HCLSoftware

Unica Campaign Tuning Guide V12.1.9

Contents

Chapter 1. Unica Campaign performance configuration overview

The purpose of this document is to improve performance of flowchart execution, which is the core of the Unica Campaign application. The performance of Unica Campaign is tied closely to database performance. Optimal settings of database-related parameters can significantly improve overall Unica Campaign application performance.

Unica Campaign is a marketing campaign management application. An installation of Unica Campaign consists of multiple components, including Unica Platform and Unica Campaign. The installation also relies on other tools such as web application servers and databases.

All of these components have properties, features, and settings that you can configure to improve performance. Unica Campaign itself has a number of configuration properties which you can use to tune your installation for best performance.

Defining "best performance" is difficult. Every environment and implementation has different requirements. Unica Campaign performance can be affected by many factors, including hardware, software, and network configuration.

The following environment was used as the basis for Unica Campaign performance configuration testing:

- Unica Campaign v11.1
- AIX® (7.1)
- WAS (8.5.5.12 ND)
- DB2® (11.1)

Chapter 2. Database tuning for Unica Campaign: DB2®

A good starting point for tuning your configuration is to use the DB2® AUTOCONFIGURE command. This command generates values for parameters based on your responses to questions about workload characteristics.

The AUTOCONFIGURE command calculates and displays initial values for the buffer pool size, database configuration, and database manager configuration parameters, with the option of applying these recommended values.

The following auto configuration script suggests the database current and recommended parameter values based on the current workload. The insights can then be used to configure the parameter values accordingly.

```
"AUTOCONFIGURE USING MEM_PERCENT 60 WORKLOAD_TYPE MIXED
NUM_STMTS 500 ADMIN_PRIORITY BOTH IS_POPULATED YES NUM_LOCAL_APPS 0
NUM_REMOTE_APPS 20 ISOLATION RR BP_RESIZEABLE YES APPLY NONE "
```

Use database load utilities

You can improve performance significantly by using a database load utility for all datasources. Database load utilities are available from your database vendors.

About this task

The basic procedure to configure Unica Campaign for use with a database loader is summarized below. Follow these steps for each datasource.

Note: These steps do not apply to every combination of database type and operating system. For detailed instructions, along with troubleshooting advice, see the *Unica Campaign Administrator's Guide*.

1. Create two load control file templates: one for adding records and one for appending records.
2. Create a script or executable to start the load utility. Examples are provided in the *Unica Campaign Administrator's Guide*.
3. In Unica Campaign, go to `Campaign|partitions|partition1|dataSources|<datasourcename>` and set the properties that begin with the word **Loader**. These properties identify the control file templates and indicate the location of the script or executable file.

Spread tablespaces across multiple disks

A table space is a logical unit of storage in a database. Generally speaking, spreading database table spaces across multiple disks improves performance.

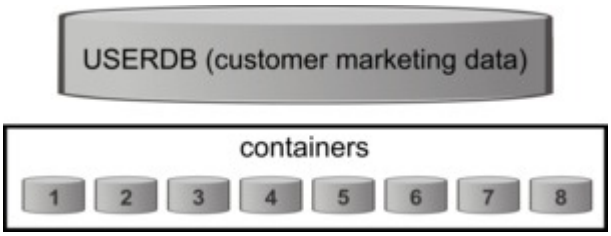
A table space can be System Managed Space (SMS) or Database Managed Space (DMS). Each table space is a collection of containers. A container is a data storage location, such as a file, directory, or device. DB2® spreads data across containers so you can store data on multiple disks for greater speed and storage capacity.

Recommendations:

- Before you create the database, make sure that you have multiple disks to split the table space containers. This approach helps to minimize I/O and improve overall performance.
- Keep database containers and LOG files in different locations.
- Split table spaces across multiple disks and keep them separate from the LOG file disk.
- Create a user temporary table space and split it across multiple disks.
- The LOGFILESIZ parameter defines the size of each primary and secondary log file. The default value of LOGFILSIZ is 1024, which might not be sufficient when deploying the Unica Campaign application and populating the data into the tables. Consider increasing the LOGFILSIZ, LOGPRIMARY, and LOGSECOND based on the number of transactions that you anticipate.

