

Unica Optimize V12.0 Troubleshooting and Tuning Guide



Contents

- Chapter 1. Unica Optimize troubleshooting tips..... 1**
 - Upgrading Unica Optimize errors..... 1
 - Unica Optimize session run errors.....2
 - Errors in Unica Optimize session results..... 4
 - Unica Optimize session takes a long time to run..... 9
 - Sample-related provisioning problems..... 11
 - setenv.sh reference..... 12
- Chapter 2. General tuning and performance tips for Unica Optimize..... 14**
 - Additional indexes for extra performance improvements..... 15
 - Use cases that can negatively affect performance..... 16
 - Optimize a transaction query for performance improvements..... 18

Chapter 1. Unica Optimize troubleshooting tips

There are common configuration issues that can interfere with running Optimize. There are many more complex situations with a mixture of configuration settings, rule configuration, and data that might cause unexpected results in your Optimize session. The following sections provide details on troubleshooting and fixing these scenarios.

Upgrading Unica Optimize errors

If you encounter any errors while you are upgrading Unica Optimize, see these possible solutions.

ZeroGlr: Cannot open or remove a file containing a running program

Description of problem

You receive the following error during installation:

```
Unica Campaign is not successfully installed.  
Check log files for more details.
```

In the installation log file, there is the following error:

```
ERROR - ZeroGlr: <location of library file>/<name of library  
file>  
(Cannot open or remove a file containing a running program.)
```

This error can occur when you are upgrading Unica Optimize on an AIX® system, even if the Unica Optimize listener is not running. The upgrade of Unica Optimize to Campaign updates specific libraries. Occasionally on AIX systems, even after you stop the Unica Optimize listener, the libraries are not unloaded from memory.

Solution

1. If you have already run the Campaign installation, run the uninstaller.
2. Confirm the Unica Optimize listener is not running.
3. Run the slibclean command.
4. Run the Campaign installation.

The slibclean command removes unused libraries from memory. This command requires root privileges.

Unica Optimize session run errors

You might have configuration or other errors that prevent a Unica Optimize session from running.

Could not connect to the Campaign server

Description of problem

This error indicates that Unica Campaign web cannot connect or communicate with the Campaign server.

Solution

Confirm that the Campaign listener of listenerType "Optimize" or "Campaign and Optimize both" is running.

If it is not running, start the Campaign listener.

If the problem still persists, confirm all configuration properties that define the connection properties are configured properly, including:

- Campaign | unicaACOListener | serverHost
- Campaign | unicaACOListener | serverPort
- Campaign | unicaACOListener | useSSL

If it is a single server installation, then value for "Campaign|unicaACOListener|serverPort" should be exactly same as "Campaign|unicaACLlistener|serverPort".

Unable to unlock the session with session ID

Description of problem

This error is written to the Unica Optimize session log as `Problem unlocking the session. ID: <session ID> CODE 11336: Accessed table not mapped.`

This error is caused when the home directory for the database is not correctly specified.

Solution

You must configure the database connection in the `setenv.sh` file appropriate for your operating system and database.

Restart the Campaign listener after you edit the `setenv.sh` file.

- Oracle: Set ORACLE_HOME and ORACLE_BASE.
- DB2@: Set DB2_INSTANCE_DIR
- MSSQLServer: Configure the DSN setting for the database.

Response XML is missing the data xml element

Description of problem

This error is written to the Unica Optimize session log as `[E] [TABLE ACC] DATA_SOURCE_CONNECT to UA_SYSTEM_TABLES(<Database Name>) as <UserID>: FAILED: CODE 11518: Database connection error [sdbsrc:2988].`

You receive this error when you attempt to run a Unica Optimize session.

This error indicates that the database credentials stored in the `<Campaign Installation Directory>/Install/jdbc.properties` file are incorrect.

Also if Optimize license is not applied, you will receive this error.

Solution

You can solve this issue in either of two ways.

