. The following code shows how to retrieve the session ID from the browser request object, and then use it to populate a `LogEntry` object. (It does not include the code for generating the sid.)

```
//Retrieve the session ID from the query string.
String sid = request.getParameter(“sid”);

//Create a log entry and populate it with the session ID.
LogEntry entry = new LogEntry();
entry.putString(“SESSION_ID”, sid);
```

A `LogEntry` can contain many key/value pairs, and they may or may not be related. For example, in the reference implementation’s `logging_functions` module, the key/value pairs required to generate all of the Endeca report elements are contained in a single `LogEntry` object.

This is good practice, for it allows the application to log a single `LogEntry` for each browser request, and then use that entry to create a complete set of report elements. You can use the logging_functions module as a reference for which key/value pairs are required for each Endeca report element.

Note: The reference implementation is located in `$ENDECA_REFERENCE_DIR` on Windows, or `%ENDECA_REFERENCE_DIR%` on UNIX. This is `C:\Endeca\PlatformServices\reference` on Windows, or `endeca/PlatformServices/reference` on UNIX.

LogEntry example in Java

```
//Retrieve the session ID from the query string.
String sid = request.getParameter(“sid”);

//Create a log entry and populate it with the session ID.
LogEntry entry = new LogEntry();
entry.putString(“SESSION_ID”, sid);
```

LogEntry example in .NET

```
String sid = Request.QueryString[“sid”];

//Create a log entry and populate it with the session ID.
LogEntry entry = new LogEntry();
entry.AddString(“SESSION_ID”, sid);
```

Notes

- If you have an existing set of key/value pairs in a Map, you can use the `LogEntry(java.util.Map)` constructor instead.
- For Java, if you have an existing set of key/value pairs in a Map, you can use the `LogEntry(Map)` constructor instead.
- For .NET, you can use `LogEntry(IDictionary)` instead.

About the LogConnection class

The `LogConnection` class functions as a repository for the hostname and port configuration for the Log Server you want to send the log request to.

The signature for a `LogConnection` constructor looks like this:

```
//Create a Log Server connection
LogConnection lsc = new LogConnection(logHost, logPort);
```

You call the following methods on a `LogConnection` object to establish a connection with a Log Server and send it a log request:

- The `log() method` sends a request to the Log Server but does not wait for a response.
- The `logAndWait()` method sends a log request to the Log Server and returns a boolean response that indicates whether the log entry was successful. Use this method if you need to make sure a log request is successful.
- The `logAsynchronously()` method queues up a log request to send to the Log Server asynchronously and without waiting for a response. Using this method can improve performance by preventing the Logging API from holding onto the thread while sending the log entry.

These methods use a LogEntry object as their argument when sending a log request to the Log Server. The following examples show the code to send a log request.

LogConnection example in Java

```
//Send the log request and do not wait for a response.
lsc.log(logEntryObject);

//Send the log request and wait for a response.
lsc.logAndWait(logEntryObject);

//Queue the log request and send it asynchronously.
lsc.logAsynchronously(logEntryObject);
```

LogConnection example in .NET

```
//Send the log request
lsc.log(logEntryObject);

//Send the log request and wait for the response.
lsc.logAndWait(logEntryObject);
```

Note: The instantiation of a `LogConnection` object does not open a persistent connection to the Log Server, nor does it initiate an HTTP socket connection. Instead, each issuance of the `log()` method opens an HTTP socket connection, which is closed after the log request has been sent. For the `logAndWait()` method, the HTTP socket connection remains open until the response indicating log entry success or failure is received. For the `logAsynchronously()` method, the HTTP socket is not opened until after the method has returned.

Using the core classes to send a log request

You use the two core classes described in the previous sections to build a log request and send it to the Log Server.

The following code shows how to build and execute a log request:

Log Request example in Java

```
//Create a Log Server connection
LogConnection lsc = new LogConnection(logHost, logPort);

//Create a log entry
LogEntry entry = new LogEntry();
entry.putString(key1, value1);
entry.putString(key2, value2);

//Send the request to the Log Server
lsc.log(entry);
```

Log Request example in .NET

```
//Create a Log Server connection
LogConnection lsc = new LogConnection(logHost, logPort);

//Create a log entry
LogEntry entry = new LogEntry();
entry.AddString(key1, value 1);
entry.AddString(key2, value 2);
```

```
//Send the request to the Log Server
lsc.log(entry);
```

About logging errors

This section describes how errors are logged by various methods.

The `log()` and `logAndWait()` methods throw a `LogException` if they encounter a problem while attempting to send a log request. The `logAsynchronously()` method throws a `LogException` only if it is called with a null argument; otherwise, it does not throw an exception because it operates asynchronously.

Chapter 3

Implementing Logging and Reporting in Oracle Endeca Workbench

The following sections discuss how you configure and run logging and reporting in Oracle Endeca Workbench. Oracle Endeca Workbench is a client of the Endeca Application Controller, which provides process coordination and control.

Implementing logging and reporting in a Web application

At a high level, you need to do the following tasks in order to implement the Endeca Logging and Reporting System:

- Add Logging API calls to your application modules. For the Endeca Application Controller environment, this is described in "Using the Endeca Logging API."
- Modify the report settings XML file if necessary. For the Endeca Application Controller environment, this is described in "Customizing report content and presentation."
- Configure and run the Log Server and Report Generator. You can provision and run these components using Oracle Endeca Workbench, or through the command line in EAC. These tasks are described in the *Oracle Endeca Application Controller Guide*.

Ways to implement logging and reporting in Oracle Endeca Workbench

You can implement and control the Log Server and Report Generator in Oracle Endeca Workbench in two ways.