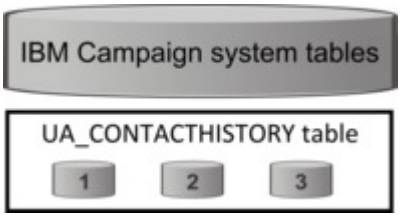
Example 1: User database

During performance testing, disk utilization on the User Database machine pertaining to I/O is observed to go up to 100%. The database has a tablespace with containers spanned over two disks. After tuning and spreading containers over a total of 8 disks, in certain cases you may still see spikes but the average consumption is brought below 20% while running 5 concurrent complex flowcharts.



Example 2: System database server

There is a possibility of Disk I/O contention on the Unica Campaign System database server as well. Depending on your flowchart, a large amount of data may be written to the `UA_CONTACTHISTORY` table. For example, say you are running five multiple concurrent complex flowcharts, which are writing a large amount of data to `UA_CONTACTHISTORY` simultaneously. In this case, spreading database table spaces across multiple disks can improve performance.



Generally speaking, spreading database table spaces across multiple disks improves performance. Whenever possible, create a database having tablespaces with containers spread across multiple disks at the time of the Unica Campaign installation.

Index databases

In general, the fastest way to access data from a database is to use an index. Indexes increase the efficiency of finding a specific piece of data. Indexing provides an efficient and fast way to identify the data (rows) in a table.

Recommendations:

- Index every primary key and most foreign keys in the database.
- Always index audience ID fields.
- Index columns that are joined in queries.
- Index columns involved in ORDER BY and GROUP BY.
- Index columns that perform sorting operations, including UNION and DISTINCT.
- Consider indexing any attributes that are frequently referenced in SQL WHERE clauses.
- Use an index for both equality and range queries.

When you use indexing, keep the following guidelines in mind:

- Add indexes only when absolutely necessary. Indexes significantly impact INSERT, UPDATE, and DELETE performance, and they require storage.
- Avoid or remove redundant indexes. For example, two indexes that use the same or similar columns make query optimization more complicated and consume more storage.
- Carefully choose one clustered index for each table.
- Avoid indexing columns that consist of LONG character strings.

Partition databases

In the case of huge data (millions of records), consider partitioning databases and objects.

The DB2® database manager allows great flexibility in spreading data across multiple database partitions of a partitioned database. You can choose how to distribute your data by declaring distribution keys. To determine which and how many database partitions your table data is spread across, you can select the database partition groups and table spaces where you want to store the data.

Partition tables

Table partitioning can improve performance. Table partitioning is a data organization scheme in which table data is divided across multiple storage objects called data partitions or ranges, according to values in one or more table columns.

With table partitioning, each index can be placed in its own table space, regardless of the table space type. Each data partition is stored separately. These storage objects can be in different table spaces, in the same table space, or a combination of both.

Without table partitioning, all indexes for a particular table are stored in the same storage object by default.

Perform database maintenance

For best performance, perform periodic maintenance activity on large tables by running a command such as RUNSTATS.

The DB2® RUNSTATS command updates statistics in the system catalog about the characteristics of a table and/or associated indexes, or statistical views. It is highly recommended that you use the DB2® RUNSTATS command to collect

current statistics on tables and indexes, especially if significant update activity has occurred or new indexes have been created since the last time the RUNSTATS command was run. This command provides the optimizer with the most accurate information with which to determine the best access plan.

Example:

```
runstats on table DB2INST2.UA_CONTACTHISTORY and detailed indexes all
```

Consider the case of a table that could have a large variation in the amount of data it contains, at any given moment. The volatility or extreme changeability of this type of table makes reliance on the statistics collected by RUNSTATS inaccurate. Statistics are gathered at, and only reflect, a single point in time.

To generate an access plan that uses a volatile table can result in an incorrect or poorly performing plan. For example, if statistics are gathered when the volatile table is empty, the optimizer tends to favor accessing the volatile table using a table scan rather than an index scan.

To prevent this type of issue, consider declaring the table as volatile using the ALTER TABLE statement. By declaring the table volatile, the optimizer will consider using an index scan rather than a table scan. Access plans that use declared volatile tables do not depend on the existing statistics for that table.

```
"ALTER TABLE <table_name> VOLATILE CARDINALITY"
```

Chapter 3. Setting in-database optimization to improve flowchart performance

Using in-database optimization can improve flowchart performance. When in-database optimization is on, processing is done on the database server and output is stored in temporary tables on the database server whenever possible.

About this task

You can apply in-database optimization in two ways: globally and for individual flowcharts. The best practice is to turn off the global configuration setting and set the option at the flowchart level.

