How-To Library

Informatica

Profiling and Discovery Sizing Guidelines

Abstract

The system resource guidelines for profiling and discovery include resource recommendations for the Profiling Service Module, the Data Integration Service, profiling warehouse, and hardware settings for different profile types. This article describes the system resource and performance tuning guidelines for profiling and discovery.

Supported Versions

- Data Quality 10.2
- Data Quality 10.2.1
- Data Quality 10.2.2
- Data Quality 10.4
- Data Quality 10.4.1
- Data Quality 10.5
- Data Quality 10.5.1

Table of Contents

Profiling Service Module

The Profiling Service Module is a module in the Data Integration Service that manages request to run profiles and generate scorecards. To create and run profiles and scorecards, you must associate the Data Integration Service with a profiling warehouse. The Profiling Service Module stores profiling data and metadata in the profiling warehouse.

Overview

When you run a profile or scorecard in the Analyst tool or Developer tool, the application sends the request to the Data Integration Service. The Profiling Service Module converts the profile into one or more mappings. The Profiling Service Module sends the mappings to the LDTM for optimization and compilation. The LDTM passes the compiled mappings to DTM instances that get the profiling rules and run the profile or generate a scorecard for the profile.

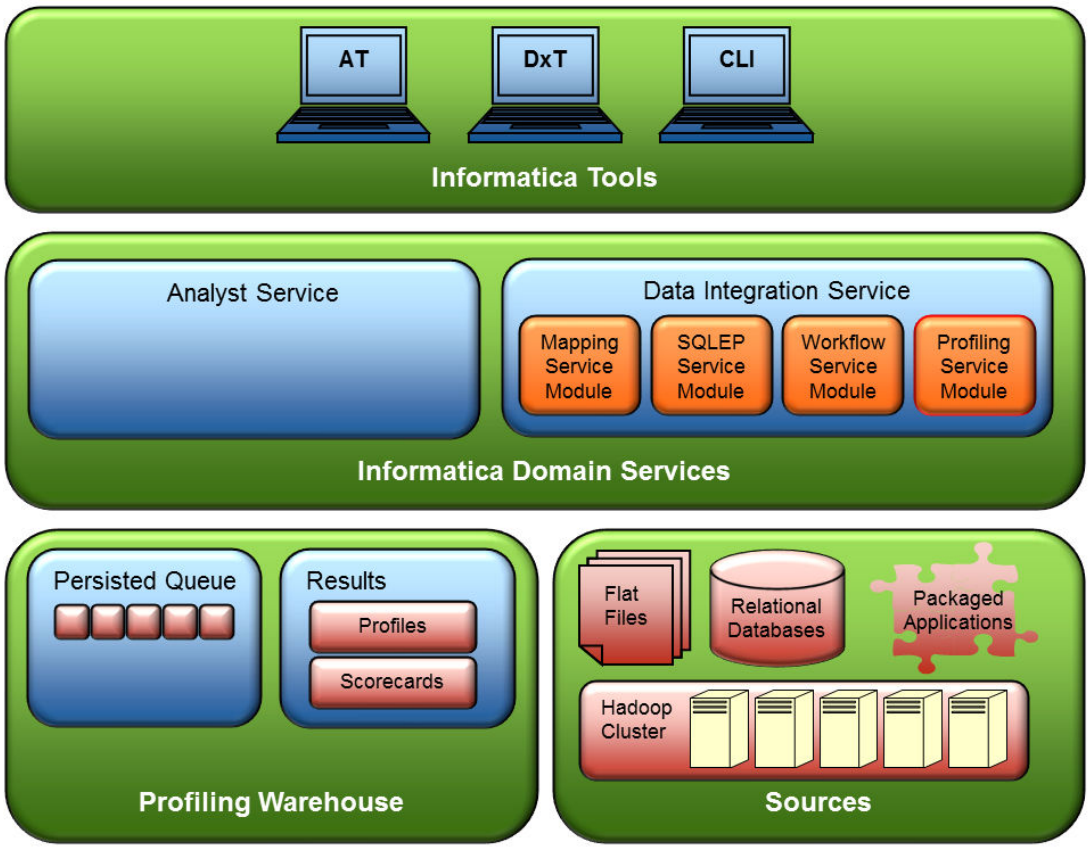
The Profiling Service Module interacts with the following components:

Components	Description
Profiling Warehouse	The database that contains the profiling results. It can also contain cached source data for drilldown.
Relational Database Sources	Databases that contain data organized for efficient data processing. Modern relational databases are optimized to process the data that is stored in them, and the Profiling Service Module takes advantage of this optimization.
Flat File Sources	Files that contain data in delimited or fixed-width form.
Non-relational Sources	Non-relational sources of data organized for efficient data processing. - The main type of non-relational database source is the mainframe, for example, an IMS or a VSAM database system. - Another type of non-relational source is a SAP database. The Profiling Service Module requires additional resources to read a non-relational database source. For mainframe sources, the Profiling Service Module takes up most of the processing tasks to minimize the data access costs.

Functional Architecture of Profiling Service Module

The Profiling Service Module is a component of the Data Integration Service and runs profile jobs and returns profile results. The functional architecture consists of tools, services, and databases.

The following figure shows the functional architecture of the Profiling Service Module:



Data Integration Service

The Profiling Service Module runs all the profiling jobs as mappings and uses the scalable and multithreaded Data Integration Service environment. The Profiling Service Module analyzes each profile job to determine the optimal number of mappings and sequence of mappings for the best job performance.

You can configure the Data Integration Service to run on a single node or multiple nodes. In a single-node configuration, all the processing happens on one node. In a multinode configuration, all the processing happens on an Informatica grid. In both configurations, the Data Integration Service is a single service. You cannot differentiate between mappings running on one node or multiple nodes.

Profiling Warehouse

The profiling warehouse stores all the profile results, computes SQL queries, and returns all SQL queries on these results. You can use a set of profiling warehouse views that you can customize for the profile statistics that you want to read. Profiling warehouse relies on standard relational database technology for scalability and performance.

Profiling warehouse maintains the persisted profile job queue. The profile job queue stores jobs based on priority so that the real-time jobs with high priority run first followed by the long-running jobs. This method of running different job types in a nonlinear manner brings out the best throughput.

Data Sources

The Data Integration Service can access all data sources. Based on the type of source data, the Data Integration Service uses pushdown optimization to transfer the profiling logic to the data source for full or partial run.

Scaling the Run-time Environment for Profiling Service Module

The Profiling Service Module is a multithreaded module in the Data Integration Service. When the Profiling Service Module scales, it uses the additional memory, temporary disk, and CPU cores that you make available to the machine.

The Profiling Service Module scales in the following ways based on the profile job type and the Profiling Service Module configuration:

Type	Description
Scaling Up	Scaling up is the ability of the Profiling Service Module to perform the same task proportionately faster after you add more resources to a single machine. - If you double the resources, the amount of time the profile job takes reduces by half. - If you double the number of CPU cores, you must add additional memory and temporary disk space.
Scaling Out	Scaling out indicates the ability of the Profiling Service Module to scale when adding more machines to it. Similar to scaling up, doubling the number of machines halves the amount of time required to run a profile job. The Profiling Service Module scales out in the following ways: - Informatica grid. Run the Data Integration Service on an Informatica grid. The Profile Service Module submits the profiling mappings in such a way that it evenly distributes the workload across the grid. - Hadoop cluster. The Profiling Service Module runs the profile jobs on a Hadoop cluster. The Hadoop cluster is a distributed and highly scalable run-time environment to run profiling jobs. To run a profiling job faster, you can add nodes to the Hadoop cluster.
Pushdown	In the pushdown method, the Profiling Service Module sends a part of the profile job to the relational database for processing in the form of an SQL query. The Profiling Service Module processes the query results. If the relational database has sufficient processing power, you can allocate more resources for the Data Integration Service where the Profiling Service Module runs. You can add more resources to a single node or add more machines to a grid. If the performance limitation is the processing power of the relational database, the machine running the relational database requires more resources.

Profiling Service Module Resources

The Profiling Service Module is a component of the Data Integration Service that manages requests to run profiles and generate scorecards. You can plan the system performance well if you understand the architecture of the Profiling Service Module and concepts of performance and data sources.

Supported Data Sources

The Profiling Service Module can access all supported data sources. Each category of data source has distinct performance characteristics. You can plan the profile deployment and troubleshoot performance well if you understand the differences of each category of data sources.

The following data sources have different performance characteristics when you run profiles:

Data Source	Description
Flat file source	The Profiling Service Module reads each row in a flat file source. The Profiling Service Module can construct rows by reading bytes from a flat file as required. When you run a profile on flat file data sources, the Profiling Service Module runs all the processing logic in the mapping, including sorting and buffering.
Relational source	Relational data sources contain an SQL query engine that you can use to view the data in a front-end application. The Profiling Service Module shares the processing logic with the relational database for some of the profile jobs. If the Profiling Service Module and relational source are in two different machines, the Data Integration Service distributes the processing logic across the resources of the two machines. You can optimize the relational source for the profile queries that results in the increase of performance.
Semi-structured source	Avro, JSON, Parquet, and XML formats are semi-structured data sources. You can create flat file data objects for JSON or XML data sources. You can create complex file data objects for Avro, JSON, Parquet, and XML data sources in Hadoop Distributed File System (HDFS).
Mainframe source	If the mainframe source is nonrelational, such as IMS or VSAM, the Profiling Service Module processes the source as a flat file. It is not recommended that you share the SQL processing queries with IBM DB2 sources because mainframe access can result in additional charges or license fees. The Profiling Service Module considers all relational mainframe sources as special flat files and performs all the processing logic. This method reduces the number of I/O operations on the mainframe source.
Other sources	The Profiling Service Module considers social media, PowerExchange, logical data object, and mapping transformation data sources as flat files.

Data Integration Service Resources

The Data Integration Service runs the Profiling Service Module, and it has base memory and variable memory requirements. The variable memory requirements are based on the number of parallel mappings.

The memory requirements are as follows:

Type	Requirements
Base Memory	The amount of memory required to run the Java Virtual Machine that the Data Integration Service uses, which is approximately 640 MB.
Variable Memory	The amount of memory required to run each Data Transformation Manager thread. One Data Transformation Manager thread is required to run each mapping that computes a part of a profile job. This overhead is dependent on the Maximum Execution Pool Size property in the service properties. The default value of this property is 10 and the overhead is approximately 1,000 MB. A mapping requires additional memory to read address or identity reference data. A profile that reads the output of an address validation rule may incur an additional 1 GB in memory to read and cache the address validation reference data.

Profiling Service Module Resources

The Profiling Service Module uses fewer resources to run a profile on a relational data source than a flat file data source.

Following are the CPU, memory, disk, and operating system requirements for the Profiling Service Module

CPU	The Profiling Service Module uses less than 1 CPU. Consider the following CPU requirements for different profile types: - Column profiles. Depends on the data source type. - Relational systems. Requires less than one CPU for each Data Transformation Manager thread. - Flat files. Use approximately 2.3 CPUs for each Data Transformation Manager thread. - Key and functional dependency discovery. Requires one CPU for each Data Transformation Manager thread. - Join, foreign key, and overlap discovery. Requires two CPUs for each Data Transformation Manager thread. When you calculate the number of CPUs required for Data Transformation Manager operations, round the total number up to the nearest integer. Disk space is a one-time cost when the Data Integration Service is installed. CPU overhead is minimal when the Data Integration Service is not running jobs.
Memory	No additional memory is required beyond the minimum needed to run the mapping.
Disk	No disk space is required.
Operating System	Use a 64-bit operating system, if possible, as a 64-bit system can handle memory sizes greater than 4 GB. A 32-bit system works if the profiling parameter fits within the memory limitations of the system.

Sizing Guidelines for Profiling

Profile sizing is an iterative process in which you analyze profile performance, use guidelines to estimate and apply resources and parameters, and monitor and adjust the results as required. The sizing guidelines in this document are for the Profiling Service Module and are independent of the resource requirements for other modules in the Data Integration Service.

Overview

You can use the sizing estimates to optimally run the profile jobs in your organization based on your performance requirements. As a profile administrator or system architect, you need to have a good understanding of the Data Integration Service and Profiling Service Module that have performance characteristics based on the hardware and software configurations.

Note: The Data Integration Service includes modules that manage data transformation processes. The Profiling Service Module is a module in the Data Integration Service that manages request to run profiles and generate scorecards.

Consider the following important factors for profile sizing:

- The total number of users on the system and typical number of users that run profiling jobs.
- The mix of profile jobs, such the number and type of profiles you plan to run.
- The general characteristics of the data sources, such as the typical number of rows, number of columns, and average column width.
- A breakdown of the data sources by type, such as flat files, relational sources, and non-relational sources.

Profile Sizing Process

The profile sizing process is a set of tasks designed to increase system performance. This process assumes that you have a profiling implementation in your organization. If you do not have a profiling implementation, the process can help you with the initial procurement of hardware resources.

The sizing requirements can differ based on the type of profile deployment and size of the organization. To increase performance, you can optimize the Data Integration Service and Profiling Service Module and set up the required hardware. You can change the Data Integration Service and Profiling Service Module properties to match the profile deployment environment in your organization and optimally run the profile jobs.

You might need to consider the sizing requirements because of different reasons such as:

- Performance issues. Profile jobs that do not meet expectations, slow response time to profile runs, and resource capacity planning for future growth can lead to sizing assessment.
- Performance factors. Number of users, volume of source data, and profile types.