The two methods are:

- You can configure and run each component and step individually.
- You can automate the end-to-end logging and reporting process by using the report generation script.

In either case, the output directory of the Log Server must not be the same directory as the input directory of the Report Generator in terms of provisioning. In other words, the output and input must reside in separate directories.

Oracle recommends using the Deployment Template for provisioning LogServer and ReportGenerator components in your implementation.

Note: You can also run logging and reporting with other Application Controller clients, such as the eaccmd tool or a custom Web services interface. For information about provisioning the LogServer and ReportGenerator components in an Application Controller provisioning file, see "Provisioning an Implementation with the Application Controller" in the *Oracle Endeca Application Controller Guide*. For eaccmd usage, see "Component and script control commands" in the Application Controller Guide. For API details, see the ""Endeca Application Controller API Class Reference" in the *Oracle Endeca Application Controller Guide*.

Configuring and running each component individually

To configure and run logging and reporting step by step in Oracle Endeca Workbench, do the following:

- Provision and start a Log Server, and provision a Report Generator.
- Specify the kind of reports you want to generate.
- View the generated reports.

Provisioning the Log Server

Before you can run the Log Server, you must provision it. You can do this using the Deployment Template, through Oracle Endeca Workbench, in a provisioning file you run with eaccmd, or programmatically.

To provision a new Log Server in Oracle Endeca Workbench:

1. Open Oracle Endeca Workbench, log in as admin, and go to the EAC **Administration** > **Admin Console** page.
2. In the Components tab, expand the Log Server component and click **New Log Server**.
3. Using the setting descriptions below, type in the settings and click **Create**.

Log Server settings and default or recommended values

Use these suggested values and guidelines when provisioning a new Log Server component.

Oracle Endeca Workbench Setting	Description	Default/recommended value
Working Directory	Working directory for the process that is launched. If it is specified, it must be an absolute path. If any of the other properties of this component contain relative paths, they are interpreted as relative to the working directory.	If working-dir is not specified, it defaults to `$ENDECA_CONF/work/<app¬Name>/<componentName>` on UNIX, or `%ENDECA_CONF%\work\<app¬Name>\<componentName>` on Windows.

Oracle Endeca Workbench Setting	Description	Default/recommended value
Log File	The path to the Log Server log file.	If the Log File is not specified, the default is component working directory plus component name plus “.log”.
Port	Required. Port on which to run the LogServer.	Typically the Dgraph component port plus two (such as 8002).
Output Prefix Name	Required. Path and prefix name for the LogServer output. For example, output_prefix = c:\temp\wine generates files that start with "wine" in c:\temp.	**Note:** The Log Server output directory cannot be the same as the Report Generator input directory, due to file usage contention issues.
Gzip	Required. Controls the archiving of log files. Possible values are true and false.	The default value is false.
Startup Timeout	Specifies the amount of time in seconds that the eaccmd waits while starting the Log Server. If it cannot determine that the Log Server is running in this timeframe, it times out.	The default is 60.
Custom Properties	An optional list of properties, consisting of a required name and an optional value. For more information, see "Adding properties to hosts and components" in the *Oracle Endeca Application Controller Guide*.	n/a

Starting the Log Server directly

You can start, run, and monitor the Log Server using Oracle Endeca Workbench. You have to start the Log Server before it can begin logging usage of your Web application.

You must provision the Log Server before starting it.

To start, run, and monitor the Log Server in Oracle Endeca Workbench:

1. Open Oracle Endeca Workbench, log in as admin, and go to the **EAC Administration** > **Admin Console** page.
2. In the Components tab, select the Log Server component and click Start. When the Log Server is started, its icon changes to green and its status changes to Running.

3. To monitor the status of the Log Server, click the Running link located next to its name. The screen displays the component’s status, its start time, and the length of time it has been running.

Provisioning the Report Generator

Before you can run the Report Generator, you must provision it. You can do this using the Deployment Template, through Oracle Endeca Workbench, in a provisioning file you run with eaccmd, or programmatically.

To provision a new Report Generator in Oracle Endeca Workbench:

1. Open Oracle Endeca Workbench, log in as admin, and go to the **EAC Administration** > **Admin Console** page.
2. In the Components tab, expand the Report Generator component and click **New Report Generator**.
3. Using the setting descriptions below, type in the settings and click **Create**.

Report Generator settings and default or recommended values

Use these suggested values and guidelines when provisioning a new Report Generator component.

Oracle Endeca Workbench Setting	Description	Default/recommended value
Host	The name of the host this component is located on.	n/a
Working Directory	Working directory for the process that is launched. If it is specified, it must be an absolute path. If any of the other properties of this component contain relative paths, they are interpreted as relative to the working directory.	If working-dir is not specified, it defaults to `$ENDECA_CONF/work/<app¬Name>/<componentName>` on UNIX, or `%ENDECA_CONF%\work\<app¬Name>/<component` on Windows.
Log File	The path to the Report Generator log file.	If the log-file is not specified, the default is component working directory plus component name plus “.log”.
Input Directory or File	Required. Path to the file or directory containing the logs to report on. If it is a directory, then all log files in that directory are read. If it is a file, then just that file is read.	**Note:** The Log Server output directory cannot be the same as the Report Generator input directory, due to file usage contention issues.
Output File	Required. Name the generated report file and path to where it is stored. For example: `C:\Endeca\reports\weekly\myreport.xml` on Windows	**Note:** If you are running the report generation script provided with the installation, the value of Output File will change with every run of the script, based on the specific output file for the day or week of the run.