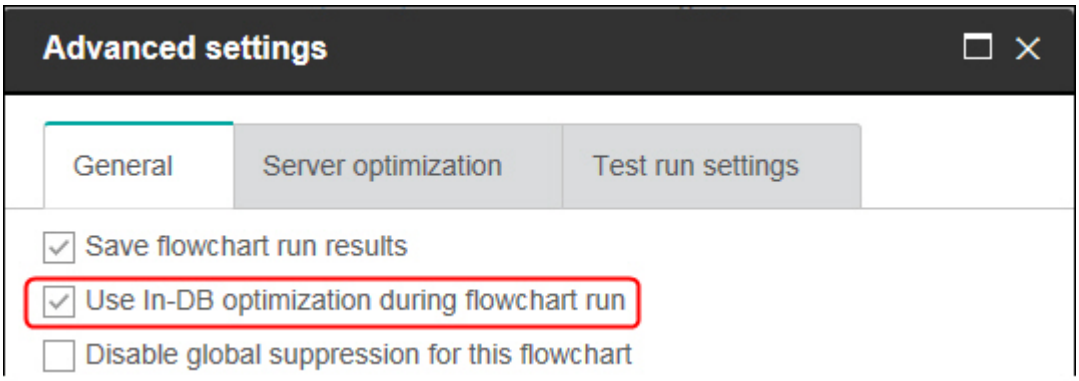
1. To adjust the option globally, at the partition level:
    a. Choose **Settings > Configuration**.
    b. Choose **Unica Campaign > partitions > partition[n] > server > optimization**.
    c. Set **useInDbOptimization** to TRUE (on) or FALSE (off).
2. To override the option for an individual flowchart:

    a. Open a flowchart in **Edit** mode.

    b. Open the **Admin** menu  and select **Advanced settings**.

    c. Select or clear **Use In-DB optimization during flowchart run**.

    Example

    When you save and run the flowchart, in-database processing will be used whenever possible, if you are using in-database optimization.

Note: In-database processing cannot be done if you specify any limitations on the output cell size or if temporary tables are disabled for a process.

Note: During a long running campaign flowchart, Campaign application loses connection to the DB2 database. Long running Campaign flowchart fails with SQL30081N error on Campaign server log file. To run

long running Campaign Flowchart, you must set the `STALE_CONN_TIMEOUT` (seconds time period) environment variable in Campaign listener (`rc.unica_ac/cmpServer.bat`). script files. Campaign server finds that if connection is idle for (`STALE_CONN_TIMEOUT`) seconds time period, it will re-open that connection.

Details about in-database optimization

In-database optimization avoids copying IDs from the database to the Unica Campaign server for processing whenever possible. This option can improve flowchart performance.

In-database optimization determines:

- Whether operations are done on the database server or the local Unica Campaign server; and
- Where the results of operations are stored.

When in-database optimization is on:

- Processing tasks such as sorting, joining, and merging data are done on the database server whenever possible.
- Output cells of processes are stored in temporary tables on the database server.

In-database optimization affects CPU consumption:

- When in-database optimization is on, more CPU is consumed on the database server.
- When in-database optimization is off, more CPU is consumed on the Unica Campaign server.

You can apply in-database optimization globally and override the global setting for individual flowcharts. The best practice is to turn off the global configuration property (**Use in-DB optimization**) and set the option at the flowchart level (**Advanced Settings > Admin > Use in-DB optimization during flowchart run**).

Important: In-database processing cannot be done if you specify any limitations on the output cell size or if temporary tables are disabled for a process.

Limitations of in-database optimization

- In-database optimization is not supported for all databases.
- Depending on the logic that is required, some functions are still performed on the Unica Campaign server, even with in-database processing turned on. Some examples are given below:
  - The query uses tables from different data sources.

    For example, if a Select process queries different data sources, Unica Campaign automatically stores the ID lists for those cases on the application server.
  - The query contains non-SQL macros or derived fields.

For example, to calculate a derived field, Unica Campaign evaluates the derived field formula to see whether any part of the calculation can be performed with SQL. If simple SQL statements can be used, the calculation is done in-database. If not, temporary tables are created on the Unica Campaign server to handle the calculations and persist the results from process to process within a flowchart.

