Querying ClearQuest Oracle 10g databases

Controlling case-sensitivity for queries on multi-line text fields

Lionel Silberman

July 05, 2007

In IBM® Rational® ClearQuest® environments using Oracle® version 10g or later, administrators can use a feature in ClearQuest version 7.0.1 to allow users to run queries on multi-line text fields and return case-insensitive results. This article also describes how to revert to case-sensitive queries for your entire database index or for only one field.

By default, IBM® Rational® ClearQuest® queries that are run against multiline text fields in Oracle databases can return only results that are case-sensitive, because the database uses the case-sensitive Character Large Object (CLOB) data type to store those fields. Therefore, a query for Test will not find a match with test or TEST.

However, if you are using Oracle Database 10g or later, you can use a feature available in both ClearQuest and ClearQuest MultiSite Version 7.0.1 to get case-insensitive results. When this feature is enabled, a query for Test will find matches with test, TEST, and any other combinations of uppercase and lowercase letters, as long as the spelling is the same.

Note: This ClearQuest feature is not available for Oracle databases earlier than the Database 10g version.

Enabling case-insensitive queries involves four basic steps:

1. Identify all multiline text fields in the user database.
2. Get necessary permissions from database and ClearQuest administrators.
3. Build indexes for all multiline text fields that will be indexed.
4. Enable the feature by adding a behavior flag in the table, creating if needed first.
5. Run an SQL script to build the indexes for all multiline text fields that will be used for queries.

This article describes this process in more detail and also explains how to revert to the default case-sensitive queries.

Prerequisites

- You will need specific software tools to create the indexes for these case-insensitive queries on multiline text fields in Oracle databases. Therefore, you must have access to this software:
• Rational ClearQuest Designer
• Oracle client software, Version 10 or later
• Oracle SQL*Plus interface, Version 10 or later

• To run the utilities that build the indexes, the Rational ClearQuest administrative host must have version 7.0.1 installed on Microsoft Windows. The ClearQuest software can be on the same system as the Oracle client and SQL*Plus interface, but it does not have to be.

• To enable queries that can return case-insensitive results, you must be using a language with a code page that includes ASCII characters and that is supported by Rational ClearQuest. These languages and their associated code page values are listed in Table 1.

### Table 1. Code pages and associated languages

<table>
<thead>
<tr>
<th>Code page</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>20127 (US ASCII)</td>
<td>English</td>
</tr>
<tr>
<td>Safe Shift-JIS (60932)</td>
<td>Japanese</td>
</tr>
<tr>
<td>932</td>
<td>Japanese</td>
</tr>
<tr>
<td>936</td>
<td>Simplified Chinese</td>
</tr>
<tr>
<td>949</td>
<td>Korean</td>
</tr>
<tr>
<td>950</td>
<td>Traditional Chinese</td>
</tr>
<tr>
<td>1250 (Eastern Europe)</td>
<td>Czech, Croatian, Hungarian, Polish, Romanian, Slovak, Slovenian, Albanian</td>
</tr>
<tr>
<td>1251 (Cyrillic)</td>
<td>Belarussian, Bulgarian, Macedonian, Russian, Serbian, Ukrainian</td>
</tr>
<tr>
<td>1252 (Western Europe)</td>
<td>Spanish, Danish, German, English, French, Italian, Norwegian, Dutch, Portuguese, Swedish</td>
</tr>
<tr>
<td>1253</td>
<td>Greek</td>
</tr>
<tr>
<td>1254</td>
<td>Turkish</td>
</tr>
<tr>
<td>1255</td>
<td>Hebrew</td>
</tr>
<tr>
<td>1257 (Baltic)</td>
<td>Estonian, Lithuanian, Latvian</td>
</tr>
</tbody>
</table>

For more information about code pages, see Administering Rational ClearQuest in the Resources section, under ClearQuest online documentation.

#### Enabling queries that are not case-sensitive

These tasks are typically performed by the Rational ClearQuest administrator, who understands the schema and the fields that are used for queries, with the assistance of an Oracle database administrator, who understands Oracle databases and the SQL*Plus interface.

