Informatica® Multidomain MDM
10.3

Performance Tuning Guide

Informatica Multidomain MDM Performance Tuning Guide
10.3
September 2018

Table of Contents

Preface

The *Multidomain MDM Performance Tuning Guide* helps you optimize the overall performance of the Informatica® MDM Hub within the database and the application server environments.

This guide is intended for MDM administrators, database administrators, system administrators, and anyone else responsible for the performance of the Multidomain MDM environment.

Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.

- View product availability information.

- Create and review your support cases.

- Find your local Informatica User Group Network and collaborate with your peers.

Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

Informatica maintains documentation for many products on the Informatica Knowledge Base in addition to the Documentation Portal. If you cannot find documentation for your product or product version on the Documentation Portal, search the Knowledge Base at https://search.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

Introduction to Performance Tuning

This chapter includes the following topics:

Overview

You can use the performance tuning recommendations to configure specific parameters that optimize the performance of the MDM Hub. Experiment with these parameters to arrive at optimum values. You can also achieve a baseline performance by using these recommendations.

Factors that Influence the Performance of the MDM Hub

The following table lists the different components that you can fine-tune to optimize the performance of the MDM Hub:

Components	Description
Java	Parameters that you can fine-tune in the Java layer. These parameters are applicable for the MDM Hub and for the Process Servers, including clustered environments.
General Database Recommendations	Parameters such as sizing and storage that you can fine-tune in the database layer.
Oracle	Parameters that you can fine-tune in the Oracle database, including the RAC environment. These parameters are applicable for both the MDM Hub Master Database schema and any Operational Reference Store (ORS) schema.

Components	Description
Microsoft SQL Server	Parameters that you can fine-tune in the Microsoft SQL Server. These parameters are applicable for both the MDM Hub Master Database schema and any ORS schema.
IBM DB2	Parameters that you can fine-tune in the IBM DB2 database. These parameters are applicable for both the MDM Hub Master Database schema and any ORS schema.
The MDM Hub Specific Configuration	Parameters that you can configure in the MDM Hub. These parameters are applicable for both the MDM Hub Master Database settings and any ORS specific settings.
Batch Job Optimization	Parameters that you can configure for the better performance of the batch jobs.
Hub Console Optimization	Parameters that you can configure for the better performance of the Hub Console.
Informatica Data Director and Services Integration Framework Optimization	Parameters that you can configure for the better performance of Informatica Data Director (IDD) and Services Integration Framework (SIF).
Environment Validation Tools and Utilities	List of tools and utilities that you can use to verify the current environment to identify the area of improvement.

Acronyms

The following table lists the different acronyms used in this guide:

Term	Definition
BO	Base Object
EJB	Enterprise Java Beans
GC	Garbage Collection
HM	Hierarchy Manager
JVM	Java Virtual Machine
MDM	Informatica Master Data Management
PAM	Product Availability Matrix
PGA	Program Global Area (Oracle database)
RAC	Real Application Cluster
RAID	Redundant Array of Independent Disks
RMAN	Recovery Manager
SAN	Storage Area Network

Term	Definition
SGA	System Global Area
SIF	Services Integration Framework
SOAP	Simple Access Object Protocol
SSD	Solid State Drive

CHAPTER 2

Recommendations

This chapter includes the following topics:

Overview

The recommendations are based on the regular volumes of data with standard hardware. For larger systems, you can adjust the settings accordingly. You can configure parameters related to Java, Oracle database, Microsoft SQL server, IBM DB2, and the MDM Hub to optimize the performance of the MDM Hub.

Recommendations for Java

You can configure the parameters related to the JVM settings and database connection pool. You can change or fine-tune the Java parameters to improve the MDM Hub performance.

The following table lists the recommendations for the JVM settings:

Parameter	Recommended Setting	Description	Informatica Knowledge Base article number
JVM optimization mode	'-server'	By default, 64-bit Java runs in "server" JVM. Since Java SE 5.0, with the exception of 32-bit Windows, the server VM is automatically selected on server-class machines. The definition of a server-class machine might change from release to release. Therefore, check the appropriate ergonomics document for the definition for your release.	144943
Heap Size (64-bit)	`-Xms1024m` `-Xmx8192m`	In a 64-bit environment with heavy SIF usage, keep the max heap size equal to or greater than 4 GB. If the number of ORSs is more (irrespective of active or obsolete), allocate more heap memory. Informatica recommends to have the value of Xmx as 8 GB. If more memory is needed, add another JVM with a new clustered node.	120835
Code Cache	`-XX:ReservedCodeCacheSize=256m` (Oracle JVM) `-XX:codecachetotal=256m` (IBM JVM)	Maximum size limit for the code cache.	352050
Stack Size (64-bit)	`-Xss512k`	Use the platform default. Adjust if required. Stack Size is the memory used by Java for each thread it spans. This memory is outside the heap. If the memory is too low, the server might fail with 'StackOverFlow' error. If the memory is too high, the server might fail with 'Out of memory' error. If you need to fine-tune the value, analyze the heap dump to arrive at an optimal value. For a 32-bit processor, fine-tune the value to get additional memory for the heap to avoid an "Out of memory" error. **Note:** If you use Informatica Address Verification, then use `-Xss2048k`.	144477 and 207128

Parameter	Recommended Setting	Description	Informatica Knowledge Base article number
Garbage Collection Policy	Use the default policy, as chosen automatically by the JVM.	Complete analysis needs to be done to determine the garbage collection (GC) policy, if you decide to change it. Garbage collectio policies include Parallel GC and Concurrent Sweep. For example, while Parallel GC in heavy real time usage can give longer pauses, Concurrent Sweep does not do so. You can use a combination of the policies. See the Java Guide to set the exact parameters.	144944
Application Server Sizing	Optimal application server sizing.	Proper analysis is required to determine the application server sizing. Different factors including average number of concurrent users, peak number of concurrent users, current data volume, rate of data growth, future data volume, Hierarchy Manager configuration, Hierarchy Manager relationship cardinality, and number of Data Director applications must be considered for sizing. For assistance on server sizing, contact the Informatica Professional Services team.	-

The following table lists the recommendations for the database connection pool:

Parameter	Recommended Setting	Description	Informatica Knowledge Base article or documentation
`CMX_SYSTEM` Database Connection Pool	`Max Connection: (N+T) ×` `1.5`, where: - N is the number of concurrent IDD and SIF API users. - T is the number of concurrent threads in Batch. `Min Connection: 0`	On average, each thread takes 2.5 connections. Therefore, multiply the number of concurrent threads by 1.5.	-
ORS Database Connection Pool	`Max Connection: (N+T) ×` `2.5`, where: - N is the number of concurrent IDD and SIF API users. - T is the number of concurrent threads in Batch. `Min Connection: 0` Test Connection on Lease: Disable Statement Caching: 10	On average, each thread takes 2.5 connections. Therefore, multiply the number of concurrent threads by 2.5. Test Connection on Lease: Disable this property to avoid additional database cost incurred. Min Connection: In some instances, application servers might have some connection leaks with respect to rollback transaction where a connection caught in a rollback is not released to the pool. If such instances are found, you can set Min Connection to 0. Statement Caching (WebLogic): Initially, set 10 as the minimal number.	121471

General Recommendations for Database

You can configure the parameters related to the database environment, virtual image environment, and database tables and data.

The following table lists the recommendations related to the database environment:

Parameter	Recommended Setting	Description
Environment Sharing	Always use non-shared environment: The Production ORS must have exclusive use of the Oracle instance. The Production Oracle instance must have exclusive use of the host machine.	The MDM Hub Oracle database instances (both the MDM Hub Master Database and ORS) need not be shared with other MDM Hub installations and must not be shared with other applications. **Note:** Each additional level of sharing compromises the best performance possible on a particular hardware.
Connectivity: Application Server to Hub Store databases	The fastest connectivity possible.	Connection latency might have a major performance impact. Enable faster connectivity to the data store by using fiber optic connections.

Parameter	Recommended Setting	Description
Connectivity: Database server to Data File storage	The fastest connectivity possible.	Connection latency might have a major performance impact. Enable faster connectivity to the data by using fiber optic connections. Have a dedicated point to point connection to avoid network contention.
Database Server Sizing	Optimal database server sizing.	You need proper analysis to determine the database server sizing. Different factors including current data volume, rate of data growth, future data volume, SIF calls, and batch volume must be considered for sizing. For assistance on server sizing, contact the Informatica Professional Services team.
MDM Hub Master Database (cmx_system) host name in cluster environments	Absolute host name or IP address	To avoid caching issues in multi-node or cluster environments, use the absolute host name or IP address in place of the default `localhost`. The host name is configured during the MDM Hub installation. You can update the host name in the `DATABASE_HOST` column of the `C_REPOS_DATABASE` table.

The following table lists the recommendations for the virtual image environment:

Parameter	Recommended Setting	Description
Data File Storage	Use a physical drive instead of a virtualized drive.	Use a physical drive to store the data files instead of a virtualized drive within the image. Use a physical drive to avoid I/O contention because of a virtualized drive and the latency caused by the introduction of another layer with no actual benefit.
Hardware and Software Specification	As good as the equivalent physical or standalone instance.	Must be as good as the equivalent physical machine and must meet all of the PAM or sizing requirements.
CPU Cores	100% allocated to virtual image.	The physical machine, where the virtual image is hosted, must allocate 100% of the CPU cores to the virtual image. Sharing is not recommended.

The following table lists the recommendations related to the database tables and data:

Parameter	Recommended Setting	Description
Custom and Backup tables created with names starting with `C_REPOS%`	Do not name the custom tables or backup tables with names starting with `C_REPOS%`.	During HUB server restart and Met Migration, the performance of the Hub Console degrades if the schema has large volume tables with names that start with `C_REPOS%`. Ensure that backup tables are not created with names starting with `'C_REPOS%'`.
Fragmentation	Minimize likelihood of fragmentation.	Maintain the Oracle schema to ensure that fragmentation is kept to a minimum. Monitor and de-fragment whenever the degree of fragmentation has an impact on the MDM Hub performance.
High volume of data in `C_REPOS_TABLES` with historical data	Perform regular maintenance on the following METADATA tables: - `C_REPOS_AUDIT` - `C_REPOS_MQ_DATA_CHANGE` - `C_REPOS_JOB_CONTROL` - `C_REPOS_JOB_METRIC` - `C_REPOS_MET_VALID_RESULT` - `C_REPOS_MET_VALID_MSG` - `C_REPOS_TASK_ASSIGNMENT_HIST`	If the number of records in the metadata tables is too high, it might cause issues such as slow startup, out of memory errors, and performance issues. It is recommended to back up and truncate or reduce the data volume on the metadata tables. If you enable raw retention on any base object then you can purge `C_REPOS_JOB_*` tables for any date beyond the maximum raw retention period. For more information, search the Informatica Knowledge Base for article number 141201.

