Performance Optimization for Informatica Data Services
Abstract
You can optimize data service performance by enabling performance features and tuning Data Integration Service properties that benefit data service requests. This article describes the data services architecture, performance features, and the steps to increase data service performance.

Supported Versions
- Data Services 9.1.0 Hotfix 2

Table of Contents
Overview ................................................. 2
Data Services Architecture. .................................. 3
Mapping Optimization. ........................................... 4
  Early Projection Optimization Method. .................... 4
  Early Selection Optimization Method. ....................... 5
  Predicate Optimization Method. .............................. 5
  Pushdown Optimization Method. ............................... 6
  Cost-Based Optimization Method. ......................... 7
  Semi-Join Optimization Method. ............................ 8
  SQL Data Service Optimization Level Configuration. .... 9
  Web Service Optimization Level Configuration. .......... 11
Data Object Caching. ....................................... 13
  Configuring Data Object Caching. .......................... 13
Result Set Caching. ........................................... 14
  Enabling Result Set Caching for an SQL Data Service. .... 14
  Enabling Result Set Caching for a Web Service. .......... 15
Transformation Cache. ....................................... 16
Data Integration Service Tuning. ......................... 17
  SQL Data Services Properties for Memory and Concurrent Requests. .. 17
  Web Service Property to Configure an Active DTM Process. .... 18
  Web Services Properties for Memory and Concurrent Requests ................ 19
  Web Service Log Management. ............................. 20
  Example Data Integration Service Configuration for Concurrent Web Service Requests ........ 21

Overview
Informatica Data Services provides the ability to access and transform data through an SQL data service or a web service. An SQL data service is a virtual database that end users can query. A web service is a collection of operations that provide access to data integration functionality. The Data Integration Service provides end-users access to data services. You can optimize data service performance by enabling performance features and tuning Data Integration Service properties that benefit each type of data service request.
Use the following performance optimization techniques to get the best performance results based on your requirements:

- Mapping optimization
- Result set caching
- Data object caching
- Transformation caching
- Data Integration Service tuning

**Note:** Components outside of the Informatica product such as network and database performance can also affect data services performance results.

## Data Services Architecture

When you create an SQL data service or a web service, the Developer tool stores the objects in the Model repository. To make a data service accessible to end users, you deploy the data service to the Data Integration Service. When you deploy a data service, you add the object to an application and deploy the application to the Data Integration Service. The Data Integration Service performs the data transformation processes for SQL data services and web services.

The following figure shows how clients connect to data services:

An external client sends SQL queries to access data in virtual tables of an SQL data service, execute virtual stored procedures, and access metadata. A web service client sends requests to run a web service operation to read, transform, or write data.

When the Data Integration Service receives an SQL query from an external client, the SQL Service Module of the Data Integration Service starts a Data Transformation Manager (DTM) process to run the query. The Data Integration Service returns the query results to the external client.

When the Data Integration Service receives a web service request message, the Web Service Module of the Data Integration Service starts a DTM process to complete the request. The Data Integration Service sends the results in a web service response message to the web service client.
Mapping Optimization

Use mapping optimization to increase the performance of SQL data service queries and web service requests. The Data Integration Service optimizes mappings based on the mapping optimizer level that you configure.

The optimizer level determines which optimization methods the Data Integration Service applies to the mapping at run time. You can choose one of the following mapping optimizer levels:

None
- The Data Integration Service does not optimize the mapping. It runs the mapping exactly as you designed it.

Minimal
- The Data Integration Service applies the early projection optimization method to the mapping.

Normal
- The Data Integration Service applies the early projection, early selection, predicate optimization, and pushdown optimization methods to the mapping. This is the default optimizer level.

Full
- The Data Integration Service applies the early projection, early selection, predicate optimization, pushdown, cost-based, and semi-join optimization methods to the mapping.

By default, the Data Integration Service applies the Normal optimizer level to mappings.

When you configure an optimization level, consider the following guidelines:

- The normal optimizer level can benefit most mappings.
- The full optimizer level benefits mappings that contain a join. It is especially beneficial when you join data from tables that are different sizes.
- The minimal or none optimizer level is useful when you verify data output or debug mapping logic.

You can configure the optimizer level in the Developer tool data viewer configuration setting, in the Administrator tool Data Integration Service properties, in the SQL data service connection, or in the header of the web service request.

Early Projection Optimization Method

The early projection optimization method causes the Data Integration Service to identify unused ports and remove the links between those ports.

Identifying and removing links between unused ports improves performance by reducing the amount of data the Data Integration Service moves across transformations. When the Data Integration Service processes a mapping, it moves the data from all connected ports in a mapping from one transformation to another. In large, complex mappings, or in mappings that use nested mapplets, some ports might not ultimately supply data to the target. The early projection method causes the Data Integration Service to identify the ports that do not supply data to the target. After the Data Integration Service identifies unused ports, it removes the links between all unused ports from the mapping.

The Data Integration Service does not remove all links. For example, it does not remove the following links:

- Links connected to a Custom transformation
- Links connected to transformations that call an ABORT() or ERROR() function, send email, or call a stored procedure

If the Data Integration Service determines that all ports in a transformation are unused, it removes all transformation links except the link to the port with the least data. The Data Integration Service does not remove the unused transformation from the mapping.