Oracle Endeca Workbench Setting	Description	Default/recommended value
	`/endeca/reports/weekly/myreport.xml` on UNIX	
Stylesheet File	Required. Filename and path of the XSL stylesheet used to format the generated report.	`%ENDECA_CONF%\etc\tools_report_stylesheet.xsl` on Windows `$ENDECA_CONF/etc/tools_report_stylesheet.xsl` on UNIX
Settings File	Path to the report_settings.xml file.	`%ENDECA_CONF%\etc\ report_settings.xml` on Windows `$ENDECA_CONF/etc/report_settings.xml` on UNIX
Start Date Stop Date	These set the report window to the given date and time. The date format should be either `yyyy_mm_dd` or `yyyy_mm_dd.hh_mm_ss`. For example, 2008_10_25.19_30_57 expresses Oct 25, 2008 at 7:30:57 in the evening. The Stop Date parameter is exclusive. This means that if you only want to report a single date (e.g., 2/17/2009), you have to specify the Start Date as 2/17 and the Stop Date as 2/18.	n/a
Charts	Turns on the generation of report charts.	Disabled by default.
Java Binary	Should indicate a JDK 1.5.x or later.	Defaults to the JDK that Endeca installs.

Java Options	Command-line options for the java_binary setting. This command is primarily used to adjust the Report Generator memory, which defaults to 1GB and to adjust the language code for reports, which defaults to English. To set the memory, use the following (ignore the linebreak):
	`java_options = -Xmx[MemoryInMb]m`
	`-Xms[MemoryInMb]m`
		n/a

Oracle Endeca Workbench Setting	Description	Default/recommended value
Arguments	Command-line flags to pass to the Report Generator, expressed as a set of arg sub-elements.	n/a

Starting the Report Generator directly

Because the Report Generator relies on Log Server output, you must start the Log Server before starting the Report Generator.

Before you run the Report Generator, you have to copy the Log Server output to the Report Generator input directory. Because of file usage contention issues, these two cannot share the same directory.

Note: Starting a Report Generator is typically done separately, and not as part of a baseline update script.

To start, run, and monitor the Report Generator in Oracle Endeca Workbench:

1. Open Oracle Endeca Workbench, log in as admin, and go to the **EAC Administration** > **Admin Console** page.
2. In the Components tab, select the Report Generator component and click **Start**.
   When the Report Generator is started, its icon changes to green and its status changes to Running. It stops once it has processed all logs in its input directory.
3. To monitor the status of the Report Generator, click the **Running** link located next to its name.
   The screen displays the component's status, its start time, and the length of time it has been running.

Provisioning the Report Generator to run in French or German

You can provision the Report Generator with settings that produce reports in French or German rather than in English (the default).

To provision the Report Generator to run in French or German:

1. Provision the Report Generator in Oracle Endeca Workbench as described in "Provisioning and starting the Report Generator."
2. In the Java Options setting:
   a) Specify the language code as `-Duser.language = fr` for French.
   b) Specify the language code as `-Duser.language = de` for German.
3. When you are finished provisioning, click **Create**. The next time you run the Report Generator, it creates reports in the language specified.

Specifying report frequency

You configure report frequency in Oracle Endeca Workbench on the Report Generation section of the Application Settings page.

To specify report frequency in Oracle Endeca Workbench:

1. In Oracle Endeca Workbench, go to the **Application Settings** > **Report Generation** page.
2. Select a frequency for your reports:

- For reports generated once a day, check **Daily Reports**.
- For reports generated once a week, first check **Weekly Reports**. Then specify the day that the weekly reports begin on in the drop-down list.

Note: If you want to generate daily and weekly reports, check both.

3. Click **OK** to save your configuration. If you select the report frequency in Oracle Endeca Workbench, then Oracle Endeca Workbench will automatically provision a host with the alias of webstudio for you, if one does not exist already. This host contains a directory provisioned with an alias of webstudio-report-dir. This is set to the following directory for report storage:
   - On Windows: `C:\Endeca\Workbench\workspace\reports\<app name>`
   - On UNIX: `usr/local/endeca/Workbench/workspace/reports/<app name>`
4. After the directory mentioned in step 3 has been created, manually add daily and/or weekly sub-directories to it. These sub-directories are where Oracle Endeca Workbench will look for reports to display.

   Note: The Deployment Template creates these sub-directories for you, as does the report generation script.

Automatically scheduling report generation

If you specified the report frequency in Oracle Endeca Workbench, Oracle Endeca Workbench runs provisioned scripts named DailyReports, for the daily reports, or WeeklyReports, for weekly reports, automatically, just after midnight, once a day or once a week, as specified.

Keep in mind the following:

- If you did not already provision either or both of these scripts before you set the report frequency in Oracle Endeca Workbench, Oracle Endeca Workbench provisions them for you automatically, using the correct aliases.
- If you change the alias name of these scripts or remove them, Oracle Endeca Workbench will not automatically run them.

If you are not using Oracle Endeca Workbench to control the report generation process, you may want to automate the process using the Scheduled Tasks control panel on Windows or crontab task scheduler on UNIX. See your operating system documentation for details about automated scheduling.

Running the report generation script

The report generation script automates much of the logging and reporting process for you. This section describes the script in detail and then explains how to use it.

About the report generation script

The report generation script is an EAC script you can use to drive and automate the entire process of report generation.

The report generation script works as follows:

- It looks at the list of files in the Log Server and determines which of these files are relevant to the report that you requested, such as daily or weekly. If a frequency of daily is specified on the command line, logs from the previous day are requested. If a frequency of weekly is specified,

logs from the seven days previous to the time the script is run are requested. It can also tell the Log Server to roll its current log and start a new one. This is useful if you want to control the size of a log file or keep it within the requested date range.
- It moves the relevant files from the Log Server host to the Report Generator host and instructs the Report Generator to generate a report. For weekly reports, it passes the Report Generator the exact dates of the seven days ending yesterday.
- It moves the report files from the Report Generator host to the Oracle Endeca Workbench host.
- It instructs the Log Server to delete all files over 30 days old.