- Ensure that the Optimize license is applied to the setup.
- Update `<Optimize Installation Directory>/Install/jdbc.properties` with the correct credentials. Use the Unica Platform encryptPasswords tool to encrypt the password.

Related reference

- [setenv.sh reference \(on page 12\)](#)

Errors in Unica Optimize session results

Many issues that you might encounter in Unica Optimize sessions can be resolved by obtaining details from the session logs. Some of these errors might cause a Unica Optimize session to fail. Some of these errors might be the cause of unexpected optimization results.

The LP solver was unable to find an optimal solution to the chunk problem

Description of problem

The outer algorithm (the part that handles the capacity rules) found that the problem given to it for a customer sample is not solvable. This error might mean that one of two conditions:

- There is a logical contradiction in the capacity rules.
- A solution to those rules is not possible with the existing data.

This error can also occur if the scores used for the proposed contacts exceed the numerical precision of the floating point math used. In general, do not exceed a range of 1.0 to 1.0e+11.

Solution

Look for problems with the logic of the capacity rules, and for mismatches with the rules and the proposed contact data. For instance, if a capacity rule requires a minimum of something, make sure that the PCT has at least that many of that item.

Also, check for sample-related provisioning problems. If there seems to be no problem with the rules and data, collect the following files from the log directory of your Unica Optimize installation:

- Unica Optimize session log
- All files that start with `ACO_`

Send these files to Technical Support so that they can analyze them.

The generation loop was unable to eliminate all slack and surplus variables

Description of problem

The outer algorithm (the part that handles the capacity rules) progresses by creating alternative solutions to the per customer rules that are generated by the inner (core) algorithm. It creates alternatives by temporarily changing offers scores, and looking for solutions that have not yet been generated. You receive this error if the following two conditions are met:

- The outer algorithm cannot satisfy the capacity rules with any of its alternate solutions.
- The core algorithm is not creating alternative solutions.

This error might mean either of the following conditions:

- The per customer rules and the capacity rules are inherently in conflict, so no solution is possible.
- The data is such that a solution is not possible.

Solution

Look for problems with the logic of the rules, and for mismatches with the rules and the proposed contact data. For instance, You have a per customer minimum of 1 on all offers, channels, and segments. This rule results in at least as many offers as there are customers in the results. If you also have a capacity rule with a maximum value that was less than the number of customers, this rule causes an inherent logical conflict between the per customer rule and the capacity rule.

Also, check for sample-related provisioning problems. If there seems to be no problem with the rules and data, collect the following files from the log directory of your Unica Optimize installation:

- Unica Optimize session log
- All files that start with `ACO_`

Send these files to Technical Support so that they can analyze them.

You can use the `ContinueOnGenerationLoopError` configuration property to have Unica Optimize skip the set of customers if this error occurs or stop the Unica Optimize session.

Unprocessable customer

Description of problem

At the end of each session run, there are some log entries that summarize the results. One of the entries is: "Total # of Unprocessable Customers". This error indicates that no solution can be found for the number of customers shown. When this error occurs, it is not an unrecoverable error. The result is that the "unprocessable" customers receive no offers.

You can get a separate file that contains details of unprocessable customers in comma-separated values (CSV) format in the Unica Optimize server log directory by setting the `Optimize | logging | enableBailoutLogging` property to `TRUE`. This file is `unprocessableables_10-digit-session-ID.csv` and is in the `partition/partition[n]/logs` directory in the Unica Optimize installation directory. Customers skipped because they cannot be processed are marked with `unprocessable` in the `unprocessableables_10-digit-session-ID.csv` file.

Solution

Unprocessable customers can occur for one of the following reasons:

- The rules and data make it impossible to find a solution.

If you set `ContinueOnGenerationLoopError` to `True` and you receive the error `The generation loop was unable to eliminate all slack and surplus variables`, the customers from the set that caused the error are marked as unprocessable.

- The algorithm exceeded the number of alternatives to solve the problem.

