ELO server – Installation and operation

Optimization

Table of contents

ELO metadata tables **3**

Getting started 3
Problem 4
Solution 5
Implementation 6
Effects on other ELO modules 10
Restrictions 11

ELO Indexserver load balancing **12**

Information 12

ELO Indexserver SSO **13**

Information 13

ELO metadata tables

Getting started

It can take a long time to search multiple metadata fields if the SQL Optimizer does not recognize the strict search criteria in the query. It can help improve performance if you organize the field values in a metadata table. Unlike the "objkeys" table, the field values are stored in the columns of the database table instead of in the rows.

> **Information**
>
> This documentation applies to ELO 10 and higher.

This document describes how to improve searching multiple metadata fields by organizing the field values as a metadata table.

This documentation:

- Describes the problem
- Presents the solution
- Explains the implementation
- Describes the effects on other ELO modules
- Explains the restrictions

Problem

It can take a long time to search multiple index fields in the *Objkeys* table if the SQL Optimizer does not recognize the strict search criteria in the query.

In our use case, we perform searches using the bank code, account number, and a time range. The *objkeys* table contains 370 million rows. The search takes 1-3 seconds if the SQL Server has 48 GB RAM. It takes 50 seconds if the server has 16 GB RAM. In Oracle Enterprise Manager, we can see that the priority ranking of the search is the bank code followed by the time range and finally the account number. This means that the Optimizer does not recognize that the search should first find the account number, which is unique.

Solution

Instead of one row per index value and multiple rows per document when storing the field values to the *objkeys* table, the field values are stored in a "flat" table with one column per field value and one row per document. Performance tests based on the e-mail data in our live repository have shown a five-fold increase in speed.

Implementation

New metadata form property: Data organization

The metadata form uses the `DocMask.dataOrganisation` field to determine how field values should be stored:

Data organization	DB storage
`DATA_ORGANISATION_OBJKEYS = 0;`	Conventional storage as name-value pairs in the *objkeys* table
`DATA_ORGANISATION_TABLE = 1;`	Metadata table, one table for the metadata form (flat table), one column per field.

Converting existing field data

You can convert the field values of an existing metadata form into a metadata table using the following code.

The index values are converted by setting the element

```
dataOrganisation = DocMaskC.DATA_ORGANISATION_TABLE
```

and then checking in the metadata form.

In a background job, the ELO Indexserver moves the data from the *objkeys* table to the metadata table. The job status is listed under the GUID of the metadata form. It can be retrieved with `queryJobState`.

Information

1. It is not possible to convert fields with invalid group names, nor is it possible to convert fields with duplicate group names.
2. Currently, it is not possible to convert a metadata table back into an *objkeys* table.
3. For fields with multiple field values (column index), only the first value is transferred to the metadata table.
4. During the update, no changes should be made to the folder or document metadata that are assigned metadata with the metadata form. However, you can create new documents and folders.
5. The original file name is stored in the *objkeys* table (ELO_FNAME).
6. On startup, the Indexserver checks whether a conversion process was interrupted. If this is the case, the Indexserver continues the conversion by evaluating an entry in the *eloixopt* table with

```
IXID=_ALL,
OPTNAME=dataOrganisationTransfer_<maskid>,
OPTVALUE=<ixid>.
```

Example

```
public static void changeDataOrganisation(IXConnection conn,
    String maskId) {
  log.info("start change data organisation");
  try {
    long t1 = System.currentTimeMillis();

    DocMask dmTable = conn.ix().checkoutDocMask(maskId, DocMaskC.mbAll,
                                                LockC.NO);
    dmTable.setDataOrganisation(DocMaskC.DATA_ORGANISATION_TABLE);
    conn.ix();

    JobState jobState = conn.ix().queryJobState(dmTable.guid, true, true,
                                                true);
    while (jobState.isJobRunning() {
      Thread.sleep(1000);
      jobState = conn.ix();
      log.info("jobState=" + jobState);
    }

    long t2 = System.currentTimeMillis();
    log.info("end change data organisation, ms=" + (t2 - t1));
  } catch (Exception e) {
  log.error("Failed to convert data organisation", e);
  }
}
```

Representation in the database

The metadata table is created under the name "Keywording_" plus the metadata form ID. The fields are represented as columns in the table.