To run both daily and weekly reports, if you updated these settings in Oracle Endeca Workbench, these will be provisioned for you.

A version of the report generation script is included with the Endeca software, and is stored in `%ENDECA_ROOT%\bin\generate-report.bat` for Windows (`$ENDECA_ROOT/bin/generate-report.sh` for UNIX). You can copy and modify this script as needed. For information, see “Editing the report generation script.”

Notes

- The report generation script, generate-report.bat or generate-report.sh, overrides the options specified for the Log Server and Report Generator components in the EAC. For instance, it ignores previously-set options for Start Date and Stop Date.
- The supplied script for report generation does not support multi-platform scenarios (although multi-machine scenarios are supported). If you want to perform multi-platform report generation, you need to update the script or provide your own.
- The Log Server’s Output Prefix should not be set to write to the same folder as the Report Generator’s Input File or Directory. If these two components are set to write to the same directory, you will receive an error.

Editing the report generation script

You can modify the existing report generation script to suit your own needs, by copying and editing its source files.

The report generation script is written in Java. Note the following details:

- The script source tree is installed as part of the Endeca reference implementation, and can be found in `%ENDECA_REFERENCE%\eac_scripts` on Windows, or `$ENDECA_REFERENCE/eac_scripts` on UNIX.
- The executable files for the script are stored in the `%ENDECA_ROOT%\bin` (Windows) or `$ENDECA_ROOT/bin` (UNIX); they depend on the `eacscript.jar` file in `%ENDECA_ROOT%\lib\java` (Windows) or `$ENDECA_ROOT/lib/java` (UNIX).

You can generate your own version of the `eacscript.jar` file by modifying the source files in the reference implementation.

High-level workflow for using the report generation script

Perform all necessary provisioning and start the Log Server before running the report generation script.

In order to run the report generation script for a given application, follow these steps:

1. Do the necessary setup. This includes provisioning the host, components, and scripts, and starting the Log Server.
2. Start the script.
3. View the generated reports.

Required setup for the report generation script

The report generation script does not include the provisioning step. Therefore, you must provision the following items in the EAC Administration Console in the order given:

1. Add a host that points to a machine on which Oracle Endeca Workbench is running, by selecting the Hosts tab and completing the fields as follows:
   a) For New Host Alias, specify `webstudio`. (The script depends on this alias, and fails if the host name is specified incorrectly.)
   b) For Host Name, specify the DNS name. Because you are typically working across machines, it's best to avoid using localhost.
   c) For Agent Port, specify the port the agent is using, such as 8888.
   d) For Custom Directories, you may specify `webstudio-report-dir` as any directory. The default path is `C:\Endeca\Workbench\workspace\reports\<app name>` on Windows, or `/usr/local/endeca/Workbench/workspace/reports/<app name>` on UNIX (assuming you installed to `/usr/local`). This is the directory where Oracle Endeca Workbench will look for report files, so it must match.
   e) Click **Create Host**.
2. Configure and start the Log Server.
3. Configure the Report Generator but do not start it.
4. Add a report generation script by selecting the Scripts tab and completing the information as follows. (If you want to generate daily and weekly reports, add two separate scripts, one for each.)
   a) For New Script Alias, specify the name of the script. If you scheduled these in the Report Generation screen of Oracle Endeca Workbench, specify either `DailyReports` or `WeeklyReports`.
   b) For Command, for a daily report script, specify `%ENDECA_ROOT%\bin\generate-report.bat daily` (Windows) or `$ENDECA_ROOT/bin/generate-report.sh daily` (UNIX).
   c) For Command, for a weekly report script, specify `%ENDECA_ROOT%\bin\generate-report.bat weekly` (Windows) or `$ENDECA_ROOT/bin/generate-report.sh weekly` (UNIX).
   d) Click **Create Script**.

   Note: If you check off the boxes in the **Application Settings** > **Report Generation** section, this step will be done for you.

Starting the report generation script

Once requests have been made to the Endeca application, you can start the report generation script based on the created log entries.

To start the report generation script in Oracle Endeca Workbench:

1. Go to the **EAC Administration** > **Admin Console** page and select the Scripts tab.
2. Select the application and the previously provisioned report generation script that you want to start, such as `daily-report-script` or `weekly-report-script`.
3. Click **Start**. Generated reports are output to the `webstudio-report-dir` directory.

Viewing reports in Oracle Endeca Workbench

Typically, you use Oracle Endeca Workbench to view the XML reports you have generated.

Note: As long as you followed the filename and location requirements in the previous steps, you will be able to see your reports in Oracle Endeca Workbench.

To view reports in Oracle Endeca Workbench:

1. On a Windows machine, open Internet Explorer.
2. In the Address box, enter the following URL: `http://WorkbenchHost:8006`

   Note: If you specified a different Endeca Tools service port, use that instead.

3. At the Oracle Endeca Workbench login page, enter your user name and password then click **Log In**.
4. In the navigation menu, click **View Reports**.

The default report is the current daily report, which is the same as the "Current (daily)" link on the navigation menu (in the left pane of the View Reports page). The navigation menu also lets you view:

- Current weekly report, via the Current (weekly) link
- List of archived daily reports, via the Daily Reports link
- List of archived weekly reports, via the Weekly Reports link

For example, you can click the Daily Reports link to display a list of daily reports. From the list of reports, you can click a specific report to display its contents.