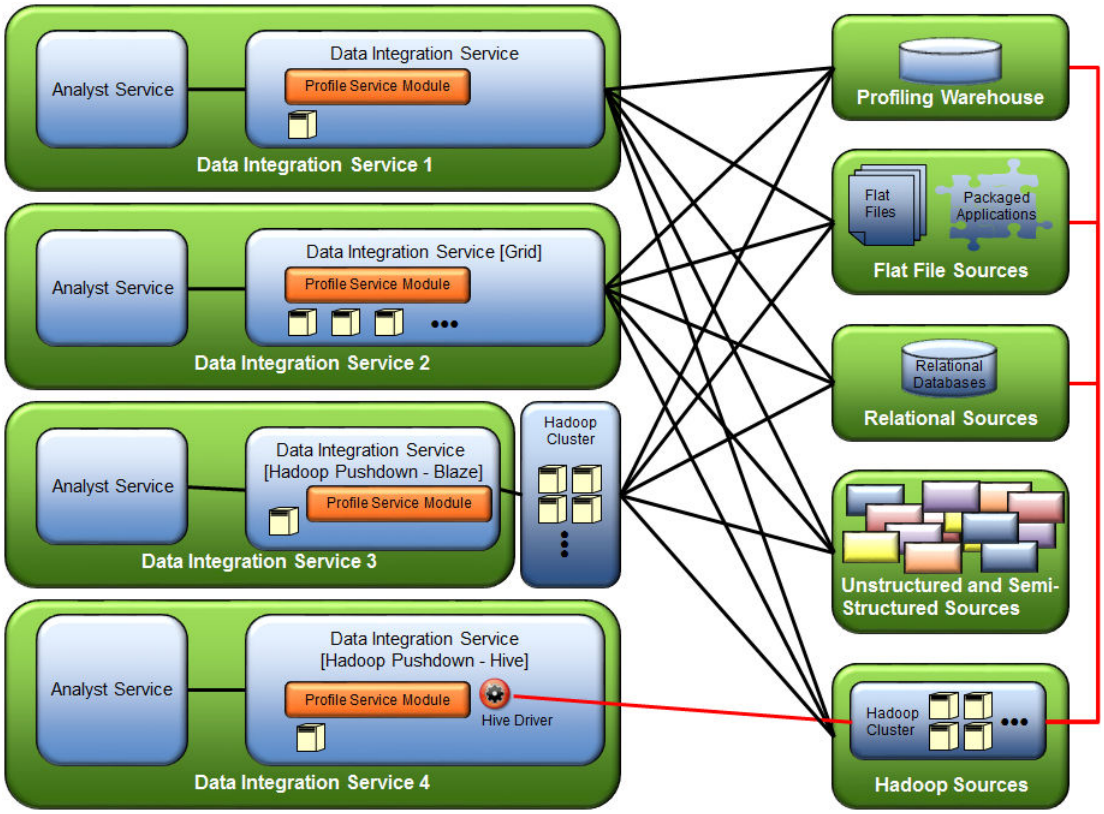
You can complete the following tasks as part of the profile sizing process:

1. Analyze the sizing requirements. You might need to consider the sizing requirements because of different reasons. For example, performance problems, such as profile jobs that do not meet expectations, slow response time to profile runs, and resource capacity planning for future growth can lead to sizing assessment.
2. Analyze the performance factors, such as the number of users, volume of source data, and profile types.
3. Analyze the Data Integration Service properties and properties specific to Profiling Service Module in the Administrator tool.
4. Use the profile sizing guidelines to calculate the required hardware resources and Informatica Administrator properties.
5. Make the hardware changes as required.
6. Make the Data Integration Service and Profiling Service Module property changes as required.
7. Monitor the changes to verify the required performance improvements. If you do not see the expected performance improvements, repeat the process from step 1.

Deployment Architecture

A typical profiling solution includes an Informatica Analyst service, the Data Integration Service configured as a single node or grid, and profiling warehouse.

The following figure describes four deployment architectures of the profiling solution:



The Informatica domain is the administrative unit for the Informatica environment. The domain is a collection of nodes that represent the machines on which the application services run. The Data Integration Service performs data integration tasks for Informatica Analyst and Informatica Developer. The Profiling Service Module runs profiles. The Profiling Service Module can run profiles on different types of source data, such as flat files, relational sources, and non-relational sources. The profiling warehouse stores profile results. The Analyst Service runs the Analyst tool. You must configure Hadoop pushdown properties for the Data Integration Service to run profiles in the Hadoop environment. You can run the profiles on the Blaze engine or Hive engine in the Hadoop environment.

Baseline System

A baseline system is a profile configuration based on the minimum recommended configuration parameters. You can use a baseline system to differentiate the scenarios for the development deployment and production deployment. You then use the baseline system to plan the resources.

The following table describes the baseline system with the recommended configuration:

Component	Data Integration Service	Profiling Warehouse
CPU	4 cores	2 cores
Memory	8 GB	8 GB
Temporary Disk	1 drive, 50 GB free	100 GB free
Users	10	-
Current Users	2 to 4	-

The following table describes the baseline system with recommended configuration for profile jobs that you run on the Blaze engine in the Hadoop environment:

Component	Data Integration Service	Profiling Warehouse	Cluster
Nodes	-	-	3 nodes / cluster
CPU	4 cores	2 cores	8 cores / node
Memory	8 GB	8 GB	64 GB / node
Temporary Disk	50 GB free	100 GB free	100 GB / node

Profile Deployment

As part of profile deployment, you need to plan the resources for profile deployment in the development environment and production environment. The Profiling Service Module has a set of parameters that controls the performance of a profiling job. You must configure the parameters for each deployment.

When you plan a profile deployment, you need to consider the profile job type, response time, user type, and data sources.

The following categories determine the system resource recommendations:

- Resource guidelines for the Profiling Service Module and the Data Integration Service, including memory, disk space, and CPU usage.
- Resource guidelines for column profiling, key discovery, functional dependency discovery, foreign key discovery, and overlap discovery based on the data source types and hardware capacity.

Profile Job Type

You can have multiple profile jobs when you run a profile on a data source. Each profile operation uses a different combination of resources. The mix of profile jobs determines the resource requirements. You need to balance the performance goals and resource costs effectively to optimize the deployment.
The following table summarizes the relative use of resources by each profile job type and data source:

Profile Operation	Data Source Type	CPU	Memory	Disk Space	RDBMS	Profiling Warehouse
Column Profile	Flat File	Medium	Low	Medium	None	Medium
Column Profile	Relational	Low	Low	None	High	Medium
Data Domain Discovery	Flat File	High	Low	Medium	None	Low
Data Domain Discovery	Relational	Medium	Low	None	High	Low
Key Discovery	-	Low	High	High	None	Low
Functional Dependency Discovery	-	Low	High	High	None	Low
Overlap Discovery	-	High	Low	None	None	Low
Foreign Key Discovery	-	High	Low	None	None	Low

Profile Operation	Data Source Type	CPU	Memory	Disk Space	RDBMS	Profiling Warehouse
Enterprise Discovery	Flat File	High	High	High	None	High
Enterprise Discovery	Relational	High	High	High	High	High
Reporting or Viewing Results	-	Low	None	None	None	Low
Drilldown	Flat File	Low	None	None	None	None
Drilldown	Relational	Low	None	None	Low	None

Response Time

The speed of a profile job run depends on the type of the profile job and resource types that the profile job uses. Most of the algorithms benefit from faster CPUs and memory because the operating system can use memory in different ways including caching data.
If the profile job has multithreaded algorithms, you can add additional CPU cores to improve the response time. Some algorithms perform better with faster or additional temporary disk.

The network speed is critical when the Data Integration Service queries or writes data to the profiling warehouse in another machine. The network speed is also important when the Data Integration Service running on one machine pushes queries to the RDBMS on another machine.

The following table summarizes the resource types for the Data Integration Service that increase response time when you add more or better resources for each resource type:

Profile Job Type	Faster CPU	Cores	Memory	Disk	Network
Column Profile	Yes	Yes	No	Yes	Yes
Data Domain Discovery	Yes	Yes	No	Yes	Yes
Key Discovery	Yes	No	Yes	Yes	No
Functional Dependency Discovery	Yes	No	Yes	Yes	No
Overlap Discovery	Yes	Yes	No	No	No
Foreign Key Discovery	Yes	Yes	No	No	No
Enterprise Discovery	Yes	Yes	Yes	Yes	Yes
Reporting or Viewing Results	Yes	No	No	No	Yes
Drilldown	Yes	No	No	No	Yes

User Types

The profile workload including system-generated profile jobs, such as periodic scorecard runs, depends on the number of users and type of users. When the number of users increases, more profile jobs run concurrently. The concurrent jobs indicate a range of the number of average profiling jobs for each profile type that can run successfully for the specified number of cores and memory. The type of profiling jobs that you need to estimate for depends on the type of user and resources.

Each user type might generate the following profile jobs:

Informatica Analyst user

Submits profile jobs, such as profile run, scorecard run, and drill-down jobs.

Informatica Developer user

Runs all the profile job types including enterprise discovery. In the Developer tool, the profile job type depends on the project.

infacmd command line utility user

Schedules scorecard runs but these users can run all profile jobs.

Pushdown Optimization for Data Sources

The effective use of the computing resource allocation depends on the data source type . When you run a profile on a relational source, the Profiling Service Module can transfer some of the profiling logic to the data source. The source system must be able to accommodate the additional workload. When you run a profile on a non-relational data source, the Profiling Service Module needs to compute the profiling job in the Data Integration Service. You can allocate all the computing resources to the system that runs the Informatica application. The pushdown of the processing logic also depends on the rule type and profile type.

The following guidelines determine the pushdown optimization for column profiles and rules:

- Pushdown optimization applies only to physical data sources.
- Pushdown optimization applies only to the following rules:
  - Rules containing a single expression transformation or internal expression rule with a single Boolean output port type.
  - Reusable validation rules that contain a single validation expression transformation.
  - Rules created in the Analyst tool.
- Pushdown optimization does not apply to the following data objects:
  - Logical data object and mapping specification

    Note: Pushdown optimization does not apply to the profiling logic. However, the Data Integration Service machine can optimize the logical data object and mapping specification mappings and push down parts of the mappings before the Data Integration Service applies the profiling logic.
  - Mapping specification
  - Flat file
  - Mainframe source
- Pushdown optimization does not apply to the following rules:
  - Rules with multiple transformations.
  - Rules with a single, non-Boolean output port.
  - Reusable rules.
  - Rules that contain IIF(), Ltrim(), or Rtrim() function.
- Pushdown optimization does not apply to columns with the Date data type.

The Profiling Service Module pushes the value frequency computation and rule logic to the data source for column profiles, data domain discovery profiles, and enterprise discovery profiles. The Profiling Service Module pushes the filter logic to the data source for key discovery and functional discovery for a single table, and overlap discovery and foreign key discovery for multiple tables.

Note: If a column profile run does not push down the value frequencies, the Data Integration Service does not push down the rules.

The following table summarizes the resource allocation between the Profiling Service Module and data source system based on the pushdown of the processing logic:

Profile Job Type	Pushdown	Database	Profile Service Module
Column Profile	Yes	Medium	Medium
Column Profile	No	None	High
Data Domain Discovery	Yes	Medium	Medium
Data Domain Discovery	No	None	High
Key Discovery	Yes	None	High
Key Discovery	No	None	High
Functional Dependency Discovery	Yes	None	High
Functional Dependency Discovery	No	None	High
Overlap Discovery	Yes	None	High
Overlap Discovery	No	None	High
Foreign Key Discovery	Yes	None	High
Foreign Key Discovery	No	None	High
Enterprise Discovery	Yes	Medium	Medium
Enterprise Discovery	No	None	High
Reporting or Viewing Results	Yes	None	Medium
Reporting or Viewing Results	No	None	High
Drilldown	Yes	Medium	Low
Drilldown	No	None	High

Hardware Guidelines for Profiling

The performance of a profile depends on the hardware architecture. You usually have predefined hardware components in the system and it might not be possible to upgrade an individual component without the replacement of the computer. However, understanding how each component functions and the implication on performance can help when you plan for increase in performance.

Consider the following hardware considerations for profiles:

CPU

An increase in the processing speed of the CPU results in faster computation of the profiling results. Usually, doubling the speed of the CPU reduces the time of the profile result computation by half for the same processor family.

The speed of the CPU depends on the micro architecture of the CPU, including the clock speed, instruction set, and whether the CPU is 32 bit or 64 bit. The speed of the CPU depends on the number of effective cores or processing threads. The Data Integration Service uses multithreading programming techniques and uses more than one core or one processing thread at a time. If the CPU has more cores, the system can scale better and have increased throughput.

Disk

The Data Integration Service can use disk to read flat files and similar nonrelational sources.

The Data Integration Service streams and processes the data. The dependent factors are the disk type and network bandwidth to the CPU. To increase the read performance in this use case, you can optimize the storage architecture for single file access.

This optimization can be complex due to the variety of storage technologies:

- Single and RAID Disk. The important factors you need to consider are the rotation speed, seek speed, controller speed, and motherboard bus speed.
- SAN. The storage technology is implementation dependent. The important factors you need to consider are the number and types of host controllers, transfer speed from the appliance, and motherboard bus speed.

You can also use disk for temporary storage. The Data Integration Service depends on temporary disk space where the Data Integration Service can write to multiple temporary files on different disks in parallel. The Data Integration Service uses this approach to increase the read and write bandwidth that results in faster processing. The system can increase performance with a single temporary directory that uses a high performance storage implementation.

Memory

If you add faster memory, the CPU can quickly retrieve data from memory and increase the performance.

If you add more memory, the operating system and database can cache more data for faster access. The increase in cache data increases the performance of the Profiling Service Module because the Profiling Service Module can quickly access the data available in memory.

Network

The Data Integration Service uses the network to access databases that do not reside on the Data Integration Service machine. Multiple profile functions depend on the network to transfer large amounts of data from a database that the Data Integration Service processes.

The following profile functions use the network:

- Column profile. The Data Integration Service can push the data processing down to the database. However, a column profile processes the unique values locally. If the source table is large and the profile runs on a key column, the Data Integration Service transfers all the values in the column.
- Rule profile. The Data Integration Service pushes some rules down to the database. If the Data Integration Service cannot push a rule down to the database, the Data Integration Service transfers all the values in each column that is part of the rule.
- Data domain discovery. The Data Integration Service can push the processing down to the database. However, a column profile processes the unique values locally. If the source table is large and the profile runs on a key column, the Data Integration Service transfers all the column values.
- Foreign key discovery. The Data Integration Service transfers all the data because the Data Integration Service reads all the values for each source table that you run a profile on.

The network speed between the Profiling Service Module and database plays an important role in the performance. The profile functions run faster if you increase the network speed and the number of host controllers. If the network has multiple nonprofile functions, the profile functions run slower.

Hardware Guidelines for Column Profiles

The factors that affect profile performance include the speed of the CPU, memory size, disk space, and the speed of the disk and network.