The Developer tool enables this optimization method by default.
Early Selection Optimization Method

The early selection optimization method applies the filters in a mapping as early as possible. Filtering data early increases performance by reducing the number of rows that pass through the mapping. In the early selection method, the Data Integration Service splits, moves, splits and moves, or removes the Filter transformations in a mapping.

The Data Integration Service might split a Filter transformation if the filter condition is a conjunction. For example, the Data Integration Service might split the filter condition "A>100 AND B<50" into two simpler conditions, "A>100" and "B<50."

When the Data Integration Service can split a filter, it attempts to move the simplified filters up the mapping pipeline, closer to the mapping source. Splitting the filter allows the Data Integration Service to move the simplified filters up the pipeline separately. Moving the filter conditions closer to the source reduces the number of rows that pass through the mapping.

The Data Integration Service might also remove Filter transformations from a mapping. It removes a Filter transformation when it can apply the filter condition to the transformation logic of the transformation immediately upstream of the original Filter transformation.

The Data Integration Service cannot always move a Filter transformation. For example, it cannot move a Filter transformation upstream of the following transformations:

- Custom transformations
- Transformations that call an ABORT() or ERROR() function, send email, or call a stored procedure
- Transformations that maintain count through a variable port, for example, COUNT=COUNT+1
- Transformations that create branches in the mapping. For example, the Data Integration Service cannot move a Filter transformation upstream if it is immediately downstream of a Router transformation with two output groups.

The Data Integration Service does not move a Filter transformation upstream in the mapping if doing so changes the mapping results.

The Developer tool enables this optimization method by default.

You might want to disable this method if it does not increase performance. For example, a mapping contains source ports "P1" and "P2." "P1" is connected to an Expression transformation that evaluates "P2=f(P1)." "P2" is connected to a Filter transformation with the condition "P2>1." The filter drops very few rows. If the Data Integration Service moves the Filter transformation upstream of the Expression transformation, the Filter transformation must evaluate "f(P1)>1" for every row in source port "P1." The Expression transformation also evaluates "P2=f(P1)" for every row. If the function is resource intensive, moving the Filter transformation upstream nearly doubles the number of times it is called, which might degrade performance.

Predicate Optimization Method

The predicate optimization method causes the Data Integration Service to examine the predicate expressions generated by a mapping or the transformations within a mapping to determine whether the expressions can be simplified or rewritten to increase performance of the mapping.

When the Data Integration Service runs a mapping, it generates queries against the mapping sources and performs operations on the query results based on the mapping logic and the transformations within the mapping. The generated queries and operations often involve predicate expressions. Predicate expressions represent the conditions that the data must satisfy. The filter and join conditions in Filter and Joiner transformations are examples of predicate expressions.

This optimization method causes the Data Integration Service to examine the predicate expressions generated by a mapping or the transformations within a mapping to determine whether the expressions can be simplified or rewritten.
to increase performance of the mapping. The Data Integration Service also attempts to apply predicate expressions as early as possible to improve mapping performance.

This method also causes the Data Integration Service to infer relationships implied by existing predicate expressions and create new predicate expressions based on the inferences. For example, a mapping contains a Joiner transformation with the join condition "A=B" and a Filter transformation with the filter condition "A>5." The Data Integration Service might be able to add the inference "B>5" to the join condition.

The Data Integration Service uses the predicate optimization method with the early selection optimization method when it can apply both methods to a mapping. For example, when the Data Integration Service creates new filter conditions through the predicate optimization method, it also attempts to move them upstream in the mapping through the early selection method. Applying both optimization methods improves mapping performance when compared to applying either method alone.

The Data Integration Service applies this optimization method when it can run the mapping more quickly. It does not apply this method when doing so changes mapping results or worsens mapping performance.

When the Data Integration Service rewrites a predicate expression, it applies mathematical logic to the expression to optimize it. For example, the Data Integration Service might perform any or all of the following actions:

- Identify equivalent variables across predicate expressions in the mapping and generates simplified expressions based on the equivalencies.
- Identify redundant predicates across predicate expressions in the mapping and remove them.
- Extract subexpressions from disjunctive clauses and generates multiple, simplified expressions based on the subexpressions.
- Normalize a predicate expression.
- Apply predicate expressions as early as possible in the mapping.

The Data Integration Service might not apply predicate optimization to a mapping when the mapping contains transformations with a datatype mismatch between connected ports.

The Data Integration Service might not apply predicate optimization to a transformation when any of the following conditions are true:

- The transformation contains explicit default values for connected ports.
- The transformation calls an ABORT() or ERROR() function, sends email, or calls a stored procedure.
- The transformation does not allow predicates to be moved. For example, a developer might create a Custom transformation that has this restriction.

The Developer tool enables this optimization method by default.

**Pushdown Optimization Method**

The pushdown optimization method causes the Data Integration Service to push transformation logic to the source database. The Data Integration Service translates the transformation logic into SQL queries and sends the SQL queries to the database. The source database executes the SQL queries to process the transformations.

Pushdown optimization improves the performance of mappings when the source database can process transformation logic faster than the Data Integration Service. The Data Integration Service also reads less data from the source.

The amount of transformation logic that the Data Integration Service pushes to the source database depends on the database, the transformation logic, and the mapping configuration. The Data Integration Service processes all transformation logic that it cannot push to a database.

The Data Integration Service can push the following transformation logic to the source database:

- Expression transformation logic
Filter transformation logic
Joiner transformation logic. The sources must be in the same database management system and must use identical connections.