Recommendations for Oracle Database

You can configure the parameters related to the Oracle database environment, tablespace, Oracle table statistics, RAC recommendations, and Oracle flashback.

The following table lists the recommendations related to the Oracle database environment:

Parameter	Recommended Setting	Description
`C_REPOS_APPLIED_LOCK`	Enable caching for this table, `C_REPOS_APPLIED_LOCK`	The application uses this table frequently so you can cache this table to improve performance.
RMAN backups	Suppress RMAN backups from running during batch processing.	RMAN backups are good for a fast backup and restore. However, performance is decidedly lower when the RMAN backup is performed.
Archive Logging	Turn off archive logging during the initial data load. For steady-state operation, you can turn on archive logging after the initial data load ends.	Archive logging is unnecessary during the initial data load and adds an overhead. If there is a failure during the initial data load, Oracle rolls back the entire transaction (the current batch cycle of a batch job). The process can be re-run with no data loss.

During the initial data load, back up the ORS schema that loads at regular checkpoints with no jobs running. For example, back up the ORS schema after major long running jobs have completed. At an absolute minimum, take backups after completion of each phase: Stage, Load, Match, and Merge. You require backups to safeguard the work already done before you proceed.

You might enable archive logging for all steady-state operations (post initial data load).

If you use a Standby database (database mirroring), disable the standby before doing the initial data load. When the initial data load is complete, copy the database to the standby site, and then enable the standby. |

The following table lists the recommendations for the tablespace:

Parameter	Recommended Setting	Description
Tablespace	Use the following recommended settings for tablespace: - Locally Managed. - Uniform Extent. - Auto Segment Management. - Default Tablespace Block Size of 8 KB or 16 KB.	Applicable for all tablespaces involved in the MDM Hub including `CMX_DATA`, `CMX_INDX`, and `CMX_TEMP`. Default Tablespace Block Size of 8 KB is good for high API workload implementations. Use 16 KB as a balanced block size to support a mix of API and batch processing (small transactions and bulk read and write transactions). If you use a Default Tablespace Block Size of 16 KB and the database was created with default block size of 8 KB, then configure an appropriate `DB_CACHE` for it.
Tablespace Storage	Use the following Oracle recommendations for storage of tablespace data files: - If you use RAID: use either RAID 1+0 or RAID 0. - If you do not use a RAID controller: each tablespace must comprise multiple data files spread across different disks.	Many small disks perform better than a few large disks if everything else remains equal. Use the following steps to improve the performance: - RAID1+0 (RAID 10) has a high degree of fault tolerance with mirroring. Use as many disks as possible. Disks must be as fast as possible. - Avoid RAID 5 due to a writing overhead and poor performance if there is a disk failure. - If there is no RAID controller: use multiple single disks to split several data files over more disks. Do not dedicate data files for one tablespace to a single disk. Use the fastest disks possible and stripe the disks. In such cases, each tablespace uses part of each disk instead of a tablespace being dependent on a single disk. Similarly, it is better to keep redo and undo logs in a different physical disk.

The following table lists the recommendations for the Oracle table statistics:

Parameter	Recommended Setting	Description

Table Statistics	Analyze the ORS schema on a regular (frequent) basis.
	Use the following options:
	- Analyze the full schema (perform outside business hours).
	- Analyze individual tables whenever 10% of the data is changed.
	- Perform unplanned table analysis.
	- Set the `DBMS_STATS .SET_GLOBAL_PREFS` (see Environment Sharing in general database recommendations).
	- With no sharing:
	`DBMS_STATS.SET_GLOBAL_PREFS( 'DEGREE ', DBMS_STATS.AUTO_DEGREE);`
	- If forced to share:
	`DBMS_STATS.SET_GLOBAL_PREFS( 'DEGREE ', <(number of CPUs on DB Server) minus 1> );`
		Analyze the entire ORS schema on a regular basis as a best practice.
		Analyze individual tables whenever 10% of the data is changed by using data sampling. This is the Oracle recommendation. See Oracle database documentation for details.
		With sampling there is a trade-off. The best execution plans are always chosen when the statistics represent the entire table. When you use sampling, the execution plan is as appropriate to the tables as the data sample. For example, you can use 10% of the table as representative of the table as a whole. If you use a sample, be sure to use a large enough sample. A sample of 1 to 2% is too small unless the table is large. A sample of 10% is more representative on smaller tables. You can switch off sampling and perform a full analyze, which provides the most appropriate execution plans. However, a full analyze is achieved at the cost of far higher overhead in terms of the time taken when running the analyze. It is best to run outside of business hours to mitigate impact.

The following table lists the recommendations for RAC:

Parameter	Recommended Setting	Description
Environment	Each node must satisfy the same recommendations as set for the standalone environment.	-
`_PKQ` Sequence	Use `NOORDER`. Set sequence cache to 20000.	Informatica recommends to increase initial data load performance. These sequences are used to populate the `ROWID_OBJECT` in the base object tables. For more information, search the Informatica Knowledge Base for article number 115788.

The following table lists the recommendations for the Oracle flashback:

Parameter	Recommended Setting	Description
Flash recovery area	Fast file system.	Use a fast file system for your flash recovery area, preferably without operating system file caching.
Disk spindles	As needed.	Configure enough disk spindles for the file system to hold the flash recovery area.
Striped storage volume	Smaller stripe size.	If flash recovery area does not have non-volatile RAM, opt for striped storage volume with smaller stripe size such as 128k. This will allow each write to the flashback logs to be spread across multiple spindles, improving performance.

For more information, search the Informatica Knowledge Base for article number 333718.

INIT.ORA Recommendations for Oracle

The INIT.ORA recommendations are based on the standard hardware of 24 GB RAM with 8-core CPU Oracle server.

The following table lists the basic initialization parameters for the Oracle database:

INIT.ORA Parameter	Applicable to version	Value	Description
`cursor_sharing`		EXACT	Only allows statements with identical text to share the same cursor.
`db_block_checking`		FALSE	To avoid additional overhead.
`db_block_size`		8192	This parameter affects the maximum value of the `FREELISTS` storage parameter for tables and indexes. Oracle uses one database block for each freelist group.

INIT.ORA Parameter	Applicable to version	Value	Description
db_cache_size		2000M	Reduces additional overhead on dynamic allocation.
db_file_multiblock_read_count		0	To be auto-determined by Oracle database.
db_writer_processes (Single Core)		1	Oracle guideline.
db_writer_processes (Multi Core)		CPU / 8	Oracle guideline.
disk_async_io		TRUE	Oracle guideline.
filesystemio_options		SETALL	This bypasses file system buffer cache, especially on Linux.
java_pool_size		0	To be auto-determined by Oracle database.
large_pool_size		400M	If undefined, RMAN would use the SHARED POOL.
log_buffer		10M	A value of 10 MB is a reasonable initial size. Increase based on Oracle AWR report taken under typical heavy load. Section:**Cache Sizes** > **Log Buffer.**
memory_target		Two thirds of available memory.	Two thirds of the available physical memory for Oracle. One third is reserved for system and other processes.
memory_max_target		Two thirds of available memory.	Two thirds of the available physical memory for Oracle. One third is reserved for system and other processes.
open_cursors		1000	The MDM Hub uses parallel processing opening up multiple cursors.
parallel_adaptive_multi_user		TRUE	The MDM Hub uses multi-sessions for the same Oracle instance.
Processes		1000	Sufficient Oracle processes are allocated to support connection, parallel thread, internal process, and other usage. If set too small, some processes might fail to run.

INIT.ORA Parameter	Applicable to version	Value	Description
Recyclebin		OFF	Set the value of the `recyclebin` parameter to OFF. The MDM Hub has many temporary tables that the recycle bin spends time trying to maintain when dropped.
shared_pool_size		400M	Use 400 MB initially. Increase this according to the value seen in the Oracle AWR report for this instance taken under typical heavy load. Refer to section:**Cache Sizes** > **Shared Pool Size.**
streams_pool_size		0	The MDM Hub does not use Oracle streams functionality. Disable the functionality to prevent Oracle from reserving memory for it.
utl_file_dir		Do not set this	Use an Oracle Directory object instead of this parameter. For more information, see the Informatica Knowledge Base article 90456.
workareas_size_policy		AUTO	To be auto-determined by Oracle.
pga_aggregate_target	11g	0 in case of Automatic Memory Management (AMM) or One third of memory allocated to Oracle in case of Manual Memory Management (MMM).	Set PGA explicitly for MMM. For AMM, you need not set PGA.
sga_target	11g	Two thirds of Max_Memory in AMM or two thirds of allocated memory to Oracle in MMM.	For SGA, Informatica recommends to allocate two thirds of memory available for Oracle regardless of AMM or MMM. In AMM configuration, the setting of SGA explicitly always reserves two thirds of memory for SGA. This makes Buffer Cache ready for large data loaded into it during the batch. If you do not SGA, AMM might allocate memory for something else. The remaining memory might also not be enough for a big batch and it might run slower.

INIT.ORA Parameter	Applicable to version	Value	Description
optimizer_capture_sql_plan_baselines		False	Setting this parameter to false makes SQL plan management to not recalculate the execution plan for each repeatable SQL statement.
optimizer_index_caching		0	Setting this value to 0 defaults the behavior of cost-based optimization to favor nested loop joins and IN-list iterators.
optimizer_index_cost_adj		100%	To make use of default optimization based on table indexes, setting to default value evaluates index access paths at the regular cost. Default value is 100%.
optimizer_adaptive_features	12c	FALSE (Scope = BOTH).	Enables or disables all of the adaptive optimizer features, including adaptive plan.
optimizer_use_sql_plan_ baselines		TRUE	Enables or disables the use of SQL plan baselines.

For more information about the INIT.ORA parameters, see Informatica knowledge base article 90408.

Recommendations for Microsoft SQL Server

You can configure parameters related to the Microsoft SQL Server to optimize the performance of the MDM Hub.

