How to Increase the Performance of Deployed Objects through Data Object Caching
Abstract

You deploy objects to make them accessible to end users. End users can make SQL queries against the virtual tables in a deployed SQL data service. Domain users can run deployed mappings at the command line. This article explains how to increase performance of the SQL queries and mappings by caching the data objects included in the SQL data service or mapping.

Supported Versions

- Informatica Data Services 9.0 - 9.0.1

Table of Contents

- Overview ................................................. 2
- Benefits of Data Object Caching ....................... 3
- Cache Tables ............................................. 3
- Cache Operations ........................................ 4
- Cache Optimization ...................................... 4
- How to Enable or Disable Data Object Caching .... 4
  - Logical Data Object and Virtual Table Properties ......................... 5
  - Enabling or Disabling Data Object Caching .................................. 5
- How to Configure Cache Properties .................... 5
  - Logical Data Object and Virtual Table Cache Properties ................. 6
  - Configuring Cache Properties ............................................... 6
- How to Monitor Cache Statistics ....................... 6
  - Viewing Logs for Logical Data Object Cache Refresh Runs .............. 6
  - Viewing Logs for Virtual Table Cache Refresh Runs ..................... 7

Overview

Deploy objects to a Data Integration Service to make mappings and SQL data services available for end users to run or query. Deployed objects can include logical data objects or virtual tables. Cache the logical data objects or virtual tables to increase the performance of mappings or SQL queries.

To make mappings and SQL data services available to end users, a developer adds the objects to an application, and deploys the application. A developers can also deploy a mapping or SQL data service directly, which causes the Developer tool to add the object to an application and deploy the application.

An application can contain mappings, SQL data services, or both. When the application is running on a Data Integration Service, domain users can run the mappings in the application, and end users can query the SQL data services.

By default, the Data Integration Service extracts source data and builds required data objects when an end user runs a mapping or queries an SQL data service. For example, a developer creates a virtual table that uses a logical data object as a source. When an end user runs an SQL query against the virtual table, the Data Integration Service extracts data from the source resources and builds the logical data object and virtual table. If the logical data object uses a read mapping to transform data, the Data Integration Service also runs the read mapping to populate the logical data object.

Logical data objects and virtual tables can use disparate resources as sources. The mappings that define data transformation can also be complicated. Therefore, logical data objects and virtual tables can take a long time to build.
You can improve mapping and query performance for end users by caching the logical data objects and virtual tables. 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, the mapping or SQL query runs more quickly. You enable data object caching for each logical data object or virtual table in an application. The Data Integration Service caches the objects in relational database tables.

**Benefits of Data Object Caching**

Enable caching for data objects that end users access frequently. When you enable data object caching, you distribute the workload involved in running a mapping or SQL query among multiple machines.

Enable data object caching to achieve the following results:

- Alleviate the load on the source system. You can schedule cache refreshes during periods of time when the load on the source system is likely to be low.
- Increase the availability of data objects. End users can run SQL queries and mappings when the source system is not available.
- Improve the performance of SQL queries and mappings. The Data Integration Service retrieves data quickly when it can access a local machine instead of a back-end data system. The Data Integration Service also avoids running the logical data object or virtual table read mappings when an end user runs an SQL query or mapping.

**Cache Tables**

The Data Object Cache Manager is the component of the Data Integration Service that creates cache tables. It creates cache tables in a relational database. Cache tables are read-only. End users cannot update the cache tables with SQL commands.

You can use the following databases to store data object cache tables:

- IBM DB2
- Microsoft SQL Server
- Oracle

Use the Administrator tool to configure the cache database for a Data Integration Service. Select a connection with the appropriate database type as the cache connection. For example, to store data object cache tables in an Oracle database, create a connection to the Oracle database, and select the connection as the cache connection for the Data Integration Service.

When you cache data objects in the cache database, the Data Object Cache Manager creates one table for each cached logical data object or virtual table in an SQL data service or mapping.

If a logical data object uses another logical data object as a source and you enable caching for the derived object, you can cache the source object as well. For example, logical data object LDO3 joins data from logical data objects LDO1 and LDO2. A developer creates a mapping that uses LDO3 as a source and includes the mapping in an application. You can cache LDO1, LDO2, and LDO3. If you cache LDO3, the Data Integration Service uses cache data when an end user runs the mapping. If you cache LDO1 and LDO2, but not LDO3, the Data Integration Service does not use cache data when an end user runs the mapping.

Objects within an application share cache tables, but objects in different applications do not. If one data object is used in multiple applications, the Data Object Cache Manager creates a separate cache table for each instance of the data object.
Cache Operations

You can configure when and how the Data Object Cache manager populates the cache through the Administrator tool or infacmd. You can configure when the Data Object Cache manager first loads data to the cache and when and how it refreshes the cache. You can also abort a cache refresh and purge cache data.

You can perform the following operations on the data object cache:

**Load the cache for the first time**

The Data Object Cache manager creates cache tables when you configure the data objects for caching and enable the application in the Administrator tool. The Data Object Cache Manager loads data for logical data objects and virtual tables into the cache tables.