The Data Integration Service cannot push transformation logic after a source in the following circumstances:

- The Data Integration Service cannot push any transformation logic if the source is a customized data object that contains a custom SQL query.
- The Data Integration Service cannot push any transformation logic if the source contains a column with a binary datatype.
- The Data Integration Service cannot push Expression or Joiner transformation logic if the source is a customized data object that contains a filter condition or user-defined join.

The Data Integration Service applies pushdown optimization to a mapping when you select the normal or full optimizer level. When you select the normal optimizer level, the Data Integration Service applies pushdown optimization after it applies all other optimization methods. If you select the full optimizer level, the Data Integration Service applies pushdown optimization before semi-join optimization, but after all of the other optimization methods.

When you apply pushdown optimization, the Data Integration Service analyzes the optimized mapping from the source to the target or until it reaches a downstream transformation that it cannot push to the source database. The Data Integration Service generates and executes a SELECT statement based on the transformation logic for each transformation that it can push to the database. Then, it reads the results of this SQL query and processes the remaining transformations in the mapping.

Cost-Based Optimization Method

The cost-based optimization method causes the Data Integration Service to evaluate a mapping, generate semantically equivalent mappings, and run the mapping with the best performance. This method is most effective for mappings that contain multiple Joiner transformations. It reduces run time for mappings that perform adjacent, unsorted, inner-join operations.

Semantically equivalent mappings are mappings that perform identical functions and produce the same results. To generate semantically equivalent mappings, the Data Integration Service divides the original mapping into fragments. The Data Integration Service then determines which mapping fragments it can optimize.

Generally, the Data Integration Service can optimize a fragment if the fragment meets the following requirements:

- The Data Integration Service can optimize every transformation within the fragment. The Data Integration Service can optimize a transformation if it can determine the number of rows that pass through the transformation. The Data Integration Service cannot optimize certain active transformations, such as some Custom transformations, because it cannot determine the number of rows that pass through the transformation.
- The fragment has one target transformation.
- No transformation in the fragment has multiple output groups.
- No two linked ports within a fragment perform an implicit datatype conversion. Therefore, the datatype, precision, and scale for each output port must match the datatype, precision, and scale of the linked input port.

The Data Integration Service optimizes each fragment that it can optimize. During optimization, the Data Integration Service might add, remove, or re-order transformations within a fragment. The Data Integration Service verifies that the optimized fragments produce the same results as the original fragments and forms alternate mappings that use the optimized fragments.

The Data Integration Service generates all or almost all of the mappings that are semantically equivalent to the original mapping. It computes data statistics for the original mapping and each alternate mapping. The Data Integration Service compares the statistics to identify the mapping that runs most quickly. The Data Integration Service performs a
validation check on the best alternate mapping to ensure that it is valid and produces the same results as the original mapping.

The Data Integration Service caches the best alternate mapping in memory. When you run a mapping, the Data Integration Service retrieves the alternate mapping and runs it instead of the original mapping.

**Semi-Join Optimization Method**

The semi-join optimization method attempts to reduce the amount of data extracted from the source by modifying join operations in the mapping.

The Data Integration Service applies this method to a Joiner transformation when one input group has many more rows than the other and when the larger group has many rows with no match in the smaller group based on the join condition. The Data Integration Service attempts to decrease the size of the data set of one join operand by reading the rows from the smaller group, finding the matching rows in the larger group, and then performing the join operation. Decreasing the size of the data set improves mapping performance because the Data Integration Service no longer reads unnecessary rows from the larger group source. The Data Integration Service moves the join condition to the larger group source and reads only the rows that match the smaller group.

Before applying this optimization method, the Data Integration Service performs analyses to determine whether semi-join optimization is possible and likely to be worthwhile. If the analyses determine that this method is likely to increase performance, the Data Integration Service applies it to the mapping. The Data Integration Service then reanalyzes the mapping to determine whether there are additional opportunities for semi-join optimization. It performs additional optimizations if appropriate. The Data Integration Service does not apply semi-join optimization unless the analyses determine that there is a high probability for improved performance.

For the Data Integration Service to apply the semi-join optimization method to a join operation, the Joiner transformation must meet the following requirements:

- The join type must be normal, master outer, or detail outer. The joiner transformation cannot perform a full outer join.
- The detail pipeline must originate from a relational source.
- The join condition must be a valid sort-merge-join condition. That is, each clause must be an equality of one master port and one detail port. If there are multiple clauses, they must be joined by AND.
- If the mapping does not use target-based commits, the Joiner transformation scope must be All Input.
- The master and detail pipelines cannot share any transformation.
- The mapping cannot contain a branch between the detail source and the Joiner transformation.

The semi-join optimization method might not be beneficial in the following circumstances:

- The Joiner transformation master source does not contain significantly fewer rows than the detail source.
- The detail source is not large enough to justify the optimization. Applying the semi-join optimization method adds some overhead time to mapping processing. If the detail source is small, the time required to apply the semi-join method might exceed the time required to process all rows in the detail source.
- The Data Integration Service cannot get enough source row count statistics for a Joiner transformation to accurately compare the time requirements of the regular join operation against the semi-join operation.

The Developer tool does not enable this method by default.
**SQL Data Service Optimization Level Configuration**

Configure the optimizer level when you want the SQL data service to use an optimizer level other than the normal optimizer level. By default, each SQL data service uses the normal optimizer level.