#### Step 1. Identify all multiline text fields in the user database

To begin the process of enabling case-insensitive queries, review the schema of the user database and identify all of the multiline text fields. Some fields that appear to be short strings are actually multiline text fields, for example, the Keywords and Symptom fields in the Defect Tracking schema. One way of determining which fields are multiline text fields is to use the Fields grid. To do this:
1. Start the Rational ClearQuest **Designer**.
2. Open the **schema**.
3. Expand each **record type**.
4. Click the **Fields** icon in the **Workspace** pane to display the Fields grid.
5. In the **Fields** grid, check for **MULTILINE_STRING** in the **Type** column and make a list of fields that are **MULTILINE_STRING**.

If you have a large number of multiline text fields in your databases, you can identify multiline text fields using the **PDSQL** utility, which is installed with Rational ClearQuest on Windows. To identify all of the multiline text fields by using the PDSQL command-line utility:

6. Create a file with one semicolon-terminated command of columns `<tablename>`. For example:

   ```
   C:\testpdsql.scr columns defect;
   ```

7. Redirect the file into PDSQL using the `<` redirection operator, and pipe through a `grep` command, looking for CLOB fields.
8. Then redirect the file into the PDSQL command-line utility by using the `<` redirection operator, and pipe the output through a `grep` command, looking for CLOB multiline text field. Here is an example of the command:

   ```
   C: ClearQuest>pdsql -v ora -s <host> -db <dbname> -u <cqschema_user>
   -p <pw> < c:\testpdsql.scr | grep CLOB
   ```

   The system generates a list of all of the CLOB fields found in the defect table (see Table 2).

**Table 2. CLOB fields found in the defect table**

<table>
<thead>
<tr>
<th>Type</th>
<th>CLOB field</th>
</tr>
</thead>
<tbody>
<tr>
<td>DESCRIPTION</td>
<td>CLOB (2147483647,0)</td>
</tr>
<tr>
<td>KEYWORDS</td>
<td>CLOB (2147483647,0)</td>
</tr>
<tr>
<td>SYMPTOMS</td>
<td>CLOB (2147483647,0)</td>
</tr>
<tr>
<td>NOTE_ENTRY</td>
<td>CLOB (2147483647,0)</td>
</tr>
<tr>
<td>NOTES_LOG</td>
<td>CLOB (2147483647,0)</td>
</tr>
<tr>
<td>IMPL_COMPLETED_BY</td>
<td>CLOB (2147483647,0)</td>
</tr>
</tbody>
</table>

For more information on how to use the **PDSQL** utility, see IBM Technote 1145079, **Information about PDSQL and how to use it**.

**Step 2. Get permissions from administrators**

Before you can begin to build indexes for multiline text fields, a database administrator must grant permissions to build indexes to the Rational ClearQuest administrator who manages the user database. The database administrator grants these permissions by specifying the Oracle database administrator's user name and password, as well as the TNSName of the user database. Then the database administrator runs the `setup_cq_ora_user_clob.sql` script and specifies the Oracle user...
login ID created for the user database. The syntax is as follows (this is a single command that may wrap across lines on your screen):

```
> sqlplus <oracle_DBA_user>/<password>@<tnsnamesentry>@setup_cq_ora_user_clob.sql
<oracle_user>
```

### Step 3. Build indexes for all multiline text fields that will be indexed.

In Oracle Database 10g, multiline text fields that do not have an index cannot be used in queries. When you build an index to enable case-insensitive queries on any multiline text field, you must build an index for every multiline text field that is used in queries. However, not all queries need to be case-insensitive. However, not every multiline text field needs to support case-insensitive queries. When you create the index for each field, you can decide whether queries will be case-insensitive or case-sensitive. When you run the `setup_cq_clob_index` SQL script, you can choose which fields will have queries that are case-insensitive and which fields will have queries that are case-sensitive.

For example, you could build an index that supports case-insensitive queries on the **Description** field and an index that supports case-sensitive queries on the **Keyword** field. However, if you do not build an index for the Keyword field, and then you try to run a query on it, the query will fail to execute and the system will report an error.