Processing raw SQL in macros

Custom macros that consist of raw SQL statements can be processed in-database, within the following guidelines:

- All raw SQL custom macros must begin with `select` and contain exactly one `from` in the rest of the text.
- For databases that only support insert into `<TempTable>` syntax, you must map at least one base table to the same data source at the same audience level as the raw SQL custom macro. If the fields that are selected by the raw SQL custom macro are too large for the fields of the temp table, a runtime error occurs.
- If you use a raw SQL query in a Select process that has an input cell, you must use the `<TempTable>` token to obtain the correct list of audience IDs. Also use the `<OutputTempTable>` token to prevent audience IDs from being retrieved from the database back to the Unica Campaign server.
- If you use raw SQL with in-database optimization, you must code the raw SQL to join with the temp table from the upstream process. Otherwise, the results are not scoped by the results from the upstream process.

Chapter 4. Adjusting configuration properties in Unica Campaign to improve performance

You can adjust configuration properties in Unica Campaign and Unica Platform to improve performance.

1. To access the configuration settings, choose **Settings > Configuration**.
2. Adjust the following configuration properties.

Configuration properties that affect Unica Campaign performance

You can improve Unica Campaign performance by adjusting configuration properties.

DB2NotLoggedInitially

Configuration category

`Campaign|partitions|partition[n]|dataSources|`*`dataSourcename`*

Description

This property determines whether Unica Campaign uses the `not logged initially` SQL syntax when populating temporary tables in DB2®.

A value of `TRUE` disables logging for inserts into temp tables, which improves performance and decreases database resource consumption. When set to `TRUE`, if a temp table transaction fails for any reason, the table will become corrupted and must be dropped. All data previously contained in the table will be lost.

If your version of DB2® does not support the `not logged initially` syntax, set this property to `FALSE`.

If you are using a DB2® 11 user database on z/OS®, set this property to `FALSE`. If you are using DB2® 10.5 with the BLU feature ON for a user database, set both **DB2NotLoggedInitially** and **DB2NotLoggedInitiallyUserTables** to `FALSE`.

Default value

`TRUE`

Valid Values

`TRUE | FALSE`

AllowSegmentUsingSQLCase

Configuration category

`Campaign|partitions|partition[n]|dataSources|`*`dataSourcename`*

Description

This property specifies whether the Segment process consolidates multiple SQL statements into a single SQL statement, when specific configuration conditions are met.

Setting this property to TRUE results in significant performance improvements when all following conditions are met:

- Segments are mutually exclusive.
- All segments come from a single table.
- Criteria for each segment are based on the macro language.

In this case, Unica Campaign generates a single SQL CASE statement to perform segmentation, followed by segment-by-field processing on the Unica Campaign application server.

Default value

TRUE

Valid Values

TRUE | FALSE

TempTablePostExecutionSQL

Configuration category

Campaign|partitions|partition[n]|dataSources|*dataSourcename*

Description

Use this property to specify a complete SQL statement that Unica Campaign runs immediately after the creation of a temporary table in a user data source or in the system tables database. For example, to improve performance, you can create an index on a temporary table immediately after its creation (see examples below). To enable the creation of temporary tables in a data source, the AllowTempTables property must be set to TRUE.

You can use tokens to substitute the table name (<TABLENAME>) and column names (<KEYCOLUMNS>) in the SQL statement, because the values are generated dynamically when the campaign runs.

This property is automatically added to the SQL expression without checking its syntax. If you use this property, make sure that it is a legal expression. You can enclose the string in quotation marks, but this is not required.

This property treats semicolons as delimiters to run multiple SQL statements. If your SQL statement contains semicolons and you want it to run as one statement, use a backslash as an escape character before the semicolons.

Note: If you are using stored procedures with this property, be sure that you use the correct syntax for your database.

Tokens available to **TempTablePostExecutionSQL** are described below.

Token	Description
<AMUSER>	This token is replaced with the Unica user name associated with the flowchart for which temp tables were created.
<CAMPAIGNCODE>	This token is replaced with the code for the campaign associated with the flowchart for which temp tables were created.
<CAMPAIGNNAME>	This token is replaced with the name of the campaign associated with the flowchart for which temp tables were created.
<DBUSER>	This token is replaced with the database user name for the database where the temp tables were created.
<FLOWCHARTNAME>	This token is replaced with the name of the flowchart associated with the temp table creation.
<KEYCOLUMNS>	This token is replaced with the temp table column name(s).
<TABLENAME>	This token is replaced with the temp table name.
<USER>	This token is replaced with the Unica Campaign user name of the user running the flowchart.