To understand how the mapping optimizer level creates an optimized query for an SQL data service, view the query plan for an SQL Data Service. When you view the query plan, the Developer tool displays a graphical representation of the optimized query based on the optimizer level and a graphical representation of the original query.

You can use one or more of the following methods to configure the optimizer level for an SQL data service:

- Configure the optimizer level for data preview of an SQL data service before you deploy it to a Data Integration Service.
- Configure the optimizer level for deployed SQL data services that run on a specific Data Integration Service.
- Configure the optimizer level in the connection string of queries that you run against a deployed SQL data service.

**Configuring the SQL Data Service Optimization Level for Data Preview**

Configure the optimization level that the Data Integration Service uses to execute SQL queries when you preview the output of a SQL data service.

1. In the Developer tool, click Run > Open Run Dialog.
   The Run dialog box appears.
2. Click Data Viewer Configuration.
3. Click the New button.
4. Enter a name for the data viewer configuration.
5. Click the Advanced tab.
6. Select an optimizer level.
7. Click Apply.
8. Click Close
   The Developer tool creates the data viewer configuration.

**Configuring the Optimizer Level for Deployed SQL Data Services**

Configure the optimizer level to specify which optimization level the Data Integration Services uses to execute SQL queries against deployed SQL data services.

1. In the Administrator tool, select the Data Integration Service.
2. In the Custom Properties section, click Edit.
   The Edit Custom Property dialog box appears.
3. Click New.
   The New Custom Property dialog box appears.
4. Configure the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter SQLDataServiceOptions.optimizeLevel.</td>
</tr>
</tbody>
</table>
| Value       | Enter the numeric value that is associated with the optimization level that you want to configure. You can enter one of the following numeric values: 
- 0 for no optimization.
- 1 for minimal optimization.
- 2 for normal optimization.
- 3 for full optimization. |

5. Click OK.

Configuring the Optimizer Level for an SQL Data Service Query

Configure the optimizer level to specify with optimization level the Data Integration Services uses to execute SQL queries against a deployed SQL data service. The optimizer level that you set in the connection string overrides the value that you configure in the Administrator tool for all SQL Data Services that run on a particular Data Integration Service.

To configure the SQL data service optimization level for a query, append the following entry to the SQL data service connect string: SQLDataServiceOptions.optimizeLevel= <numeric_optimizer_level>.

You can enter one of the following numeric optimizer levels:
- 0 for no optimization.
- 1 for minimal optimization.
- 2 for normal optimization.
- 3 for full optimization.

View SQL Data Service Query Plan

When you view the query plan for an SQL data service, you view the graphical representation of your original query and the graphical representation of the optimized query. The graphical representation describes how the Data Integration Service processes the query. The graphical representation includes the transformations and the order that the Data Integration Services processes each transformation.

The Developer tool uses the optimizer level that you set in the Developer tool to generate the optimized query. The optimized query displays the query as the Data Integration Service runs it.

For example, you want to query the CUSTOMERS virtual table in an SQL data service. In the Data Viewer view, you choose the default data viewer configuration settings, which sets the optimizer level for the query to normal.

You enter the following query in the Data Viewer view:

```
select * from CUSTOMERS where CUSTOMER_ID > 150000 order by LAST_NAME
```

When you view the SQL query plan, the Developer tool displays the following graphical representation of the query:
The non-optimized view displays the query that you enter. The Developer tool displays the WHERE clause as a Filter transformation and the ORDER BY clause as a Sorter transformation. The Developer tool uses the pass-through Expression transformation to rename ports.

When you view the optimized query, the Developer tool displays the following graphical representation of the query:

```
ReadDataObject -> expr_CUSTME -> OrderByExpr -> OrderBy -> expr_output_ -> output
RSDSMap_CUST CMERSD0Map
```

The optimized view displays the query that the Data Integration Service runs. Because the optimizer level is normal, the Data Integration Service pushes the filter condition to the source data object. Pushing the filter condition increases query performance because it reduces the number of rows that the Data Integration Service reads from the source data object. Similar to the non-optimized query, the Developer tool displays the ORDER BY clause as a Sorter transformation. It uses pass-through Expression transformations to enforce the datatypes that you specify in the logical transformations.

**Viewing an SQL Query Plan**

Display the SQL query plan to view a mapping-like representation of the SQL query you enter when you preview virtual table data.

1. Open an SQL data service that contains at least one virtual table.
2. Click the **Data Viewer** view.
3. Enter an SQL query in the **Input** window.
4. Optionally, select a data viewer configuration that contains the optimizer level you want to apply to the query.
5. Click **Show Query Plan**.
   - The Developer tool displays the SQL query plan for the query as you entered it on the **Non-Optimized** tab.
6. To view the optimized query, click the **Optimized** tab.
   - The Developer tool displays the optimized SQL query plan.

**Web Service Optimization Level Configuration**

Configure the optimizer level when you want the web service to use an optimizer level other than the normal optimizer level. By default, each web service uses the normal optimizer level.

You can use one or more of the following methods to configure the optimizer level for a web service:

- Configure the optimizer level for data preview of a web service before you deploy it to a Data Integration Service.
- Configure the optimizer level for deployed web services that run on a specific Data Integration Service.
- Configure the optimizer level in the header of the web service request.