Consider the following hardware considerations for column profiles:

Component	Requirement
CPU	The Profiling Service Module uses the multithreaded environment of the Data Integration Service. Therefore, the CPU speed is less important than the number of cores in the CPU. To calculate the number of cycles that the Profiling Service Module uses each second, add the clock speeds of the cores.
Memory	Profile operations run faster with more memory. When you run a profile on a flat file source, the Profiling Service Module uses memory to sort the value frequency data and buffer data. The Profiling Service Module performs multiple read operations to the same part of a file by reading from the memory buffer and not from the flat file on disk. This method applies to rule profiling for flat file and relational sources.
Disk	The Profiling Service Module uses disk space for temporary storage when memory cannot store all the intermediate profile results. The Profiling Service Module uses multiple temporary directories in a single profile job. The Profiling Service Module divides the storage and Input/Output operations among multiple disks in parallel. The profile performance increases if separate physical disks have the temporary directories. Disk technologies, such as rotational speed and on-disk buffering, affect profile performance.
Input/Output	The Input/Output speeds for memory, disk, and network affect the performance of the Profiling Service Module. Higher speeds allow the Profiling Service Module to quickly access large amounts of data. Network speed affects the relational databases that are not on the Data Integration Service machine and flat files that are on a storage device attached to the network.

Flat File and Mainframe Sources for Column Profiles

When you run a profile on a flat file, the Profiling Service Module divides the job into multiple mappings that infer the metadata for the columns and virtual columns. Each mapping can run serially, or two or more mappings can run in parallel. In addition, the Profiling Service Module generates another type of mapping to cache the source data. This mapping always runs in parallel with the column profile mappings because it takes longer than a column profile mapping.

When you run a profile on a mainframe data source, the Profiling Service Module groups as many columns as possible into a single mapping. This grouping minimizes the number of table scans on the data source. Mainframe data sources require more disk space than flat files to store the temporary computations.

You can compute the total resources required by column profiles after you consider the following requirements:

Column Profile Mapping Requirements

The CPU, memory, and disk space requirements for a column profile mapping are as follows:

Component	Requirements
CPU	Column profiling consumes approximately 2.3 CPUs for each mapping. When you calculate the number of CPUs you need, round up the total to the nearest integer.
Memory for Mappings	The Profiling Service Module uses two methods for profile mappings. First, it applies a method that requires approximately 2 MB of memory for each column. If the first method does not work, it uses the second method of sorting columns with a buffer of 64 MB. The minimum resource required is 10 MB, representing 2 MB • 5 columns. The maximum resource required is 72 MB, representing a 64 MB buffer for one high-cardinality column and 8 MB for the remaining four low-cardinality columns.
Memory for the Buffer Cache	The Profiling Service Module caches the flat file data as it reads the data. Profiling speed increases if the Profiling Service Module can cache all the file data. The exception to using cache memory is when two or more mappings read a file concurrently. In this case, add 100 MB of memory. This enables the mappings to share the read operations and increase performance.
Disk	A profile mapping may need disk space to perform profiling computations. The following formula calculates the disk space for a single mapping: `2 X number of columns per mapping X maximum number of rows X ((2 bytes per character X maximum string size in characters) + frequency bytes)` where - 2 indicates two passes. Some analyses need two passes. - The default value for the number of columns for each mapping is 5. - Maximum number of rows is the maximum number of rows in any flat file. - 2 bytes per character is the typical number of bytes for a single Unicode character. - Maximum string size in characters is the maximum number of characters in any column in any flat file, or 255, whichever is less. - Frequency bytes value is 4. The frequency bytes store the frequency calculation during the analysis. Perform the above calculation and allocate the disk space to one or more physical disks. Use one disk for each mapping, and use a maximum of four disks.
Operating System	Use a 64-bit operating system to accommodate memory sizes greater than 4 GB. A 32-bit system works if the profiling parameter fits within the memory limitations of the system.

Note: These guidelines covers the optimal flat file profiling case, which uses five columns for each mapping. In some cases, the Profiling Service Module must run the profile for more than five columns in one mapping, for example, when running a profile on mainframe data where the financial cost of accessing the data can be high.

Profile Cache Mapping Requirements

A profile cache mapping caches data to the profiling warehouse and has different resource requirements than a column profile mapping.

The CPU, memory, and disk space requirements for a profile cache mapping are as follows:

Component	Requirements
CPU	The cache mapping requires approximately 1.5 CPUs.
Memory	The cache mapping requires no additional memory beyond the Data Transformation Manager thread memory.
Disk	The cache mapping requires no disk space.

Aggregate Profile Mapping Resources

To compute the total resources required by profiling, add the profile mapping requirements to the cache mapping requirements.

Use the following formula to determine the total profiling resources:

```
(number of concurrent profile mappings X resources for each mapping) + cache mapping resources
```

Relational Databases for Column Profiles

The Profiling Service Module transfers as much processing as it can to the machine where the database resides. You need to consider the division of work between the Profiling Service Module and database when you estimate resources for each machine.

Depending on the rule logic, rules can be pushed down to the database or handled internally by the Profiling Service Module. If a rule is pushed down to the database, it is treated like a column during the profile run. Rules that run inside the Profiling Service Module are treated like columns in a flat file profile run. The rules are grouped into mappings of five output columns at a time before running the profile. The flat file calculations apply in this case.

The network between the relational database and the Profiling Service Module must be able to handle the data transfers. For large databases, the bandwidth required can be considerable.

The resource guidelines are for a single mapping that pushes the profiling logic to the relational database for each column.

Component	Requirement
CPU	At least one CPU processes each query based on the relational database. If the relational database can increase the processing power, such as the parallel hint in Oracle, the number of CPUs that the mapping uses increases.
Memory	The relational database requires memory in the form of a buffer cache. The greater the buffer cache, the faster the relational database runs the query. Use at least 512 MB of buffer cache.

Component	Requirement
Disk	Relational systems use temporary table space. The formula for the maximum amount of temporary table space required is: `2 X maximum number of rows X (maximum column size + frequency bytes)` where - Two indicates two passes. - Maximum number of rows is the maximum number of rows in any table. - Maximum column size is the number of bytes in any column in a table that is not one of the very large data types that you cannot run a profile on. An example of the very large datatype is CLOB. The column size must take into account the character encoding, such as Unicode or ASCII. - Frequency bytes is 4 or 8 bytes. Frequency bytes store the frequency during the analysis. This is the default size that the database uses for COUNT(*). In many situations, the mapping uses less disk space. Perform the disk computation and assign the temporary table space to one or more physical disks. Use one disk for each mapping, and use a maximum of four disks.
Operating System	Use a 64-bit operating system to accommodate memory sizes greater than 4 GB. A 32-bit system works if the profiling parameter fits within the memory limitations of the system.

Hardware Guidelines for Data Domain Discovery

The system resource guidelines for data domain discovery depend on the type of data source and the capacity of the hardware. The type of data source and the complexity of the data domain rules determine the resource requirements for the Data Integration Service machine.

When you run data domain discovery, the Data Integration Service machine processes all of the data domain rules. When you run data domain discovery on some relational sources, the Data Integration Service computes the unique values to reduce processing load due to data domain rules. This division of work is similar to the division of work when you run a column profile.

One performance consideration that differentiates data domain discovery from a column profile is the amount of processing required for columns. When you run data domain discovery, the Data Integration Service processes all the column values to find matching inference for each data domain and accurate conformance. The Profiling Service Module limits the number of data domains for each profile mapping because the complexity of each domain rule can vary. More data domains for each profile mappings can affect the processing resources of the Data Integration Service machine.

When you run data domain discovery, the Profiling Service Module analyzes the data source and profile parameters that you configured to determine the required set of mappings. The Profiling Service Module first categorizes the volume of data in the data source and then generates the required number of mappings.

Low Volume Sources for Data Domain Discovery

Low volume sources contain up to 100,000 rows. The Profiling Service Module does not differentiate between flat file and relational sources. The Profiling Service Module uses a mapping strategy that runs a profile on up to 20 columns against 50 domains for each mapping.

Consider the following hardware considerations for low volume sources:

Component	Requirements
CPU	The CPU utilization is low. Additional CPU cores increase performance because the Profiling Service Module uses the multithreaded environment of the Data Integration Service.
Memory for Mapping	The low volume mapping uses approximately 128 MB without considering the data domain rules. The transformations in the data domain rules might require additional memory.
Disk	The Profiling Service Module does not generate intermediate results when you run data domain discovery for low volume sources. The disk requirement is negligible. However, the transformations in the data domain rules might require some amount of temporary disk.

High Volume Sources for Data Domain Discovery

High volume sources contain more than 100,000 rows. To reduce the aggregate computation in the data domain rules, the Profiling Service Module generates value frequencies for each column and runs them through the data domain rule. The Profiling Service Module pushes the computation to the database when you perform data domain discovery on relational data sources. When you run data domain discovery on flat files, the Profiling Service Module performs the computation.

Flat Files

The Profiling Service Module generates one mapping containing two columns in the data source and up to five data domains. This ratio maintains a balance between performing data domain discovery on a set of columns and the processing required to run the data domain rules on the columns. The Profiling Service Module runs these mappings sequentially by default. You can set the Maximum Concurrent Profiling Threads to a higher number to change the sequential run of mappings.

Consider the following hardware considerations for flat files:

Component	Requirements
CPU	The CPU usage is about 2.3 CPUs, which is the same for a column profile, for each mapping. The CPU usage can increase based on the complexity of the data domains.
Memory for Mapping	Each mapping requires at least 132 MB. The memory requirement for all data domain discovery mappings is 128 MB. The two columns in the mapping requires 2 MB each as buffer memory. If these buffers overflow, data domain discovery uses a secondary buffer of 64 MB. The total memory requirement is 196 MB. The memory requirement can be more because the transformations in data domain rules might have additional memory requirements.
Disk	A data domain discovery mapping might need disk space to perform profiling computations. The following formula calculates the disk space for a single mapping: `Number of columns for each mapping X maximum number of rows X ((2 bytes for each character X maximum string size in characters) + frequency bytes)` where - Number of columns for each mapping is 2. - Maximum number of rows is the maximum number of rows in any flat file. - Two bytes for each character is the typical number of bytes for a single Unicode character. - Maximum string size in characters is the maximum number of characters in any column in any flat file, or 255, whichever is less. - Frequency bytes is 4. Frequency bytes store the frequency calculation during the analysis.

Perform the calculation and assign the disk space to one or more physical disks. Use one disk for each mapping. You can consider a maximum of four disks.

Relational Sources

The Profiling Service Module usually generates one mapping that contains a single column in the relational table and up to five domains. Sometimes, the Profiling Service Module might generate multiple mappings for the same column.

Each mapping pushes the value frequency query to the database to minimize the number of redundant values. This pushdown method avoids duplicate computation in the data domain rules. The Profiling Service Module runs five of these mappings in parallel. The Maximum DB Connections parameter controls the number of parallel mappings.

Consider the following hardware considerations for relational sources:

Component	Requirements
CPU	The CPU usage is about 1 CPU for each mapping. The CPU usage can increase based on the complexity of the data domains.
Memory for Mapping	Similar to the low volume processing, each mapping requires at least 128 MB if you do not consider the data domain rules. The memory requirement for each data domain rule can be more because the transformations in the data domain rules might have additional memory requirements.
Disk	A data domain discovery mapping might need disk space for temporary tablespace on the database machine to perform profiling computations. The following formula calculates the temporary tablespace for a single mapping: `maximum number of rows X (maximum column size + frequency bytes)` where - maximum number of rows is the maximum number of rows in any table. - maximum column size is the number of bytes in any table column that is not one of the very large data types that you cannot run a profile on. An example of the very large datatype is CLOB. The column size must take into account the character encoding, such as Unicode or ASCII. - Frequency bytes is 4 or 8 bytes. Frequency bytes store the frequency during the analysis. This is the default size that the database uses for COUNT(*).

Compute the disk space and assign the temporary tablespace to one or more physical disks. Use one disk for each mapping. You can consider a maximum of four disks.

Hardware Guidelines for Foreign Key and Overlap Discovery

The hardware resource guidelines depend on whether column signatures are available before the profile run.

The Profiling Service Module uses a couple of strategies to infer foreign keys or overlap columns. If column signatures are available before the profile run, the Profiling Service Module does not require additional resources. If the signatures need to be computed, the profile needs additional resources.

The hardware resource requirements are as follows:

Component	Requirements
CPU	Requires two CPUs for each mapping.
Memory	Requires 64 MB of additional memory for internal caches If no column profile is run. Requires no additional memory if column profile is run.
Disk	Does not require temporary disk space.

Hardware Guidelines for Key and Functional Discovery

The Profiling Service Module processes a data source sample to infer the keys and functional dependencies. The bandwidth requirement for flat files and relational databases is less because the data size is usually small.

Both key and functional dependency discovery algorithms have large CPU resource and temporary disk space requirements. The algorithms use memory to cache between the intermediate results and temporary disk.

The factors that affect profile performance include CPU, memory, disk size, and disk speed:

Component	Requirements
CPU	Uses one CPU for each mapping.
Memory	Requires 256 MB of memory in addition to the mapping memory.
Disk Size	Caches intermediate profile results to the disk and the required amount of disk space depends on the complexity of data and the number of columns. You can consider a minimum of 128 GB disk space.
Disk Speed	The input/output speeds, for both memory and disk, affect the Profiling Service Module performance. Higher speeds allow the Profiling Service Module to quickly access large amounts of data.

Hardware Guidelines for Enterprise Discovery

The Profiling Service Module processes a data source sample to infer the keys and functional dependencies. The bandwidth requirement for flat files and relational databases is less because the data size is usually small.

Enterprise discovery is a process that systematically runs profiles in multiple ways across multiple sources. The profile types that are a part of the enterprise discovery are column profile, data domain discovery, primary key discovery, and foreign key discovery. Enterprise discovery runs a profile on each data source and then runs a foreign key discovery on all sources. A mixture of column profiles, data domain discovery and primary key discovery determines the required resources based on the value you configure for the Max Concurrent Profiling Jobs parameter.
The primary use case for enterprise discovery is to run a profile on relational sources. You might also need to consider a secondary use case for flat file sources and other non-relational sources.

Relational Source

When multiple data sources in enterprise discovery are relational sources, the Profiling Service Module pushes some parts of the column profile computation and data domain computation to the database. The Profiling Service Module performs all the computation for primary key discovery and foreign key discovery.