**Refresh the cache**

You can refresh the cache for a data object according to a schedule or manually. To refresh data according to a schedule, set the cache refresh period for the logical data object or virtual table in the Administrator tool. To refresh the cache manually, use the infacmd dis RefreshDataObjectCache command.

When the Data Object Cache Manager refreshes the cache, it creates a new cache. If an end user runs a mapping or queries an SQL data service during a cache refresh, the Data Integration Service returns information from the existing cache.

**Abort a refresh**

To abort a cache refresh, use the infacmd dis CancelDataObjectCacheRefresh command. If you abort a cache refresh, the Data Object Cache Manager restores the existing cache.

**Purge the cache**

To purge the cache, use the infacmd dis PurgeDataObjectCache command. You must disable the application before you purge the cache.

Cache Optimization

Optimal performance for the cache depends on the speed and performance of the cache database, and the cache size. Configure the cache size within the cache database.

Because the Data Object Cache Manager must maintain the old cache during a refresh, the cache must be large enough to store two sets of data. Use the following formula to estimate the required minimum cache size:

\[
2 \times \text{average data object size} \times \text{number of data objects}
\]

For example, you want to cache 20 logical data objects and 10 virtual tables. The average data object size is 15 MB. The required cache size is \(2 \times 15 \text{ MB} \times (20 + 10) = 900 \text{ MB}\).

How to Enable or Disable Data Object Caching

You enable or disable data object caching for each logical data object or virtual table in an application through the Administrator tool.

When you enable caching for a logical data object or virtual table, you also set the cache refresh time for the object. You must disable the SQL data service or application that contains the object before you configure logical data object or virtual table properties.
Logical Data Object and Virtual Table Properties

Logical data object and virtual table properties control whether to cache a logical data object or virtual table and how often to refresh the cache.

The following table describes the logical data object and virtual table properties:

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

Enabling or Disabling Data Object Caching

To enable or disable caching, stop the application that contains the logical data object or virtual table, edit the data object properties, and restart the application.

1. In the Administrator tool, select the Data Integration Service.
2. Click the Applications view.
3. Select the application that contains the logical data object or virtual table for which you want to enable or disable caching.
4. Stop the application.
5. Expand the application that contains the data object for which you want to enable or disable caching.
6. Select the logical data object or virtual table.
7. Edit the logical data object or virtual table properties.
8. Click OK.
9. Restart the application.

How to Configure Cache Properties

You configure cache properties for a Data Integration Service through the Administrator tool.

You configure the cache removal time, cache connection, and maximum number of concurrent refresh requests for the Data Integration Service. You do not have to stop the Data Integration Service to configure the logical data object and virtual table cache properties.
Logical Data Object and Virtual Table Cache Properties

Logical data object and virtual table cache properties control the cache properties for the Data Integration Service. The following table describes the data object and 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</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>

Configuring Cache Properties

To configure the cache removal time, cache connection, and maximum number of concurrent refresh requests, select the Data Integration Service and edit the cache properties.

1. In the Administrator tool, select the Data Integration Service.
2. Click the Properties view.
3. Edit the Logical Data Object/Virtual Table cache properties.
4. Click OK.

How to Monitor Cache Statistics

When you enable data object caching, you can monitor cache statistics such as the cache run ID, the request count, and the cache hit rate. The cache hit rate is the total number of requests on the cache divided by the total number of requests for the data object.

Use the Monitoring tab in the Administrator tool to view information about data objects in an application. The Cache Refresh Runs view displays cache information for the selected data object.

Viewing Logs for Logical Data Object Cache Refresh Runs

You can download the logs for logical data object cache refresh runs to view the cache refresh run details.

1. In the Administrator tool, click the Monitoring tab.
2. In the Navigator, expand a Data Integration Service.
3. In the Navigator, expand an application and select Logical Data Objects.
   The contents panel displays a list of logical data objects.
4. In the contents panel, select a logical data object.
   Details about the selected data object appear in the details panel.
5. In the details panel, select the Cache Refresh Runs view.
6. In the details panel, click View Logs for Selected Object.
Viewing Logs for Virtual Table Cache Refresh Runs

You can download the logs for virtual table cache refresh runs to view the cache refresh run details.

1. In the Administrator tool, click the Monitoring tab.
2. In the Navigator, expand a Data Integration Service.
3. In the Navigator, expand an application and select SQL Data Services.
   The contents panel displays a list of SQL data services.
4. In the contents panel, select an SQL data service.
5. Click the Virtual Tables view.
6. In the contents panel, select a virtual table.
   Details about the selected virtual table appear in the details panel.
7. In the details panel, select the Cache Refresh Runs view.
8. In the details panel, click View Logs for Selected Object.

Authors

Lori Troy
Senior Technical Writer

Andy Lai
Staff Engineer

Acknowledgements

The authors would like to acknowledge Chris Lay and Raymond To for their technical assistance.