**Configuring the Web Service Optimizer Level for Data Preview**

Configure the optimizer level to specify which optimization level the Data Integration Services uses to preview the output of a web service.

1. In the Developer tool, click **Run > Open Run Dialog**.
   - The **Run** dialog box appears.
2. Click **Data Viewer Configuration**.
3. Click the **New** button.
4. Enter a name for the data viewer configuration.
5. Click the **Advanced** tab.
6. Select an optimizer level.
7. Click **Apply**.
8. Click **Close**

The Developer tool creates the data viewer configuration.

When you run the data viewer to preview the output of an operation mapping, select the data viewer configuration that includes the optimizer level that you want to use.

### Configuring the Optimizer Level for Deployed Web Services

Configure the optimizer level to specify which optimization level the Data Integration Services uses to run deployed web services.

1. In the Administrator tool, select the Data Integration Service.
2. In the **Custom Properties** section, click **Edit**.
   
   The *Edit Custom Property* dialog box appears.
3. Click **New**.
   
   The *New Custom Property* dialog box appears.
4. Configure the following properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Enter <code>WebServiceOptions.optimizeLevel</code></td>
</tr>
</tbody>
</table>
| Value    | Enter the numeric value that is associated with the optimization level that you want to configure. You can enter one of the following numeric values:  
- 0 for no optimization.  
- 1 for minimal optimization.  
- 2 for normal optimization.  
- 3 for full optimization. |

5. Click **OK**.

### Configuring the Optimizer Level for a Web Service Request

Configure the optimizer level in the header of the web service request to specify which optimization level the Data Integration Services uses to process the request. The optimizer level that you set in the web service header overrides the value that you configure in the Administrator tool for web services that run on the Data Integration Service.

To override the web service optimization level for a web service request, include the following entry in the HTTP header of the web service SOAP request: 

`WebServiceOptions.optimizeLevel= <numeric_optimizer_level>`.

You can enter one of the following numeric optimizer levels:

- 0 for no optimization.
- 1 for minimal optimization.
- 2 for normal optimization.
Data Object Caching

The Data Integration Service can use cached data for data objects. Users that access data objects frequently might want to enable caching for data objects.

You can increase mapping and query performance for end users by caching the logical data objects and virtual tables. If the application contains an SQL data service, you can cache logical data objects and virtual tables. If the application contains a web service, you can cache logical data objects. When you enable data object caching, the Data Integration Service accesses pre-built logical data objects and virtual tables. When the Data Integration Service can access pre-built data objects, performance increases for SQL queries or web service requests.

The Data Integration Service can use the following databases to store data object cache tables:

- IBM DB2
- Microsoft SQL Server
- Oracle

To optimize data object cache performance, consider the following techniques:

- Define primary keys and foreign keys for logical data objects. When the Data Integration Service generates cache for logical data objects with keys, it creates indexes. The indexes can increase the performance of queries on the cache database.
- Cache logical data objects that you join in a mapping. When you join cached logical data objects, the Data Integration Service can pushdown the Joiner transformation logic to the cache database even when the source data originates from different databases.

For more information about data object caching, see the Informatica How-To Library article "How to Increase the Performance of Deployed Objects through Data Object Caching."
https://kb.informatica.com/h2l/HowTo%20Library/1/0272_DataObjectCaching.pdf

Configuring Data Object Caching

To configure data object caching, configure the cache properties for the Data Integration Service and the application that contains the logical data object and virtual tables.

1. In the Administrator tool, select a Data Integration Service.
2. Click the Properties view.
3. Edit the following Logical Data Object/Virtual Table Cache properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Removal Time</td>
<td>The amount of milliseconds the Data Integration Service waits before cleaning up cache storage after a refresh. Default is 3,600,000.</td>
</tr>
<tr>
<td>Cache Connection</td>
<td>The database connection name for the database that stores the data object cache. Select a valid connection object name.</td>
</tr>
<tr>
<td>Maximum Concurrent Refresh Requests</td>
<td>Maximum number of cache refreshes that can occur at the same time. Limit the concurrent cache refreshes to maintain system resources.</td>
</tr>
</tbody>
</table>

4. Click OK.
5. Click the Applications view.
6. Select the application that contains the logical data object or virtual table for which you want to enable or disable caching.
7. Stop the application.
8. Expand the application that contains the data object for which you want to enable or disable caching.
9. Select the logical data object or virtual table.
10. Edit the following logical data object or virtual table properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Caching</td>
<td>Cache the logical data object or virtual table.</td>
</tr>
<tr>
<td>Cache Refresh Period</td>
<td>Number of milliseconds between cache refreshes.</td>
</tr>
</tbody>
</table>

11. Click **OK**.
12. Restart the application.

**Result Set Caching**

Result set caching enables the Data Integration Service to use cached results for SQL data service queries and web service requests. Users that run identical queries in a short period of time may want to use result set caching to decrease the runtime of identical queries.

When you configure result set caching, the Data Integration Service caches the results of the DTM process associated with each SQL data service query and web service request. The Data Integration Service caches the results for the expiration period that you configure. When a client makes the same query or request before the cache expires, the Data Integration Service returns the cached results.

The Data Integration Service stores the result set cache in memory and in cache files on disk. When the Data Integration Service caches data, it first writes the cache to memory and then writes the overflow cache to cache files on disk. Data Integration Service allocates cache memory based on the result set cache **Maximum Total Memory Size** property. Data Integration Service allocates disk space based on the result set cache **Maximum Total Disk Size** property. When the amount of cache exceeds the cache memory and disk space allocation, the Data Integration Service does not store additional cache.