Consider the following hardware configuration for the relational sources:

Component	Requirement
CPU	Data domain discovery performs better if the Data Integration Service machine has multiple CPU cores or processing threads. If primary key discovery results in large intermediate results, primary key performs better with a Data Integration Service machine that has faster CPU speed and faster disk access.
Memory	Memory requirements for profile jobs except primary key discovery are minimal because the profile jobs do not buffer data. When you perform primary key discovery, the Data Integration Service machine requires some memory. You can add memory to speed up the pushdown operations for column profiles and data domain discovery.
Disk	If primary key discovery leads to large intermediate profile results, the profile jobs use some amount of temporary disk. If the Data Integration Service machine has faster access to two or more physical disks, the primary key profile jobs perform better. Column profiles and data domain discovery do not use temporary disk space.

Non-relational Sources

When you run enterprise discovery on non-relational sources, the Profiling Service Module performs all the profile computations. The CPU, memory, and temporary disk requirements depend on the specific profile function that enterprise discovery runs. The primary key discovery and foreign key discovery consumes temporary disk space.

Development Environment

The development deployment is a profile implementation scenario where you interact with the Profiling Service Module using the Analyst tool or Developer tool.

In development deployment, you can run profiling jobs, verify results, and drill down on the results to understand the content, structure, and quality of the source data. When you run enterprise discovery, the source of the profile jobs can vary from one database table to many database tables. You can use the profile results to build and troubleshoot scorecards.

The development deployment scenario requires understanding the types and numbers of concurrent profiling jobs that you need to run. You need to consider the average concurrency or maximum concurrency based on the requirements for the system. The expected mix of profile jobs determines the required amount of resources

CPU Cores Worksheet

As part of the development deployment, estimate the number of CPU cores for the nodes of the Data Integration Service machine required for a specific combination of profile jobs. The estimated CPU cores is the computed total number of CPU cores required for the nodes of the Data Integration Service machine.

You can use the following worksheet for all profile jobs except enterprise discovery. Enterprise discovery jobs runs at a lower priority than the standard profile jobs. Concurrency is the expected number of jobs of each profile type that run in parallel. Weight is the percentage of the profile jobs that run for each profile type.

In the Concurrency column in the following worksheet, enter the expected number of jobs of each profile type that run in parallel. The Average Cores column lists the values for each profile job type. In the Weight column, enter the percentage of the profile jobs that run for each profile type. Split the weight values for each profile job type so that the total of all the weight values in the worksheet is 100. Multiply the values in each column and update the "A x B x C" column.

Use the following worksheet to record values:

Profile Operation	Concurrency (A)	Average Cores (B)	Weight (C)	A x B x C
Scorecard (flat file)		2.3		
Scorecard (relational source)		2.0		
Column Profile (flat file)		2.3		
Column Profile (relational source)		2.0		
Data Domain Discovery (flat file)		2.3		
Data Domain Discovery (relational source)		2.0		
Primary Key Discovery		1.0		
Functional Dependency Discovery		1.0		
Foreign Key Discovery		2.0		
Overlap Discovery		2.0		

Calculation

To calculate the number of estimated CPU cores, add all the values in the "A x B x C" column and divide the total value by 100. You can then plan for the nodes of the Data Integration Service machine with the calculated number of CPU cores.

Memory Worksheet

When you estimate the amount of memory required for a combination of profile jobs, the recommended approach is to start with the minimal amount of recommended memory. The base temporary disk recommendation is 8 GB. You can then add more memory to the machine based on the types of profile jobs and the data that the profile jobs use.

Flat file sources perform better if the entire data source can fit into the buffer pools of the operating system. If the data sources are large, you can add additional memory.

Some of the profile operations require memory to cache the source flat files or for the profiling algorithms of column profile, key discovery, functional dependency discovery, and data domain discovery. Enter the expected number of parallel jobs for each profile job type in the "Concurrency" column. Concurrency is the expected number of jobs of each profile type that run in parallel. Multiply the values in the columns A and B for each row in the worksheet and update the "A x B" column.

Use the following worksheet to record the values:

Profile Operation	Concurrency (A)	Memory (B)	A x B
Scorecard (flat file)		2 GB	
Column Profile (flat file)		2 GB	
Data Domain Discovery (flat file)		2 GB	
Primary Key Discovery		1 GB	
Functional Dependency Discovery		1 GB	

The base memory recommendation of 8 GB meets the memory requirements of the following profile operations:

- Run a scorecard on a relational source
- Run a column profile on a relational source
- Perform data domain discovery on a relational source
- Perform foreign key discovery
- Perform overlap discovery

Calculation

Add all the values in the "A x B" column. You can then add the recommended base memory of 8 GB with the total value of the "A x B" column to compute the required memory.

Temporary Disk Space Worksheet

When you estimate the temporary disk space required for a specific mix of profile jobs, you can start with the minimal amount of recommended temporary disk space. The base temporary disk recommendation of 50 GB. You can then add more temporary disk memory based on the types of profile jobs and data that the profile jobs use. Performance improves when the additional disk memory is split across multiple physical disks.

Some of the profile operations require temporary disk space to cache the intermediate computation for flat files and profile types, such as column profile and data domain discovery.

In the Concurrency column in the following worksheet, enter the expected number of jobs for each profile type that run in parallel. The Temporary Disk column lists the values for each profile job type. Multiply the values in the Concurrency and Temporary Disk columns for each row and update the "A x B" column.

Use the following worksheet to record the values:

Profile Operation	Concurrency (A)	Temporary Disk (B)	A x B
Scorecard (flat file)		5 GB	
Column Profile (flat file)		10 GB	
Data Domain Discovery (flat file)		10 GB	
Primary Key Discovery		25 GB	
Functional Dependency Discovery		25 GB	

The base temporary disk recommendation of 50 GB meets the memory requirements of the following profile operations:

- Run a scorecard on a relational source
- Run a column profile on a relational source
- Perform data domain discovery on a relational source
- Perform foreign key discovery
- Perform overlap discovery

Calculation

Add all the values in the "A x B" column. Then, add the base temporary disk size of 50 GB with the total value of the "A x B" column to compute the required disk space for the development deployment.

Relational Data Sources Worksheet

When you run column profiles and scorecards on relational data sources, you need to consider the additional CPU and temporary tablespace requirements on the database. The profile queries require sorting the data that results in further memory requirements for buffer pools.

The following settings determine how effectively the profiling and scorecard tasks run on relational sources:

Maximum DB Connections

The number of database connections. The setting indicates a measure of maximum concurrency for the profile query, which consumes the maximum tablespace.

Maximum Column Size

The maximum size of any column that you can run a profile on. Double the length if the column is Unicode. You cannot run a profile on columns, such as large text columns, CLOBS, and BLOBS.

Maximum Number of Rows

The number of rows in the largest relational table you run the profile on.

Use the following worksheet to record the values:

Metric	Value
Maximum DB Connections	
Maximum Column Size	
Maximum Number of Rows	
Sorting pass*	2
*Sorting pass is the maximum number of passes that the profile queries perform to sort the data.	

Calculation

To compute the required relational tablespace that you can use in development deployment, multiply all the values in the Value column.

Profiling Warehouse Worksheet

You can use this worksheet to estimate the amount of tablespace required to store the profiling results. The worksheet contains two parts and includes multiple worksheets. The first part gathers tentative values on the expected data characteristics and result sizes. The second part uses the values in the first part to calculate the final estimate.

Worksheet - Part 1

Enter the values for each metric in the Value column of the following worksheet:

Metric	Description	Value
Average Value Length for a Profile (AVL)	The average length of a value in characters across all columns and all tables that you run a profile on.	
Average Value Length for a Scorecard (AVLsc)	The average length of a value in characters across all columns and all tables that you run a scorecard on.	
Average Cardinality for a Profile (AC)	The cardinality of a column is the number of unique values in each column expressed as a percentage. This is the average cardinality across all columns and all tables that you run a profile on.	
Average Cardinality for a Scorecard (ACsc)	The average cardinality across all columns and all tables that you run a scorecard on.	
Average Number of Columns Across All Tables (NC)	The average number of columns across all tables that you run a profile on.	
Average Number of Columns across All Scorecards (NCsc)	The average number of columns across all scorecards.	

Metric	Description	Value
Average Number of Tables or Schema (NT)	The average number of tables or schema. For overlap discovery and foreign key discovery, this parameter represents the number of tables.	
Max Value Frequencies (MVF)	The maximum number of value frequencies in the profiling warehouse for each column. The same number applies to both profiles and scorecards. The default value is 16000.	
Average Number of Primary Keys (APK)	The average number of primary keys for each primary key discovery profile. A general guideline is to set this parameter to 100.	
Average Number of Functional Dependencies (AFD)	The average number of functional dependencies for each functional dependency discovery profile. A general guideline is to set this parameter to 1000.	
Average Overlapping Pairs (AOPP)	The estimated average number of overlapping pairs between tables as a percentage. A general guideline is to set this parameter to .01 (1%).	
Average Foreign Keys for Each Table (FKT)	The estimated average number of foreign keys for each table. A general guideline is to set this parameter to 4.	

Worksheet - Part 2

After you complete part 1 of the worksheet, update the worksheets in part 2.

In the Value column of the following worksheets, enter the values based on the formula in the Calculation column. You can enter the values for specific metrics, such as the number of scorecards and number of column profiles, based on your development deployment.

Scorecard

Use the following worksheet to record the values for a scorecard:

Metric	Calculation	Value
Average scorecard size	`NCsc X [((2 X AVLsc) + 64) X (MVF X ACsc)]`	
Number of scorecards	-	
Average number of versions for each scorecard	-	

Calculation

To calculate the required tablespace to store the scorecard results in the profiling warehouse, multiply all the values in the Value column.

Column Profile

Use the following worksheet to record the values for a column profile:

Metric	Calculation	Value
Average profile size	NC X [((2 X AVL) + 64) X (MVF X AC)]	
Number of column profiles	-	

Calculation

To calculate the required tablespace to store the column profile results in the profiling warehouse, multiply both the values in the Value column.

Data Domain Discovery

Use the following worksheet to record the values for data domain discovery:

Metric	Calculation	Value
Average data domain size	NC X 254	
Number of data domain discovery profiles	-	

Calculation

To calculate the total tablespace required to store the data domain discovery results in the profiling warehouse, multiply both the values in the Value column.

Primary Key Discovery

Use the following worksheet to record the values for primary key discovery:

Metric	Calculation	Value
Average primary key result size	APK X [(128 + (32 X AVL))]	
Number of primary key discovery profiles	-	

Calculation

To calculate the total tablespace required to store the primary key discovery results in the profiling warehouse, multiply both the values in the Value column.

Functional Dependency Discovery

Use the following worksheet to record the values for functional dependency discovery:

Metric	Calculation	Value
Average functional dependency result size	APK X [(160 + (32 X AVL))]	
Number of functional dependency discovery profiles	-	

Calculation

To calculate the total tablespace required to store the functional dependency discovery results in the profiling warehouse, multiply both the values in the Value column.

Overlap Discovery

Use the following worksheet to record the values for overlap discovery:

Metric	Calculation	Value
Signatures	(NC X NT) X 3600	
Overlapping pairs	(NC X NT)square X AOP	

Calculation

To calculate the total tablespace required to store the overlap discovery results in the profiling warehouse, add both the values in the Value column.

Foreign Key Discovery

Use the following worksheet to record the values for foreign key discovery:

Metric	Calculation	Value
Signatures	(NC X NT) X 3600	
Foreign keys	(FKT X NT) X [(224 + (2048 X AVL))]	

Calculation

To calculate the total tablespace required to store the foreign key discovery results in the profiling warehouse, add both the values in the Value column.

Note: If you run overlap discovery and foreign key discovery on the same set of tables, both the jobs share the disk space for signature computation.

Final Calculation

To calculate the total tablespace required for all the profile operations, add the tablespace values for the following profile operations:

- Scorecard
- Column profile
- Data domain discovery
- Primary key discovery
- Functional dependency discovery
- Overlap discovery
- Foreign key discovery

Enterprise Discovery Resources

When you estimate the enterprise discovery resources, you need to optimize running each individual profile quickly with additional resources and the costs and limitations of scaling the additional resources.

Enterprise discovery also requires evaluating the relational sources where the column profile SQL queries run. If you have many tables, enterprise discovery can quickly affect the performance of a relational source.

Relational Resources

You must limit the number of profiling queries to the number of CPU cores of the database or fewer than the number of cores. In relational data source profiles, the Profiling Service Module runs a profile on every column in a separate query to the relational source. The optimal run of enterprise discovery depends on the size of the relational source.

If you have lesser resources than the recommended guidelines, the time that the Profiling Service Module takes to finish the enterprise discovery job increases.

The following table indicates the recommended number of cores based on the number of database tables in a typical enterprise discovery job:

Number of Tables	CPU Cores	Memory	Concurrent Jobs
Less than 100	4	8 GB	3
Between 100 and 500	8	16 GB	5
Between 500 and 1000	16	32 GB	10
Between 1000 and 2000	32	64 GB	20
More than 2000	>=64	>=128 GB	40

Data Integration Service Resources

You can estimate the Data Integration Service resources for enterprise discovery in two ways. The first approach is to decide the level of concurrency for the number of tables in the enterprise discovery job. Then, you can use the worksheets for the specific mix of profile jobs.

The second approach is to use the general recommendations in the following table:

The following table lists the recommended number of cores based on the number of database tables in a typical enterprise discovery job:

Number of Tables	CPU Cores	Memory	Temporary Disk / Spindles	Concurrent Jobs
Less than 200	4	8 GB	20 GB / 1	3
Between 200 and 1000	8	32 GB	80 GB / 2	5
Between 1000 and 2000	16	64 GB	160 GB / 3	10
More than 2000	>=32	>=128 GB	>= 320 GB / 4	20

When you plan for a deployment that has both single-table discovery and enterprise discovery jobs, choose the maximum configuration between the two use cases.

Profiling Warehouse Usage Worksheet

To estimate the profiling warehouse usage of enterprise data discovery, you need to consider the estimates you use for non-enterprise discovery jobs as well.

In the Value column of the following worksheets, enter the values based on the formula in the Calculation column. You can then estimate the profiling warehouse tablespace and use it along with the estimates for non-enterprise discovery jobs.

