



This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

Outbound Contact Reference Manual

Recommended DBMS Optimizations

12/12/2025

Recommended DBMS Optimizations

Contents

- **1 Recommended DBMS Optimizations**
 - 1.1 Optimizing Calling List Tables for Weight Rules
 - 1.2 Temporary Tables Considerations
 - 1.3 Maintaining Indexes for Large Calling Lists
 - 1.4 Configuring IBM DB2 for Use of the LANGUAGE SQL Option
 - 1.5 Handling of the Backslash Escape Sequences

This topic explains how to perform recommended DBMS optimizations.

Optimizing Calling List Tables for Weight Rules

If you use a Sybase Adaptive Server as your DBMS and more than one calling list is using the same database table—that is, the calling lists refer to the same Table Access object in the configuration—Genesys recommends that you modify the default locking scheme on the database table in order for OCS to correctly maintain the weight rules for these calling lists.

Follow these procedures:

1. Stop and unload all campaigns/campaign groups that include calling lists referencing this database table.
2. Execute this SQL statement:

```
alter table <table_name> lock datarows
```

Note:

If the table contains several thousand calling records, the above SQL statement might take some time to execute.

Temporary Tables Considerations

In the Calling List Details view, to enable the user to navigate in a timely manner through calling list tables with a large number of customer contacts (three to four thousand or more records in the calling list table), OCM or Genesys Administrator uses an auxiliary temporary table that is generated when the calling list is opened.

The suggestions below reference the different DBMS types that use temporary tables (Microsoft SQL, Sybase, Informix, and DB2).

Microsoft SQL Server

Microsoft SQL Server 7.0 and above: Temporary tables are stored in the tempdb database. On Microsoft SQL (MS SQL) Server 7.0 and higher, the size of tempdb is automatically enlarged by default if required.

Microsoft SQL Server 6.5 and below: Microsoft SQL Server 6.5 requires manual adjustment of the size of this database. When a large calling list is opened, the size of tempdb may become insufficient; in that case an Out of space in tempdb error message will be returned by the DBMS and displays on OCM or Genesys Administrator's GUI. To resolve this problem and to enable GUI to process large calling lists, increase the size of the tempdb database.

Sybase

The same considerations regarding the size of tempdb as described above for Microsoft SQL Server 6.5 also apply to Sybase.

For additional details, please refer to your Sybase system documentation.

Informix

Operations on large calling lists will require sufficient space in the dbspace where temporary tables reside. You may need to increase this size. The dbspace where temporary tables are stored is defined by the "DBSPACETEMP" ONCONFIG parameter.

For additional details, please refer to your Informix system documentation.

DB2

Operations on calling lists require a temporary table. The DB2 engine stores this temporary table in User Temporary table space, which the database administrator (DBA) should explicitly create.

If this table space is absent, an attempt to open a Calling List in Genesys Administrator or OCM results in the following error message: DBServer returned SQL error [IBM][CLI Driver][...] SQL0286N. A default table space could not be found with a page size of at least <pagesize> that authorization ID <user-name> is authorized to use.

To resolve this problem, ensure that a table space of the correct type (User Temporary) with a page size of at least <pagesize> exists, and that the authorization ID <user-name> has USE privilege on this table space.

Oracle

Oracle 8.1 and below: On versions of Oracle before 8.1.7, Oracle does not use temporary tables, and OCM or Genesys Administrator performs sorting in its memory when an ORDER BY clause is issued.

Calling Lists containing several thousand records may require additional processing time, which may cause DB Server to force Genesys Administrator or OCM to disconnect if the application is too slow to respond. In this scenario, Genesys suggests that you add more conditions to the filter to limit the number of retrieved records (less than two to three thousand).

Note:	If you are using OCM, you can also increase the value of the db_timeout option.
--------------	---

Maintaining Indexes for Large Calling Lists

When working with large calling lists consisting of several thousand or more records, the performance of the DBMS for outbound dialing with OCS might be affected. Follow these guidelines to improve the performance of the DBMS in this scenario.

Creating Indexes

If a dialing filter is used, Genesys recommends that you create indexes for all fields used in the dialing filter into WHERE and ORDER BY clauses in the filter. Create indexes using the available DBMS tools or using the following SQL statements:

```
CREATE INDEX <unique__index_name> ON  
<calling_list_table_name> (<field_used_in_filter>)
```

Updating Indexes

If the data in a calling list table significantly changes, earlier indexes might become out of date. For example, if you create a calling list table and manually insert 10 customer contacts, then import into this table an additional 10,000 customer contacts, the indexes built on the table will become out of date.

To avoid possible performance issues, before starting a campaign/campaign group, update indexes in a table that has been changed significantly. To update indexes, you should recalculate statistics in the database.

Below are recommendations for recalculating statistics for the calling list table. To recalculate statistics, follow the recommendations for your DBMS type.

Note:	Statistics are updated automatically when you create or recreate an index for an existing table.
--------------	--

Microsoft SQL Server

Microsoft SQL Server versions 7.0 and higher automatically recalculate statistics when a table is significantly changed. The `auto update statistics` option (default = ON) controls recalculation.

If you are using an earlier version of SQL or if automatic statistic recalculation is switched off, you can manually recalculate statistics by executing the following statement:

```
UPDATE STATISTICS <calling_list_table_name>
```

This operation might be time-consuming depending on the size of the table.

For additional details, please refer to Microsoft SQL Server product documentation or your Database Administrator.

Sybase

Use the following statement to manually update statistics for a calling list table:

```
UPDATE STATISTICS <calling_list_table_name>
```

This operation might be time-consuming depending upon the size of the table.

For additional details, please refer to your Sybase system documentation.

Oracle

To manually recalculate statistics for a calling list table that has been significantly changed, use the following statement:

```
ANALYZE TABLE <calling_list_table_name> COMPUTE STATISTICS
```

This operation might be time-consuming depending upon the size of the table. You can also use procedures from the DBMS_STATS package to recalculate statistics.

For additional details, please refer to your Oracle documentation or consult your Database Administrator.

Informix

Use the following statement to manually update statistics for a calling list table:

```
UPDATE STATISTICS HIGH FOR TABLE <calling_list_table_name>
```

This operation might be time-consuming depending upon the size of the table.

For additional details, please refer to your Informix documentation or consult your Database Administrator.

Configuring IBM DB2 for Use of the LANGUAGE SQL Option

IBM DB2 must be properly configured to provide the ability to create and execute Stored Procedures with the LANGUAGE SQL option in the CREATE PROCEDURE statement. Refer to the IBM Support Website or DB2 product documentation for additional details.

Handling of the Backslash Escape Sequences

OCS stores the data in the DBMS as is, without processing of the backslash escape sequences. The DBMS must be configured to omit the processing of the backslash escape sequences; otherwise, the data might be altered in the DBMS or some unexpected errors might occur on the DBMS side while processing the queries containing any backslash escape sequences—for example, "\0".