When you enable the Data Integration Service to use cached results, data service performance increases. However, to further improve the data service processing time for identical queries and requests, allocate enough space to store the cache in memory. When you configure the amount of cache memory to be equal to or greater than what is required to cache the results, you increase performance by reducing the system I/O overhead. When the Data Integration Service writes cache files to disk, the data service processing time increases due to system I/O overhead.

**Enabling Result Set Caching for an SQL Data Service**

To use cached results for identical SQL data service queries, configure the Data Integration Service to use result set caching.

1. In the Administrator tool, select a Data Integration Service.
2. Click the **Process** view to configure the following result set cache properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Total Disk Size</td>
<td>Maximum number of bytes allowed for the total result set cache file storage. Default is 0.</td>
</tr>
<tr>
<td>Storage Directory</td>
<td>Absolute path to the directory that stores result set cache files.</td>
</tr>
<tr>
<td>File Name Prefix</td>
<td>The prefix for the names of all result set cache files stored on disk. Default is RSCACHE.</td>
</tr>
<tr>
<td>Maximum Per Cache Memory Size</td>
<td>Maximum number of bytes allocated for a single result set cache instance in memory. Default is 0.</td>
</tr>
<tr>
<td>Maximum Total Memory Size</td>
<td>Maximum number of bytes allocated for the total result set cache storage in memory. Default is 0.</td>
</tr>
<tr>
<td>Maximum Number of Caches</td>
<td>Maximum number of result set cache instances allowed for this Data Integration Service process. Default is 0.</td>
</tr>
<tr>
<td>Enable Encryption</td>
<td>Indicates whether result set cache files are encrypted using 128-bit AES encryption. Valid values are true or false. Default is true.</td>
</tr>
</tbody>
</table>

3. Click the **Application** view and then click the SQL data service to configure the following property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result Set Cache Expiration Period</td>
<td>The number of milliseconds that the result set cache is available for use. If set to -1, the cache never expires. If set to 0, result set caching is disabled. Changes to the expiration period do not apply to existing caches. If you want all caches to use the same expiration period, purge the result set cache after you change the expiration period. Default is 0.</td>
</tr>
</tbody>
</table>

**Enabling Result Set Caching for a Web Service**

To use cached results for identical web service requests, configure the Data Integration Service to use result set caching:

1. In the Administrator tool, select a Data Integration Service.
2. Click the **Process** view to configure the following result set cache properties:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Total Disk Size</td>
<td>Maximum number of bytes allowed for the total result set cache file storage. Default is 0.</td>
</tr>
<tr>
<td>Storage Directory</td>
<td>Absolute path to the directory that stores result set cache files.</td>
</tr>
<tr>
<td>File Name Prefix</td>
<td>The prefix for the names of all result set cache files stored on disk. Default is RSCACHE.</td>
</tr>
<tr>
<td>Property</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Maximum Per Cache Memory Size</td>
<td>Maximum number of bytes allocated for a single result set cache instance in memory. Default is 0.</td>
</tr>
<tr>
<td>Maximum Total Memory Size</td>
<td>Maximum number of bytes allocated for the total result set cache storage in memory. Default is 0.</td>
</tr>
<tr>
<td>Maximum Number of Caches</td>
<td>Maximum number of result set cache instances allowed for this Data Integration Service process. Default is 0.</td>
</tr>
<tr>
<td>Enable Encryption</td>
<td>Indicates whether result set cache files are encrypted using 128-bit AES encryption. Valid values are true or false. Default is true.</td>
</tr>
</tbody>
</table>

3. Click the **Application** view, click the web service, and then click the operation to configure the following property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Result Set Cache Expiration Period</td>
<td>The number of milliseconds that the result set cache is available for use. If set to -1, the cache never expires. If set to 0, result set caching is disabled. Changes to the expiration period do not apply to existing caches. If you want all caches to use the same expiration period, purge the result set cache after you change the expiration period. Default is 0.</td>
</tr>
</tbody>
</table>

4. If you want the Data Integration Service to cache the results by user, enable WS-Security in the web service properties.

**Transformation Cache**

To increase performance for a data service, consider configuring the transformation cache properties for the Aggregator, Joiner, Lookup, Rank, or Sorter transformations in mappings that are associated with the SQL data service or the web service.

The Data Integration Service stores cache in memory and in cache files on disk. When the Data Integration Service caches data, it first writes the cache to memory and then writes the overflow cache to cache files to disk.

To improve the data service processing time for mappings that contain the Aggregator, Joiner, Lookup, Rank, or Sorter transformation, configure the transformations to allocate enough space to store the cache in memory. When you configure the amount of cache memory to be equal to or greater than what is required to cache the data and index, you increase performance by reducing the system I/O overhead. When the Data Integration Service writes cache files to disk, the data service processing time increases due to system I/O overhead.

When you configure numeric values for transformation cache properties, the value that you configure is equal to the amount of memory that the Data Integration Service can use to store data cache and index cache.