The following table lists the recommendations related to the Microsoft SQL Server environment:

Database Parameters	Recommended Setting	Description
AUTO_UPDATE_STATISTICS_ASYNC	ON	To enforce statistics update in asynchronous mode.
PARAMETERIZATION	FORCED	To parameterize SQL statements. Parameterized statements reduce the frequency of query compilations and recompilations to improve performance.
READ_COMMITTED_SNAPSHOT	ON	To allow other sessions to see data that is not yet a committed transaction.

Recommendations for IBM DB2

You can configure the parameters related to the IBM DB2 environment, registry variables, and database file configuration parameters.

The following table lists the recommendations related to the IBM DB2 environment:

Parameter	Recommended Setting	Description
Physical Disk Drives (Tablespace)	Different physical drives for different tablespaces.	To reduce the amount of blocked input/output, you can increase input/output parallelism. Achieve input/output parallelism by storing user data tablespaces, temporary tablespaces, and transaction logs on different physical disk drives. Batch operations can access all the paths in parallel, which increases the throughput by reducing the input wait times and output wait times.
Physical Disk Drives (Container)	Different physical drives for different containers.	If more physical disk drives are available, you can increase input/output parallelism by extending parallelism to the container level. To increase input/output parallelism, place all containers for a tablespace on different physical disks. The IBM DB2 prefetchers and input/output cleaners access these containers in parallel without blocking each other, thereby increasing the throughput.
Processing Large Dataset		When you process a large data set, use the following command to rebind packages: `db2 bind @db2cli.lst blocking all grant public sqlerror continue CLIPKG 10` Perform the step every time you process a large data set. It is better to do so after the initial data load.
Reorganize Match Tables		Routinely determine and re-organize the match key tables (C_<Base Object>_STRP). To improve the performance of the SearchMatch API, you need to reorganize match key tables based on their primary key column, `SSA_KEY`. To determine if a match key table needs reorganization, perform a reorganization check and analyze the results. The cluster ratio of the primary key index appears in the `CLUSTERRATIO` column, F4, of the reorganization check result. The cluster ratio must be close to 100% for optimal performance. Determine when to reorganize the match key table by noting the cluster ratio at which you observe degradation in the SearchMatch API performance. Use the IBM `DB2 REORGCHK` and `REORG` commands to reorganize tables. Update the match key table statistics so that the IBM DB2 optimizer can use the table layout that the reorganization generates.

The following table lists the recommendations for the registry variables:

Database Parameters	Recommended Setting	Description
`DB2_INLIST_TO_NLJN`	NO	Configures the optimizer to prefer or not prefer nested loop joins. The DB2 SQL compiler might rewrite an IN list predicate as a join. The rewrite might provide better performance if you define an index on the joined columns. When the optimizer does not have accurate statistics, it might not be able determine the best join for the rewritten join. Set the variable to NO to prevent the optimizer from favoring nested loop joins in such cases.
`DB2_ANTIJOIN`	YES	Configures the optimizer to transform subqueries into anti-joins. The MDM Hub has queries that use NOT EXISTS subqueries. Set the registry variable to YES to look for possibilities to transform NOT EXISTS subqueries into anti-joins that IBM DB2 processes more efficiently.
`DB2_REDUCED_OPTIMIZATION`	`REDUCE_LOCKING`	Configures reduced optimization features or rigid use of optimization features at the specified optimization level. Set the registry variable to `REDUCE_LOCKING` to favor `NLJOIN` over `MSJOIN` whenever possible to reduce the amount of locking on the outer table.
`DB2_EXTENDED_OPTIMIZATION`	`ON`, `ENHANCED_MULTIPLE_DISTINCT`, `IXOR`, `SNHD`	Configures whether or not the query optimizer uses optimization extensions to improve query performance. The `ON`, `ENHANCED_MULTIPLE_DISTINCT`, and `SNHD` values specify different optimization extensions.
`DB2_HAS_JOIN`	YES	Configures hash join as a possible join method when compiling an access plan. Tune hash join to get the best performance. For best performance of a hash join, avoid hash loops and overflows to disk. To tune hash join performance, perform the following tasks: - Estimate the maximum amount of memory available for the sheapthres configuration parameter. - Tune the sortheap configuration parameter.

The following table lists the recommendations for the database file configuration parameters:

Database Parameters	Recommended Setting	Description
LOCKLIST	AUTOMATIC	Allocates the amount of storage to the lock list of a database. Multiple MDM Hub processes use "locks at the row-level" to complete tasks and to support concurrency. The number of locks that IBM DB2 needs to acquire depends on the number of rows to process. If the incoming volume differs greatly in size, set the parameter to AUTOMATIC to allow the database manager to determine the appropriate value. If you conservatively tune the LOCKLIST parameter value, lock escalations can occur, and some of the MDM Hub operations can fail due to lock timeouts.
MAXLOCKS	AUTOMATIC	Configures the percentage of the lock list that one application can use. Most MDM Hub processes run under the scope of a single application. Such single applications can acquire many row-level locks, consuming most of the available lock list. If the incoming volume differs greatly in size, predicting the MAXLOCKS parameter value is difficult. Set the parameter to AUTOMATIC to allow the database manger to determine the appropriate value. If you conservatively tune the MAXLOCKS parameter value, lock escalations can occur, and some of the MDM Hub operations can fail due to lock timeouts.
CATALOGCACHE_SZ	25000 or higher	Configures the maximum memory that the catalog cache can use from the shared memory of the database. IBM DB2 stores system catalog information in the catalog cache. The MDM Hub comprises many dynamic SQL queries that reference multiple metadata objects. If the catalog cache is large, IBM DB2 can retain information for some of the metadata objects from the system catalogs in the memory. If subsequent dynamic SQL queries require the same metadata objects, the compilation is quick. The MDM Hub comprises many frequently accessed metadata objects. Therefore, you must set the CATALOGCACHE_SZ parameter value to 25000 or higher.
LOGBUFSZ	4096 or higher	Configures the amount of the database heap to use as a buffer for log records before writing the records to disk. The MDM Hub creates logs for most of its operations in the IBM DB2 transaction logs. IBM DB2 buffers the log records in the log buffer before writing them to the disk. If the log buffer is large, IBM DB2 writes the log records to the disk less frequently. This makes disk input/output for log records more efficient. The default value for the database configuration parameter is not sufficient for an average MDM Hub environment. Set the parameter to 4096 pages or higher.

Database Parameters	Recommended Setting	Description
LOGFILSIZ	128000 or higher	Configures the number of log records written to the log files. A single MDM Hub transaction can contain many DML queries resulting in many log records that might span across many log files. A large log file size avoids the need to create new log files frequently. If IBM DB2 creates new log files frequently, it adversely influences the performance of input/ output bound systems. The total log space for a database is equal to the total number of primary and secondary log files multiplied by the log file size. The database must have adequate log space to ensure that the MDM Hub transactions do not run out of log space and fail. If the MDM Hub transactions fail, the database needs more time to roll back the transactions. Set LOGFILSIZ to 128000 or higher to ensure that the MDM Hub transactions do not fail and need a roll back. Also, you must consider the number of primary and secondary logs for a complete equation of log space.
LOGPRIMARY	100	Configures the number of primary log files to be pre-allocated. IBM DB2 creates primary log files when you activate the database. If the uncommitted transaction exhausts the primary log space, IBM DB2 creates secondary log files as needed. Set the LOGPRIMARY parameter value to 100 to efficiently handle the MDM Hub processes. Secondary log files act as a backup in cases where long transactions can exhaust the entire primary log space.
LOGSECOND	100	Configures the number of secondary log files that IBM DB2 can create and use for recovery log files. Log file creation can adversely impact the performance based on the size of the log file. If you allocate sufficient primary log space, transaction performance increases because the database does not create secondary log files frequently. Set the LOGSECOND parameter value to 100 to cover unexpected long transactions due to large incoming volumes, especially during batch processes. The sum of the values of LOGPRIMARY and LOGSECOND must be 200.
PCKCACHESZ	128000	Configures the package cache size which is allocated out of the database shared memory. The MDM Hub has many dynamic SQL statements. Each dynamic SQL statement has a compiled package associated with it. IBM DB2 caches these packages in the package cache memory. You must configure an appropriate package cache size to avoid package cache overflows, which adversely influence performance. Experiment with the values for the package cache size. Initially, set the parameter value to 50000 and monitor the different phases of the MDM Hub processes. If you observe frequent package overflows, tune the parameter again.
STMHEAP	AUTOMATIC	Configures the limit of the statement heap, which is used during the compilation of an SQL statement. If the statement heap is not sufficient, it might prevent the optimizer from evaluating all possible access plans for an SQL query. This might result in a suboptimal plan and adversely influence performance. Set the STMTHEAP parameter to automatic to allow the optimizer to weigh all possible access plans for the compilation an SQL query.

Database Parameters	Recommended Setting	Description
SORTHEAP	AUTOMATIC	Configures the sort heap size. The MDM Hub processes perform many sorts. If the sort heap size is not sufficient, large sorts can spill from the memory to disk. Disk input/output is slower compared to memory, and such sort spills can cause queries to run longer. Sort spills to disk can adversely influence performance and is more evident with larger spills. Set the parameter value to AUTOMATIC for the memory tuner to dynamically size the memory area as the sort requirements change.
SHEAPTHRES_SHR	AUTOMATIC	Configures the limit on the total amount of database shared memory that the sort memory consumers can use at a time. Set the SHEAPTHRES_SHR parameter to AUTOMATIC if you set the SORTHEAP parameter to AUTOMATIC.
UTIL_HEAP_SZ	50000 or higher	Configures the maximum amount of memory that the BACKUP, RESTORE, and LOAD utilities can use simultaneously. During some batch operations, the MDM Hub uses the IBM DB2 LOAD utility to move data between tables. The LOAD utility uses the utility heap to complete the data movement process. The size of the utility heap has an impact on the performance of the LOAD operation. Set the UTIL_HEAP_SZ parameter to an appropriate value to provide better throughput for the MDM Hub processes.

Recommendations for the MDM Hub

You might be able to improve performance by changing how the MDM Hub operates within the environment.