Use the `setup_cq_clob_index.sql` script to build indexes for multiline text fields. **You must build an index for every multiline text field in the user database where a text query is used as a filter.**

**Tip:** The same script creates indexes for both case-insensitive and case-sensitive queries.

The `setup_cq_clob_index.sql` script is stored in the Rational ClearQuest directory, which may have this path or a different one in your setup:

```
C:\Program Files\Rational\ClearQuest
```

For each multiline text field, run the `setup_cq_clob_index.sql` script and follow these steps:

1. Specify the ClearQuest administrator's Oracle user login and password and the TNSName of the user database, as follows:

```
> sqlplus <oracle_user>/<password>@<tnsnamesentry>@setup_cq_clob_index.sql
```

2. The script will prompt you to either drop an existing index or create a new one. Indicate that you want to create a new multiline text index.
3. Enter the Rational ClearQuest record **type name**. For example, enter **defect**.
4. Enter the Rational ClearQuest **field name**. For example, enter **description**.
5. Indicate that the user database is Oracle Version 10 or later.
6. Enter the **BASIC_LEXER** type. A **lexer** is a software component that divides text strings into individual words, or tokens, so that the individual words can be indexed. A listing of supported lexer types follows. (The other choices support indexing for non whitespace delimited languages which do not have case.)
7. Indicate that the index should be **case-insensitive**. With this last step, your index for the multiline field is complete.

These are the types of lexers that are supported:

- **BASIC_LEXER**: The lexer for English and most western European languages that use white space delimited words.
- **CHINESE_VGRAM**: A lexer for extracting tokens from Chinese text.
- **CHINESE_LEXER**: A lexer for extracting tokens from Chinese text. An improved version of CHINESE_VGRAM.
- **JAPANESE_VGRAM**: A lexer for extracting tokens from Japanese text.
- **JAPANESE_LEXER**: A lexer for extracting tokens from Japanese text. An improved version of JAPANESE_VGRAM.
- **KOREAN_LEXER**: A lexer for extracting tokens from Korean text.
- **KOREAN_MORPH_LEXER**: A lexer for extracting tokens from Korean text. An improved version of KOREAN_LEXER.

For information on these lexers, consult the Oracle documentation.

**Step 4. Enable the feature by adding a behavior flag in the table, creating if needed first.**

Before you can build an index for a multiline text field, you must add a new table to the database and add a behavior value that allows indexing to that table. Using Oracle's SQL*Plus interface, follow these steps:

1. Log on to the user database with the login ID of an Oracle user who has permission to create tables and to specify the TNSName. The Oracle user is the same user name and password that you entered when you created the user database with the Rational ClearQuest Maintenance Tool. The **tnsnamesentry** is the database name stored in the **tnsnames.ora** file.

   ```sql
   > sqlplus <oracle_user>/<password>@<tnsnamesentry>
   ```

2. Add a table with these column and row entries.

   ```sql
   > sqlplus create table ratl_diagnostics (diag_name VARCHAR2(16), diag_value VARCHAR2(255));
   ```

   **Note**: The system fails to create the table if a table with the same name already exists in the user database.

3. Insert the following row into the **ratl_diagnostics** table. This row includes a value that allows indexes to be created for multiline text fields.

   ```sql
   > sqlplus insert into ratl_diagnostics(diag_name, diag_value)
   values ("Behavior", "CLOB_CASE_AND_INDEX");
   ```
4. Commit the change to the database.

> sqlplus commit;

**Note:** For information on the syntax parameters for SQL*Plus, see Oracle documentation.

**Other index considerations**

- **Test performance:** Be sure to test the index performance with a test database. Indexes typically speed up queries, sometimes reducing the time required to execute the query to a fraction of the former elapsed time. However, indexes can either increase or decrease the performance of operations to add, change, and delete database records. To make sure that indexes have not unduly decreased performance of your user database for these operations, create a test database, build the indexes, and then measure performance when you add, change, and delete Rational ClearQuest records. In the unlikely event that the overall change in database performance is negative, you can consider dropping indexes from some fields and informing users which multiline text fields are not available for queries.

- **Indexing for ClearQuest MultiSite:** If you are using Rational ClearQuest MultiSite, then when you create and move databases you must re-index all multiline text fields in the new databases. The indexes that support multiline fields are not replicated from the working master database to user databases. Therefore you must go through the entire process for enabling case-insensitive queries for every user database. To simplify the first step of this process (identifying all multiline text fields in the user database), maintain a list of all multiline text fields that have been indexed in the working master database.