Viewing reports produced by other Report Generators

Typically, you use the View Reports page to read reports that were produced by the Report Generator that is running for the specific application you are accessing in Oracle Endeca Workbench.

However, you can also view reports that were generated by another client to the Endeca Application Controller, as long as its ReportGenerator component uses the `tools_report_stylesheet.xsl` stylesheet. To view these reports, copy the report XML files to the appropriate daily or weekly subdirectory in the reports directory on the Oracle Endeca Workbench server.

Archiving and deleting log files and reports

Oracle recommends that you do the following for the existing log files and reports:

- Archive your log files on a weekly basis.

  When the Report Generator processes reporting information, it processes all log files contained in the logs directory you specified and any of its subdirectories. This processing has performance implications as the size of your log data grows. To minimize log processing time, Oracle recommends that you archive your log files on a weekly basis to a directory that is not under the logs directory.

- Delete outdated log files.

  The EAC script for report generation retains 30 days of log files, in case a report does not generate properly. More specifically, after a report has been successfully generated, any log files that are more than 30 days older than the start of the report’s time period are deleted. By extension, if a report is not successfully generated, no log files are deleted, ensuring that no data is lost.

If you are not using the report generation script, you need to purge log files manually.

- Delete outdated reports.

  Reports are never deleted by the Endeca Application Controller or Oracle Endeca Workbench. Therefore, it is the administrator’s responsibility to check the contents of the reports directory on the Oracle Endeca Workbench server periodically and manually delete any obsolete or unwanted reports.

Chapter 4

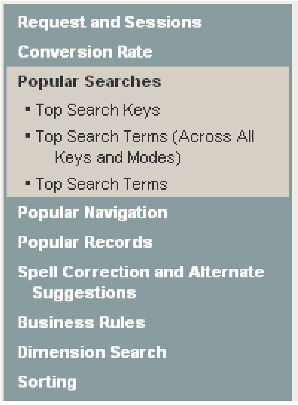
Customizing Report Content and Presentation

This section describes how to customize reports for your particular needs.

Report contents overview

In Oracle Endeca Workbench, a report describes the search and navigation usage of your Web site on a daily or weekly basis.

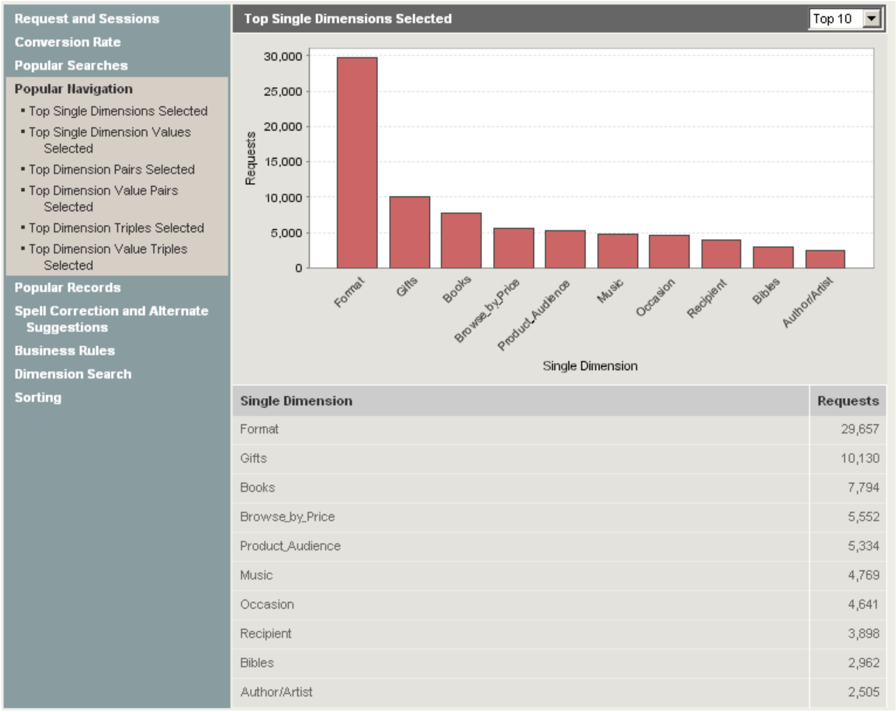
Reporting data is organized into report sections and further organized by individual report items. For example, Popular Searches and Popular Navigation are report sections. Report sections contain report items, for example the Popular Searches section contains the Top Search Keys item, among others. The following image shows the report sections available in a typical Endeca application. It also shows the report items that display when Popular Searches is selected.



Selecting a report item in the left-hand pane displays report entries in the body of the report. This information includes the report's name, the reporting data in tabular format, and in some cases a chart of the reporting data displayed in the table.

If a chart appears with the reporting data, the chart may have a drop-down list to view subsets of the reporting data. For example, if you view Top Single Dimensions Selected item, under the Popular Navigation section, you can view subsets of the reporting data by selecting Top 10, Top 20, Top 50, and so on from the list. If you choose a value for Top N that is greater than 30, the chart displays only the first 30 items. However, the table displays all items.

The following image shows the report item named Top Single Dimensions Selected. The report item contains a bar chart representation of the tabular data and it contains a drop-down list to display subsets of the data. In this image, the report items shows the top single dimensions selected and the number of requests for each.