The number of alternatives is configured by the `Optimize | AlgorithmTuning | MaxAlternativesPerCustomerEvaluated` property in the session level advanced settings and the configuration properties. Setting the value of this property to a higher number lowers the likelihood that the customer is unprocessable (assuming that it is not inherently unsolvable). However, when it occurs, it also makes the performance penalty higher.

No offers

Description of problem

If a particular customer receives no offers, it is not necessarily an error. Unless a per customer minimum is found in the per customer rules, it is perfectly legal to reject all offers of some customers, if no rules are violated and the overall score is maximized. It can also be a side-effect of an error, as when

there is no legal combination of offers given the rules, or when the customer is unprocessable.

Solution

Check the following conditions:

- Whether a per customer minimum exists in the rules
- Ensure that given the rules, all combinations are legal
- Whether there are any unprocessable customers

Invalid size provided to the init count table. (1,0): CODE 5: Internal Error 5

Description of problem

No channel offer attribute values are defined.

Solution

You must define some channel offer attribute values.

Unica Optimize does not satisfy Max limit of Min/Max # Offers Capacity rule

Description of problem

Unica Optimize session does not meet the maximum setting of the Min/Max # Offers Capacity rule.

The minimum and maximum limits of the Min/Max # Offers Capacity rule are not hard limits. The Unica Optimize session generates less than the maximum number of offers if there are not enough proposed contacts.

However, occasionally even when there are enough proposed contacts that meet criteria, the maximum is not met.

Solution

There are many reasons why solutions are not found, dependent on your set of rules and data. Some possible reasons include:

- Using fractional scores between -1 and 1. Fractional scores between -1 and 1 might give suboptimal results. Fractional scores greater than 1 and less than -1 are allowed.
- Using a range of scores greater than 100,000. If your range of scores is outside of this limitation, the Unica Optimize might not meet capacity rules.

Unica Optimize session takes a long time to run

Here are troubleshooting steps that you can follow if you believe your Unica Optimize session is taking too long to run.

Before you start

1. Make sure that the session is not running with `Optimize|Debug|ExtraVerbose` enabled, as this setting causes slow run times. Check both the session-level advanced settings and the configuration properties.
2. Make sure that you are using a DB loader, and that it is properly configured.
3. If you are using time intervals with your rules, make sure that the contact history tables for your audience level are indexed.
4. Set `Optimize|Logging|loggingLevels` to MEDIUM or LOW. Check both the session-level advanced settings and the configuration properties.

Run a session to generate a clean log for troubleshooting

If your Unica Optimize session takes a long time to run after you follow the steps in [Before you start \(on page 9\)](#), you need to collect detailed information to analyze.

Run a session to generate a clean Unica Optimize server log with the HIGH or ALL setting on. While your session is running, do not access any Unica Optimize reports, as this action adds data to the log that can confuse things. This session might take a longer time to

run because you are collecting detailed log data. Remember to reset `Optimize | Logging | loggingLevels` to `MEDIUM` or `LOW` after you collect the clean log.

When you have the generated log, you check for two things:

- The amount of time spent accessing the database to set up the data that is needed for the session.
- The amount of time spent processing customer samples (chunks).

How to check the amount of time spent accessing the database

Using the clean log you generated, follow these steps to find out how long Unica Optimize is taking to access the database to set up the data that is needed for the session.

1. In the Unica Optimize server log, search for the string: `LRE Starting chunk: 0`
2. Take the timestamp of this entry, and subtract from it the timestamp of the first entry in the log. The difference is the amount of time spent accessing the database to set up the data that is needed to run the session.

If the value seems too high, look at the start and end timestamps for the queries that comprise the log section preceding `LRE Starting chunk: 0` to identify which one is taking too long.

3. Then, troubleshoot the task that took too long in the same way as you would any other database performance issue.

How to check the amount of time spent processing customer samples (chunks)

Using the clean log you generated, follow this procedure to find out how long Unica Optimize is taking to process customer samples.

