Version 11 Release 0.1
31 August, 2018

IBM Contact Optimization Troubleshooting and Tuning Guide

IBM

Note

Before using this information and the product it supports, read the information in “Notices” on page 17.

This edition applies to version 11, release 0, modification 1 of IBM Contact Optimization and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Contact Optimization troubleshooting tips 1

Contact Optimization installation errors 1

Contact Optimization session run errors 1

Errors in Contact Optimization session results . . . 3

Contact Optimization session takes a long time to run 5

Sample-related provisioning problems 7

Displaying Contact Optimization listener output to a console 8

ACOServer reference 8

Chapter 2. General tuning and performance tips for Contact Optimization 11

Additional indexes for extra performance improvements 11

Use cases that can negatively affect performance . . 12

Optimize a transaction query for performance improvements 13

Before you contact IBM technical support 15

Notices 17

Trademarks 19

Privacy Policy and Terms of Use Considerations . . 19

Chapter 1. Contact Optimization troubleshooting tips

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

Contact Optimization installation errors

If you encounter any errors while you are installing Contact Optimization, 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:

```
IBM Optimize 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 Contact Optimization on an AIX® system, even if the Contact Optimization listener is not running. The Contact Optimization upgrade installation updates specific libraries. Occasionally on AIX systems, even after you stop the Contact Optimization listener, the libraries are not unloaded from memory.

Solution

1. If you have already run the Contact Optimization installation, run the uninstaller.
2. Confirm the Contact Optimization listener is not running.
3. Run the **slibclean** command.
4. Run the Contact Optimization installation.

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

Contact Optimization session run errors

You might have configuration or other errors that prevent a Contact Optimization session from running.

Could not connect to the Contact Optimization server

Description of problem

This error indicates that Campaign cannot connect or communicate with the Contact Optimization server.

Solution

Confirm that the Contact Optimization listener is running.

If it is not running, start the Contact Optimization 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`

Unable to unlock the session with session ID

Description of problem

This error is written to the Contact Optimization 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 `ACOServer` file appropriate for your operating system and database.

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

Restart the Contact Optimization listener after you edit the `ACOServer` file.

Response XML is missing the data xml element

Description of problem

This error is written to the Contact Optimization 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 Contact Optimization session. This error indicates that the database credentials stored in the `<`*`Contact Optimization Installation Directory`*`>/Install/jdbc.properties` file are incorrect.

Solution

You can solve this issue in either of two ways.

- Replace `<`*`Contact Optimization Installation Directory`*`>/Install/jdbc.properties` file with the `<`*`Campaign Installation Directory`*`>/Install/jdbc.properties` file.
- Update `<`*`Contact Optimization Installation Directory`*`>/Install/jdbc.properties` with the correct credentials. Use the Marketing Platform **`encryptPasswords`** tool to encrypt the password.

Related reference:

"ACOServer reference" on page 8

Errors in Contact Optimization session results

Many issues that you might encounter in Contact Optimization sessions can be resolved by obtaining details from the session logs. Some of these errors might cause a Contact Optimization 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 Contact Optimization installation:

- Contact Optimization session log
- All files that start with `ACO_`

Send these files to IBM® 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 Contact Optimization installation:

- Contact Optimization session log
- All files that start with `ACO_`

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

You can use the `ContinueOnGenerationLoopError` configuration property to have Contact Optimization skip the set of customers if this error occurs or stop the Contact Optimization 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 Contact Optimization server log directory by setting the `Optimize|logging|enableBailoutLogging` property to `TRUE`. This file is `unprocessables_`*`10-digit-session-ID`*`.csv` and is in the `partition/partition[n]/logs` directory in the Contact Optimization installation directory. Customers skipped because they cannot be processed are marked with `unprocessable` in the `unprocessables_`*`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.

Contact Optimization does not satisfy Max limit of Min/Max # Offers Capacity rule

Description of problem

Contact Optimization 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 Contact Optimization 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 Contact Optimization might not meet capacity rules.

Contact Optimization session takes a long time to run

Here are troubleshooting steps that you can follow if you believe your Contact Optimization 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 Contact Optimization session takes a long time to run after you follow the steps in “Before you start” on page 5, you need to collect detailed information to analyze.

Run a session to generate a clean Contact Optimization server log with the HIGH or ALL setting on. While your session is running, do not access any Contact Optimization 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 Contact Optimization is taking to access the database to set up the data that is needed for the session.

1. In the Contact Optimization 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 Contact Optimization is taking to process customer samples.

In the Contact Optimization 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 Contact Optimization 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 Contact Optimization 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 Contact Optimization 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`.

Displaying Contact Optimization listener output to a console

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

1. Open the Contact Optimization listener file, `ACOServer.bat` (Windows) or `ACOServer.sh` (UNIX), in the `bin` directory under your Contact Optimization installation directory for editing.
2. To display the Contact Optimization 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 Contact Optimization server output to a console, do the following:

- **Windows**: Configure Contact Optimization 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 Contact Optimization listener manually.

The script to start or stop the Contact Optimization listener is the `ACOServer.bat` (Windows) or `ACOServer.sh` (UNIX) file in the `bin` directory of your Contact Optimization installation.

See "Displaying Contact Optimization listener output to a console" for instructions on editing the Contact Optimization listener to show status in the console.