Request and Sessions
Conversion Rate
Popular Searches
Popular Navigation
▪ Top Single Dimensions Selected
▪ Top Single Dimension Values Selected
▪ Top Dimension Pairs Selected
▪ Top Dimension Value Pairs Selected
▪ Top Dimension Triples Selected
▪ Top Dimension Value Triples Selected
Popular Records
Spell Correction and Alternate Suggestions
Business Rules
Dimension Search
Sorting
Top Single Dimensions Selected
Top 10
30,000
25,000
20,000
15,000
10,000
5,000
0
Requests
Format
Gifts
Books
Browse_by_Price
Product_Audience
Music
Occasion
Recipient
Bibles
Author/Artist
Single Dimension

Single Dimension	Requests
Format	29,657
Gifts	10,130
Books	7,794
Browse_by_Price	5,552
Product_Audience	5,334
Music	4,769
Occasion	4,641
Recipient	3,898
Bibles	2,962
Author/Artist	2,505

You can configure which report sections and report items you want to display in Oracle Endeca Workbench by modifying the `report_settings.xml` file.

About report settings XML

The report settings XML file (`report_settings.xml`) determines which report sections and report items, if any, are excluded from a report.

Any report sections or report items that you do not explicitly disable are included in the report. Also this file sets chart display increments and session queue size. You can customize the content of a report, by changing the `report_settings.xml` file to:

- Disable unnecessary report sections or items
- Adjust the number of items considered in top_n report data
- Modify the session queue size

The `report_settings.xml` file is installed on the Report Generator server in `%ENDECA_CONF%\etc` on Windows and `$ENDECA_CONF/etc` on UNIX.

Disabling unnecessary report sections or items

You can disable report sections and items that are not needed in a report by adding report section names or report item names to the `disable` element in the `report_settings.xml` file.

Each report item name and its corresponding report entry name is listed in “Report sections.” Disabling a report section disables all the report items in that section. For example, to remove the Conversion

Rate report section, and all its report items from a report, add conv_rate_summ to the disable element as shown here.

```
<report_settings>
 <disable list="conv_rate_summ" />
</report_settings>
```

You can disable any number of report sections or report items. The following example disables the conversion_rate_by_hour and conversion_rate_by_date but not the other conversion rate items in the report section.

```
<report_settings>
 <disable list="conversion_rate_by_hour,conversion_rate_by_date" />
</report_settings>
```

Changing the number of items considered in a top_n report

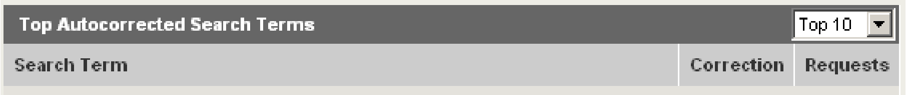
Several report items have a top_n element that sets the number of results returned for a report item. The Report Generator displays results for up to that number, depending on the content of the log files comprising the report.

The report items that have a configurable top_n element available contain top_ in the name, for example, top_search_keys, top_dimensions, conversion_rate_by_top_dimensions.

Each top_n report item has a initial value that you specify and you can specify the increments that appear in the drop-down list. For example, you can set up a report item to have an initial value that displays Top 10 and increments in the drop-down list with Top 10, Top 20, and Top 50. The following XML configures the Top Autocorrected Search Terms report item to display the Top 10, Top 25, Top 50 choices in the drop-down list:

```
<report_settings>
 <top_n report_item="top_autocorrected_terms" n="10, 25, 50"
  initial_value="10" />
</report_settings>
```

The following image shows how the configured report item appears in Oracle Endeca Workbench.



Top Autocorrected Search Terms

Search Term	Correction	Requests

Customizing the report generation file

When generating reports for viewing on the View Reports page, Oracle Endeca Workbench uses the report_settings.xml file as the display configuration file.

This file is located on the Report Generator server in the \workspace\etc directory on Windows (/workspace/etc on UNIX).

Because the Endeca Application Controller specifically looks for this name and path, you cannot change the name or location of the report configuration file.

You can customize the context and appearance of your reports by modifying the report elements in the report_settings.xml file. However, both daily and weekly reports must use the same configuration file.

About report presentation XSLT

The stylesheet `tools_report_stylesheet.xsl` transforms report content into XML that Oracle Endeca Workbench can load and display on the View Reports tab.

Do not edit `tools_report_stylesheet.xsl`. Oracle Endeca Workbench depends on its contents. The stylesheet `report_stylesheet.xsl` controls the appearance of an HTML report. `report_stylesheet.xsl` transforms report content into browser-readable HTML output. `report_stylesheet.xsl` is not discussed in detail in this document. However, the file is written in standard XSLT. If you are familiar with XSLT, you can open it and make modifications to the styles it contains.

HTML reports outside Oracle Endeca Workbench

You can set up the Report Generator to output one or more HTML reports using the command line.

You could have several versions of `report_settings.xml`, one for each report that you generate. When you run a given report, you specify the appropriate `report_settings.xml` file. For example, Operations might want to check MDEX Engine and overall site performance, while Marketing might only need to investigate merchandising results.

There are two limitations to using HTML reports: they do not contain charts for any of the reporting data, and they do not support interactive views of the report data that use Top N drop-down lists.

Oracle Endeca Workbench uses one `report_settings.xml` file to generate all reports that display on the View Reports page.

Report sections

The XML report sections that display in Oracle Endeca Workbench are divided into several groups.

The XML report section groups are:

- Request and Sessions
- Conversion Rate
- A/B Testing
- Popular Searches
- Popular Navigation
- Popular Records
- Spell Correction and Alternate Suggestions
- Business Rules
- Dimension Search
- Sorting

The tables in following sections list each report section and item, along with its title in the report, and the required log entry key/value pair.

Log entry key/value pairs

Each report item (except for num_requests) has a corresponding key/value pair.