In the Unica Optimize server log, subtract the time stamp from the line matching `LRE Starting chunk: 0` from the time stamp of the line matching `Run Thread terminated`.

This log entry tells you the total time that is spent in the CPU-intensive optimization section. If this action is where the bulk of the time is being spent, you can get a better idea of what is going on by looking at the chunk iterations.

The optimal solution for each chunk is found by iterating through the following steps:

1. Apply a set of scores to the offers in that chunk
2. Find the optimal solutions with those scores for the customers in the chunk by using the core algorithm.
3. Use the result in the outer algorithm to find a new set of scores to try.

Each time Unica Optimize applies a set of scores, it counts as one chunk iteration. The amount of time that is spent in the CPU-intensive section is roughly proportional to the average number of iterations per chunk.

Sample-related provisioning problems

To handle large volumes of data while not sacrificing the quality of the results, and at the same time getting the results in an acceptable amount of time, certain requirements are made regarding the makeup of the proposed contacts in a session.

One of the strategies Unica Optimize uses is to break the proposed contact data into random subsets of approximately equal numbers of customers; it then optimizes the proposed contacts of each of these samples independently. If multiple threads are configured and supported by your hardware, these customer samples are processed concurrently.

There is a class of problems that can result in errors or suboptimal results that are a side-effect of the customer sample approach. The number of customer samples that are used for a session run is determined by dividing the number of customers in the PCT by the value of the configuration parameter `Optimize|AlgorithmTuning| CustomerSampleSize`. It is important that there are enough proposed contacts matching each capacity rule to allow each random customer sample to be statistically similar relative to every feature used by a capacity rule.

For example, say that we have 1 million customers, and have a configured customer sample size of 1000. This configuration implies that we have 1000 customer samples. Imagine that we have a capacity rule that is set up as: minimum 1 email, maximum 5000 emails.

What Unica Optimize does in this example is to take the rule constraints and modify them to spread that rule across the customer samples. In this example, the maximum 5000 emails constraint is divided by the number of samples, so that each sample is processed with a maximum 5 emails constraint. But what do we do with the minimum 1 email constraint? We cannot have each sample requiring a minimum 1/1000 of an email!

Instead, we randomly pick one sample to process with a minimum 1 email constraint, while the other 999 samples are processed with no minimum email constraint. This process all works fine, unless there are not enough proposed contacts using email to make sure that all 1000 samples get at least one email. If your proposed contacts contains only 500 contacts using email, there is a smaller than 50% chance that a particular sample contains an email. That means you have a greater than 50% chance that the session exits with an error because the minimum cannot be satisfied, even though 500 times that minimum were present in the proposed contacts.

To avoid this situation, any feature that is used in a capacity rule should be well-represented relative to the number of samples. The following scenario is an example of this. You have an input cell 1 that contains 100,000 contact IDs, which are all 100,000 distinct audience IDs. Offer 1 is assigned to input cell 1. You also have input cell 2 that contains one contact ID, and the audience ID is not in input cell 1. Offer 2 is assigned to input cell 2. The one capacity rule sets the minimum number of offers to 1, and the minimum number of contact IDs is set to 100,000.

setenv.sh reference

If there are complications during installation, or if you move your database installation, you might need to configure the Campaign listener manually.

The script to start or stop the Campaign listener is the `cmpServer.bat` (Windows™) or `rc.unica_ac` (UNIX™) file in the `bin` directory of your Campaign installation.

Set following environment variables in `setenv.sh` script.

Table 1. `setenv.sh` script reference

A two-column table listing the settings available in the setenv.sh script and their descriptions.