Default value

No default value defined.

Examples

The following value creates an index on the temp table just after its creation, to improve the data retrieval process: `CREATE INDEX IND_<TABLENAME> ON <TABLENAME> (<KEYCOLUMNS>)`

The following example for Oracle calls a stored procedure and uses backslashes to escape the semicolon: `begin dbms_stats.collect_table_stats()\; end\;`

AllowTempTables

Configuration category

`Campaign|partitions|partition[n]|dataSources|`*`dataSourcename`*

Description

This property specifies whether Unica Campaign creates temporary tables in the database. Creating temporary tables can significantly improve the performance of campaigns.

When the value is `TRUE`, temporary tables are enabled. Each time a query is issued against the database (for example, by the Segment process), the resulting IDs are written to a temporary table in the database. When an additional query is issued, Unica Campaign can use that temporary table to retrieve rows from the database.

A number of Unica Campaign operations, such as `useInDbOptimization`, rely on the ability to create temp tables. If temporary tables are not enabled, Unica Campaign retains the selected IDs in the Unica Campaign server memory. The additional query retrieves IDs from the database and matches them to the IDs in server memory. This can negatively impact performance.

You must have appropriate privileges to write in the database to use temporary tables. Privileges are determined by the database login that you provide when you connect to the database.

Default value

`TRUE`

Note: Typically, you set **AllowTempTables** to `TRUE`. To override the value for a specific flowchart, open the flowchart in Edit mode, select **Admin**  **> Advanced settings**, click the **Server optimization** tab, and select **Disallow use of temp tables for this flowchart**.

MaxRowFetchRecords

Configuration category

`Campaign|partitions|partition[n]|dataSources|`*`dataSourcename`*

Description

For performance reasons, it is best to keep this number low.

When the selected number of IDs is less than the value specified by the `MaxRowFetchRecords` property, Unica Campaign passes the IDs to the database one at a time, in separate SQL queries. This process can be very time-consuming. If the number of selected IDs is greater than the value specified by this property, Unica Campaign uses temporary tables (if allowed on the database source), or it pulls down all the values from the table, not including any unnecessary values.

Default value

`100`

UseMergeForTrack

Configuration category

`Campaign|partitions|partition[n]|dataSources|`*`dataSourcename`*

Description

This property implements SQL MERGE syntax to improve the performance of the Track process in flowcharts. This property can be set to `TRUE` for DB2®, Oracle, SQL Server 2008, and Teradata 12. It can also be used with other databases that support the SQL MERGE statement.

Default value

```
TRUE (DB2 and Oracle) | FALSE (all others)
```

Valid Values

```
TRUE | FALSE
```

MaxQueryThreads

Configuration category

`Campaign|partitions|partition[n]|dataSources|`*`dataSourcename`*

Description

This property specifies the upper limit on the number of simultaneous queries allowed to run against each database source from a single Unica Campaign flowchart. Higher values generally improve performance.

Unica Campaign runs database queries using independent threads. Because Unica Campaign processes run in parallel, it is common to have multiple queries running simultaneously against a single data source. If the number of queries to be run in parallel exceeds the MaxQueryThreads, the Unica Campaign server limits the number of simultaneous queries to the specified value.

The maximum value is unlimited.

Note: If `maxReuseThreads` is set to a non-zero value, it should be greater than or equal to the value of `MaxQueryThreads`.

Default value

Varies depending on the database

maxVirtualMemory

Configuration category

`Campaign|partitions|partition[n]|server|optimization`

Description

It is used for internal locking of memory which prevents it from being swapped as temporary files.

Set a value equal to (80% x available memory) / (number of expected concurrent flowcharts). For example:

If available virtual memory on server = 32 GB
Number of concurrent flowcharts = 10
Set virtual Memory = (80 % x 32) / 10 = approximately 2.5 GB / flowchart

Default value

`128` (MB)

maxVirtualMemory is a global configuration setting. To override the value for a specific flowchart, open the flowchart in Edit mode, select **Advanced settings** from the **Admin** menu  , select the **Server optimization** tab, and change the **Campaign virtual memory usage** value.

doNotCreateServerBinFile

Configuration category

`Campaign|partitions|partition[n]|server|optimization`

Description

To improve performance, set this property to `TRUE`. When this property is `TRUE`, strategic segments create Segment temp tables in the data source rather than creating binary files on the Unica Campaign server. You must specify at least one data source in the Create Segment (CreateSeg) process configuration dialog to hold the temp tables. Also, you must set the `AllowTempTables` property to `TRUE` to enable the creation of temporary tables in a data source.

