

IBM Campaign
Version 9 Release 1
January 2014

Tuning Guide

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 23.

This edition applies to version 9, release 1, modification 0 of IBM Campaign and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2013, 2014.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. IBM Campaign performance configuration overview 1

Chapter 2. Setting up IBM Campaign to use database load utilities 3

Troubleshooting database load utilities for IBM Campaign 5
 Timeout and locking problems: DB2 database load utility. 5
 "Check pending" problems: DB2 database load utility. 6

Chapter 3. Database tuning for IBM Campaign: DB2 7

Spread tablespaces across multiple disks 7
Index databases 8
Partition databases 9
Partition tables 9
Perform database maintenance 9

Chapter 4. Setting in-database optimization to improve flowchart performance 11

Details about in-database optimization 11

Chapter 5. Adjusting configuration properties in IBM Campaign to improve performance 13

Configuration properties that affect IBM Campaign performance 13

Before you contact IBM technical support 21

Notices 23

Trademarks 25
Privacy Policy and Terms of Use Considerations . . . 25

Chapter 1. IBM Campaign performance configuration overview

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

IBM Campaign is a marketing campaign management application. An installation of IBM Campaign consists of multiple IBM components, including IBM Marketing Platform and IBM 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. IBM 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. IBM Campaign performance can be affected by many factors, including hardware, software, and network configuration.

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

- IBM Campaign v9.1
- AIX (7.1)
- WAS (7.0 ND)
- DB2 (9.7)

Chapter 2. Setting up IBM Campaign to use database load utilities

You can improve performance by using a database load utility for all datasources.

About this task

IBM Campaign supports the use of database load utilities, which are available from your database vendors. You must obtain licensed copies of any database load utilities.

Database load utilities can improve performance when pushing ID lists into temporary tables and when exporting data from IBM Campaign to the database. For example, data is exported during a Snapshot, Mail List, or Call List process.

Load utilities can provide significant performance improvements. Testing on DB2 indicates that without the load utility, insertion of a million rows requires about 5 times more CPU utilization and significant disk I/O. Results vary depending on the hardware in use.

The following instructions assume that you are using a DB2 database. If you are using a different database, adjust the instructions accordingly.

Important: The following adjustments can impact system resources and potentially impact performance numbers.

Procedure

To set up IBM Campaign to use a database load utility, there are three main steps to perform for each datasource: Create two load control file templates; create a script or executable to start the load utility; then set the loader configuration properties in IBM Campaign.

1. Create two load control file templates.

Most database load utilities require the use of a control file. IBM Campaign can generate control files dynamically based on control file templates that you create.

- a. Create a load control file template for adding records. The template must consist of the following lines. This example template is named

loadscript.db2:

```
connect to <DATABASE> user <USER> using <PASSWORD>;
load client from <DATAFILE> of del modified by coldel| insert into <TABLE>(
<FIELDNAME><,>
)
nonrecoverable;
```

- b. Create a load control file template for appending records. The template must consist of the following lines. This example template is named

loadappend.db2:

```
connect to <DATABASE> user <USER> using <PASSWORD>;
load client from <DATAFILE> of del modified by coldel| insert into <TABLE>(
<FIELDNAME><,>
)
nonrecoverable;
```

You now have templates for loading data into a new or empty database table and appending data to an existing database table.

IBM Campaign will fill in the DATABASE, USER, PASSWORD, DATAFILE, TABLE and FIELDNAME tokens in the templates and create a configuration file called CONTROLFILE for the DB2 Load.

2. Create a script or executable to start the load utility.

To invoke the load utility, IBM Campaign uses a shell script (or an executable, for Windows), which is identified in the **Loadercommand** configuration property. You can specify either a direct call to the database load utility executable or a call to a script that launches the database load utility.

- a. For this example, create a shell script called db2load.sh, to start the loader. For the /tmp path, you can substitute the directory of your choice:

```
#!/bin/sh
cp $1 /tmp/controlfile.tmp
cp $2 /tmp/db2load.dat
db2 -tvf $1 >> /tmp/db2load.log
```

- b. Change permissions on the script file so it has execute permission:
chmod 755 db2load.sh

3. Set loader configuration properties in IBM Campaign.

The loader configuration properties identify the control file templates and indicate the location of the script or executable file.