MDM Hub Environment

The following table lists the recommendations related to the MDM Hub environment:

Parameter	Recommended Setting	Description
Obsolete Data Director applications and Operational Reference Stores databases	Remove obsolete items.	Obsolete Data Director applications and Operational Reference Stores schemas impact the performance of server startups, run time memory, and Security Access Manager profile caching.
Order of authentication provider in Security Providers	Configure the security providers in order with the first provider being the provider that authenticates the heaviest user load.	**Configuration** > **Security Providers** > **Authentication Providers.** The MDM Hub authenticates the user based on the order of the security providers configured. If most of the users are authenticated by using the custom security provider (if applicable), it is recommended to move it to the first position. **Note:** Each authentication request has a cost of few milliseconds associated with it. The number of authentication requests is reduced significantly by using the User Profile Cache.

Application Server

The following table lists the recommendations for the application server configuration:

Parameter	Recommended Setting	Description
Maximum thread count for the thread pool	300 or higher	For example, in JBoss set the following property in the `standalone-full.xml` file: `<thread-pools> <thread-pool name="default"> <max-threads count="300"/> </thread-pool></thread-pools>`
Maximum connections in HTTP connection pool	300 or higher	For example, in JBoss set the following property in the `standalone-full.xml` file: `<connector name="http" protocol="HTTP/1.1" scheme="http" socket-binding="http" max-connections="300"/>`

JDBC logging level	OFF	For example, in JBoss, set the following log level property in the `standalone-full.xml` file:
		
		`<subsystem xmlns="urn:jboss:domain:logging:1.2">:`
		
		` <logger category="com.microsoft.sqlserver.jdbc"> <level name="OFF"/></logger>`

Transaction timeout	Greater than 3600 seconds.	Set the transaction timeout to at least 3600 seconds (1 hour). For example, in JBoss set the following property in the `standalone-full.xml` file: `<coordinator-environment default-timeout="3600"/>`

Operational Reference Store

The following table lists the recommendations for the ORS configuration:

Parameter	Recommended Setting	Description
Production Mode	Enable this property in Production.	**[Configuration** > **Database** > **Database Properties].** Enable this property to remove additional overhead of pre-scheduled daemons that refresh the metadata cache.
Batch API Inter-operability	Enable if both real time and batches are used.	**[Configuration** > **Database** > **Database Properties].** Enabling this configuration has an impact on performance. Enable the configuration if batches and real time API calls are used or if Data Director is used. If your application uses neither real time API updates nor Data Director, do not enable API Batch Interoperability. **Tip:** During Initial Data Load, disable this property for faster loading of data.

Parameter	Recommended Setting	Description
Auditing	Disable the auditing completely.	Auditing introduces additional overhead. You must disable auditing completely.
Write lock monitor Interval	`cmx.server.writelock.monitor.interval=10`	When more than one Hub Console uses the same ORS, a write lock on a Hub Server does not disable caching on the other Hub Servers. The unit is in seconds. For more information, see the *Multidomain MDM Configuration Guide*.

Schema Design

The following table lists the recommendations for the schema design:

Parameter	Recommended Setting	Description
Child Base Objects	Avoid too many child base objects for a particular parent base object.	The performance of load, tokenize, and automerge batch jobs decreases as the number of child base objects for a base object increases.
Match columns	Avoid too many match columns.	The performance of tokenize and match jobs decreases with the increase in the number of match columns.
Lookup Indicator	Enable Lookup Indicator only for 'Lookup' tables and not for any other base objects unrelated to lookup.	**Schema** > **[base object]** > **Advanced** > **Lookup indicator.** Enabling lookup indicator for non-lookup base objects unnecessarily caches the base object data in the memory. Doing so results in out of memory errors, slow Data Director performance, and slower rate of lookup cache refresh.
Lookup Display Name	Configure Lookup Display Name to be the same as the lookup column.	For high volume lookup tables: If you set the lookup display name to any column other than the column on which the relationship is built, SIF PUT calls must send the lookup display name values in the SIF call. When inserting data into the base object, the lookup value is validated by querying the lookup table. The order of lookup is predefined: the lookup display column value comes first followed by the actual column value second. In high volume lookup tables this becomes an overhead.
History	Enable History if you want to retain historical data for the specific base object. Otherwise, disable it.	If you enable History for a base object, the MDM Hub additionally maintains history tables for base objects and for cross-reference tables. The MDM Hub already maintains some system history tables to provide detailed change-tracking options, including merge and unmerge history. The system history tables are always maintained. Over a period of time, history in the database keeps growing. Consider keeping months or at most a few years of history in the system and to preserve database access performance.
History	To avoid very large history tables that cause performance issues, you can partition the tables.	For more information, search the Informatica Knowledge Base for article number 306525.

Parameter	Recommended Setting	Description
Cross Reference Promotion History	Enable Cross Reference Promotion History if you want to retain historical data for the specific base object.	**Schema** > **[base object]** > **Advanced** > **Enable History of Cross Reference Promotion.** Enabling history incurs performance cost both to real time and to batch operations. Use the history option cautiously and if required.
Trust	Configure trust only for required columns.	A higher number of trust columns and validation rules on a single base object incur higher overhead during the Load process and the Merge process. If the more trusted and validated columns are implemented on a particular base object: Longer SQL statements (in terms of lines of code) are generated to update the `_CTL` control table and the `_VCT` validation control table. Minimize the number of trust and validation columns to conserve good performance.
Case Insensitive Search	Enable Case Insensitive Search only for VARCHAR2 columns. Ensure that you do not include any column with a data type other than VARCHAR in the Search Query.	Enabling Case Insensitive Search for non-VARCHAR2 columns hinders performance.
Message Trigger Setup	Avoid configuring multiple message triggers for different event types.	Do an in-depth analysis before configuring message triggers. There is a performance cost associated with them during the execution of load jobs.
	Tune Message Trigger Query.	The best approach to tuning the query used in the Package Views is to use Explain Plan. Add custom indexes wherever required to avoid full table scans, and analyze tables/schema on a regular basis. When you use Explain Plan, retrieve the plan by wrapping the query around an outer query that contains a "where" clause for a `rowid_object` equal to. For more information about message triggers, search the Informatica Knowledge Base for article number 142115.
	Throughput can be greatly improved if you increase the Receive Batch Size and reduce the Message Check Interval.	The Message Queue Monitoring settings have a major impact related to the throughput message posting time. Configure these settings from the Hub Console in the Master Reference Manager (MRM) Master Database `(CMX_SYSTEM)` in the Configuration section.
	Avoid unnecessary column selection in message trigger.	Do not select "Trigger message if change on any column" if you do not need to monitor all the columns. Also, try to minimize the selection of columns.

Parameter	Recommended Setting	Description
'Read Database' Cleanse Function	Use the following recommended settings: - Use with caution. - Enable 'cache' if used.	The 'Read Database' cleanse function incurs a performance overhead compared to using a similar MDM Cleanse Function to perform the same function. The performance overhead is more pronounced on a high volume table. The overhead is caused by the creation of a new database connection and the corresponding transmit to, processing by, and receipt of the results from the database. These would otherwise be managed within the Process Server application layer. If use of this function cannot be avoided, if applicable, enable caching behavior of the Read Database function. Pass a Boolean 'false' value to the 'clear cache' input field of the Read Database function'. Doing so reduces performance lag by enabling future operations to use the cached value rather than creating a new database connection on each access of the function.
Cleanse Functions	Do not make it very complex.	The performance of batch jobs increases with reduced number and reduced complexity of cleanse functions.
Timeline (Versioning)	The Dynamic Timeline or Versioning must be enabled on an Entity base object (regular base object, not a Hierarchy Manager Relationship base object) if strictly required. Versioning has a performance impact on the base object associated with it. For Hierarchy Manager Relationship base objects, versioning is enabled with no option to disable.	Versioning must only be enabled on those Entity base objects (regular MDM Hub base object) which strictly need it to maintain the fastest performance possible. With the Versioning functionality, the additional associated metadata and processing carry a significant amount of complex processing when running any process on a version-enabled base object. Enabling versioning on a base object brings an additional performance cost to all processing performed on that base object. For more information, search the Informatica Knowledge Base for article numbers 138458 and 140206.
State Management	Disable state management if you do not require it.	State Management carries an associated performance overhead. If you use Data Director with workflows, you must enable State Management. However, enabling History for State Management Promotion at the cross-reference level is optional.
Delta Detection	Enable it only on the minimum number of columns that strictly need it.	Delta Detection carries a sizable associated overhead on performance. If a Landing Table has only new and updated records in every staging job, you can disable delta detection. If you want to enable Delta Detection the least impactful approach is to use the `last_update_date`. If you need additional columns, for each additional column you enable, analyze if the involvement of this additional column is worth the associated performance overhead. Avoid blindly enabling Delta Detection on all the columns.
Cleanse Mappings	Minimize the complexity of Mappings.	Minimized the complexity in the mapping to have better performance. If you use a Cleanse List in a Cleanse mapping, use static data in the Cleanse List. Consider using lookup tables only for dynamic data.

Parameter	Recommended Setting	Description
Validation Rules	Optimize Validation Rule SQL code.	The detection piece of each Validation Rule SQL runs against every record during the Load process to determine if it applies. Poorly performing Validation Rule SQL influences performance on every Loading record.
User Exits	Optimize user exit code for performance.	User exit code influences performance if not optimized. Applicable to both Data Director and batch user exits.
Packages	Optimize the SQL code written in each MDM Queries which is called from an MDM Hub Package.	These MDM Hub Packages are used in Data Director, SIF API calls, Data Manager, Merge Manager, and in search operations. If you do not tune the MDM Hub for performance, it results in an expensive operation whenever it is called.

Custom Indexes	Use caution when adding custom indexes. Each index added has an associated cost. Ensure that gain received outweighs the cost of each additional custom index.	Index management has a performance cost associated.
		
		Perform the following steps to improve the performance cost:
		1. Get a log of the real queries run on the base object (content data) and base object shadow tables (content metadata) on a typical day. Ignore temporary `T$%` tables and system `C_REPOS_%` tables;
		2. Identify indexes which exist on these tables to avoid unnecessary overlap.
		3. Before adding any indexes, review a regular day of logs and take an inventory of:
		  a. SIF API call duration.
		  b. Data Director process durations.
		  c. Batch jobs:
		    a. Duration of each batch job.
		    b. Duration of each cycle within that batch job.
		    c. Duration of longest running statements within a batch job.
		4. Consider the longest running process for potential benefit from a custom index.
		5. Consider adding indexes so the longest running SQL query or queries hit the new index in their execution plan. Avoid indexing fields which have many updates or inserts.
		After each new custom index added return to Step 3 and assess if there is still potential to improve performance through adding more custom indexes.