Column Profiles

Use the following worksheet to record the values for column profiles:

Metric	Calculation	Value
Average profile size	`NC* X [((2 X AVL*) + 64) X (MVF* X AC*)]`	
Total number of tables for all the enterprise discovery profiles	-	
*Metrics have the following definitions: - NC. The average number of columns across all tables that you run a profile on. - AVL. The average length of a value in characters across all columns and all tables that you run a profile on. - MVF. The maximum number of value frequencies that the profiling warehouse saves for each column. The default value is 16000. - AC. The cardinality of a column is the number of unique values in each column expressed as a percentage. This is the average cardinality across all columns and all tables that you run a profile on.		

Calculation

To calculate the required profiling warehouse tablespace for a column profile, multiply both the values in the Value column.

Data Domain Discovery

Use the following worksheet to record the values for data domain discovery:

Metric	Calculation	Value
Average data domain size	`NC* X 254`	
Total number of tables for all the enterprise discovery profiles	-	
*NC. The average number of columns across all tables that you run a profile on.		

Calculation

To calculate the required profiling warehouse tablespace for data domain discovery, multiply both the values in the Value column.

Primary Key Discovery

Use the following worksheet to record the values for primary key discovery:

Metric	Calculation	Value
Average key result size	APK* X [(128 + (32 X AVL*))]	
Total number of tables for all the enterprise discovery profiles	-	
*Metrics have the following definitions: - APK. The average number of primary keys for each Primary Key Discovery profile. A general guideline is to set this parameter to 100. - AVL. The average length of a value in characters across all columns and all tables that you run a profile on.		

Calculation

To calculate the required profiling warehouse tablespace for primary key discovery, multiply both the values in the Value column.

Foreign Key Discovery

Use the following worksheet to record the values for foreign key discovery:

Metric	Calculation	Value
Signatures	(NC* X NT*) X 3600	
Foreign keys	(FKT* X NT*) X [(224 + (2048 X AVL*)]	
*Metrics have the following definitions: - NC. The average number of columns across all tables that you run a profile on. - NT. The total number of tables for all enterprise discovery profiles. - FKT. The estimated average number of foreign keys for each table. - AVL. The average length of a value in characters across all columns and all tables that you run a profile on.		

Calculation

To calculate the required profiling warehouse tablespace for foreign key discovery, add both the values in the Value column.

Enterprise Discovery

Calculation

To calculate the total profiling warehouse tablespace for enterprise discovery, add the tablespace for non-enterprise discovery jobs to the total of the tablespace values for the following profile job types:

- Column profile
- Data domain discovery
- Primary key discovery
- Foreign key discovery

Production Environment

The production deployment is a profile implementation scenario where you implement the scorecards and reports built in the development phase on a production system. You can run scheduled jobs on the production system periodically to monitor data quality.

Scaling

Scaling for profiling indicates linear scaling. Doubling the size of the input doubles the time it takes to process the input. Similarly, doubling the processing power halves the time to process the same data.

You can consider two ways of scaling. The first way is how a fixed set of data scales when the hardware changes. You can either add more resources to a single machine or create a grid of machines. The second way is how a profile function performs when the number of columns or number of rows change for a fixed hardware configuration.

The worksheets that estimate the resources are based on general recommendations. Sometimes, the recommendations might not result in a single Informatica Data Integration Service node due to your budget constraints.

For example, a node with 128 cores might exceed your budget. However, the Profiling Service Module can scale across an Informatica grid. In this example, you can replace the 128 core machine with a grid of eight nodes with 16 cores each. The Profiling Service Module divides the workload evenly among these nodes. To realize the performance gains with more hardware, you can adjust the profiling configuration parameters appropriately.

The scaling strategy depends on the profile job type:

Column Profile

Column profile uses both small source and large source strategies. When the data source contains fewer than 100,000 rows, the profile runs on all the columns for all the rows at the same time. This small source strategy applies to both relational and flat file data sources. When the data source contains more than 100,000 rows, the profile makes successive passes over the data to compute the column aggregate statistics, such as value frequencies, patterns, and data types. Each pass processes one or more columns based on the data source.

Relational sources, where you can push down the column profiling query, process one column at a time. If the RDBMS can handle queries, you can double the number of connections to reduce the processing time by half. Flat file sources can batch process five columns at a time by default. To reduce the time required to run a column profile by half, you can double the number of cores. When you double the cores, you need to adjust the other hardware components. For example, you need to add memory for additional buffering while sorting and increase the number of temporary disks to increase the disk throughput.

Data Domain Discovery

The scaling strategy for data domain discovery is similar to that of column profiles. You can use the small source approach to run all the rows using the data domain discovery algorithm. The large source approach differs based on the data source type.

When you run data domain discovery on relational sources, the profile pushes the query processing for each column to the database. Therefore, the profile scales in the number of connections to the RDBMS, assuming the RDBMS can handle the queries. Data domain discovery runs batch jobs of data domains against columns with default values of 20 data domains for every 50 columns. Additional cores help the profile scale if the data source has a large number of columns and data domains. When you compare with column profiles, data domain discovery does not need additional memory or temporary disk spindles unless the data domain rule logic requires the additional resources.

Key and Functional Dependency Discovery

Both key discovery and functional dependency discovery algorithms usually operate on a small sample of the data source, typically less than 100,000 rows. The algorithms are not multithreaded and do not benefit from scaling the number of CPU cores.

Each algorithm reads from the sample and then makes successive passes over the data. Then, the Profiling Service Module writes large intermediate results to temporary files. You can add more memory and faster disks of up to two spindles to make the algorithms perform faster. An increase in the number of CPU cores does not affect scaling.

Overlap and Foreign Key Discovery

Both overlap and foreign key discovery algorithms operate in two steps. The first step computes the signatures of the data sources and second step computes either the overlaps or foreign keys.

The signature computation is CPU intensive and scales with the number of cores. Signature computation is the most time-consuming step. When you double the number of cores, signature computation takes half the time. Overlap discovery cannot scale by design because it is single threaded. Foreign key discovery partitions the job and scales with additional CPU cores.

Enterprise Discovery

Enterprise discovery scales to the extent that the column profile, data domain discovery, key discovery, and foreign key discovery scale. You can add more CPU cores for the enterprise discovery to scale because multiple profile functions can run in parallel. However, when you add more CPU cores, you must increase the number of disk spindles because the Profiling Service Module writes intermediate to the temporary disk.

Data Sizes and Profiling Functions Overview

Usually, the distribution and correlation of source data values control how the profile algorithms perform. The section contains general guidelines and therefore, specific profiling jobs might not conform to these guidelines.

Number of Rows

Linear scaling has the effect of "2X," which indicates that the profile algorithm takes double the time to run a profile if you double the number of rows in a data source. Non-linear scaling has an effect greater than "2X."

The following table summarizes how doubling the number of rows affects the performance of different profiling job types in terms of how the job types scale:

Profile Job Type	Effect	Description
Column profile	>= 2X	Sorting is the major component and scaling is not exactly 2X. However, for certain use cases, other components make it closer to linear scaling.
Data domain discovery	>= 2X	For flat file data sources, scaling is exactly 2X. However, similar to a column profile, scaling might be more than 2X.
Key discovery	~ 2X	Usually, scaling is linear. However, scaling is dependent on data and the complexity of relationships in the data.
Functional dependency discovery	~ 2X	Usually, scaling is linear. However, scaling is dependent on data and the complexity of relationships in the data.
Overlap discovery	Step 1. 2X Step 2. Constant	The first step of computing the signatures is directly proportional to the number of rows. The second step takes the same amount of time.

Profile Job Type	Effect	Description
Foreign key discovery	Step 1. 2X Step 2. Constant	The first step of computing the signatures is directly proportional to the number of rows. The second step takes the same amount of time.
Enterprise discovery	~ 2X	Enterprise discovery is a mixture of column profiling, data domain discovery, key discovery, and foreign key discovery. Enterprise discovery scales as the average of these functions.

Number of Columns

Usually, the profile algorithm takes double the time to run a profile if you double the number of columns in a data source. Linear scaling has the effect of "2X." Non-linear scaling has an effect greater than "2X."

The following table summarizes how doubling the number of columns affects the performance of different profiling job types in terms of how the profile job types scale.

Profile Job Type	Effect	Description
Column profile	2X	Columns are independent of each other.
Data domain discovery	2X	Columns are independent of each other.
Key discovery	2 to the power of X	Sometimes, key discovery must compare all combinations of columns to find the keys. The effect is exponential in the number of columns.
Functional dependency discovery	2 to the power of X	Sometimes, key discovery must compare all combinations of columns to find the keys. The effect is exponential in the number of columns.
Overlap discovery	Step 1. 2X Step 2. X to the power of 2	The first step of computing the signatures is directly proportional to the number of columns. The first step is linear scaling. The second step relies on comparing all columns with other columns and scales as the square of the number of columns. The second step runs faster than the first step.
Foreign key discovery	Step 1. 2X Step 2. Constant	The first step of computing the signatures is directly proportional to the number of columns. The first step is linear scaling. The second step relies on comparing all columns with other columns and scales as the square of the number of columns. The second step runs faster than the first step.
Enterprise discovery	~ 2X	Enterprise discovery is a mixture of column profiling, data domain discovery, key discovery, and foreign key discovery. Enterprise discovery scales as the average of these functions.

Scaling on Hadoop

Hadoop implements a grid in a different way than the Data Integration Service grid. Both systems achieve performance by scaling out in the number of nodes. Hadoop uses a distributed file system so that the computation engine can quickly access the data for each node.

When you run a profile, the data might not be on a particular grid node. Hadoop minimizes the number of network I/Os that the system needs when profiles run on a Data Integration Service grid. Unlike the Data Integration Service grid, only column profiles and data domain discovery run on Hadoop. The Profiling Service Module must run other data discovery processes in the native Data Integration Service mode. Additionally, Hadoop performs the computation of column profiles and data domain discovery on the Hadoop grid. Native sources do not support pushdown optimization.

Hadoop processes all data locally. If the data source is not in the Hadoop environment, the Profiling Service Module stages the data first in Hadoop and then runs a profile on it. This approach requires a fast network connection between the Hadoop cluster and database.

Profiling Warehouse Sizing Guidelines

Profile sizing is an iterative process in which you analyze profile performance, use guidelines to estimate and apply resources and parameters, and monitor and adjust the results as required.

Overview

The profiling warehouse database stores profiling and scorecard results. You create the profiling warehouse when you create the Data Integration Service.

You can create the profiling warehouse database in the following relational database systems:

- Oracle
- IBM DB2
- Microsoft SQL Server

More than one Data Integration Service can share the same profiling warehouse. The main resource for the profiling warehouse is disk space. The disk size calculations depend on the expected storage sizes of integers. Some databases, such as Oracle, use a compressed number format and require less disk size.

Hardware Guidelines

The profiling warehouse contains a set of tables into which the profile mappings write the results. SQL queries fetch results from the profiling warehouse and send to the client application. In addition, you can use profiling warehouse views to run external reports on the profiling warehouse data.

The usage pattern for profile runs vary from small sets of results to multiple tables when the Data Integration Service writes the results to the profiling warehouse. The largest of these results is the value frequency data, at most 16,000 rows for relational sources and 80,000 for flat file sources. The exception is the mappings that stage the source data. These mappings can store millions of rows at one time and can consume significant resources over a longer period of time.

CPU Cores

The queries to the profiling warehouse usually do not include aggregate and other large computations. Most of the computational costs account for storing and selecting data. The CPU usage might not be ongoing or significant due to the brevity of the SQL queries.

To determine the number of CPU cores, you can use the following procedure:

1. Choose the minimum number of CPU cores. To find out the minimum number of CPU cores, choose the larger value between 4 and the value you get when you divide Maximum Profile Execution Pool Size value by 4.
2. Add 1 to 2 cores to the minimum number of CPU cores if the dominant column profile use case is to stage the data.
3. To the result value in step 2, add 1 core for every 2 to 4 concurrent reports on the profiling warehouse.

The sum of all the values is the recommended number of CPU cores for the profiling warehouse.

Note: The recommended number of CPU cores applies to the maximum resource usage of the Profiling Service Module. The number of CPU cores can vary based on the usage of the Profiling Service Module.

Memory

The profiling warehouse performs better with additional memory as the database can cache more data.

The following table provides guidelines for the minimal amount of memory for the buffer cache:

Use Case	Memory
Single user or Laptop	0.5 GB
2 to 5 users	2 GB
5 to 10 users	4 GB
> 10 users	8 GB

Disk

The profiling warehouse performs better with the tablespace spread among more disks, which is a general recommendation that works for all use cases. Since most of the queries involve fetching or storing data instead of analysis, there is little need for temporary tablespace. Therefore, there are no specific recommendations for disk.

Network

The speed of data transfer between the Profiling Service Module and profiling warehouse affects the profile performance. Slow network speed can increase the time required to write the results to the profiling warehouse and retrieve the results back to the client application. You need to have a fast and dedicated network or switch between the profiling warehouse and the Profiling Service Module for better profile performance.

Mixed Databases

The database that hosts the profiling warehouse usually hosts other repositories, such as the Model Repository Service. You need to take the minimum configuration into account when you analyze the expected usage in the environment and plan accordingly.

Database Management

You need to periodically review and manage the profiling warehouse database growth. You can remove profile information that you no longer need and monitor or maintain the profiling warehouse tables.

The need for maintenance depends on different scenarios, such as short-term projects or when you no longer need the profile results. You can remove unused profile results and recover disk space used by the results so that you can reuse the database space for other purposes.

Purge

Purges profile and scorecard results from the profiling warehouse.

The infacmd ps Purge command uses the following syntax:

```
Purge
<-DomainName|-dn> domain_name
[<-Gateway|-hp> gateway_name]
[<-NodeName|-nn>] node_name
```

```
<-UserName|-un> user_name

<-Password|-pd> Password

[<-SecurityDomain|-sdn> security_domain]

<-MrsServiceName|-msn> MRS_name

<-DsServiceName|-dsn> data_integration_service_name

<-ObjectType|-ot> object_type

<-ObjectPathAndName|-opn> MRS_object_path

[<-RetainDays|-rd> results_retain_days]

[<-ProjectFolderPath|-pf> project_folder_path]

[<-ProfileName|-pt> profile_task_name]

[<-Recursive|-r> recursive]

[<-PurgeAllResults|-pa> purge_all_results]
```

The following table describes infacmd ps Purge options and arguments:

Option	Argument	Description
-DomainName -dn	domain_name	Required. The name of the Informatica domain. You can set the domain name with the -dn option or the environment variable INFA_DEFAULT_DOMAIN. If you set a domain name with both methods, the -dn option takes precedence.
-Gateway -hp	gateway_name	Optional if you run the command from the Informatica installation \bin directory. Required if you run the command from another location. The gateway node name. Use the following syntax: `[Domain_Host]:[HTTP_Port]`
-NodeName -nn	node_name	Required. The name of the node where the Data Integration Service runs.
-UserName -un	user_name	Required if the domain uses Native or LDAP authentication. User name to connect to the domain. You can set the user name with the -un option or the environment variable INFA_DEFAULT_DOMAIN_USER. If you set a user name with both methods, the -un option takes precedence. Optional if the domain uses Kerberos authentication. To run the command with single sign-on, do not set the user name. If you set the user name, the command runs without single sign-on.
-Password -pd	Password	Required if you specify the user name. Password for the user name. The password is case sensitive. You can set a password with the -pd option or the environment variable INFA_DEFAULT_DOMAIN_PASSWORD. If you set a password with both methods, the password set with the -pd option takes precedence.