Default value

FALSE

Valid Values

`TRUE | FALSE`

httpCompressionForResponseLength

Configuration category

`Campaign|partitions|partition[n]|server|optimization`

Description

This property enables and configures compression for HTTP responses from the Unica Campaign web application to the client browser for flowchart-specific messages. The Unica Campaign web application reads this property only once for each partition. If you modify this property, you must restart the web application for the change to take effect.

Compression can improve page load and interaction times by reducing the amount of data that is sent over HTTP.

All responses that have a data length greater than or equal to the `httpCompressionForResponseLength` value (in KB) are candidates for compression. Any other responses are not compressed.

Compression reduces network transfer, but it requires resources on the server side. Therefore, compression makes sense only for large amounts of data, when sufficient server-side resources are available. If you typically have network delays that can slow large data transfers, you can analyze how much time it takes to load a given amount of data. For example, suppose that some of your HTTP requests are <100 KB in size, but most are 300 to 500 KB. In this case, you would increase the value of this property to 500 KB so that only responses >= 500 KB in size are compressed.

To disable compression, set the value to 0.

Default value

`100` (KB)

Valid Values

`0` (disables compression) or higher

cacheSystemDSQueries

Configuration category

`Campaign|partitions|partition[n]|server|optimization`

Description

To improve performance, set this value to TRUE. When set to TRUE, this property reduces multiple execution of queries on the Unica Campaign system tables by caching the query results. When set to FALSE, query results are not cached.

Default value

`TRUE`

Valid Values

`TRUE | FALSE`

keepFlowchartLogOpen

Configuration category

`Campaign|partitions|partition[n]|server|logging`

Description

This property specifies whether Unica Campaign opens and closes the flowchart log file each time a line is written to the log file.

A value of `TRUE` can improve performance of real-time interactive flowcharts. When the value is `TRUE`, Unica Campaign opens the flowchart log file only once, and closes it when the flowchart's server process exits. A side

effect of using the TRUE value is that recently-logged messages may not be immediately visible in the log file, as Unica Campaign flushes the log messages to file only when its internal buffer becomes full or when the number of logged messages equals the value of the logFileBufferSize property.

If the value is FALSE, Unica Campaign opens and closes the flowchart log file.

Default value

FALSE

Valid Values

TRUE | FALSE

loggingLevels

Configuration category

Campaign|partitions|partition[n]|server|logging

Description

The **loggingLevels** property controls the amount of detail written to the Unica Campaign server log file, based on severity.

Default value

MEDIUM

Valid Values

LOW: represents the least detail (the most severe errors only)

MEDIUM

HIGH

ALL: includes trace messages and is intended primarily for diagnostic purposes

Note: You may want to set **loggingLevels** to ALL during configuration and testing. This value generates a large amount of data and therefore may not be advisable for production operation. Setting any logging level higher than its default can adversely affect performance.

You can adjust these settings from within a flowchart by using **Tools > Logging options**.

logFileBufferSize

Configuration category

Campaign|partitions|partition[n]|server|logging

Description

This property is used when **keepFlowchartLogOpen** is TRUE. Specify a value to indicate the number of messages to buffer before writing to the log. If the value is 1, every log message is written immediately to file, effectively disabling buffering but having a negative impact on performance.

This property is ignored if **keepFlowchartLogOpen** is FALSE.

Default value

5

cellCodeBulkCreation

Configuration category

Campaign|partitions|partition[n]|server|systemCodes

Description

A value of TRUE improves performance of the cell code generation utility during bulk creation of cell codes, because multiple cell codes are generated with a single invocation of the cell code generator. This is more efficient and is the recommended setting. A value of TRUE also improves performance when copying flowcharts, templates, and process boxes.

When the value is FALSE, the cell code generator is invoked once for each cell code generation. If cell code generation seems to take a long time for Segment, Sample, and Decision process boxes, or for the target cell spreadsheet, set this value to TRUE.

The default setting is FALSE to support existing customized implementations. If you are using a legacy custom-made cell code generation utility, leave this setting at its default value of FALSE until you implement a new custom utility. Then you can change its value to TRUE.

If you are not using a custom cell code generation utility, change the value to TRUE to take advantage of the efficiency improvements.