- a. Choose **Settings > Configuration**, then select Campaign|partitions|partition1|dataSources|<datasourcename>.
- b. Set the properties that begin with the word **Loader**. For important information, see the topic *Configuration properties for IBM Campaign* in the *IBM Campaign Administrator's Guide* or the online help.
 - **LoaderCommand**: The path to the script or executable to invoke the database load utility. Most database load utilities require several arguments to be launched successfully. The tokens required by DB2 are shown in angle brackets in the following example. Type them exactly as shown. They will be replaced by the specified elements when the command is run. Example: /Unica/Campaign/partition/partition1/db2load.sh <CONTROLFILE> <DATAFILE>
 - **LoaderCommandForAppend**: The path to the script or executable to invoke the database load utility for appending records to a database table. Example: /Unica/Campaign/partition/partition1/db2load.sh <CONTROLFILE> <DATAFILE>
 - **LoaderDelimiter** and **LoaderDelimiterForAppend**: The delimiter used in the loader control file template.
 - **LoaderControlFileTemplate**: The path to the control file template, relative to the current partition. Example: loadscript.db2
 - **LoaderControlFileTemplateForAppend**: The path to the control file template for appending records, relative to the current partition. Example: loadappend.db2
 - All other **Loader** settings: Specify as required for your implementation, according to the information provided in the topic *Configuration properties for IBM Campaign*.

The following image shows an example of Loader configuration settings.

LoaderCommand	/Unica/Campaign/partition/partition1/db2load.sh <CONTROLFILE><DATAFILE> (path of Shell script to start loader)
LoaderCommandForAppend	/Unica/Campaign/partition/partition1/db2load.sh <CONTROLFILE><DATAFILE> (path of Shell script to start loader)
LoaderControlFileSpecifiesFields	TRUE
LoaderControlFileTemplate	loadscript.db2
LoaderControlFileTemplateForAppend	loadappend.db2
LoaderDelimiter	
LoaderDelimiterAtEnd	FALSE
LoaderDelimiterAtEndForAppend	FALSE
LoaderDelimiterForAppend	
LoaderNULLValueInDelimitedData	
LoaderUseLocaleDP	FALSE

c. Be sure to adjust the **Loader** configuration settings for each datasource.

Results

IBM Campaign performs the following actions when it writes to the database: First, it creates a temporary data file as fixed width or delimited text. If specified by the **LoaderControlFileTemplate** property, a temporary control file is created dynamically based on the template file and the list of fields to be sent to the database. Next, it issues the command specified by the **LoaderCommand** configuration property. Finally, it cleans up the temporary data file and control file.

Troubleshooting database load utilities for IBM Campaign

Some known issues with database loader utilities are listed below, along with workarounds or solutions.

Timeout and locking problems: DB2 database load utility

Use the following suggestions to help troubleshoot timeout and locking problems that might occur when you use a DB2 database load utility with IBM Campaign.

Symptoms

Multiple flowcharts are running concurrently and the flowcharts write to the same table. A flowchart run fails with the following errors:

- IBM Campaign UI: "Loader command exited with an error status 4" and
- Loader logs: "SQL0911N The current transaction has been rolled back because of a deadlock or timeout".

For example, you are using multiple flowcharts to insert records to the UA_ContactHistory table using the Mail List process box.

Causes

The load utility does not support loading data at the hierarchy level. When you concurrently run multiple flowcharts that load data into the same table, each individual load process locks the table. Each load process has to wait until the previous load finishes. If a process takes a longer time to finish, the next load process in the queue times out and produces the errors listed above.

Table locking during load operations: In most cases, the load utility uses table level locking to restrict access to tables. The level of locking depends on the stage of the load operation and whether it was specified to allow read access.

