IBM Unica Optimize
Version 8 Release 5
January 20, 2012

Troubleshooting Guide

Note
Before using this information and the product it supports, read the information in "Notices" on page 11.

This edition applies to version 8, release 5, modification 0 of IBM Unica Optimize (product number 5725-D22) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Optimize session troubleshooting tips

Many issues that you might encounter in Optimize sessions can be resolved by obtaining details from the session logs. You can then fix the specific problems that are identified.

Error	Description of Problem	Action to Take
Could not connect to the IBM® Unica Optimize server	This error indicates that IBM Unica Marketing cannot connect or communicate with the Optimize server.	Confirm that the Optimize listener is running. If it is not running, start the Optimize 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`, and `Campaign|unicaACOListener|useSSL`.
Failure to start Solver engine	This error usually indicates a missing or invalid license file.	Check for the license file in the `/bin` directory for `xpauth.xpr`
The LP solver was unable to find an optimal solution to the chunk 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.	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, use the `configtool` utility to change the `Optimize|Debug|SolverDebugEnable` item so it is no longer hidden. Then select this property in Configuration Manager, and set its value to `TRUE`. Now run the session again (making sure that you are in single thread mode). In the log directory, some files are created that start with the letters `ACO_`. The rest of the file name is made up of two numbers. The first number is the customer sample number and the second number is the LRE iteration for that customer sample. The file with the highest iteration number for the highest sample number contains the linear programming model that got the error. Send this file to IBM Unica Technical Support so that they can analyze it.

Error	Description of Problem	Action to Take
The generation loop was unable to eliminate all slack and surplus variables	The outer algorithm (the part that handles the capacity rules) progresses by creating alternative solutions to the per customer rules 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.	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.
Unprocessable customer	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 per customer rules 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 containing details of unprocessable customers in comma-separated values (CSV) format in the Optimize server log directory by setting the `Optimize|logging|enableBailoutLogging` property to `TRUE`. This file is `unprocessables_`*`10-digit-session-ID`*`.csv` and is located in the *`OptimizeInstallationDirectory`*`/partition/partition[n]/logs` directory.	Unprocessable customers can occur either because the rules and data make it impossible to get a solution, or because the algorithm exceeded the number of alternatives to solve the problem. The number of alternatives is configured by the `Optimize|AlgorithmTuning|MaxAlternativesPerCustomerEvaluated` property. Setting the value of this property to a higher number lowers the likelihood of the customer being unprocessable (assuming that it is not inherently unsolvable). However, but when it occurs, it also makes the performance penalty higher.
No offers	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, as long as 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.	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	No channel offer attribute values are defined.	You must define some channel offer attribute values.

Optimize session takes a long time to run

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

Before you start

1. Make sure that the session was not running with `Optimize|Debug|ExtraVerbose` enabled, as this setting causes slow run times.
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 the Optimize server logging level to MEDIUM or LOW.

Run a session to generate a clean log for troubleshooting

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

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

- The amount of time spent accessing the database to set up the data 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 Optimize is taking to access the database to set up the data needed for the session.

1. In the 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 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 these steps to find out how long Optimize is taking to process customer samples.

1. In the 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 spent in the CPU-intensive optimization section. If this action is where the bulk of the time is being spent (which is typically the case), 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 applying a set of scores to the offers in that chunk, finding the optimal solutions with those scores for the customers in the chunk by using the core algorithm, then using the result in the outer algorithm to find a new set of scores to try. Each time Optimize applies a set of scores, it counts as one chunk iteration. The amount of time 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 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 which can result in errors or suboptimal results that are a side-effect of the customer sample approach. The number of customer samples 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 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. In order to avoid this situation, any feature used in a capacity rule should be well-represented relative to the number of samples.

To display Optimize listener output to a console

Occasionally when debugging an issue or configuring performance, it might be useful to view the Optimize listener output in a console window.

1. Open the Optimize listener file, `ACOServer.bat` (Windows) or `ACOServer.sh` (UNIX), located in the `bin` directory under your Optimize installation directory for editing.
2. To display the Optimize server output to a console, keep the following line:
   - **ACOserver.bat**: `unica_aolsnr` (around line 44)
   - **ACOserver.sh**: `unica_aolsnr` (around line 160)
3. Save and close the file.