When you configure the transformation cache properties to use the **Auto** setting, the Data Integration Service allocates cache memory based on the Data Integration Service process Maximum Memory Size and Maximum Session Memory properties. When the maximum memory size is 0, the Data Integration Service divides the maximum session memory allocation among transformations that use the **Auto** setting. When the maximum memory size value is greater than zero, the Data Integration Service allocates the required amount of cache memory for each transformation, until the memory requirement exceeds the maximum memory size value.
Data Integration Service Tuning

The Data Integration Services provides end users access to data services and performs the data transformation processes for data services. By tuning the Data Integration Service based on your requirements, performance can increase.

Consider the following performance bottlenecks and solutions:

- The Data Integration Service can consume large amounts of memory while processing SQL data services and web services. To configure the memory setting that the Data Integration Services uses when it processes SQL data services and web services, configure the Data Integration Service process properties that impact memory consumption.
- The Data Integration Service consumes system resources to spawn a DTM process for each web service request. To configure the Data Integration Service to use one DTM process to process more than one web service request, configure the web service DTM keep alive time property for the Data Integration Service.
- The Data Integration Service, each SQL data service, and each web service that runs on the Data Integration Service consumes system and memory resources for each concurrent request. To configure the number of concurrent request that the Data Integration Service, each SQL data service, and each web service can accept, configure the Data Integration Service process properties and the web service properties.
- The number of web service log files that the Data Integration Service writes and maintains can decrease performance. Configure the web service trace level and the amount of web service run-time log files that the Data Integration Service stores on disk.

**SQL Data Services Properties for Memory and Concurrent Requests**

To configure how the Data Integration Service manages memory and concurrent requests for SQL data service queries, tune the Data Integration Service properties.

In the Administrator tool, configure the following advanced properties for the Data Integration Service process:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Maximum Heap Size         | Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Data Integration Service. Use this property to increase the performance. Append one of the following letters to the value to specify the units:  
  - b for bytes  
  - k for kilobytes  
  - m for megabytes  
  - g for gigabytes  
  Default is 512 megabytes.  
  **Note:** Consider increasing the maximum heap size when the Data Integration Services needs to process large amounts of data. |

In the Administrator tool, configure the following SQL properties for the Data Integration Service process:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum # of Concurrent Connections</td>
<td>Limits the number of database connections that the Data Integration Service can make for SQL data services. Default is 100.</td>
</tr>
</tbody>
</table>
In the Administrator tool, configure the following execution options for the Data Integration Service process:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Execution Pool Size</td>
<td>The maximum number of requests that the Data Integration Service can run concurrently. Requests include data previews, mappings, profiling jobs, SQL queries, and web service requests. Default is 10. Note: When you increase the pool size value, the DTM process of the Data Integration Service uses more hardware resources such as CPU, memory, and system I/O. Set this value based on the resources available on your machine and your data service requirements. For example, consider the number of CPUs on the machine that hosts the Data Integration Service and the amount of memory that is available to the Data Integration Service.</td>
</tr>
<tr>
<td>Temporary Directories</td>
<td>Location of temporary directories for Data Integration Service process on the node. Default is &lt;Informatica Services Installation Directory&gt;/tomcat/bin/disTemp. Add a second path to this value to provide a dedicated directory for temporary files created in profile operations. Use a semicolon to separate the paths. Do not use a space after the semicolon.</td>
</tr>
<tr>
<td>Maximum Memory Size</td>
<td>The maximum amount of memory, in bytes, that the Data Integration Service can allocate for running requests. If you do not want to limit the amount of memory the Data Integration Service can allocate, set this threshold to 0. When you set this threshold to a value greater than 0, the Data Integration Service uses it to calculate the maximum total memory allowed for running all requests concurrently. The Data Integration Service calculates the maximum total memory as follows: Maximum Memory Size + Maximum Heap Size + memory required for loading program components. Default is 512,000,000. Note: If you run profiles or data quality mappings, set this threshold to 0.</td>
</tr>
<tr>
<td>Maximum Session Size</td>
<td>The maximum amount of memory, in bytes, that the Data Integration Service can allocate for any request. For optimal memory utilization, set this threshold to a value that exceeds the Maximum Memory Size divided by the Maximum Execution Pool Size. The Data Integration Service uses this threshold even if you set Maximum Memory Size to 0 bytes. Default is 50,000,000.</td>
</tr>
</tbody>
</table>

**Web Service Property to Configure an Active DTM Process**

To increase performance, you can configure the Data Integration Service to keep a DTM process active so that it can process more than one web service request.

In the Administrator tool, add the following custom property in the Data Integration Service Custom Property area:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebServiceOptions.DTMKeepAliveTime</td>
<td>The value of this custom property is the number of milliseconds that a DTM process remains active in order to process incoming web service requests. Even when there are no incoming web service requests, the DTM process remains active for the amount of time that you configure. When the time is exceeded, the existing DTM process is no longer active and the DIS starts a new DTM process for the next incoming request. This value defaults to 5000 milliseconds. Note: The ability to use an existing DTM process increases performance. The DIS requires additional resources to start a DTM process for each request. Keeping the DTM active consumes memory. Therefore, users should consider the memory consumption when configuring this option.</td>
</tr>
</tbody>
</table>
**Web Services Properties for Memory and Concurrent Requests**

To increase performance, configure concurrency and memory properties for the Data Integration process and each web service.