Option	Argument	Description
-SecurityDomain -sdn	security_domain	Required if the domain uses LDAP authentication. Optional if the domain uses native authentication or Kerberos authentication. Name of the security domain to which the domain user belongs. You can set a security domain with the -sdn option or the environment variable INFA_DEFAULT_SECURITY_DOMAIN. If you set a security domain name with both methods, the -sdn option takes precedence. The security domain name is case sensitive. If the domain uses native or LDAP authentication, the default is Native. If the domain uses Kerberos authentication, the default is the LDAP security domain created during installation. The name of the security domain is the same as the user realm specified during installation.
-MrsServiceName -msn	MRS_name	Required. The Model Repository Service name.
-DsServiceName -dsn	data_integration_service_name	Required. The Data Integration Service name
-ObjectType -ot	-	Required. Enter profile or scorecard.
-ObjectPathAndName -opn *	MRS_object_path	Optional. Do not use with ProjectFolderPath or Recursive. The path to the profile or scorecard in the Model repository. Use the following syntax: `ProjectName/FolderName/.../{SubFolder_Name/ObjectName|ProjectName/ObjectName}`
-RetainDays -rd	results_retain_days	Optional. The number of days that the profiling warehouse stores profile or scorecard results before it purges the results.
-ProjectFolderPath -pf *	project_folder_path	Optional. Do not use with ObjectPathAndName or ProfileTaskName. The names of the project and folder where the profile or scorecard is stored. Use the following syntax: `ProjectName/FolderName`
-ProfileName -pt *	profile_task_name	Optional. The name of the profile task that you want to purge. If you specified the ProjectFolderPath, you do not need to specify this option because the ProjectFolderPath includes the name of the enterprise discovery profile that contains the profile task.
-Recursive -r	recursive	Optional. Do not use with ObjectPathAndName. Applies the command to objects in the folder that you specify and its subfolders.
-PurgeAllResults -pa	purge_all_results	Optional. Set this option to purge all results for the profile or scorecard object. Use with the -recursive option to apply the command to profile and scorecard results in the folder that you specify and its subfolders.
* To run the command, you need to specify ObjectPathAndName or ProjectFolderPath or ProfileTaskName.		

Tablespace Recovery

As part of the regular profile operations, the Data Integration Service writes profile results to the profiling warehouse and deletes results from the profiling warehouse. The indexes and base tables can become fragmented over a period of time. You need to reclaim the unused disk space, especially for Index Organized Tables in Oracle database.

Most of the profiling warehouse tables contain relatively small amount of data and you do not need to recover the tablespace and index space.

The following tables store large amounts of profile data and deleting the tables can leave the tables fragmented:

Name	Description
IDP_FIELD_VERBOSE_SMRY_DATA	Stores the value frequencies
IDP_VERBOSE_FIELD_DTL_RES	Stores the staged data

When you perform the tablespace recovery, ensure that no user runs a profile task. After you recover the data, update the database statistics to reflect the changed structure.

IBM DB2

The recommendation is to shut down the Data Integration Service when you reorganize the tables and indexes.

To recover the database for a table, run the following command:

```
 REORG TABLE <TABLE NAME>

REORG INDEXES ALL FOR TABLE <TABLE NAME> ALLOW WRITE ACCESS CLEANUP ONLY ALL
```

Oracle

You can rebuild Index Organized Tables in Oracle. This action reclaims unused fragments inside the index and applies to the IDP_FIELD_VERBOSE_SMRY_DATA and IDP_FIELD_VERBOSE_SMRY_DATA profiling warehouse tables.

To recover the database for a table, run the following command:

```
ALTER TABLE <Table Name> MOVE ONLINE
```

Microsoft SQL Server

Microsoft SQL Server reclaims unused space back into the tablespace and compacts indexes when rows are deleted. You do not need to maintain the database.

Database Statistics

Update the database statistics to allow the database to quickly run the queries on the profiling warehouse.

IBM DB2

To update the statistics, run the following command:

```
RUNSTATS ON TABLE <TABLE NAME> WITH DISTRIBUTION AND DETAILED INDEXES ALL
```

Oracle

By default, Oracle gathers database statistics and therefore, you do not need to perform any action. For more information, refer the documentation on Oracle `DBMS_STATS` command.

Microsoft SQL Server

By default, Microsoft SQL Server gathers statistics and therefore, no action is required. To update the statistics more frequently than the default recommended option, refer the documentation on SQL Server UPDATE STATISTICS command.

Profiling Warehouse Guidelines for Column Profiling

The profiling warehouse stores profiling results. More than one Profiling Service Module may point to the same profiling warehouse. The main resource for the profiling warehouse is disk space.

Column profiling stores the following types of results in the profiling warehouse:

- Statistical and bookkeeping data
- Value frequencies
- Staged data

Statistical and Bookkeeping Data Guidelines

Each column contains a set of statistics, such as the minimum and maximum values. It also contains a set of tables that store bookkeeping data, such as profile ID. These take up very little space and you can exclude them from disk space calculations. Consider the disk requirement to be effectively zero.

Value Frequency Calculation Guidelines

Value frequencies are a key element in profile results. Value frequencies list the unique values in a column along with a count of the occurrences of each value.

Columns with low cardinality have few values. Columns with high cardinality can have millions of values. The Profiling Service Module limits the number of unique values it identifies to 16,000 by default. You can change this value.

Use this formula to calculate disk size requirements:

```
Number of columns X number of values X (average value size + 64)
```

where

- Number of columns is the sum of columns and virtual columns in the profile run.
- Number of values is the number of unique values. If you do not use the default of 16,000, use the average number of values in each column.
- Average value size includes Unicode encoding of characters.
- The value 64 bytes indicates 8 bytes for the frequency and 56 bytes for the key.

Staged Data Guidelines

Staged data or cached data is a copy of the source data that the Data Integration Service uses for drill-down operations. The staged data might use large amount of disk space based on the type of data source.

Use the following formula to calculate the disk size requirements for staged data:

```
number of rows X number of columns X (average value size + 24)
```

Note: The cache key size is 24.

Sum the results of this calculation for all cached tables.

The following table lists the required disk space for an 80 column table that has 100 million rows with an equal mixture of high cardinality columns and low cardinality columns:

Type of Data	Disk Space
Value frequency data	50 MB
Cached data	327,826 MB
Total	327,876 MB

The Data Integration Service stages the source data when you choose to cache data. If you do not cache data for drilldown, the disk space requirement is significantly less. All profiles store the value frequencies.

Memory and CPU Needs

The profiling warehouse does not have significant memory requirements.

Memory	The queries run by the Profiling Service Module do not use significant amounts of memory. Use the manufacturer's recommendations based on the table sizes.
CPU	You can use the following CPU recommendations for the profiling warehouse: - 1 CPU for each concurrent profile job. This applies to each relational database or flat file profile job, not to each profile mapping. - 2 CPUs for each concurrent profile job if the data is cached.

Profiling Warehouse Guidelines for Data Domain Discovery

When you run data domain discovery, the Data Integration Service stores statistical and book-keeping data in the profiling warehouse in addition to the data domain discovery results.

Statistical and Bookkeeping Data for Data Domain Discovery

Each data domain discovery run stores a copy of the data domain names and the associated groups. In addition, each column contains a set book-keeping data stored in their own tables, such as profile ID and sequence numbers. This data takes up very little space and you can exclude it from disk space calculations.

Consider the disk requirement to be effectively zero.

Data Domain Discovery Result Calculation Guidelines

Each column stores the confidence computation and other metadata for each data domain. The required disk size is not significantly large. However, the disk space requirement can add up if there are many data domains.

Use the following formula to calculate the disk size requirements:

```
2 X number of columns X number of data domains X 254
```

where

- The value 2 indicates the number of rules for each domain except the column name rule and data membership rule.
- Number of columns is the sum of columns and virtual columns in the data domain discovery run.
- Number of data domains is the number of data domains in the data domain discovery run.

- The value 254 indicates the size of the statistics and keys.

Profiling Warehouse Guidelines for Key and Functional Dependency Discovery

The disk space for key and functional dependency discovery depends on the number of inferred keys, functional dependencies, and their dependency violations. These items take up large space in the profiling warehouse if you set a large number for key and functional dependency discovery.

You can use the following formulas to compute the disk space. If you set the confidence parameter to 100%, the profiling warehouse does not store violating rows and you can exclude the computation for key violation.

Keys

```
Number of Keys X Average Number of Key Columns X 32 + Number of Keys X ( 32 + (2 Bytes
for Each Character X Average Column Size ) X Average Number of Key Columns X Average
Number of Violating Rows
```

Where

- Number of Keys is the number of inferred keys.
- Average Number of Key Columns is the average number of columns in the key.
- The value 32 is the number of bytes used to store one column in the key.
- Average Column Size is the average number of characters in the columns if the numbers and dates are converted to the String datatype.
- The value 2 Bytes for Each Character is the typical number of bytes used for a single Unicode character.
- Average Number of Violating Rows is the average number of rows that violate the key.

Functional Dependency

```
Number of FDs X (Average Number of LHS Columns + 1) X 32 + Number of FDs X (32 + (2
Bytes for Each Character X Average Column Size ) X (Average Number of LHS Columns ) X
Average Number of Violating Rows
```

Where

- Number of FDs is the number of inferred functional dependencies.
- Average Number of LHS Columns is the average number of columns in the determinant of the functional dependency. Add one column for the dependent column.
- The value 32 is the number of bytes used to store one column in the functional dependency.
- Average Column Size is the average number of characters in the columns if the numbers and dates are converted to the String datatype.
- The value 2 Bytes for Each Character is the typical number of bytes used for a single Unicode character.
- Average Number of Violating Rows is the average number of rows that violate the functional dependency.

Profiling Warehouse Guidelines for Foreign Key and Overlap Discovery

The disk space for foreign key discovery and overlap discovery is dependent on the number inferred foreign keys and overlapping column pairs. These items take up large space in the profiling warehouse if you set a large number for foreign key discovery and overlap discovery.

You can use the following formulas to compute the disk space. The Profiling Service Module computes column signatures one time for foreign key discovery and overlap discovery.

Signatures

```
Number of Columns in Schema X 3600
```

Where

- Number of Columns in Schema is the total number of columns in the profile model. After the Profiling Service Module generates the column signature for a profile task, subsequent profile tasks reuse the signature.
- 3600 is the amount of space required to store the signatures for one column.

Foreign Keys

```
Number of Foreign Keys X 2 X (Average Number of Key Columns) X 32 + Number Of Foreign Keys X ( 32 + (2 Bytes for Each Character X Average Column Size ) X Average Number of Key Columns X Average Number of Violating Rows
```

Where

- Number of Foreign Keys is the number of inferred foreign keys.
- Average Number of Key Columns is the average number of columns in the primary or foreign key.
- The value 2 is the multiplier to get the total number of columns for the foreign key.
- The value 32 is the number of bytes required to store one column in the key.
- Average Column Size is the average number of characters in the columns if the numbers and dates are converted to the String datatype.
- The value 2 Bytes for Each Character is the typical number of bytes for a single Unicode character.
- Average Number of Violating Rows is the average number of rows that violate the foreign key either in the parent table or child table.

Overlap Discovery

```
Number Of Overlap Pairs X 2 X 32
```

Where

- Number of Overlap Pairs is the number of inferred overlap pairs.
- The value 2 is the number of columns in the pair.
- The value 32 is the number of bytes required to store one column in the overlap pair.

Profiling Warehouse Guidelines for Enterprise Discovery

Enterprise discovery automates the profiling functions it runs. The required profiling warehouse resources are the sum of the resources for the individual functions.

Enterprise discovery automates the following profile functions:

- Column profile with no data caching
- Data domain discovery
- Primary key discovery
- Foreign key discovery

In the following worksheet, use the formula in the Calculation column to enter the metric values in the Value column.

Use the following worksheet to record the values:

Metric	Calculation	Value
Book-keeping overhead	-	100, 000
Column profile	`Average number columns for each table X number of values X (average value size + 64)`	
Data domain discovery	`2 X number of columns X number of domains X 254`	
Primary key discovery	`Average number of keys for each table X average number of key columns X 32 + number of keys X ( 32 + (2 bytes for each character X average column size ) X average number of key columns X average number of violating rows`	
Signatures	`Average number columns for each table X 3600`	
Foreign key discovery	`Average number of foreign keys for each table x 2 X (average number of key columns) X 32 + number of foreign keys X ( 32 + (2 bytes for each character X average column size ) X average number of key columns X average number of violating rows`	

Calculation

To estimate the profiling warehouse resources based on average size estimates, perform the following steps:

1. Add the values for book-keeping overhead, column profile, data domain discovery, primary key discovery, signatures, and foreign key discovery.