Parallel Degree on Base Object	Between one and number of CPU cores on database machine.	Parallel degree is an advanced base object property. For optimum performance of batch jobs, set a value between one and the number of processor cores on the database server machine. For more information, search the Informatica Knowledge Base for article number 181313.

Match and Merge

The following table lists the recommendations for the match and merge configuration:

Parameter	Recommended Setting	Description
Match Path Filter	Filter on root path instead of at the match rule level.	If you need to exclude records from the match process, filter on the root path instead of at the match rule level. When you filter at the root level, it excludes the records from tokenization and they, therefore, do not participate in the match.
Check for missing children	Use it with caution.	This match patch property indicates if parent records must be considered for matching based on the existence of child records. If you need a fuzzy match on a base object, tokenization of a parent base object record must occur. Tokenization of a parent base object record occurs if all child base objects that have the option to check for missing children disabled have a related child base object record. If a parent base object record has a child, where the option to check for missing children is disabled yet contains no record, the parent record is not tokenized. The MDM Hub performs an outer join between the parent and the child tables when the option to check for missing children is enabled. This option has an impact on the performance on each match path component on which the option is enabled. Therefore, when not needed, it is more efficient to disable this option.
Match Key	The tighter the key the better the performance.	The width of the match key determines the number of rows in the tokenization table (number of tokenized records which are used to match each record to be matched) and the number of records to be considered for each match candidates. Usually the standard key width is enough. **Search Level**: Use the narrowest possible search level to generate acceptable matches. Usually the typical search level is enough. **Match Rules**: For each match rule, add one or more exact match columns to act as a filter to improve the performance for each rule.
Dynamic Match Analysis Threshold (DMAT)	Change if required. Default is 0.	Although DMAT helps improve performance, take care when setting this limit. If you set the level too low it might cause under matching. It is recommended that clients first analyze the data to assess why a particular search range contains a large count. Sometimes the reason might be due to a noise-word or phrase like "do not send" or a valid-word or phrase such as "John." When setting this value, identify any large ranges that are causing bottlenecks and then use the "Comparison Max Range" count to set the DMAT. Proper analysis is required to change this value. See the Informatica Knowledge Base article 90740 on changing this value. For more information, search the Informatica Knowledge Base for article number 90740.

Parameter	Recommended Setting	Description
STRIP_CTAS_DELETE_RATIO	Change if required. Default is 10%.	**C_REPOS_TABLE [base object] > STRIP_CTAS_DELETE_RATIO.** Proper analysis is required if you decide to change the default 10% value. If the volume of data change in the _STRP table is more than this percentage, tokenization would instead use 'Create Table … As Select …' code to recreate the _STRP table with the needed changes rather than delete and/or insert operations to arrive at the same result in less time. The optimal value might vary for each implementation and it depends on the size of the table and the percentage of records that must be updated.
COMPLETE_STRIP_RATIO	Change if required. Default is 60%.	**[Model > Schema > [base object] Advanced > Complete Tokenize Ratio].** Proper analysis is required if you decide to change the default 60% value. If the volume of data change in the _STRP table is more than this percentage, the tokenization process would drop and re-create the entire _STRP table rather than (re)tokenizing only the updated records.
AUTOMERGE_CTAS_RATIO	Change if required. Default is -1.	**C_REPOS_TABLE [base object] > AUTOMERGE_CTAS_RATIO.** Proper analysis is required before you decide to change the default -1 value. If the volume of records queued for merge is greater than this percentage, automerge uses the 'Create Table As Select' option for faster merging instead of the regular delete/insert operations. **Note:** Default value is -1 which indicates that this Create Table as Select (CTAS) feature is OFF for Automerge.

For more information about tuning match and merge, search the Informatica Knowledge Base for article number 357214.

Services Integration Framework (SIF) APIs

The following table lists the recommendations for the SIF API:

Parameter	Recommended Setting	Description
Protocol	Use EJB protocol over HTTP or SOAP.	EJB Protocol is faster and reliable. For more information, search the Informatica Knowledge Base for article number 138526.
Disable Paging	Set to True if not required.	SIF Request parameter: disablePaging If paging is not required (if results are not going to return many records), it is better to set this flag to true. If set to false (default value), this would incur two database calls for each SIF call.
Return Total	Do not set any value.	SIF request parameter: returnTotal If you do not require total count, it is better to not set this flag. If set to true, it would incur two database calls for each SIF call.

Hub Server Properties

The following table lists the recommendations for the Hub Server properties:

Parameter	Recommended Setting	Description
User Profile Cache	True Default is true.	`cmx.server.provider.userprofile.cacheable` This property is found in the `cmxserver.properties` file. When you set this flag to true, once a user profile is authenticated, it is cached. Set the flag to true to suppress the need for explicit user authentication requests for every SIF call.
User Profile Life Span	60000 Default is 60000.	`cmx.server.provider.userprofile.lifespan` This property is found in the `cmxserver.properties` file. Time to retain the cached user (milliseconds) before refreshing. A few minutes is adequate-avoid setting this to longer durations.
Security Access Manager (SAM) cache refresh interval	5 clock ticks Default is 5 clock ticks at a rate of 60,000 milliseconds for 1 clock tick, which is equivalent to 5 minutes.	`cmx.server.sam.cache.resources.refresh_interval` This property is found in the `cmxserver.properties` file. Refreshes the SAM cache after the specified clock ticks. To specify the number of milliseconds for 1 clock tick, use the `cmx.server.clock.tick_interval` property. For more information, see the *Multidomain MDM Configuration Guide*.
Cleanse Poller	30 (seconds) Default is 30.	`cmx.server.poller.monitor.interval` This property is found in the `cmxserver.properties` file. For every number of seconds configured, the MDM Hub Server would poll the availability of the Process Server and accordingly flag the status of the Process Server as valid or invalid. For more information, search the Informatica Knowledge Base for article number 151925.

Infinispan

The following table lists the recommendations for Infinispan parameters, which are located in the `inifinspanConfig.xml` file:

Parameter	Recommended Setting	Description
`expiration lifespan`	86400000 (milliseconds)	Maximum lifespan of a cache entry in milliseconds. When a cache entry exceeds its lifespan, the entry expires within the cluster. You can increase the lifespan for the following caches: DISABLE_WHEN_LOCK, DATA_OBJECTS, and REPOS_OBJECTS. For example, you can increase a lifespan from one hour (3600000) to one day (86400000). Each cache has its own default value for this parameter. To find the default values, open the `inifinspanConfig.xml` file.
`expiration interval`	300000 (milliseconds)	Maximum interval for checking the lifespan. For example, you can increase an interval from five seconds (5000) to five minutes (300000).

For more information about Infinispan parameters, search the Informatica Knowledge Base for article number 509572.

Logging

The following table lists the recommendations for the logging:

Parameter	Recommended Setting	Description
Hub Server Logging	Set to ERROR mode.	Change the `log4j.xml` file to use ERROR mode. If clustered, update the `log4j.xml` file in all nodes. For JBoss, use `<JBoss node>/conf/jboss-log4j.xml`. For other application servers, update `<INFAHOME>/hub/server/conf/log4j.xml`. Once the log4j configuration file is updated, changes are reflected in the log within a few minutes.
Process Server Logging	Set to ERROR mode.	Change the `log4j.xml` file to use ERROR mode. If clustered, update the `log4j.xml` file in all nodes. For JBoss, use `<node>/conf/jboss-log4j.xml`. For other application servers, update the `<INFAHOME>/hub/cleanse/conf/log4j.xml` file. Once the log4j configuration file is updated, changes are reflected in the log within a few minutes.

For more information about logging, search the Informatica Knowledge Base for article number 120879.

Smart Search

The following table lists the recommendations for smart search:

Parameter	Recommended Setting	Description
Solr index directory	Use separate disk for Solr index.	Keeping the Solr index in a separate disk will improve the performance. It is preferable to keep Solr index files on SSDs instead of traditional spinning hard disks for better performance of the smart search requests.
ZooKeeper directory	Use separate disk for ZooKeeper directory.	Keeping ZooKeeper directory in a separate disk improves the performance. It is preferable to keep ZooKeeper directory on SSDs instead of traditional spinning hard disks for better performance of the smart search requests.
Limit Searchable Fields	Do as needed.	Do not index unnecessary searchable fields. Multiple searchable fields increase the indexing and searching time, so configure only the required fields as searchable fields. Also keep only the required fields and facets. Facets should only be on the fields with low entropy. Also limit the number of fuzzy fields.

Task Assignment

The following table lists the recommendations for task assignments:

Parameter	Recommended Setting	Description
`task.creation.batch.size`	Default is 1000.	In MDM 10.0 and earlier, the default value is 50. Available in `cmxserver.properties`. Sets the maximum number of records to process for each match table. If more tasks need to be assigned on the run, you can increase this value.

Operational Reference Store and SIF APIs

The following table lists the recommendations for ORS-specific SIF API generation:

Parameter	Recommended Setting	Description
Required objects	As needed.	ORS specific API generation depends on the number of objects selected. It is preferable to add only the required objects to gain performance during the SIF API generation.
SIF API (Java Doc Generation) Heap Size	Default is 256m.	`sif.jvm.heap.size` Available in `cmxserver.properties`. Sets the heap size used during the creation of Java Doc. As Java Doc creation takes a lot of heap memory, you can increase this to a higher value if required. Note that this heap size setting is not connected to the heap size of the MDM application, which is set during the server startup.

Informatica Data Quality

The following table lists the recommendations for Informatica Data Quality cleansing:

Parameter	Recommended Setting	Description
Batch Size	Default is 50.	`cmx.server.cleanse.number_of_recs_batch` Available in `cmxcleanse.properties`. If the workflow supports minibatch, then you can set this value to any desirable value depending on the number of records to be cleansed at a time. If this attribute is set, then MDM automatically groups the records for cleansing. **Note:** This property can be used in other cleansing engines if they support minibatch.

For more information, search the Informatica Knowledge Base for article number 153419.