A load operation in ALLOW NO ACCESS mode uses a super exclusive lock (Z-lock) on the table for the duration of the load. Before a load operation in ALLOW READ ACCESS mode begins, the load utility waits for all applications that began before the load operation to release their locks on the target table. At the beginning of the load operation, the load utility acquires an update lock (U-lock) on the table. It holds this lock until the data is being committed. When the load utility acquires the U-lock on the table, it waits for all applications that hold locks on the table prior to the start of the load operation to release them, even if they have compatible locks. This is achieved by temporarily upgrading the U-lock to a Z-lock, which does not conflict with new table lock requests on the target table as long as the requested locks are compatible with the load operation's U-lock. When data is being committed, the load utility upgrades the lock to a Z-lock, so there can be some delay in commit time while the load utility waits for applications with conflicting locks to finish.

Note: The load operation can time out while it waits for the applications to release their locks on the table prior to loading. However, the load operation does not time out while waiting for the Z-lock needed to commit the data.

Resolving the problem

Workaround: IBM Campaign uses a shell script (or an executable, for Windows) specified in the **Loadercommand** configuration property to invoke the database load utility. You can add queuing logic to your shell script or executable to work around this issue. This logic checks to see if one loader is running and performing load operation on a table; if this is the case, it does not allow other loaders to start loading until the previous one completes.

"Check pending" problems: DB2 database load utility

"Check pending" problems can occur when using a DB2 database load utility with IBM Campaign. Use the following suggestions to help troubleshoot these types of issues.

Symptoms

You get a SQL0668N error.

Causes

If a database loader is used to insert records in a table and the table has referential constraints, the table will be left in a "check pending" state after the load operation. Referential constraints include unique constraints, range constraints for partitioned tables, generated columns, and LBAC security rules. When a table is in this state, you get a SQL0668N error if any select query is performed on the table.

Resolving the problem

To get the table out of a "check pending" state, run the following command:

```
SET INTEGRITY FOR TABLE <TABLENAME> IMMEDIATE CHECKED
```

The following code can be used in the script:

```
load client from <DATAFILE> of del modified by coldel | insert into <TABLE>(  
  <FIELDNAME><,>  
)  
nonrecoverable;  
set integrity for <TABLE> immediate checked;
```