Default value

FALSE

Valid Values

TRUE | FALSE

Campaign | caching

Certain objects, such as offers, are cached in the web application server to improve response times in the Unica Campaign user interface. The Campaign|caching configuration properties specify the length of time that cached data is retained. Smaller values result in more frequent cache updates, which can adversely affect performance by consuming processing resources on both the web server and the database.

Client polling interval (ms)

Configuration category

`Platform|Scheduler`

Description

Unica Campaign polls the Unica Scheduler for jobs at regular intervals, specified in milliseconds by this value. The default value is 60 seconds. Avoid setting this property to any value less than 10000 (10 seconds), because doing so can decrease campaign performance.

Default value

`60000`

Status polling interval

Configuration category

`Platform|Scheduler|Schedule registrations|[Product]|[Object type]`

For Unica Campaign flowcharts, the path for this property is `Platform|Scheduler|Schedule registrations|Campaign|Flowchart`

Description

The Unica Scheduler polls the product at regular intervals to obtain the run status of scheduled objects (for example, flowcharts or mailings) that have not reported a status. The interval is specified in milliseconds. The default value is 10 minutes. A more frequent polling interval (a smaller value) can negatively affect system performance. A less frequent polling interval (a larger value) reduces the load on the system. For Unica Campaign, set a less frequent polling interval when you have a large number of Unica Campaign flowcharts that take more than 10 minutes to complete.

Default value

`600000`

Chapter 5. Troubleshooting Unica Campaign Performance

Performance depends on many factors, including your database and web server configuration, network connectivity, and Unica Campaign and Unica Platform configuration.

The following list provides a number of suggestions that may help to improve performance. Use this list to quickly identify possible areas for improvement, so you can make adjustments and rule out possible causes. In cases where more information is available, each suggestion points to the appropriate guide where you can find detailed information.

Web application server

- If you are using WebSphere, check the JVM heap size specified in the WebSphere profile. Typically, an initial setting of 512 and a maximum of 1024 (or depending on the server configuration) should suffice.
- If you are using WebLogic, set the JVM memory heap size parameters to 1024 by adding the following line to the setDomainEnv script: Set MEM_ARGS=-Xms1024m -Xmx1024m -XX:MaxPermSize=256m
- Under certain circumstances, deploying older legacy interactive channels or interactive channels with large deployment histories can stress the system and require 2048mb or greater of Campaign designtime and/or Interact runtime Java heap space.

  System administrators can adjust the amount of memory available to the deployment systems via the following JVM parameters:

  -Xms####m -Xmx####m -XX:MaxPermSize=256m

  Where the characters #### should be 2048 or higher (depending on their system load.) Note that a 64-bit application server and JVM are usually necessary for values greater than 2048.
- If you are using WebLogic, depending on which version of Campaign you are running, you may need to apply a patch (for Weblogic 10gR3). Or, for WebLogic 11gR1, you may need to explode the campaign war file, make certain changes, then rebuild the war file. For details, see the Installation or Upgrade Guide for the version of Unica Campaign that you are running. Also, see the *Unica Recommended Software Environments and Minimum System Requirements*.

Database

- Check with your DBA to see if your database is heavily loaded with other applications.
- Perform database tuning, as described in the *Unica Campaign Tuning Guide*.
- Configure database load utilities, as described in the *Unica Campaign Administrator's Guide*.
- If you created a new audience level, then your DBA created a table in the Unica Campaign system database to store response history for that audience level. Be sure the new table is indexed to improve performance.

Unica Campaign tools

- Delete orphaned temp files and tables on the application server. You can use the Unica Campaign cleanup utility (`unica_acclean`) to identify and then delete all orphaned temporary files and database tables in the current partition. The cleanup utility can be used on both the Unica Campaign system tables database and on user tables databases. For instructions, see the *Unica Campaign Administrator's Guide*.
- Use the Unica Campaign Server Manager (`unica_svradm`) to see if any unnecessary `unica_acsvr` processes are running in the background. The Status command identifies disconnected or orphaned processes. The kill command (kill -p processid#) removes the unnecessary processes. For a list of available commands and syntax, use the Help command or see the *Unica Campaign Administrator's Guide*.