Initial Data Load

The following table lists the recommendations for the initial data load:

Parameter	Recommended Setting	Description
Database/ Environment settings		Distributed Matching: Set the `cmx.server.match.distributed_match=1` parameter in the `cmxcleanse.properties` file. This often improves performance by spreading the match load across multiple servers. Ensure that your Cleanse Match servers are configured so that match processing is performed in the Batch mode on the servers you expect to spread the match across.
		UNDO Tablespace: The Match process utilizes a lot of UNDO tablespace. Ensure that sufficient UNDO tablespace is available in the database. Adjust your batch match size accordingly.
		Database Archive Log: Switch off or Disable to improve the performance during IDL.
		Batch API Interoperability: Disable during IDL.
		Application server performance: JVM, thread counts, block size. Refer to the corresponding sections in the guide.
		Performance parameter: Set the `siperian.performance` parameter in the `LOG4J.XML` file to OFF.
		Match Rules Settings: Performance of the match job majorly depends on the match rule configuration. Use the default settings of match configuration. Change it only if it is required and extensively tested. Make the search level as exhaustive and key width as extended.
		Connection pool size: Ensure that there is sufficient connection pool available for the sizing. Refer to connection pool recommendations in this guide.
		Indexes: Disable any custom indexes
		Constraints: Choose 'Allow constraints to be disabled' option to disable the constraints on NI indexes on base object and all indexes on XREF
		History: Disable history
		Analyze Schema: Analyze database schema prior to IDL
		Batch API Interoperability: Disable this flag.
		Production mode: Enable production mode flag.

For more information, search the Informatica Knowledge Base for article numbers 158622 and 158822.

Recommendations for Batch Job Optimization

A batch job is a program in the MDM Hub that you can run to complete a discrete unit of work. You can launch batch jobs individually or as a group from the Hub Console or with the SIF APIs. You can configure settings to optimize the performance of batch jobs.

The following table lists the different batch job parameters and their recommended settings to achieve a base-level performance:

Parameter	Recommended Setting	Description
Cleanse Thread Count Used in the following batch jobs: - Match Job - Generate Match Tokens process on Load job - Stage job	Start with the number of cores available. Based on CPU utilization, number of threads can be increased. Default is 1.	Available in **"Process Server > Threads for Cleanse Processing"**. Total number of threads used by the Master or Slave Process Server when executing. Generate Match Tokens after Load, Match, and Stage jobs.
Threads for Batch Processing Used in the following batch jobs: - Automerge Job - Load Job - Batch Delete - Batch Unmerge - Batch Revalidate	Specify a value that is equivalent to four times the number of CPU cores on the system on which the Process Server is deployed. Default is 20.	Available in **"Process Server > Threads for Batch Processing"**. Maximum number of threads to use for a batch process. For example, if the host machine has 16 CPU cores, set the Threads for Batch Processing in the Process Server registration to 64. Applicable only if the Process Server is marked for batch processing. **Note:** From the total number of threads available on the Process Server, dedicate n threads for Batch jobs by setting a value for the property number of threads for Batch processing.
Controller Thread Time Out Used in the following batch jobs: - Automerge Job - Load Job - Batch Delete - Batch Unmerge - Batch Recalculate	300000 (5 minutes). Default is 300000.	`com.informatica.mdm.loadbalance.ControllerThread.timeout` This property is found in the `cmxcleanse.properties` file. When distributing the load to different slave Process Servers, after the last block is sent to a slave Process Server, all slave Process Servers which are processing the blocks MUST complete the job within the timeout period. **Note:** If not completed, such blocks are marked with 'No Action' in the batch result. Note that, the batch is not marked as failed because the remaining blocks are successfully loaded.

Parameter	Recommended Setting	Description
Load analyze threshold rate Used in the following batch jobs: - Automerge Job - Load Job - Batch Delete - Batch Unmerge - Batch Recalculate	Default is 10.	`cmx.server.batch.load.analyze_threshold_rate` Available in `cmxserver.properties` **For ORACLE only. Available from MDM 10.0 HotFix 1.** Specifies the frequency that the MDM Hub gathers analytical statistics for tables affected by a batch Load job. Set to 0 to disable statistic collection. Set to 1 to collect statistics only at the end of a Load job for base object and cross-reference tables. For example, if the threshold is 10, then statistics would be gathered at every 10^n records. For example, new statistics would be gathered whenever the insert record count reaches 100, 1000, 10000, and so on.
Recycler Thread Max Idling Used in the following batch jobs: - Automerge Job - Load Job - Batch Delete - Batch Unmerge - Batch Recalculate	300000 (5 minutes). Default is 300000 (5 minutes).	`com. informatica.mdm.batchserver.RecyclerThread.max_idling` This property is found in the `cmxcleanse.properties` file. If a slave Process Server is processing a block of batch job and is idle for a duration specified in this attribute then the specific thread is marked as 'dead.' **Note:** If a slave Process Server is timed out as noted earlier, the corresponding block is marked with 'No Action' in the batch result. Note that the batch is not marked as failed as the remaining blocks are successfully loaded.
Automerge: Automerge Threads Per Job	Default is 1.	`cmx.server.automerge.threads_per_job` This property is found in the `cmxserver.properties` file. Maximum number of threads distributed across different Process Servers to process the automerge job. For example, if this value is 20, automerge would be distributed across two Process Servers each with 10. The distribution depends on factors such as CPU weightage of the Process Server and other jobs running on the Process Server. This value must be less than the value in 'Threads for Batch' attribute specified for the Process Server. **Note:** The optimum value for a database server with a 16 core processor and a solid-state drive (SSD) set up in a RAID is 20. Based on CPU utilization on different Process Servers, you can increase the threads.
Automerge: Automerge Block Size	Default is 250.	`cmx.server.automerge.block_size` This property is found in the `cmxserver.properties` file. Maximum number of records to be sent for merges to each Process Server in one block. For example, consider the scenario of two Process Servers with 1000 records to be merged. If this value is 250, each Process Server gets 250 records first followed by another 250 records next. Increasing this value can provide performance improvement based on how powerful the application servers and database servers are.

Parameter	Recommended Setting	Description
Automerge: STRP table management	Default is 10.	`cmx.server.batch.strp_upd_threads_per_job` This property is found in the `cmxcleanse.properties` file. Applicable for IBM DB2. For larger data set of data participating in Automerge, you can make faster updates to the STRP table by increasing this thread count based on the CPU of your Process Server.
Load: Batch Threads Per Job	Default is 1.	`cmx.server.batch.threads_per_job` This property is found in the `cmxserver.properties` file. Maximum number of threads distributed across different Process Servers to process the load job. For example, if this value is 20 then load process would be distributed across two Process Servers each with 10. The distribution depends on factors such as CPU weightage of the Process Server and other jobs running on the Process Server. This value must be less than the value in 'Threads for Batch' attribute specified for the Process Server. **Note:** The optimum value for a database server with a 16 core processor and a solid-state drive (SSD) set up in a redundant array of independent disks (RAID) is 20. Based on CPU utilization on different Process Servers, you can increase the threads.
Load: Batch Block Size	Default is 250.	`cmx.server.batch.load.block_size` This property is found in the `cmxserver.properties` file. Maximum number of records to be sent for load, to each Process Server in one block. For example, consider the scenario of two Process Servers with 1000 records to be loaded. If this value is 250, each Process Server gets 250 records first followed by another 250 records next. Increasing this value can provide performance improvement based on how powerful the application servers and database servers are.
Load: Threads per job for generate tokens, if 'Generate Match Tokens on Load' attribute is enabled on the base object	Same as "Threads for cleanse processing".	See 'Threads for Cleanse Processing' attribute described earlier. Note that, this thread attribute is different from the core threads per job attribute of the load job described earlier. If 'Generate Match Tokens on Load' is not selected, this attribute does not have any impact on the performance of the Load job.
Batch Recalculate (SIF API Request): Recalculate Threads Per Job	Same property, re-used from LOAD Job. See LOAD Job section for more details.	`cmx.server.batch.threads_per_job` This property is found in the `cmxserver.properties` file. Same property, re-used from LOAD Job. See LOAD Job section for more details.

| Parameter	Recommended Setting	Description
Batch Recalculate (SIF API Request): Recalculate Block Size	Default is 250.	`cmx.server.batch.recalculate.block_size` This property is found in the `cmxserver.properties` file. Maximum number of records to be sent, to recalculate BVT, to each Process Server in one block. For example, consider the scenario of two Process Servers with 1000 records to be recalculated. If this value is 250, each Process Server gets 250 records first followed by another 250 records next. Increasing this value can provide performance improvement based on how powerful the application servers and database servers are.
Batch Recalculate (SIF API Request): Threads Per Job	Same property, re-used from LOAD Job. Refer to LOAD Job section for more details.	`cmx.server.batch.threads_per_job` Available in `cmxserver.properties` Same property, re-used from LOAD Job. Refer to LOAD Job section for more details.
Batch Unmerge (SIF API Request): Unmerge Block Size	Default is 250.	`cmx.server.batch.batchunmerge.block_size` This property is found in the `cmxserver.properties` file. Maximum number of records to be sent for unmerges, to each Process Server in one block. For example, consider the scenario of two Process Servers with 1000 records to be unmerged. If this value is 250, each Process Server gets 250 records first followed by another 250 records next. Increasing this value can provide performance improvement based on how powerful the application servers and database servers are.
Batch Delete (SIF API Request): Threads per job	Same property, re-used from LOAD Job. See LOAD Job section for more details.	`cmx.server.batch.threads_per_job` This property is found in the `cmxserver.properties` file. Same property, re-used from LOAD Job. See LOAD Job section for more details.
Batch Delete (SIF API Request): Delete Batch Block Size	Default is 250.	`cmx.server.batch.delete.block_size` This property is found in the `cmxserver.properties` file. Maximum number of records to be sent for deletion, to each Process Server in one block. For example, consider the scenario of two Process Servers with 1000 records to be deleted. If this value is 250, each Process Server first gets 250 records and then another 250 records. Increasing this value can provide performance improvement. This performance improvement depends on how powerful the application servers and database servers are.