Chapter 3. Database tuning for IBM 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 "
```

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.

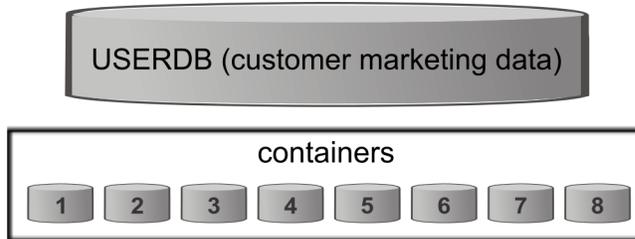
A table space can be System Managed Space (SMS) or Database Managed Space (DMS). Each table space is a collection of containers. A container is a data storage location, such as a file, directory, or device. DB2 spreads data across containers so you can store data on multiple disks for greater speed and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

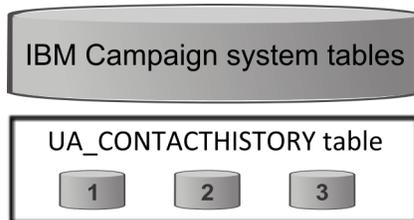
Example 1: User database

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 IBM 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 IBM 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 4. 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.

Procedure

1. To adjust the option globally, at the partition level:
 - a. Choose **Settings > Configuration**.
 - b. Choose **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**.



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.

Details about in-database optimization

In-database optimization avoids copying IDs from the database to the IBM 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 IBM 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 IBM 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 (**useInDbOptimization**) 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 required, some functions are still performed on the IBM 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, IBM 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, IBM 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 IBM 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 IBM 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 5. Adjusting configuration properties in IBM Campaign to improve performance

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

Procedure

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

Configuration properties that affect IBM Campaign performance

You can improve IBM Campaign performance by adjusting configuration properties.

DB2NotLoggedInitially

Configuration category

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

Description

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

When set to TRUE, this property disables logging for inserts in to 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.

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 of the following conditions are met:

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

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

Token	Description
<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 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 IBM 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, IBM Campaign can use that temporary table to retrieve rows from the database.

A number of IBM Campaign operations, such as `useInDbOptimization`, rely on the ability to create temp tables. If temporary tables are not enabled, IBM Campaign retains the selected IDs in the IBM 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.

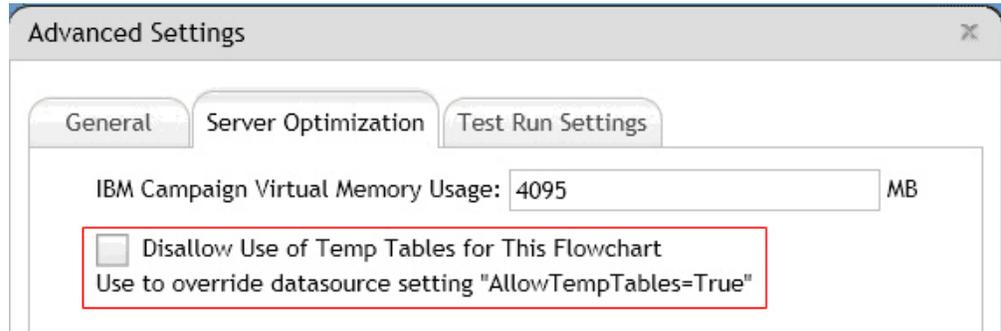
You must have appropriate privileges to write in the database to use temporary tables. Privileges are determined by the database login that you provide when you connect to the database.

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 **Advanced Settings** from

the **Admin** menu , then select **Disallow Use of Temp Tables for This Flowchart** on the Server Optimization tab.



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, IBM 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, IBM 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 IBM Campaign flowchart. Higher values generally improve performance.

IBM Campaign runs database queries using independent threads. Because IBM 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 IBM 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

This property specifies a default value for the maximum number of megabytes of system virtual memory to use when running flowcharts. You can increase the value to improve performance or decrease the value to limit the resources used by a single flowchart.

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

If available virtual memory on server = 32 GB

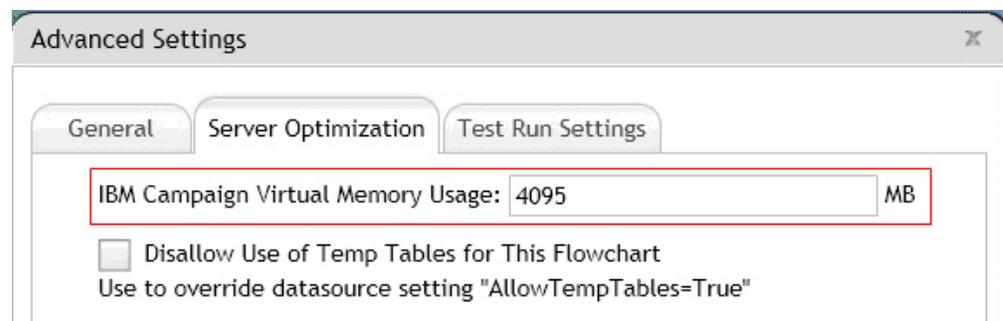
Number of concurrent flowcharts = 10

Set virtual Memory = (80 % x 32) / 10 = approximately 2.5 GB / flowchart

Default value

128

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 , and change the **IBM Campaign Virtual Memory Usage** value on the Server Optimization tab. The units are in megabytes.



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 IBM 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

keepFlowchartLogOpen

Configuration category

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

Description

This property specifies whether IBM 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, IBM 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 IBM 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, IBM Campaign opens and closes the flowchart log file.

Default value

FALSE

Valid Values

TRUE | FALSE

logFileBufferSize

Configuration category

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

Description

This property is used when the value of **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

Client polling interval

Configuration category

Platform|Scheduler

Description

IBM Campaign polls the IBM EMM 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 IBM Campaign flowcharts, the path for this property is
Platform|Scheduler|Schedule registrations|Campaign|Flowchart

Description

The IBM EMM 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 IBM Campaign, set a less frequent polling interval when you have a large number of Campaign flowcharts that take more than 10 minutes to complete.

Default value

600000

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.

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 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.
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 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](http://www.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