These relationships are described in the tables below using the following syntax:

Key/Value pair	Description
(none)	No specific keys are required
KEY=<string>	Specific key required with no specific value
KEY=<number>	Specific key required with numeric value
KEY=<string>, <string>...	Specific key required with multiple values
KEY1=<string>,KEY2=<string>...	Multiple keys required
KEY=VALUE	Specific key required with specific value
KEY={VALUE1, VALUE2...}	Specific key must equal one of list of values
KEY=<null>	Specific key must NOT be present
KEY!=VALUE	Specific key must NOT equal specific value

Requests and sessions

The Request and Session section shows request and session summaries.

They allow you to answer questions like, “How many search-only requests occurred?” and “What was the average number of requests by date?”

Report item name	Corresponding report entry	Required log entry key/value pairs
request_and_session_summary	Request and Session Summary	SESSION_ID=<string>
num_requests_by_hour	Requests by Hour	None
num_requests_by_date	Requests by Date	None
num_unique_sessions_by_hour	Sessions by Hour	SESSION_ID=<string>
num_unique_sessions_by_hour	Sessions by Date	SESSION_ID=<string>
requests_by_type	Requests by Type	TYPE=<string>
num_search_only_ requests_by_num_results	Search-Only Requests with 0 results with 1-19 results with 20-99 results with 100+ results Total	TYPE=S, NUM_RECORDS= <number>
num_nav_only_requests_by_ num_results	Navigation-Only Requests with 0 results with 1-19 results	TYPE=N, NUM_RECORDS= <number>

Report item name	Corresponding report entry	Required log entry key/value pairs
	with 20-99 results with 100+ results Total	
num_search_then_nav_ requests_by_num_ results	Search-Then-Navigation Requests... with 0 results with 1-19 results with 20-99 results with 100+ results Total	TYPE=SN, NUM_RECORDS= <number>
dimension_values	Dimension Values	NUMREFINEMENTS= <value>

About Conversion Rate

A conversion represents an end user's success in locating Web site content. In other words, a new search converts to a successful search when end users find what they are looking for.

You report these conversions with the Conversion Rate section. The Conversion Rate section shows conversion rates by date, type, terms, dimensions, and so on.

You define exactly what constitutes a conversion in the implementation of your Endeca application. Here are several examples of common conversion events:

- An end user selects a specific record page. This is the simplest definition of a search converting to a successful search.
- An end user performs a key action. For example, this action could be selecting an option such as "Add item to cart" or it could be downloading a long-form document.
- An end user performs a secondary action, such as purchasing the contents of the cart. This is a common definition of conversion: a search becomes a successful search when it results in a purchase by the end user.

Your implementation then indicates to the Endeca logging API when a conversion event has occurred in a particular session.

The conversion rate is the number of converted sessions (i.e., user sessions that are converted into sales) divided by the number of total user sessions.

Each user session is indicated in the log entries by a SESSION_ID key/value pair. The SESSION_ID key is used to recall the previous search terms and navigation requests made during the user’s session. The results are automatically gathered and interpreted into the report.

The Conversion Rate section allows you to answer sales-related questions like, “What percentage of searches resulted in sales?”, “What was the conversion rate of the most popular search terms?”, and “Which dynamic business rules resulted in the most sales?”

All Conversion Rate items use a session queue. To make its conversion rates, the Report Generator keeps a given number of sessions in memory. You can adjust the size of the queue in the session_queue element of the report_settings.xml file. The default size of the session queue is set to 5000.

It is recommended that you set the session queue size to a number that is slightly higher than the maximum number of concurrent users that you expect on your site at a given time. Setting the size too high uses more memory. Setting the size too low can result in imprecise rate calculations.

Conversion Rate

Report item name	Corresponding report entry	Required log entry key/value pairs

conversion_rate	Conversion Rate Summary
	• Total Number of Sessions
	• Sessions Converted
	• Overall Conversion Rate
		SESSION_ID=<string>
		CONVERTED=TRUE

conversion_rate_by_hour	Conversion Rate by Hour	SESSION_ID=<string>
		CONVERTED=TRUE

conversion_rate_by_date	Conversion Rate by Date	SESSION_ID=<string>
		CONVERTED=TRUE

conversion_rate_by_ type	Conversion Rate by Type
	Search
	Search-Then-Navigation Sessions
	Navigation
		SESSION_ID=<string>
		TYPE=S,
		CONVERTED=TRUE
		TYPE=SN,
		CONVERTED=TRUE
		TYPE=N,
		CONVERTED=TRUE

conversion_rate_top_ by_search_terms	Conversion Rate for Top Search Terms	SESSION_ID=<string>
		SEARCH_TERMS=<string>
		CONVERTED=TRUE

conversion_rate_by_ top_dimensions	Conversion Rate for Top Dimensions	SESSION_ID=<string>
		DIMS=<string>, <string>...
		CONVERTED=TRUE

Report item name	Corresponding report entry	Required log entry key/value pairs

conversion_rate_by_ top_merch_rules	Conversion Rate for Top Business Rules	SESSION_ID=<string>
		MERCH_RULES=<string>, <string>...
		CONVERTED=TRUE
		IN_MERCH=<string>

About A/B Testing

The A/B Testing section generates a summary of the results of rules marked as A/B testing rules.

A/B testing is a method of comparing the effectiveness of two or more merchandising or content spotlighting alternatives. Individual rules are presented to subsets of end-users, and the A/B Testing reports allows business users to compare the results.

The report section allows you to answer questions like, “How does the merchandising performed by the rules in set A compare to the merchandising performed by the rules in set B?” or “Which of these competing rules has the best conversion rate?”