Logging

- Confirm that the logging level is not set to DEBUG in the `log4j.properties` file for Unica Campaign (`<Campaign_home>/conf/campaign_log4j.properties`) and Unica Platform (`<Platform_home>/conf/log4j.properties`).
- Confirm that the configuration property `Campaign|partitions|partition [n]|server| logging|loggingLevels` is not set to **ALL**. This setting generates a large amount of data and therefore is not advisable for production operation.
- Setting any logging level higher than its default can adversely affect performance.
- Examine the Unica Campaign log files to identify possible issues. For example, look for warnings that occur repeatedly. There are log files for listeners, the web application, web connections, flowcharts, sessions, and other areas of the application. By default, most log files are in `<Campaign_home>/logs` and `<Campaign_home>/partitions/partition [n]/logs`. If you have a clustered listener configuration, additional log files are in the equivalent directories under `<campaignSharedHome>`. For more information, read about logging administration in the *Unica Campaign Administrator's Guide*.

Configuration

- Adjust the performance-related configuration settings as described in the *Unica Campaign Tuning Guide*.
- Look at the configuration properties in the `Campaign|caching` category (for example, **offerTemplateDataTTLSeconds**) to see how often the cache is refreshed. The default value is 600 (10 minutes). Smaller values result in more frequent cache updates, which can adversely affect performance by consuming processing resources on both the web server and the database. Configuration properties are described in the *Unica Campaign Administrator's Guide*.

Reports

If you have Cognos reports on your dashboards, be aware that reports require additional processing resources. Performance can become an issue when many users access dashboards that contain many reports on a regular basis. Cognos report portlets are the most resource-intensive.

For improved performance, use Cognos to schedule reports, then configure the portlet in Unica Platform so it uses the schedule. For more information, read about Cognos report performance considerations in the *Unica Platform Administrator's Guide*.

Flowcharts

- Use in-database optimization, as described in the *Unica Campaign Tuning Guide*.
- Adjust configuration settings that affect flowchart performance, as described in the *Unica Campaign Tuning Guide*.
- Consider using the Extract process to select fields from one table and write them out to another table for subsequent processing. The Extract process is designed to pare down a large amount of data to a manageable size for subsequent operations, which can result in performance improvements. For more information, see the *Campaign User's Guide*.
- Use the Unica Scheduler instead of the Schedule process in flowcharts. The Unica Scheduler is more efficient, as it does not consume server system resources when the flowchart is not running.

  Use throttling to manage performance when many processes are likely to place high demands on the system. Throttling is based on scheduler groups that you set up on the Settings > Configuration page. You assign a throttling threshold to a group, and associate schedules with that group. For more information, see the *Unica Platform Administrator's Guide*.
- Avoid profiling fields whose values are mostly unique, such as the Audience ID field. The Profile feature is more efficient (and useful) on fields with a smaller number of distinct values.

Table mapping

- Map an audience's segment membership table only if you plan to use that audience in flowcharts or Optimize sessions that use strategic segments. Strategic segments are persistent segments that can be used in multiple flowcharts or sessions. You create strategic segments by running the CreateSeg process in a session flowchart and saving the results. If you are not using strategic segments, do not map the segment membership table. Using strategic segments in Unica Campaign flowcharts or Optimize sessions is optional. If you map the segment membership table, Unica Campaign or Contact Optimization updates the table each time that you run the flowchart or Optimize session. This is unnecessary processing overhead if you are not using strategic segments. For more information, see the *Unica Campaign Administrator's Guide*.
- When you map user tables, be aware that the **Allow real-time profiling** option requires a database query each time a user clicks **Profile**, which can potentially degrade performance. The **Allow real-time profiling** option applies to all table fields, not just the checked ones. You can remap a user table to change the profiling characteristics. For details, see the *Unica Campaign Administrator's Guide*.

Network and components

- Use network monitoring tools to identify potential issues. For example:

  netstat (network statistics) is a command line tool that displays network connections (both incoming and outgoing), routing tables, and network interface statistics. This utility is available for use on both UNIX and Linux operating systems.

  tracert (Windows) / traceroute (UNIX) is a network diagnostic tool for displaying route paths and measuring delays of packets across a network.
- Windows Performance Monitor can generate reports on processor, memory, disk and network utilization.
- If you are using Interact, there is a performance cost if you configure any part of Interact to communicate using SSL. does not recommend configuring Interact to use SSL.
- You may want to look into the speed of the storage devices or appliance (at both the WebSphere or WebLogic and Application server levels) as this plays a role in performance.
- As with any application, problems can often be resolved by restarting the software and hardware. Try restarting the listener. Also try restarting the web application server. In some cases, you may need to reboot the physical servers as well.