Parameter	Recommended Setting	Description
Tokenize: Tokenization File Loader Option	Default is true.	`cmx.server.tokenize.file_load` This property is found in the `cmxcleanse.properties` file. Applicable for Oracle and DB2. If true, DB2 file loader or Oracle SQL Loader is used to load the records during the tokenization job. If file writing is causing performance issue, this can be changed to false, thereby, data is directly written to the database every time instead of file loader option. Generally, file loader is faster than the direct database write. You might choose the option according to your environment.
Tokenize: STRP table management	Default is 10.	`cmx.server.batch.strp_ins_threads_per_job` Available in `cmxcleanse.properties`. Currently applicable only for DB2. For larger dataset, Insertion to the STRP table can be made faster by increasing this thread count based on the CPU of your Process Server.
Tokenize: STRP table management	Default is 10.	`cmx.server.batch.strp_del_threads_per_job` Available in `cmxcleanse.properties`. Currently, applicable only for DB2. For larger dataset, deletion from the STRP table can be made faster by increasing this thread count based on your Process Server's CPU.
Tokenize: STRP table management	Default is 2500.	`cmx.server.batch.sql.block_size` This property is found in the `cmxcleanse.properties` file. Currently applicable only for DB2. For larger data set, you can make faster inserts or deletes on the STRP table by increasing this block size. Each thread would insert or delete as many number of records as specified in this block size. For example, for an insert of 50 k records, with 10 threads and a block size of 2.5 k, each thread might run a SQL batch of 2.5 k records twice (roughly).
Stage: Threads per job	See 'Cleanse Thread Count' attribute described earlier.	See 'Cleanse Thread Count' attribute described earlier.
Stage: Cleanse Minimum Distribution	Default is 1000.	`cmx.server.cleanse.min_size_for_distribution` This property is found in the `cmxcleanse.properties` file. The MDM Hub distributes the cleanse job across different cleanse server only if the number of records is higher than this minimum size. When distributing the load, each slave Process Server would use the Cleanse Thread Count for the number of worker threads.

Parameter	Recommended Setting	Description
Stage: Stage JDBC Loader	Default is false. Usually, file writing must be faster than the direct database writing.	`cmx.server.java_jdbc_loader` Applicable for Oracle and DB2. Default is false. This property is found in the `cmxcleanse.properties` file. If true, DB2 and Oracle use direct database connections during the stage job instead of DB2 file loader or Oracle SQL loader options If file writing is causing performance issue, this can be changed to true. On doing so, data gets directly written to the database every time instead of file loader option. Note that, generally, file loader is faster than the direct database write. You might choose the option according to your environment.
Match: Threads per job	See 'Cleanse Thread Count' attribute described earlier.	See 'Cleanse Thread Count' attribute described earlier.
Match: Match Distribution Flag	Enable this flag to 1, if the MDM Hub has to distribute the match job load across different cleanse servers.	`cmx.server.match.distributed_match` This property is found in the `cmxcleanse.properties` file. The MDM Hub distributes the match job across different cleanse server only if this value is set to 1. When distributing the load, each slave Process Server would use the Cleanse Thread Count for the number of worker threads.
Match: Match File Loader Option	Default is true. Usually, file writing must be faster than the direct database writing.	`cmx.server.match.file_load` Applicable for Oracle and DB2. Default is true. This property is found in the `cmxcleanse.properties` file. If true, DB2 file loader or Oracle SQL Loader is used to load the records during the tokenization job. If file writing is causing performance issue, this can be changed to false, thereby, data will be directly written to the database every time instead of file loader option. Generally, file loader is faster than the direct database write. You might choose the option according to your environment.
Match: Match Loader Batch Size	Default is 250.	`cmx.server.match.loader_batch_size` This property is found in the `cmxcleanse.properties` file. Applicable if JDBC load is used in match processing instead of file loader option. Maximum number of records to be sent for match in each worker thread. Increasing this value can provide performance improvement based on how powerful the application servers and database servers are.
Match: Match Elapsed Time	Default is 20 (minutes).	**Hub Console** > **Base Object** > **Max Elapsed Match Minutes.** The execution timeout in minutes when executing a match rule. If this time is reached, the match process will exit. This must be increased only if the match rule and the data are very complex. Generally rules must be able to complete within 20 minutes.

Parameter	Recommended Setting	Description
Match: Match Batch Size	Default is 20000000.	**Hub Console** > **Base Object** > **Match/Merge Setup** > **Number of rows per match job batch cycle.** Maximum number of records to be processed by the MDM Hub for matching. This number would affect the duration of match process. Also, lower the match batch size, you have to run the match process more times. **Note:** When running large Match jobs with large match batch sizes, if there is a failure of the application server or the database, you must re-run the entire batch.
Match: Maximum records per ranger node	Default is 5000.	`max_records_per_ranger_node` This property is found in the `cmxcleanse.properties` file. Number of records per match ranger node (limits memory use). Ranger is an internal component used within the match process where sorting and merging operations are performed based on this maximum records attribute. You can optimize this value to get better performance based on the memory available in your application server.
Initially Index Smart Search Data: Block Size	10000. Default is 250.	`cmx.server.batch.smartsearch.initial.block_size` Available in `cmxserver.properties`. Maximum number of records that the "Initially Index Smart Search Data" batch job can process in each block. This property is not applicable through regular indexing outside this specific batch job. When you index a large data set, you can set the value to 10000. **Note:** This property is available only from MDM 10.0 Hot Fix 2.
Initially Index Smart Search Data: Smart search threads	Default is 1. Same property, re-used from LOAD Job.	`cmx.server.batch.threads_per_job` Available in `cmxserver.properties`. Maximum number of threads distributed across different Process Servers to process the batch job "Initially Index Smart Search Data". You can increase this value to achieve more performance during this batch job. This property is not applicable for regular indexing outside this specific batch job.

Multithreaded Batch Job – Process Flow

The following image shows the process flow of Automerge, Load, Batch Delete, Batch Unmerge, and Batch Recalculate jobs:
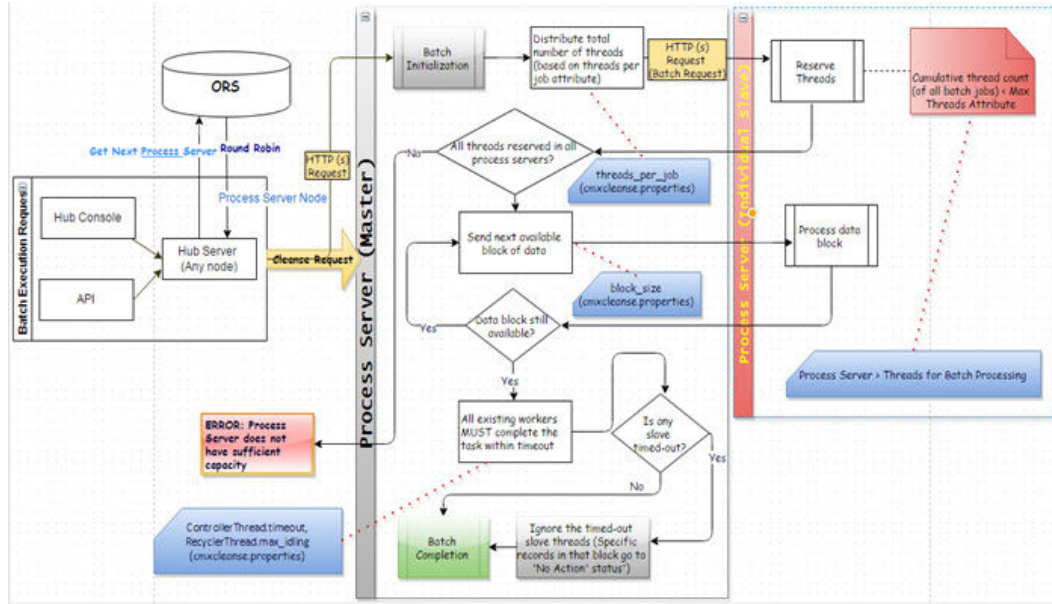
Master Process Server

The following list describes the properties to configure for the multithreaded batch jobs:

- Threads allocated in Slave Process Server #1 + Threads allocated in Slave Process Server #2 + … [All Slave Process Servers] must be equal to the specific "Threads Per Job" parameter in `cmxcleanse.properties` file.

  The following properties are related to threads:

  - `cmx.server.automerge.threads_per_job`

  - `cmx.server.batch.threads_per_job`

- Each Slave Process Server would get number of records as specified in the block size.

  The following properties are related to block_size:

  - `cmx.server.automerge.block_size`

  - `cmx.server.batch.block_size`

  - `cmx.server.batch.recalculate.block_size`

  - `cmx.server.batch.batchunmerge.block_size`

  - `cmx.server.batch.delete.block_size`

- After the last block is sent to the next available Slave Process Server, all Slave Process Servers that process the blocks MUST complete the job within the timeout period.
  `com.informatica.mdm.loadbalance.ControllerThread.timeout`

Slave Process Server

The following list describes the properties to configure for multi-threaded batch jobs:

- Threads allocated for Batch Job 1 + Threads allocated for Batch Job 2 + … [All parallel batch jobs] must not exceed "Threads for Batch Processing" of the specific Process Server.

- The `com.informatica.mdm.batchserver.RecyclerThread.max_idling` property specifies the idle time for a Process Server thread. The Process Server recycles the thread when it is idle for more than the configured value.

Recommendations for the Hub Console Optimization

The Hub Console parameters can be optimized for better performance.

The following table lists the recommendations for the Hub Console optimization:

Parameter	Recommended Setting	Description
Hub Console - Java (JNLP) Heap Size	`Initial Heap Size:` `128m` (Default) `Maximum Heap Size:` `512m` (Default).	`jnlp.initial-heap-size`, `jnlp.max-heap-size` This property is found in the `cmxserver.properties` file. If the client box, where the Hub Console is launched, can handle more Java memory, this value can be increased to add more memory to the Hub Console. Note that this memory is completely different from the JVM memory of the application server which is the server side memory.
Client Java Version (JRE)	As specified in the Product Availability Matrix (PAM).	Ensure that the client Java version is as specified in the PAM document for the product. It is not required for the Hub Console to pick the latest JRE that the client box has. The JRE version selected depends on the PATH variable. To ensure that the Hub Console uses the correct JRE, temporarily enable the console log as listed in the Informatica Knowledge Base article. When enabled, the next launch of Hub Console opens a Java console window where you can see the used JRE version in the top of the console.
Console log	Disable (Default).	Disable the logging by launching the javaws -viewer option on the client box in the run command. In the javaws window, go to Java Control Panel in the Advanced tab and perform the following steps: - Clear 'Enable tracing' and 'Enable logging' options in the Debugging section. - Check 'Hide Console' option in the Java Console section.

Parameter	Recommended Setting	Description
High Data Volume	See the parameter "High Volume of Data in `C_REPOS_TABLES` with historical data" in Database - General Recommendation Section.	Having huge number of records in the tables listed in the referenced section might cause multiple issues on Hub console. The following lists are the issues that might be include: - batch groups screen loads too slowly. - base object record is saved too slowly. - cleanse function is saved too slowly.
Network latency	Good network connection.	Hub Console communicates with the Hub Server (application server) frequently. Therefore, the Hub Console must process a considerable amount of data to and from the Hub Server. Have good network connection between the client box and the application server.