Example

```
CREATE TABLE [dbo].[keywording_239](
  [parentid] [int] NULL,
  [rechnr] [nvarchar](255) NULL,
  [rechdat] [nvarchar](255) NULL,
```

```
  [liefernr] [nvarchar](255) NULL,
  [bestellnr] [nvarchar](255) NULL,
  [betreff] [nvarchar](255) NULL,
  [kdnr] [nvarchar](255) NULL,
  [lieferant] [nvarchar](255) NULL,
  [barcode] [nvarchar](255) NULL,
  [betrag] [nvarchar](255) NULL,
  [mitarb] [nvarchar](
)
```

Database indexes

The Indexserver does not create database indexes on metadata tables. This is the sole responsibility of the database administrator.

Case-insensitive search with Oracle

To perform a case-insensitive search with Oracle, function-based indexes have to be used. Text columns in the index are added as an UPPER function argument.

A suitable index could look as follows:

```
CREATE INDEX "ARCHIV1"."IXKEYWORDING_32_01" ON "ARCHIV1"."KEYWORDING_32"
 (
    UPPER(BANK),
   UPPER(KTO),
   PARENTID
 )
TABLESPACE "TS_ELO_INDEX";
```

Editing the metadata form definition

When you add a field in the ELO Administration Console, a column is added to the metadata table.

> **Please note**
>
> This column contains NULL values. Setting a default value for a field would take a very long time for large tables. Existing objects are not assigned default values for this reason. They only apply for new folders and documents.
>
> Inversely, when you delete a field, the Indexserver removes the corresponding column from the table.
>
> All field data is lost.

Information

Like the metadata forms with *objkeys* data, metadata forms with metadata tables can only be deleted if no more objects have this metadata.

When deleting a metadata form, if you specify an alternative one, the data is not converted automatically.

MEtadata forms are not deleted permanently.

Searching metadata tables with "findfirstSords"

1. The table to be searched is defined based on the metadata form specified in `FindByIndex`.
2. If no metadata form is specified in the search, the metadata forms are determined using the field group names from the search query.
    - A search is run on the metadata tables that contain all the field groups specified in `FindByIndex.objKeys`.
    - If a metadata form with *objkeys*-based data contains all of the fields, the search is also performed on the *objkeys* table.
    - If no metadata form contains all of the fields in the search, only *objkeys* is searched.
3. If metadata forms are specified in the `FindByIndex`, only these forms are searched. Enter the forms to `FindByIndex.maskId` or the `FindByIndex.maskIds` array.
4. It is not currently possible to search all metadata table fields, nor is it possible to combine fields in the search with an OR.

Effects on other ELO modules

Indexserver client applications, ELO Java Client, ELO Web Client, etc.

The field values from metadata tables are also returned or accepted in `Sord.objKeys`. Changes to existing client applications are not necessary.

ELO Windows Client

ELO is not planning to add metadata table functions to the ELO Windows Client. For this reason, metadata tables can only be used in installations without the ELO Windows Client.

ELO Replication

The ELO Replication module does not currently support metadata table synchronization.

Restrictions

- Only one value per field possible (no column index)
- During conversion, only the first value in a field is taken from the *objkeys* table
- It is not possible to search all fields
- (Re)conversion of metadata tables into the *objkeys* table not possible
- OR-linked search not possible
- During conversion, no information can be added to or edited in existing metadata form entries
- Only installations without the ELO Windows Client or ELO Replication

ELO Indexserver load balancing

Information

This documentation has been moved.

You can now find the documentation on the ELO Indexserver page.

ELO Indexserver SSO

Information

This documentation has been moved.

You can now find the documentation on the ELO Indexserver page.