In the Administrator tool, configure the following HTTP properties for the Data Integration Service Process:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Backlog Request</td>
<td>Maximum number of HTTP or HTTPS connections that can wait in a queue for this Data Integration Service process. Default is 100. You must restart the Data Integration Service to apply your changes.</td>
</tr>
<tr>
<td>Maximum Concurrent Requests</td>
<td>Maximum number of HTTP or HTTPS connections that can be made to this Data Integration Service process. Default is 200. You must restart the Data Integration Service to apply your changes. <strong>Note:</strong> For a web service, this property impacts the number of web service requests that the Data Integration Services accepts before it sends the requests to the Data Integration Service backlog.</td>
</tr>
</tbody>
</table>

In the Administrator tool, configure the following advanced property for the Data Integration Service Process:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
</table>
| Maximum Heap Size       | Amount of RAM allocated to the Java Virtual Machine (JVM) that runs the Data Integration Service. Use this property to increase the performance. Append one of the following letters to the value to specify the units:  
  - b for bytes  
  - k for kilobytes  
  - m for megabytes  
  - g for gigabytes  
  Default is 512 megabytes. **Note:** Consider increasing the maximum heap size when the Data Integration Services needs to process large amounts of data. |
In the Administrator tool, configure the following execution options for the Data Integration Service Process:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Execution Pool Size</td>
<td>The maximum number of requests that the Data Integration Service can run concurrently. Requests include data previews, mappings, profiling jobs, SQL queries, and web service requests. Default is 10. Note: When you increase the pool size value, the DTM process of the Data Integration Service uses more hardware resources such as CPU, memory, and system I/O, you should set this value based on the resources available on your machine and your data service requirements. For example, you may want to consider the number of CPUs on the machine that hosts the Data Integration Service and the amount of memory that is available to the Data Integration Service.</td>
</tr>
<tr>
<td>Maximum Memory Size</td>
<td>The maximum amount of memory, in bytes, that the Data Integration Service can allocate for running requests. If you do not want to limit the amount of memory the Data Integration Service can allocate, set this threshold to 0. When you set this threshold to a value greater than 0, the Data Integration Service uses it to calculate the maximum total memory allowed for running all requests concurrently. The Data Integration Service calculates the maximum total memory as follows: Maximum Memory Size + Maximum Heap Size + memory required for loading program components. Default is 512,000,000. Note: If you run profiles or data quality mappings, set this threshold to 0.</td>
</tr>
<tr>
<td>Maximum Session Size</td>
<td>The maximum amount of memory, in bytes, that the Data Integration Service can allocate for any request. For optimal memory utilization, set this threshold to a value that exceeds the Maximum Memory Size divided by the Maximum Execution Pool Size. The Data Integration Service uses this threshold even if you set Maximum Memory Size to 0 bytes. Default is 50,000,000.</td>
</tr>
</tbody>
</table>

In the Administrator tool, configure the following web service property:

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum Concurrent Requests</td>
<td>Maximum number of requests that a web service can process at one time. Default is 10. Note: To avoid web service request failures, set the value of this property equal to the Data Integration Service Maximum Concurrent Requests.</td>
</tr>
</tbody>
</table>

**Web Service Log Management**

System I/O performance can decrease when the Data Integration Service writes and maintains a large number of log files. The Data Integration Service generates web service run-time logs based on the trace level that you configure. Consider managing the number of log files that the Data Integration Service writes and maintains.

When you configure web service properties for a deployed web service, you can specify the log trace level. The trace level determines the types of logs that the Data Integration Service writes to the run-time log location. The default web service trace level is INFO. When the trace level is set to FINEST or ALL, performance can decrease because the Data Integration Service writes more logs to the log file. The performance impact of setting the trace level to FINEST or ALL is the greatest when the web service uses HTTPS and WS-Security.

Also, consider archiving log files that are no longer required. System I/O is impacted by storing too many log files. By default, the Data Integration Services writes the web service run-time logs in the following directory: `<InformaticaInstallationDir>/tomcat/bin/disLogs/ws`

**Note:** If you delete the ws folder to empty the logs, you must re-create the ws folder. Stop the Data Integration Service before you delete and re-create the ws folder.
**Example Data Integration Service Configuration for Concurrent Web Service Requests**

When you configure how the Data Integration Service processes concurrent web services requests, verify that the value for the maximum number of concurrent requests is the same for the web service and the Data Integration Service process.

For example, in the following configuration the Data Integration Service accepts 200 concurrent HTTP requests but only 10 web service concurrent requests:

<table>
<thead>
<tr>
<th>Property Type</th>
<th>Property Name</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Integration Service Process</td>
<td>Maximum Concurrent Requests</td>
<td>200</td>
</tr>
<tr>
<td>Data Integration Service Process</td>
<td>Maximum Backlog Request</td>
<td>500</td>
</tr>
<tr>
<td>Data Integration Service Process</td>
<td>Maximum Execution Pool Size</td>
<td>100</td>
</tr>
<tr>
<td>Web Service</td>
<td>Maximum Concurrent Request</td>
<td>10</td>
</tr>
</tbody>
</table>

When the Data Integration Service receives 20 web service requests, 10 web service requests fail because the web service can only accept 10 concurrent requests.

To avoid web service requests failing when the web service reaches its maximum number of concurrent requests, configure the same maximum value for the Data Integration Service process and the web service. When the number of requests sent to the Data Integration Service exceeds the maximum concurrent requests value, the additional requests remain in the backlog until the Data Integration Service process is available to process the requests.

**Authors**

Padma Heid  
Lead Technical Writer

Lalitha Sundaramurthy  
Product Manager