For more information, search the Informatica Knowledge Base for article numbers 310913 and 139923.

Recommendations for Data Director and SIF Optimization

The following table lists the recommendations for Data Director and Services Integration Framework (SIF) optimization:

Parameter	Recommended Setting	Description
Cache: Lookup Cache Update Period	Default is 300000 (5 minutes).	lookupCacheUpdatePeriod in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` The amount of time the lookup data is cached in the server. Change it to a higher value if your lookup data is not going to change in real time frequently. Keep a higher value as it would improve the performance as the cache refresh is reduced.
Cache: SAM Cache Update Period	Default is 600000 (10 minutes).	samCacheUpdatePeriod in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` The amount of time the data about Security Access Manager (SAM) roles (no users) is cached in the server. Change it to a higher value if your role data is not going to change in real time very frequently. Keeping a higher value improves the performance, as the cache refresh is reduced.
Cache: Cache Refresh Daemon Thread Monitor (both Lookup and SAM Cache)	Default is 5000 (5 seconds).	threadSchedulerIdleTime in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` Time interval of the daemon thread that checks and triggers the Lookup cache and the Security Access Manager cache operations based on their update period. That is, every 5 seconds, the daemon thread would check and trigger the lookup cache and Security Access Manager cache if their update period is reached. This period can be increased to improve performance, if it is not necessary to check the SAM and Lookup Cache time periods every 5 seconds.

Parameter	Recommended Setting	Description
Search Operation: Maximum number of records retrieved at a time	Default is 100.	serverPageSize in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` Number of records to be retrieved from the Hub Server when a search operation is performed. Note that even though Hub Server shows 10 records per page, internally it loads these many numbers of records (given by this parameter). And the Hub Server caches the remaining records that are not shown in the view. **Note:** This parameter is used in Search Results and in functional areas such as potential matches, hierarchy relationship records, wherever a list of records is retrieved for the user. A higher value could make the search results performance better, provided the Hub Server can handle the load and cache.
Search Operation: Maximum number of records during search results export	Default is 5000.	maxSearchResultsExportedRows in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` Number of rows that are exported when using the export option in the search results page. Do not increase the value unless there is a valid use case. The greater the value, the lower will be the performance of the export operation.
Search Operation: Case Insensitive Search	Default is false. Adds performance overhead when enabled.	`case.insensitive.search` This property is found in the `cmxserver.properties` file. If set to true, users can enable the case insensitiveness attribute for individual columns in the base object, to enable case insensitive query search in Data Director. Enabling this flag would incur a new index created for such columns. As index management has its own performance overhead, use this flag with caution. **Note:** If there is any system column that you want to include with case insensitive search, create a functional index manually for such columns. For more information, search the Informatica Knowledge Base for article numbers 154132 and 154139.
Search Operation: Default effective date	Default is true.	isEffectiveDateIncluded in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` By default, effective date is automatically populated with current date. This may impact performance for normal queries that do not need effective date. In such scenarios, you can set this flag to false to remove the default value. For more information, search the Informatica Knowledge Base for article number 347953.
Search Operation: Remove Duplicates Flag	As applicable. Default is true.	`cmx.server.remove_duplicates_in_search_query_results` Default is true. This property is found in the `cmxserver.properties` file. By default, Data Director filters the duplicate results from the search query that joins child records. However, this flag if enabled, can affect the performance depending on the time it takes to de-duplicate the search results. If duplicates are not an issue in the search result, you can disable this flag. For more information, search the Informatica Knowledge Base for article number 122829.
Data View: Asynchronous Child Load	Default is false. Adds performance overhead when enabled.	asyncChildLoading in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` If true, all child records are opened (expanded) automatically when you open a parent subject area.

Parameter	Recommended Setting	Description
Hierarchy: Maximum records in Relationship Flyover	Default is 10000.	`sif.api.hm.flyover.max.record.count` This property is found in the `cmxserver.properties` file. This number denotes the maximum number of relationships that are to be shown in Relationship Flyover in Data Director. Internally, the best version of the truth (BVT) calculation is done for all such relationships. Therefore, having a high number here would impact the performance of relationship flyover. For more information, search the Informatica Knowledge Base for article number 157628.
Hierarchy: Inactive relationship records	Default is false.	hmInactiveRelationshipsAvailable in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` If true, Hierarchy Manager in the Data Director would show all the inactive relationship records also. Do not keep this value as true unless there is a valid use case to do so.
Task Context: Parallel Promote Threads	Same as the number of Hub Server Cores.	maxParallelPromoteThreads in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` Default is 1. Determines the maximum number of parallel threads when a task is approved (Promote API). This is helpful when you promote parent, child, and grandchild subject areas together.
Task Context: Parallel BVT Threads	Same as the number of Hub Server Cores.	maxParallelBvtThreads in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` Default is 1. Determines the maximum number of parallel threads when a task is viewed (PreviewBVT API).
Task Context: Maximum tasks assigned to a user	Default is 25.	`sip.task.maximum.assignment` This property is found in the `cmxserver.properties` file. Maximum number of tasks assigned to each user when automatic task assignment is enabled. Keeping this a high number would impact the performance of task context screens in Data Director. For more information, search the Informatica Knowledge Base for article number 134833.
Bulk Import: Bulk Import Threads	Same as the number of Hub Server Cores.	maxImportThreads in `CMX_SYSTEM.C_REPOS_DS_PREF_DETAIL` Default is 5. Maximum number of threads to be used while using the Bulk Import module.
Bulk Import: User Exit	As applicable.	PUT user exits are called for each and every record individually. Having a complex logic in the PUT user exit would deteriorate the performance. Use user exit with caution.
Bulk Import: Children, Foreign Keys, and Lookups	As applicable.	More number of children, foreign key relationships, and lookup columns in the data import would impact the performance.

Parameter	Recommended Setting	Description
Search Operation and HM Hierarchy: BVT Optimization Limit	`temptableTimeToLive` default is 20. `MaxRowCount` default is 5000.	This property is found in the `cmxserver.properties` file. Applicable for SearchQuery, SearchHMQuery, GetOneHop, and GetEntityGraph SIF APIs (also the relevant Data Director modules) when effective dates are specified. `searchQuery.buildBvtTemp.MaxRowCount`: Specifies the maximum number of records to be used in the BVT calculation when processing a search operation with effective date filters. `sif.search.result.query.temptableTimeToLive.seconds`: Specifies the number of seconds, the temporary table of the specific API must reside in the server. That is, the earlier mentioned APIs would store the full results in its own temporary tables which are cleaned-up automatically based on this timeout. Pagination would work based on the data from this temporary table instead of re-querying the API repeatedly from the core tables. For search operations involving larger number of records, it is recommended to increase this timeout value. Example: If a hierarchy entity can have more than 10,000 related records, set the values as shown: `sif.search.result.query.temptableTimeToLive.seconds=3600 searchQuery.buildBvtTemp.MaxRowCount =100000` For more information, see the *Multidomain MDM Configuration Guide* and the *Multidomain MDM Services Integration Framework Guide*.
Traffic compression	Default is TRUE.	`cmx.bdd.server.traffic.compression_enabled` This property is found in the `cmxserver.properties` file. Specifies if Data Director server traffic compression is enabled. For more information, see the *Multidomain MDM Configuration Guide*.
SearchMatch API: Thread count	Default is 1.	`cmx.server.match.searcher_thread_count` Available in `cmxcleanse.properties`. Configures the thread count for the SearchMatch API. Optimal value can be decided based on your environment. You may need to test with an incremental value of thread count until you see an optimal performance.

Recommendations for Environment Validation Tools and Utilities

You can verify the current environment and identify the area of improvement by using different tools and utilities.

The following table lists the recommendations related to the database:

Parameter	Recommended Setting	Description
Test IO (utility) applicable for Oracle	Available from Informatica Support.	Test the datafile input/output (I/O) capability of the data files of the database tablespaces by referring to the Informatica Knowledge Base article number 506035. The test result must meet or exceed the standards documented for 'Good Performance' seen in the Informatica Knowledge Base article number 506035.
Ping/TraceRt	Response Latency must be less than 10 milliseconds.	Run a ping or tracert (Trace Route) to MDM Database (`CMX_SYSTEM`) and individual ORS from application server box.
Test specific disk partition IO	Reasonable MB/ second.	Follow the instructions in the Informatica Knowledge Base article number 139805 to test all disk partitions involved in the MDM Hub, including partitions where tablespaces reside and where the database debug log is written. If greater than 10 Mb/second, that specific partition needs to be fixed.

The following table lists the recommendations for the Hub Server and the Process Server:

Parameter	Recommended Setting	Description
Ping/TraceRt	Response Latency must be less than 10 milliseconds.	Run a ping or tracert (Trace Route) to the MDM Hub Server or the Process Server from client box or the Hub Server box. Repeat this step for the following boxes wherever applicable: - All nodes in a cluster - Load balancer - Web Server

Glossary

_PKQ sequence

Sequence that is used to populate the ROWID_OBJECT of base object record. For example, the C_PARTY BO uses C_PARTY_PKQ sequence is used to populate the party records.

<INFAHOME>

Physical location where the Hub Server and the Process Server are installed.

heap size

Amount of memory allocated to Java processes which are created on the same JVM.

Hub Server

The server that manages core and common services for the MDM Hub.

master database

Database instance that stores metadata to manage individual domain schemas called ORS schemas. The database instance is unique to each the MDM Hub environment.

MaxPermGen

A JVM parameter that indicates size of the maximum memory where class metadata information is loaded.

ORS

Operational Reference Store. A database instance where you store domain data.

PermGen

A JVM parameter that specifies the size of initial memory where class metadata information is loaded.

Process Server

The server that cleanses and matches data and performs batch jobs such as load, recalculate BVT, and revalidate.

response latency

The time duration between request and response.

Tracert

Network diagnostic tool that displays the route to a particular destination with the transit delay information.

user profile

An internal object of the MDM Hub that stores the user details including authentication and associated roles.

Xms

A JVM parameter that specifies the initial Java heap size.

Xmx

A JVM parameter that specifies the maximum Java heap size.

XREF

Cross reference. Data that relates the base object data with the relevant source information.

Xss

A JVM parameter that specifies the memory assigned for stacking the threads created within the Java process.

INDEX