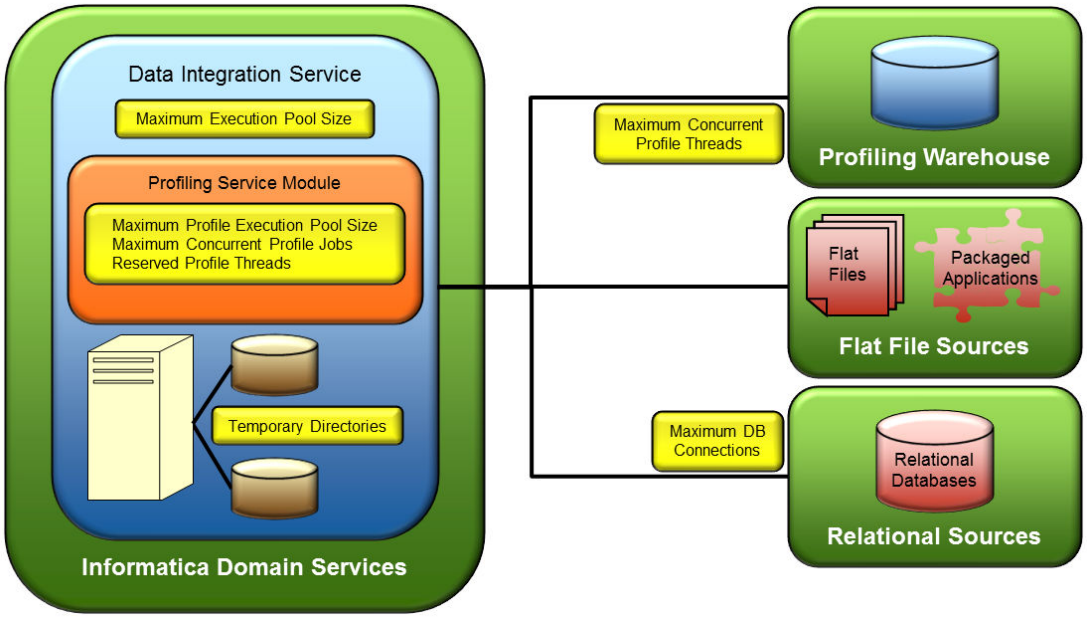
2. Multiply the result value in step 1 with the total number of tables in enterprise discovery.

Profiling Performance Tuning Parameters for Data Integration Service

The Data integration Service has a set of parameters specific to the Profiling Service Module that controls the performance of a profiling job. You must configure the parameters for each deployment.

Overview

The following figure shows the important parameters for the Data Integration Service and the components of the profiling and discovery installation that the parameters have an impact on:



You can use the following sections in the Administrator tool to configure the profiling warehouse database and advanced profiling parameters for the Data Integration Service:

- Profiling warehouse database properties on the Properties tab.
- Advanced profiling properties on the Properties tab.
- Execution options on the Processes tab.

Profiling Warehouse Database Properties

The profiling warehouse database properties apply to the Profiling Service Module across the deployment.

You can set the following parameters:

Profiling Warehouse Database

Connection name to the profiling warehouse database. In addition to the profile results, the profiling warehouse holds the persisted profile job queue. Verify that no profile job runs when you change the connection name. Otherwise, the profile jobs might stop running because the profile jobs run on the Data Integration Service where the Profiling Service Module submitted the profile jobs.

You set the default value when you create the instance.

Maximum Ranks

The number of minimum values and maximum values to display for a profile based on the datatype of the column. Ranks are useful to understand the range that the values of a column might take and whether the column has pseudonull values. You can retain the default value.

Default is 5.

Maximum Patterns

The maximum number of patterns that each column stores. Sometimes, it is important to store as many patterns as possible. You can set the **Maximum Patterns** parameter to a large number, such as 10,000, and adjust the **Pattern Threshold Percentage** parameter to **.01**. Setting a high value for this parameter has negligible impact on performance.

Default is 10.

Maximum Profile Execution Pool Size

The number of profile mappings that the Profiling Service Module can run concurrently when the Data Integration Service runs on a single node or on a grid. The pool size is dependent on the aggregate processing capacity of the Data Integration Service, which you specify for each node on the **Processes** tab of the Administrator tool. The pool size cannot be greater than the sum of the processing capacity of all nodes.

When you plan for a deployment, consider the following types of threads:

- Threads used for profile tasks.
- Reserved threads for drill-down tasks on the source data and similar, quick real-time profile tasks.
- Threads for all other nonprofiling purposes, such as SQL endpoints, preview, and deployed mappings.

It is important to understand the mixture of mappings and profile jobs so that you can configure the Maximum Execution Pool Size parameter. For optimal performance, verify that the total number of threads in the three categories adds up to the aggregate total of the Maximum Execution Pool Size parameter.

Default is 10.

Maximum DB Connections

The number of parallel queries across all the profiling jobs for a relational source. If two profile jobs run on the same database, each profile job gets half the number of connections. If two profile jobs run on different databases, each profile job gets the maximum number of connections.

The Profiling Service Module verifies the connection name to recognize the different databases. If you have different connection names, the Profiling Service Module considers the connection names as different databases. Therefore, you might not want to create database aliases with two different connection names.

All databases use the same number of connections. Therefore, if you run a profile that runs on two databases with different performance characteristics, consider the database with the lowest concurrent profile run requests to configure this parameter.

Default is 5.

Profile Results Export Path

The default path where the Profiling Service Module stores the exported objects, such as Microsoft Excel spreadsheets and DDL files.

The default value is `<Installation Directory>/tomcat/bin/ProfileExport`.

Advanced Profiling Properties

The advanced profiling properties apply to a single Data Integration Service node. You must configure the parameters for each node in the Data Integration Service.

You can configure the following advanced profiling properties:

Pattern Threshold Percent

The minimum percentage of rows matching up to two decimal places for a pattern to appear in the results.

Default is 5.00.

Maximum # Value Frequency Pairs

The maximum number of value frequency pairs stored in the profiling warehouse. This parameter does not control whether the Profiling Service Module computes all the value frequency pairs and the basic characteristics of a column profile run. However, an increase in the parameter value adds additional time to write more value frequency pairs to the profiling warehouse.

Default is 16,000.

Maximum String Length

The maximum length of a string that the Profiling Service Module mappings process internally. The default is set to the maximum value of 255. If you decrease the value, the Data Integration Service truncates the value. Decreased string lengths can have a minor impact on the amount of tablespace required for the profiling warehouse and negligible impact on the overall performance.

Default is 255.

Maximum Numeric Precision

The maximum precision, which is the number of significant digits in the number, for numeric decimal datatypes. If you set a low value for this parameter, the Data Integration Service might process additional numeric datatypes as strings instead of numbers.

Default is 38.

Maximum Concurrent Profile Jobs

The number of profile jobs that can run in parallel, even if there are more threads available to run mappings. You can use the parameter to control the number of concurrent jobs. Use the parameter to optimize the Profiling Service Module resources so that the Profiling Service Module resources do not affect the resource usage for the Data Integration Service.

When you run a column profile on a relational source, the Maximum DB Connections parameter determines the number of mappings that the Profiling Service Module uses. The Profiling Service Module uses one mapping each for the other profiling jobs. When you run a column profile on flat file sources or relational sources, the Maximum Concurrent Profile Threads parameter determines the number of mappings that the Profiling Service Module uses.

You can configure the Maximum Concurrent Profile Jobs based on the capabilities and other uses of the nodes that you run the Profiling Service Module on.

Default is 5.

Maximum Concurrent Columns

The number of columns that a mapping runs in parallel. The default value of 5 is optimal for most of the profiling use cases. You can increase the default value for columns with cardinality lower than the average value. Decrease the default value for columns with cardinality higher than the average value. You might also want to decrease this value is when you consistently run profiles on large source files where temporary disk space is low.

Default is 5.

Maximum Concurrent Profile Threads

The number of mappings that run in parallel when you run a column profile on a flat file data source or relational data source. Each mapping simultaneously runs a profile on a number of columns equal to the value you set for the Maximum Concurrent Columns parameter. If you increase this parameter value, the Profiling Service Module simultaneously runs the profile on more number of columns and reduces the overall time for the profile run.

Default is 1.

Maximum Column Heap Size

The cache size for each column profile mapping for flat files. You can increase this value to prevent the Data Integration Service from writing some parts of the intermediate profile results to temporary disk. However, this effect does not apply to large data sources. The default setting is optimal for most of the profiling use cases.

Default is 64.

Reserved Profile Threads

The number of threads that the Data Integration Service reserves to perform drill-down operations. The parameter ensures that a thread is always available for a drill-down operation, which is a quick and real-time operation. If enterprise discovery is a large part of the profile jobs, you can increase the parameter value. You can also increase the parameter value if multiple users might perform drill-down operations when the Profiling Service Module runs a profile.

Default is 1.

Data Integration Service Parameters

You need to configure the Temporary Directories and Maximum Execution Pool Size parameters for the Data Integration Service. You can configure many parameters, such as Reserved Profile Threads and Maximum DB Connections, that apply to the Profiling Service Module. Before you use the parameter recommendations, verify that you have identified a node or grid and the requirement is to configure the node or grid optimally to run profiles.

Profiling Service Module and Performance

When you configure the Data Integration Service parameters specific to Profiling Service Module, you must choose whether to optimize the Profiling Service Module for the average usage or peak usage.

If you optimize the Profiling Service Module for average usage, each profile job gets more resources. When the number of profile jobs exceeds the average threshold, the persistent queue queues up the remaining profile jobs until one or more of the jobs that run finish. When you optimize the Profiling Service Module for peak usage, more profile jobs run concurrently with reduced amount of resources. Therefore, the profile jobs might take longer time to run. The peak usage configuration increases the throughput. The Data Integration Service runs profile jobs with higher priority before the lower priority jobs irrespective of the Profiling Service Module optimization.

Temporary Directories

Location of temporary directories for the Data Integration Service process on the node. This parameter has an impact on the Data Integration Service machine.

Reserved Profiling Threads

Number of threads of the Maximum Execution Pool Size parameter that are for priority requests. This parameter has an impact on the Profiling Service Module. The reserved profiling threads are the threads that the Profiling Service Module keeps in reserve to perform drill-down tasks and profile export tasks. The Profiling Service Module uses these reserved

threads when it uses all the other threads for profile jobs. If there are no profiling jobs to run, the Profiling Service Module does not use the reserved threads.

The Profiling Service Module uses the nonprofile threads to perform a data preview job because the data preview job is a common service for all tools. The Mapping Service Module runs the service. The Data Integration Service considers the preview jobs as part of the total number of threads and not the Profiling Service Module.

Maximum Concurrent Profile Jobs

The maximum number of concurrent profile threads to run a profile on flat files. If you do not set this parameter, the Profiling Service Module determines the best number based on the set of running profile jobs and environment factors. This parameter has an impact on the Profiling Service Module. You can consider the default value of 5 for the Maximum Concurrent Profile Jobs parameter as the minimum value. To increase performance, you can increase the parameter value to match the number of CPU cores in the Data Integration Service machine. If you use a grid configuration, you can aggregate the number of cores for all the nodes in the grid to determine the total number of CPU cores.

Maximum DB Connections

The Maximum DB Connections parameter controls the number of parallel queries across all profiling jobs for a relational source. You can set the parameter to a value that the source database with the lowest concurrent profile run requests. The Profiling Service Module then does not impact the performance of any of the databases that you run a profile on. This parameter has an impact on the relational database sources that you run a profile on.

In the Database Server column of the following worksheet, enter the names of the databases. You can then enter the number of cores for each database server in the Cores column. The Max % Utilization is the percentage of the maximum use of the CPU cores in the database server. If the database has multiple cores, such as 64 or 128, you can set the Max % Utilization value to 90%. Otherwise, you can set the value to 80%. A guideline to increase performance is to verify that the temporary tablespace resides on a separate hard drive.

Enter the percentage values for each database server in the Max % Utilization column. Multiply the values in the Cores and Max % Utilization columns for each row and update the Cores x Max % Utilization column.

Use the following worksheet to record the values:

Database Server	Cores	Max % Utilization	Cores x Max % Utilization

Set the Maximum DB Connections parameter to the minimum value of all the values in the Cores x Max % Utilization column.

Maximum Execution Pool Size

The Maximum Execution Pool Size parameter determines the maximum number of requests that the Data Integration Service can run concurrently. Requests include data previews, mappings, and profiling jobs. This parameter has an impact on the Data Integration Service.

To calculate the total number of DTM threads for the Data Integration Service, consider all the threads for the Profiling Service Module and mappings from the other plugins and clients.

After you implement an initial configuration, you must monitor the Data Integration Service instance nodes. You need to verify that the Data Integration Service performs as expected and that you do not need to make additional adjustments. You must also perform long-term monitoring to verify that the requirements do not change over time. If the requirements change, you might need to make additional adjustments to the settings. You might also need to change other Profiling Service Module parameters that control how profiling jobs run on the Data Integration Service.

Use the following worksheet to calculate the total number of threads for the Data Integration Service. Enter the values in the Estimated Maximum Threads column for each row.

Use the following worksheet to record the values:

Activity	Estimated Maximum Threads
Profile run	
Preview	
Unplanned mappings	
Deployed mappings	
SQL Endpoint	

Calculation

To calculate the Maximum Execution Pool Size parameter value, add the values in all the rows.

Maximum Profile Execution Pool Size

The Maximum Profile Execution Pool Size parameter determines the total number DTM threads that the Data Integration Service uses to run profiles. This parameter has an impact on the Profiling Service Module. You can calculate the number of DTM threads allocated to the Profiling Service Module based on the expected mix of profile jobs and the number of CPU cores.

To calculate the expected mix pf profile jobs, divide the mix of profile jobs into two groups before you calculate the number of DTM threads. The first group can include the profile jobs that the Data Integration Service cannot transfer to the relational source. The second group can include the profile jobs that the Data Integration Service can transfer to the relational source. You can use the first group proportion to compute the number of threads for the jobs that the system cannot transfer to the relational sources. You can then add this number to the Maximum DB Connections value to compute the final estimate for the number of DTM threads. Each profiling type uses a different number of CPU cores. Use different weights for different profile job types.

In the following worksheet, enter the values in the A and B columns as required. Multiply the values in the A and B columns for each row and then update the A x B column.

Use the following worksheet to record the values for profile operations that the Data Integration Service cannot transfer to the relational sources:

Profile Operation	Estimated Number for Each 100 Runs (A)	Factor (B)	A x B
Scorecard on a relational source Column profile on a relational source			
Scorecard on a flat file source Column profile on a flat file Source		0.4	
Data domain discovery		0.5	

Profile Operation	Estimated Number for Each 100 Runs (A)	Factor (B)	A x B
Key discovery		1.0	
Functional dependency discovery		1.0	
Overlap discovery		1.0	
Foreign key discovery		1.0	

Calculation

Add all the values in the A x B column and divide the total value by 100. You can then multiply the result with the number of CPU cores for the nodes in the Data Integration Service machine and the recommended load factor of 2.5. The final value is the number of DTM threads for profile jobs that the Data Integration Service cannot transfer to the relational source.