Report item name	Corresponding report entry	Required log entry key/value pairs

conversion_rate_by_top_abtest_rules	Conversion Rate for All A/B Testing Rules	SESSION_ID=<string>
		CONVERTED=TRUE
		IN_MERCH=<string>
		ABTEST_RULES=<string>, <string>...

Popular Searches

The Popular Searches section generates a summary of the top search terms. They allow you to answer questions like, “What search terms were chosen most often?”

Report item name	Corresponding report entry	Required log entry key/value pairs
top_search_keys	Top Search Keys	SEARCH_KEY=<string>
top_search_terms	Top Search Terms (Across All Keys and Modes)	SEARCH_TERMS=<string>
top_search_terms_w_ results	Top Search Terms	SEARCH_KEY=<string>, SEARCH_MODE=<string>, SEARCH_TERMS=<string>, NUM_RECORDS=<number>

Popular Navigation

The Popular Navigation section shows a summary of the top dimensions and dimension values selected.

They allow you to answer questions like, “Which dimensions and dimension values are most often selected by users?” and “Which combinations of dimensions and dimension values are most popular?”

Report item name	Corresponding report entry	Required log entry key/value pairs
top_dimensions	Top Single Dimensions Selected	DIMS=<string>
top_dvals	Top Single Dimension Values Selected	DVALS=<string>
top_dimension_pairs	Top Dimension Pairs Selected	DIMS=<string>,<string>
top_dval_pairs	Top Dimension Value Pairs Selected	DVALS=<string>,<string>
top_dimension_triples	Top Dimension Triples Selected	DIMS=<string>,<string>,<string>
top_dval_triples	Top Dimension Value Triples Selected	DVALS=<string>,<string>,<string>

Popular Records

The Popular Records section shows top record searches. They allow you to answer questions like, “What were my site’s top 20 records?”

Report item name	Corresponding report entry	Required log entry key/value pairs
top_records	Top Records	RECORD_NAMES=<string>

Spell Correction and Alternate Suggestions

The Spell Correction and Alternate Suggestion section shows spell correction and alternate suggestion reports.

They allow you to answer questions like, “What were the top five search terms that returned no results?” and “How many of the alternative suggestions that we offered were actually selected by users?

Report item name	Corresponding report entry	Required log entry key/value pairs
spell_correct_alt_sugg_ summary	Number of Search Requests Returning No Results	NUM_RECORDS=0, SEARCH_TERMS=<string>
Same as above.	Number of Search Requests Autocorrected	AUTOCORRECT_TO=<string>
Same as above.	Number of Search Requests with Alternate Suggestions	DYM_TO=<string>

Report item name	Corresponding report entry	Required log entry key/value pairs
Same as above.	Number Alternate Suggestions Selected by User	IN_DYM=<string> **Note:** Requires the "in_dym" URL parameter to be added to any links where "Did you mean" suggestions are selected.
top_search_terms_no_ results	Top Search Terms Returning No Results	NUM_RECORDS=0, SEARCH_TERMS=<string>
top_autocorrected_ terms	Top Autocorrected Search Terms **Note:** This report displays the user's term and the correction.	AUTOCORRECT_TO=<string>, SEARCH_TERMS=<string>
top_search_terms_dym_ engaged	Top Search Terms with Alternate Suggestions **Note:** This report displays the user's term and the suggestion.	DYM_TO=<string>, SEARCH_TERMS=<string>

Business Rules

The Business Rule section shows a merchandising performance summary.

The report items allow you to answer questions like, "Which of my dynamic business rules was triggered most often by user queries?" and, "Of the number of requests with merchandised records, how many of the merchandised records were selected?"

Report item name	Corresponding report entry	Required log entry key/value pairs
business_rules_summary	Number of Requests with Highlighted Records	MERCH_RULES=<string>
Same as above.	Number of Business Rule Results Selected IN_MERCH=<string>,	IN_MERCH=<string>, RECORD_NAMES=<string> **Note:** Requires the "in_merch" URL parameter to be added to any links where products are selected

Report item name	Corresponding report entry	Required log entry key/value pairs
		from merchandising results.
top_merch_rules	Top Business Rules Engaged	MERCH_RULES=<string>
top_records_selected_ through_merch	Top Highlighted Records Selected	IN_MERCH=<string>, RECORD_NAMES=<string> **Note:** Requires the "in_merch" URL parameter to be added to any links where products are selected from merchandising results.
top_dvals_no_merch	Top Dimension Values Selected Without Business Rules	MERCH_RULES=<null>, DVALS=<string>
top_search_terms_no_ merch	Top Search Terms Entered Without Engaging Business Rules	MERCH_RULES=<null>, SEARCH_TERMS=<string>

Dimension Search

The Dimension search section shows dimension search summaries.

The report items allow you to answer questions like, “Which dimensions are users searching for?” and, “How often are dimension search results selected?”

Note: This requires the `in_dim_search` URL parameter to be added to any links where dimension search results are selected.

Report item name	Corresponding report entry	Required log entry key/value pairs
dim_search_summary	Number of Dimension Search Results Selected by User	IN_DIM_SEARCH=<string>
top_dimension_search_	Top Dimension Search Results Selected	results_selected IN_DIM_SEARCH=<string>, DVALS=<string>

Sorting

The Sorting section shows a sorting summary.

They allow you to answer questions like, “How often are users specifying a sort key for their results?” and, “What were the top sort keys selected?”

Note: All requests with a sort key specified are counted, not just the initial sort request.

Report item name	Corresponding report entry	Required log entry key/value pairs
sorting_summary	Number of Requests with Sort Key Specified	SORT_KEY=<string>
top_sort_keys	Top N Sort Keys Selected	SORT_KEY=<string>

Index