Settings	Description
CAMPAIGN_HOME	You must set <code>CAMPAIGN_HOME</code> .
ORACLE_HOME	If using Oracle database, set to your Oracle home directory
ORACLE_BASE	UNIX only: If using an Oracle database, set to your Oracle base. This value is normally <code>\$ORACLE_HOME</code> .
DB2_INSTANCE_DIR	UNIX only. If using a DB2 database, set to the path to your DB2 instance directory script.
LIBPATH for AIX® SHLIB_PATH for HP-UX LD_LIBRARY_PATH for Linux™ or Sun Solaris	<p>CAMPAIGN_HOME/bin:<Database Lib path>:<Existing library paths></p> <p>Example for Linux:</p> <pre>LD_LIBRARY_PATH=CAMPAIGN_HOME/ bin:ORACLE_HOME/lib/:/usr/lib:/lib export LD_LIBRARY_PATH</pre>

Chapter 2. General tuning and performance tips for Unica Optimize

Keep in mind these points when you make data or configuration decisions, if you are concerned about performance.

- In general, larger PCTs take longer to process than smaller ones, in both the IO-intensive data setup and CPU-intensive sections.
- Larger numbers of proposed contacts per customer makes the core algorithm work harder in the CPU-intensive section.
- A larger value of `Optimize | AlgorithmTuning | CustomerSampleSize` takes more memory and longer CPU-intensive processing than a smaller value. There is a tradeoff here, since larger values can give more optimal results. Also, smaller values increase the likelihood of encountering sample-related provisioning problems. Check both the session level advanced settings and the configuration properties.
- If you use a time interval in your rules, this interval adds processing time in two ways:
 1. Contact history is queried, and this query can be slow since those tables are often large.
 2. The number of rules is multiplied by the number of time windows required by the interval. This condition makes the CPU-intensive part do more work.

Configure multiple threads

You can decrease run time of the CPU-intensive section by configuring multithreading if the following is true:

- You can see from the log timestamps that much of the session run time is in the CPU-intensive section.
- The Unica Optimize server is running on hardware that supports data-intensive processing in multiple threads.

Additional indexes for extra performance improvements

Beyond indexing the segment membership, contact history, and detailed contact history tables for each audience, and the PCT, POA, and RC tables for each session, you can index and update table statistics on other tables to improve the Unica Optimize session run performance.

By default, the Unica Optimize installer will not add these extra indexes on campaign tables. These indexes should be added only if you find that Unica Optimize session queries are running slower than expected. Adding more indexes can degrade the performance for update queries, such as insert or update, on these tables. You can add these extra indexes and update table statistics by manually running the following database-specific SQL scripts:

- `aco_extra_indexes_db2.sql`
- `aco_extra_indexes_sqlsvr.sql`
- `aco_extra_indexes_ora.sql`

These scripts can be found in the `<CAMPAIGN_HOME>/ddl` folder. Each script has three sections: delete index, create index, and update table statistics.

Index scripts need to be run only one time. However, update table statistics queries need to be run periodically to keep table statistics up-to-date. It is recommended to run these table statistics queries before you trigger a large Unica Optimize session run.

 **Note:** All three scripts contain an SQL script for the default detailed contact history table, `UA_DTLCONTACTHIST`. If a Unica Optimize session is defined on any other audience level, then the index and statistics SQL queries should be run on the appropriate detail contact history table.

Use cases that can negatively affect performance

This section lists various use cases that can negatively affect the performance of Unica Optimize.

Smart offer lists with rules using offer versions

If you use smart offer lists with rules that use offer versions, there are additional queries that are used in the IO-intensive data setup section. When the number of offers in the lists is large, and the number of attributes per offer is large, the time that is taken to run these queries can be great.

High maximum for iterations per customer sample

The maximum number of iterations to use for each customer sample is configurable by using the `Optimize|AlgorithmTuning|MaxIterationsPerCustomerSample` property. Check both the session level advanced settings and the configuration properties.

Depending on the rules and data, this limit might not be reached by a customer sample. High values guarantee the highest level of optimality of the results, but often the use of a greater number of iterations does not make a large enough improvement in optimality to justify the performance penalty. Typically, five iterations yield an acceptable degree of optimality, and it is unusual to see more than about a dozen or so iterations ever required.