Final Calculation

The Max DB Connections parameter impacts the profile jobs that the Data Integration Service can transfer to the relational sources. To calculate the Maximum Profile Execution Pool Size value, add the following values:

- The number of DTM threads for profile jobs that the Data Integration Service cannot transfer to the relational source
- The Max DB Connections parameter value
- The Reserved Profiling Threads parameter value

Maximum Concurrent Profile Threads

The Maximum Concurrent Profile Threads parameter determines the number of mappings that run in parallel when you run a column profile on a flat file data source or relational data source. This parameter has an impact on the profiling warehouse.

You can retain the default value of 1 for the parameter because each column profile job that runs on a flat file data source consumes approximately 2.3 CPU cores.

You can consider increasing the parameter value in the following scenarios:

- There are not many column profile jobs on flat file sources and the Data Integration Service node has many CPU cores.
- If you run the column profile jobs on a grid with many nodes that has an aggregate CPU core count and CPU usage.
- Each column profile needs to run as fast as possible when the resources are available.

In the Value column of the following worksheet, enter the number of CPU cores that the Data Integration Service can use to run a column profile. You can then enter a value for the expected level of concurrency.

Use the following worksheet to record the values:

Metric	Value
Number of CPU cores available for column profiling	
Expected level of concurrency that you want to configure	

Calculation

To calculate the number of CPU cores available for each column profile job, divide the number of CPU cores available for column profiling value with the level of concurrency value. To calculate the Maximum Concurrent Profile Threads value, divide the number of CPU cores available for each column profile job with 2.3 and round off the value. The number of CPU cores that the Data Integration Service uses for each column profiling thread is 2.3.

Data Integration Service Parameters for the Blaze Engine

You can run profiles on the Blaze engine in the Hadoop environment after you configure the Data Integration Service parameters specific to the Blaze engine.

The following table lists the custom properties for the Data Integration Service and the values you need to configure to run the profiles on the Blaze engine:

Parameter	Value
ExecutionContextOptions.Blaze.AllowSortedPartitionMergeTx	false
ExecutionContextOptions.GridExecutor.EnableMapSideAgg	true
ExecutionContextOptions.Partitioning.PreferredPartitionSize	268435456
ExecutionContextOptions.RAPartitioning.AutoPartitionResult	true
ExecutionContextOptions.Optimizer.PushdownType	none
ExecutionContextOptions.Optimizer.NativeExpressionTx	false
ExecutionContextOptions.Optimizer.MaxMappingPorts	-1

Sizing Guidelines for Profile Deployments

The sizing guidelines help you to determine the system resources for the Profiling Service Module and Data Integration Service for different profile deployments.

The examples include multiple profile deployment scenarios based on the type of profile deployment and size of the organization. Each deployment scenario describes the typical setup definition and recommended configuration settings for the sizing parameters.

Consider the following examples:

Individual Profile Sizing Example

Few individual users in a department of a small organization. Profile activities include running a column profile on flat files and relational databases with up to 10 million rows.

Departmental Profile Sizing Example

A competency center in a medium to large organization or a small department in a large organization. Profile activities include running profiles on medium to large data sources with 50 million to 1 billion rows.

Corporate Profile Sizing Example

Large scale organization with data stewards that run profiles and scorecards daily. Users include 40 to 60 data stewards that use the Analyst tool and one to three power users that use the Developer tool. Profile operation is on a single, high-end node connected to the database servers.

Enterprise Profile Sizing Example

Large scale organization with data stewards performing profile and scorecard runs daily. Users include 30 to 40 data analysts that use the Analyst tool and 5 to 10 data architects that use the Developer tool. Profile operation is on a grid configuration.

Individual Profile Sizing Example

This use case summarizes the initiative of a few users in a department of a small organization to understand the quality of the data assets that they control. The users mostly run column profiles on flat files and database tables that have up to 10 million rows. The setup environment has a shared Windows Server on which the users run profiles.

The following table describes the setup environment:

Setup Component	Description
Users	One or two data analysts use the Analyst tool.
Hardware	1 node, 4 cores, 8 GB, 4 x 2 TB disks, and a 50% shared Windows Server.
Data	- Flat files with 10 million rows. - Relational tables with 10 million rows on an 8 core, 32 GB Linux machine.
Profile type	Column profile and data domain discovery.
Profiling warehouse	Set up on the same server.
Model Repository Service	Set up on the same server.
Analyst Service	Set up on the same server.

The following table summarizes the recommended configuration parameters for the setup environment:

Parameter	Value
Maximum Execution Pool Size	>= 10
Maximum Profile Execution Pool Size	4
Maximum Concurrent Profile Jobs	2

Parameter	Value
Maximum DB Connections	2
Maximum Concurrent Profile Threads	1

Analysis

In this basic profile scenario, the configuration depends on the concurrent profile runs that users perform on up to three smaller flat files. Column profiles on flat files with 10 million rows run quickly. However, the expected peak CPU usage can exceed the power of the Profiling Service Module hardware for short periods of time.

To continuously run a profile on three or more flat files simultaneously, you can scale down the Maximum Concurrent Profile Jobs parameter to 2. This parameter value ensures adequate throughput for all uses of the node. In this use case, one or two data analysts use the Profiling Service Module. Therefore, the users can communicate with each other to ensure that machine overload does not occur at the higher setting of three concurrent profile jobs.

The configuration also depends on the production database. You can set the Maximum DB Connections parameter value to 2 so that the two concurrent profiles do not affect the performance of the relational database. If you run all the three relational profiles, the Maximum Profile Execution Pool Size parameter limits the number of concurrent profile queries to 10.

To avoid performance issues with the production database, you can reduce the Maximum Profile Execution Pool Size parameter value to 4. An example of this scenario is where the database cannot handle more than four concurrent profile queries. When you set the Maximum Profile Execution Pool Size parameter value to 4, the column profile jobs continue to run and limits the number of concurrent queries to the database.

You can set the Maximum Execution Pool Size parameter to a value higher than the Maximum Profile Execution Pool Size parameter value to enable quicker drill-down operations and previews.

Departmental Profile Sizing Example

This use case describes a small competency center in a medium to large size organization or a department in a larger organization. The data analysts need to understand and monitor current data for data quality purposes. Data analysts need to help developers in the front-end analysis of data-migration projects.

Most of the data sources range from medium to large files with 50 million to 1 billion rows. A few of the data sources can have up to 10 billion rows. A single dedicated node runs the profiles.

The following table describes the setup environment:

Setup Component	Description
Users	Five data analysts use the Analyst tool and three developers use the Developer tool.
Hardware	1 node, 16 cores, 32 GB, 6 x 2 TB disks, and Linux.
Data	- Flat files with up to 10 billion rows. - Relational tables with up to 10 billion rows on multiple large servers.
Profile type	Column profile, data domain discovery, and scorecards.
Profiling warehouse	Set up on a different server.

Setup Component	Description
Model Repository Service	Set up on a different server.
Analyst Service	Set up on a different server.

The following table describes the recommended configuration parameters for the setup environment:

Parameter	Value
Maximum Execution Pool Size	25
Maximum Profile Execution Pool Size	15
Maximum Concurrent Profile Jobs	8
Maximum DB Connections	5
Maximum Concurrent Profile Threads	1
DIS: Temporary Directories*	3
*Each directory on a different disk.	

Analysis

The profile operations, such as drill-down operations and previews, in the Developer tool require additional Data Integration Service threads. To meet this requirement, you can set the Maximum Execution Pool Size parameter to a higher value than the Maximum Profile Execution Pool Size parameter. In this use case, the requirement is three times higher because the number of developers is three. A value in the range of 5 to 10 might be appropriate for the use case. If the use case varies, analyze the requirements of the developers and choose an appropriate number.

The Profiling Service Module node can run up to eight concurrent flat file profiles during peak usage. You can set the Maximum Concurrent Profile Jobs parameter to 8. This setting ensures that the analysts can run approximately one and a half profile or scorecard jobs at a time, which is more than adequate for most situations.

To improve the flat file performance, verify that the temporary directory includes at least three different directories on different physical disk drives.

If you run a profile on relational sources, you can retain the default value of 5 for the Maximum DB Connections parameter because this profile environment has multiple relational database servers. If all the profiles use one database server, the Maximum Profile Execution Pool Size parameter prevents more than 15 concurrent profile queries running at the same time. If the database servers have more than 16 cores, increase both Maximum Profile Execution Pool Size and Maximum Execution Pool Size parameters by up to 15 additional threads.

Corporate Profile Sizing Example

This example summarizes the scenario of a large-size organization where data stewards use profiles and scorecards as part of the daily jobs. The data stewards have a specific area of competency.

In addition, a few power users, such as data stewards and data architects, use all the profile types to ensure data quality on a larger set of data assets. The power users also perform detailed assessments of some of the critical data sources. The profile implementation consists of a single high-end node connected to the database servers and ample disk space for flat files.

The following table describes the setup environment:

Setup Component	Description
Users	The setup environment has 40 to 60 data stewards using the Analyst tool and one to three power users.
Hardware	1 node, 32 cores, 128 GB, 8 x 2 TB disks, Linux, and 128 TB SAN.
Data	- Flat files with 1 million to 1 billion rows. - Relational tables with 1 million to 1 billion rows on multiple medium to high-end servers.
Profile type	- Data stewards run column profiles and scorecards. - Power users run enterprise discovery that includes column profile, data domain discovery, primary key discovery, and foreign key discovery. Power users also run unplanned overlap discovery and functional dependency discovery jobs.
Profiling warehouse	Set up on a different high-end server.
Model Repository Service	Set up on a different server.
Analyst Service	Set up on a different server.

The following table describes the recommended configuration parameters for the setup environment:

Parameter	Value
Maximum Execution Pool Size	100
Maximum Profile Execution Pool Size	75
Maximum Concurrent Profile Jobs	15
Maximum DB Connections	5
Maximum Concurrent Profile Threads	5
DIS: Temporary Directories*	5
*Each directory on a different disk.	

Analysis

All data stewards cannot concurrently run a profile or scorecard at the same time on a Data Integration Service machine with 32 cores. If all data stewards run a concurrent profile job or scorecard job, the persisted queue of the Profiling Service Module might contain up to 35 queued jobs.

The assumption in this use case is that a data steward performs the following jobs daily:

- Runs a profile or scorecard
- Performs data analysis or other basic operations
- Runs tasks that do not depend on the Profiling Service Module

The assumption for the maximum number of concurrent profile jobs is 25.

When a power user submits an enterprise discovery job, the Profiling Service Module adds the discovery job to the queue of column profile jobs that data stewards run. However, enterprise discovery jobs natively run with a reduced priority and minimize the interference with the profile jobs that data stewards run. Power users can analyze and plan to run the larger jobs when the Profiling Service Module is not at job capacity with the profile jobs that data stewards run.

Set the Maximum Profile Execution Pool Size parameter to the value of the Maximum Concurrent Profile Jobs parameter value multiplied by the Maximum DB Connections parameter value. The basis for this recommendation is that many data stewards might use different database servers and profile jobs do not overload any single server. The maximum number of current profile queries that you can run in this configuration is 125.

If there are fewer database servers than the number of database serves in this use case, you can decrease the Maximum DB Connections parameter value to 3 or 4. You can also adjust the Maximum Execution Pool Size and Maximum Profile Execution Pool Size parameters. You can set the Maximum Execution Pool Size parameter to a higher value than the Maximum Profile Execution Pool Size parameter value. Use the recommended setting for effective drill-down operations and previews even if other profile jobs use all the remaining Data Integration Service threads.

The data stewards might concurrently run profiles on multiple flat files. To ensure scalability, you can increase the number of temporary directories on separate disks so that the temporary, intermediate profiling results have the maximum I/O throughput.

Enterprise Profile Sizing Example

This example summarizes a large size organization with a competency center where data stewards run profiles and scorecards as part of the daily jobs. Data stewards have a specific area of competency.

In addition, the organization has a large number of data architects that ensure consistency in quality, structure, and content across all data assets in the organization. The number of data assets is up to many thousands. The organization has a grid configuration to process the scalable profile and enterprise discovery jobs.

The following table describes the setup environment:

Setup Component	Description
Users	The setup environment has 30 to 40 data analysts on the Analyst tool supported by 5 to 10 data architects on the Developer tool.
Hardware	Grid - 4 x 1 node, 12 cores, 64 GB, 6 x 2 TB disks, Linux, and 128 TB SAN.
Data	- Flat files with 1 million to 1 billion rows. - Relational tables with 1 million to 1 billion rows on multiple medium to high-end servers.
Profile type	- Data stewards run column profiles, data domain discovery, and scorecards. - Data architects run enterprise discovery that includes column profile, data domain discovery, primary key discovery, and foreign key discovery. Data architects also run unplanned overlap discovery and functional dependency discovery jobs.
Profiling warehouse	Set up on a different high-end server.
Model Repository Service	Set up on a different server.
Analyst Service	Set up on a different server.

The following table describes the recommended configuration parameters for the setup environment:

Parameter	Value
Maximum Execution Pool Size	100
Maximum Profile Execution Pool Size	50
Maximum Concurrent Profile Jobs	24
Maximum DB Connections	5
Maximum Concurrent Profile Threads	5
DIS: Temporary Directories **Note:** You must configure this value separately for each process node in the Grid. Each directory is on a different disk.	3

Analysis

This example requires a challenging configuration for optimal enterprise discovery on a single database server. You need to balance the power of the grid configuration to run many profile jobs that push the computation of the column profile jobs to the relational database.

When you run enterprise discovery, the profile jobs can run up to the Maximum Concurrent Profile Jobs value. Each profile job might run multiple queries equal to the value of the Maximum DB Connections parameter. This profile job run mechanism can easily overload the database if you do not set the configuration appropriately. To prevent database overload, you can decrease the Maximum DB Connections parameter value to 3.

The data architects that submit the enterprise discovery jobs can monitor the jobs. The data architects can then adjust the Maximum DB Connections parameter value based on the database server usage in the profile deployment environment.

The grid environment distributes the column profile mappings for flat files to the grid nodes in a round robin method. Therefore, you can increase the Maximum Concurrent Profile Threads parameter value from 1 to 2 to allow more mappings for column profile jobs on possibly different nodes. In addition, you must have a minimum of three temporary directories. You can set the value for temporary directories to a higher number to increase performance.

Author

Lavanya S