- **Re-index when moving databases:** Whenever user databases are moved, the indexes that support multiline text fields are lost. Therefore, you must re-index all multiline text fields after moving databases. You can re-index the multiline text fields by following the same procedures used to create the indexes on the original user database. Save a list of all of the multiline text fields in the original database so that you don't have to create it again each time you move a database. Alternatively, you can write a script to build all of the indexes in one pass on the database that has been moved or upgraded. See the **Downloads** section for a sample script.

- **Noise words:** When you create indexes, you need to consider what Oracle calls "noise words." These are grammatically necessary words that provide little useful information, such as **of, the, it, is, or you.** To save space, noise word references are not maintained within the index, but they are stored in a customizable list by language. When you create indexes, they will automatically use a default stop list to ignore noise words. A default list of noise words is supplied for each major European language. You can alter the supplied list of noise words or define your own. For more information on noise words, check the Oracle documentation.

**Disabling the feature**

After case-insensitive queries have been enabled, there are also procedures for changing all or just part what you set up by using several methods:
• Dropping indexes
• Revert to case-sensitive queries for only one field
• Disabling the feature
• Revoking a user’s permission to build indexes

Note: Disabling the feature should be done before dropping all indexes and if desired, revoking permissions of the user account, so that no user errors will result.

Drop indexes

After you have built indexes to enable case-insensitive queries, you may later want to be able to remove individual indexes. You can drop an index by following these steps:

1. Specify the ClearQuest administrator's Oracle user login ID and password and the TNSName of the user database, and then run the `setup_cq_clob_index.sql` script:

```
> sqlplus <oracle_user>/<password>@<tnsnamesentry>@setup_cq_clob_index.sql
```

2. The script will prompt you to either drop an existing multiline text index or to create a new multiline text index. Indicate that you want to drop an existing multiline text index.
3. Enter the Rational ClearQuest record type name. For example, enter defect.
4. Enter the Rational ClearQuest field name. For example, enter description.

Important: When you drop the index for a multiline text field, that field can no longer be used for queries. Therefore, you may need to notify users and to modify queries that use this field.

Revert to case-sensitive queries for only one field

After you have enabled case-insensitive queries for a multiline text field, you can revert to case-sensitive queries. If you no longer want to enable case-insensitive queries on any field in the table, you can disable indexes for the entire table. You can also revert to case-sensitive queries for only one multiline text field by changing the index:

1. Use the `setup_cq_clob_index.sql` script to drop the existing index.
2. Run the `setup_cq_clob_index.sql` script again
3. In the final step, indicate that the new index should support case-sensitive queries.

Disabling the feature

If you no longer want to enable case-insensitive queries on any field in a table, you can revert to the Oracle default setting of case-sensitive queries by disabling indexes on all of the multiline text fields in the table. To disable the use of all indexes in a table, delete the value in the `ratl_diagnostics` table that allows indexes to be created for multiline text fields:

1. At the prompt, log on to the user database with your Oracle user login ID, password, and the TNSName of the user database as follows:

```
> sqlplus <oracle_user>/<password>@<tnsnamesentry>
```
The **tnsnamesentry** is the database name stored in the **tnsnames.ora** file.

2. Enter this command to delete the value in the table that allows indexes to be created for multiline text fields.

```sql
> sqlplus delete from ratl_diagnostics where diag_name "Behavior" and diag_value = "CLOB_CASE_AND_INDEX"
```

3. Commit the change to the database:

```sql
> sqlplus commit;
```

### Revoke permission to build indexes

The database administrator can revoke the permission that was granted to a Rational ClearQuest user to build indexes. This person needs to specify the Oracle user login ID of the user. To revoke permission to build indexes, use these SQL commands:

```sql
> sqlplus <Oracle_DBA_user>/<password>@tnsnamesentry
> sqlplus revoke SELECT ON ctxsys.ctx_indexes from <cq_username>;
> sqlplus revoke ctxapp from <oracle_user>;
```
## Downloadable resources

<table>
<thead>
<tr>
<th>Description</th>
<th>Name</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample script to build database indexes</td>
<td>ClearQuestScript-OracleDB10g.zip</td>
<td>2KB</td>
</tr>
</tbody>
</table>

© Copyright IBM Corporation 2007  
**Trademarks**  
(www.ibm.com/developerworks/ibm/trademarks/)