To not display the Optimize server output to a console, do the following:

- **Windows**: Configure Optimize listener as a windows service.
- **UNIX**: Use the following line in `ACOServer.sh` (the default):

```
unica_aolsnr > /dev/null &
```

ACOServer reference

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

The Optimize listener is the `ACOServer.bat` (Windows) or `ACOServer.sh` (UNIX) file located in the `bin` directory of your Optimize installation.

See "To display Optimize listener output to a console" on page 4 for instructions on editing the Optimize listener to show status in the console.

Setting	Description
OPTIMIZE_HOME	Full path to the Optimize installation directory
CAMPAIGN_HOME	Full path to the Campaign installation directory. If Campaign is installed on a separate host, the `CAMPAIGN_HOME` directory must be mounted (UNIX) or mapped as a network drive (Windows) and the full path must be specified. Configure the directory to have execute permissions.
ORACLE_HOME	If using Oracle database, set to your Oracle home directory
ORACLE_LIB	UNIX only- If using Oracle database, set to your Oracle lib directory. This value is normally $ORACLE_HOME/lib on 64-bit installations and $ORACLE_HOME/lib32 on 32-bit installations.
DB2_INSTANCE_DIR	UNIX only. If using DB2® database, set to the path to your DB2 instance directory script.

Setting	Description
UNICA_ACSYSENCODING	If you have a Chinese, Japanese, or Korean (CJK) character in your user name, you must set the environment variable `UNICA_ACSYSENCODING` equal to `UTF-8` so that the Optimize session runs without errors. However, if a user without a CJK character in their user name attempts to run an Optimize session while `UNICA_ACSYSENCODING` is equal to `UTF-8`, the Optimize session fails. One possible workaround is to create all user names without CJK characters. You can set `UNICA_ACSYSENCODING` in the Optimize listener file. The Optimize listener file, `ACOServer.bat` (Windows) or `ACOServer.sh` (UNIX), is located in the `bin` directory of your Optimize installation. **`ACOServer.bat`** - To set the variable for users with CJK characters in their user name, remove the comment to the line `set UNICA_ACSYSENCODING=UTF-8`. To configure the Optimize listener for users without CJK characters in their user name, add a comment to the line `set UNICA_ACSYSENCODING=UTF-8`. **`ACOServer.sh`** - To set the variable for users with CJK characters in their user name, remove the comments to the lines `UNICA_ACSYSENCODING=UTF-8` and `export UNICA_ACSYSENCODING`. To configure the Optimize listener for users without CJK characters in their user name, add comments to the lines `set UNICA_ACSYSENCODING=UTF-8` and `export UNICA_ACSYSENCODING`. You must stop and restart the Optimize listener in a new command prompt for these changes to take effect.

Chapter 2. General performance tips for Optimize

Keep in mind these points when making 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.
- 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

If you see from the log timestamps that much of the session run time is in the CPU-intensive section, and the Optimize server is running on hardware that supports data-intensive processing in multiple threads, the run time of the CPU-intensive section can be greatly decreased by setting up the configuration for multithreading.

Additional indexes for additional 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, there are other tables you can index to improve the Optimize session run performance.

The Optimize installer indexes these tables properly. However, if you have issues with installation or upgrade, you might need to index these tables manually.

The specific tables and columns you need to index depends on your installation and configuration of IBM Unica Campaign and Optimize, your specific data, and the specific optimization rules you are using. The following table lists tables and columns to index that have improved performance in several instances.

Table	Columns	Details
*audience*_dtlContactHist	• *Audience ID* • ContactDateTime • TreatmentInstID • ContactStatusID	Each *Audience ID* column must match the corresponding Audience ID defined in Campaign.

Table	Columns	Details
UA_Treatment	• OfferID • TreatmentInstID • HasDetailHistory • PackageID • CellID • OfferHistoryID	You should also allow reverse scans on these indexes.
UA_OfferAttribute	AttributeID	You should also allow reverse scans on these indexes.
UA_Offer	OfferTemplateID	You should also allow reverse scans on these indexes.
UA_OfferTemplAttr	OfferTemplateID	You should also allow reverse scans on these indexes.

Use cases that can negatively affect performance

This section lists various use cases that can negatively affect the performance of 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 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 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 using the Optimize|AlgorithmTuning|MaxIterationsPerCustomerSample property.

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

In Optimize version 7.5.3 and later, there is more detailed logging to show the minimum, maximum, and average number of alternatives 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 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 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 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.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information about the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
170 Tracer Lane
Waltham, MA 02451
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not

been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Printed in USA