Table 1. ACOServer script reference

Setting	**Description**
OPTIMIZE_HOME	Full path to the Contact Optimization installation directory
CAMPAIGN_HOME	You must set `CAMPAIGN_HOME` to point to `OPTIMIZE_HOME` in `ACOServer.bat/.sh`.
ORACLE_HOME	If using Oracle database, set to your Oracle home directory
ORACLE_LIB	UNIX only: If using an 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 a DB2® database, set to the path to your DB2 instance directory script.

Table 1. ACOServer script reference (continued)

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 Contact Optimization session runs without errors. You can set `UNICA_ACSYSENCODING` in the Contact Optimization listener file. The Contact Optimization listener file, `ACOServer.bat` (Windows) or `ACOServer.sh` (UNIX), is in the `bin` directory of your Contact Optimization 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 Contact Optimization 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 Contact Optimization 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 Contact Optimization listener in a new command prompt for these changes to take effect.

Chapter 2. General tuning and performance tips for Contact Optimization

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 Contact Optimization 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 Contact Optimization session run performance.

By default, the Contact Optimization installer will not add these extra indexes on campaign tables. These indexes should be added only if you find that Contact Optimization 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 `<OPTIMIZE_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 Contact Optimization session run.

Note: All three scripts contain an SQL script for the default detailed contact history table, `UA_DTLCONTACTHIST`. If a Contact Optimization 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 Contact Optimization.

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 Contact Optimization 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 Contact Optimization 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 Contact Optimization 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 Contact Optimization session is degraded in which the agent rules are used. It is recommended to redesign your campaign and the related Contact Optimization session by splitting the agents into different Agent Information tables and Agent Customer Relationship tables. You must create different campaigns and Contact Optimization 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 IBM Contact Optimization, you can add business rules on offer version attributes to control and manage investments and resources that are allocated for offer versions. During a Contact Optimization session, for offer version rules, IBM Contact Optimization 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 Contact Optimization 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, IBM Contact Optimization 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.

Before you contact IBM technical support

If you encounter a problem that you cannot resolve by consulting the documentation, your company's designated support contact can log a call with IBM technical support. Use these guidelines to ensure that your problem is resolved efficiently and successfully.

If you are not a designated support contact at your company, contact your IBM administrator for information.

Note: Technical Support does not write or create API scripts. For assistance in implementing our API offerings, contact IBM Professional Services.

Information to gather

Before you contact IBM technical support, gather the following information:

- A brief description of the nature of your issue.
- Detailed error messages that you see when the issue occurs.
- Detailed steps to reproduce the issue.
- Related log files, session files, configuration files, and data files.
- Information about your product and system environment, which you can obtain as described in "System information."

System information

When you call IBM technical support, you might be asked to provide information about your environment.

If your problem does not prevent you from logging in, much of this information is available on the About page, which provides information about your installed IBM applications.

You can access the About page by selecting **Help > About**. If the About page is not accessible, check for a `version.txt` file that is located under the installation directory for your application.

Contact information for IBM technical support

For ways to contact IBM technical support, see the IBM Product Technical Support website: (http://www.ibm.com/support/entry/portal/open_service_request).

Note: To enter a support request, you must log in with an IBM account. This account must be linked to your IBM customer number. To learn more about associating your account with your IBM customer number, see **Support Resources > Entitled Software Support** on the Support Portal.

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 on 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.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, 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 Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites 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
B1WA LKG1
550 King Street
Littleton, MA 01460-1250
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.

Privacy Policy and Terms of Use Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. A cookie is a piece of data that a web site can send to your browser, which may then be stored on your computer as a tag that identifies your computer. In many cases, no personal information is collected by these cookies. If a Software Offering you are using enables you to collect personal information through cookies and similar technologies, we inform you about the specifics below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name, and other personal information for purposes of session management, enhanced user usability, or other usage tracking or functional purposes. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

Various jurisdictions regulate the collection of personal information through cookies and similar technologies. If the configurations deployed for this Software Offering provide you as customer the ability to collect personal information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for providing notice and consent where appropriate.

IBM requires that Clients (1) provide a clear and conspicuous link to Customer's website terms of use (e.g. privacy policy) which includes a link to IBM's and Client's data collection and use practices, (2) notify that cookies and clear gifs/web beacons are being placed on the visitor's computer by IBM on the Client's behalf along with an explanation of the purpose of such technology, and (3) to the extent required by law, obtain consent from website visitors prior to the placement of cookies and clear gifs/web beacons placed by Client or IBM on Client's behalf on website visitor's devices

For more information about the use of various technologies, including cookies, for these purposes, See IBM's Online Privacy Statement at: http://www.ibm.com/privacy/details/us/en section entitled "Cookies, Web Beacons and Other Technologies."

Printed in USA