To analyze the customer sample iteration behavior, search in the Unica Optimize log for the string `Iteration:`. This log entry is followed by a number, indicating which iteration it is. Each chunk starts at iteration 1 and counts up. It helps to see what is going on if you get a count of each iteration number in the log, and use the results to construct a histogram.

High number of unprocessable customers

Another major factor in performance is the number of unprocessable customers. If the value of the `Optimize|AlgorithmTuning|MaxAlternativesPerCustomerEvaluated` property is a large number (over 100 or so), the time penalty is high whenever a customer is unprocessable.

When you have many unprocessable customers, look for logical errors in the rules or data. However, it is possible, especially with large numbers of proposed transactions per customer, that the time needed to get some per-customer solutions is high. If so, it might be best to reduce the value of the `MaxAlternativesPerCustomerEvaluated` parameter, accepting more unprocessable customers as a trade-off to improve performance.

In Unica Optimize version 7.5.3 and later, there is more detailed logging to show the minimum, maximum, and average number of alternatives that are evaluated for each customer sample.

Solver subroutine calls

If certain combinations of per-customer rules are used, a major performance penalty might be seen in some cases. This situation can occur when there is at least one per-customer Min/Max # Transactions rule where the minimum constraint is not zero, combined with one or more package rules.

 **Note:** In versions older than 7.5.3, "Never A with B" counts as a package rule here.

In addition to having these two rules, their scopes must overlap so both are applied to the same proposed transactions. In addition, the scores must be such that the preferred solution for a package rule causes the "Min/Max" rule to fall below its minimum. If all these conditions are met, the core algorithm cannot find the optimal results in an efficient way, and must use a slower call to the solver engine. You know that this condition is occurring if you see this message in the server log: `Solver subroutine parameters:`

If you are seeing performance issues from using "Never A with B" rules, the best way to improve performance is to upgrade to Unica Optimize version 7.5.3 or later.

Many cases where scores are the same

If there are many cases where the scores are the same, decision-making in the LRE can sometimes get inefficient. You can tell that this scenario is happening if you see this string in the server log: `Additional alternative generated:`

To avoid this situation, try assigning more varied scores to the proposed transactions.

High number of agents in the Agent Information table

When there is a high number of agents in the Agent Information table, the performance of the Unica Optimize session is degraded in which the agent rules are used. It is recommended to redesign your campaign and the related Unica Optimize session by splitting the agents into different Agent Information tables and Agent Customer Relationship tables. You must create different campaigns and Unica Optimize sessions according to the split list of agents.

Optimize a transaction query for performance improvements

You can use temporary tables instead of database-specific views to improve your performance.

In Unica Optimize, you can add business rules on offer version attributes to control and manage investments and resources that are allocated for offer versions. During a Unica Optimize session, for offer version rules, Unica Optimize creates a query to get a list of transactions that are in scope of the offer version conditions defined in the rules by marketers. This query is called Attribute Filter Query (AFQ).

The AFQ query uses PAV and APA database views. These views are created, used, and deleted during a Unica Optimize session run. The query mentioned in the PAV and APA views runs whenever the AFQ query runs. The result of these view queries is held in memory for the AFQ query, and if the result of these queries is millions of records, it is difficult for the database to keep this result in memory. This might result in an Out-Of-Memory error on the database server.

If there is at least one rule that has an Over The Period value greater than zero, Unica Optimize creates an HAV view.

To optimize the AFQ and improve your performance, you can enable the use of temporary tables. You can enable or disable the temporary table by going to **Advanced settings > Database tuning** and setting **Allow temporary tables** to `True`. You can also configure this parameter in the Configuration settings at `Affinium|Campaign|partitions|`

`partition1|Optimize|DatabaseTuning| AllowTempTables`. By default, this setting is enabled because there is an improvement in the performance of larger sessions when you use temporary tables.