Subject: In Oracle9 and higher, TIMESTAMP data type will cause ORA-1402 exception w/ Oracle8 Oci8.cpp
Posted by alendar on Fri, 24 Dec 2010 00:35:44 GMT
View Forum Message <> Reply to Message

Hi,

I was getting a "ORA-01406: fetched column value was truncated" when I used the Oracle8 session driver, but not with the Oracle7 code. I am connecting to an Oracle 11 database.

The bug appears when a column of datatype TIMESTAMP is referenced in an Execute command, and a Fetch command is called.
Here's the client code I wrote that triggered the exception:

```
if (!cursor.Execute(sql)) {
 HandleDbError(ACTNDB_EXECSEL, cursor, &sql);
 return false;
}

int colCount = cursor.GetColumns();

for (int i = 0; i < colCount; i++) {
 const SqlColumnInfo& ci = cursor.GetColumnInfo(i);
 int w = GetTextSize(ci.name, StdFont()).cx + 14;
 outputGrid->AddColumn(ci.name).Width(w);
 cw[i] = w;
}

Progress pi;
pi.SetText("Fetched %d line(s)");

outputGrid->Ready(false);

while(cursor.Fetch()) {

...

}

if (cursor.WasError()) {
 HandleDbError(ACTNDB_EXECSEL, cursor, &sql);
 return false;
}
```

The error occurs at the Fetch, not at the Execute.

Test SQL was:

select t.SRCS_FD_TS from ...

I traced the DLL calls to:

GetProcAddress(0x025D0000 [ORACLIENT10.DLL], "OCIDefineByPos") called from "OCI.DLL" at address 0x02572E69 and returned 0x025D68CE.
GetProcAddress(0x025D0000 [ORACLIENT10.DLL], "OCIStmtFetch") called from "OCI.DLL" at address 0x02572F95 and returned 0x025D6962.
GetProcAddress(0x025D0000 [ORACLIENT10.DLL], "OCIErrorGet") called from "OCI.DLL" at address 0x02573295 and returned 0x025D6B1E.

OCIDefineByPos is called in Oracle\Oci8.cpp in OCI8Connection::GetColumnInfo().

Snippet from Oci8.cpp:

```
 ...
 switch(type) {
 ...
 case SQLT_CLOB:
  ii.type = ORA_CLOB_V;
  AddColumn(SQLT_CLOB, sizeof(OCILobLocator *));
  blob = true;
  break;
 case SQLT_RDD:
  ii.type = STRING_V;
  AddColumn(SQLT_STR, 64);
  break;

 default:
  ii.type = STRING_V;
  AddColumn(SQLT_STR, ii.width ? ii.width + 1 : longsize);
  break;
 }

 Item& c = column.Top();
 oci8.OCIDefineByPos(stmthp, &c.define, errhp, i,
 blob ? (void *)&c.lob : (void *)c.Data(), blob ? -1 : c.total_len,
 c.type, c.ind, c.len, NULL, OCI_DEFAULT);
}
```

The column type returned from
oci8.OCIAttrGet(pd, OCI_DTYPE_PARAM, &type, NULL, OCI_ATTR_DATA_TYPE, errhp);
is 187, or SQLT_TIMESTAMP.

Since type 187 is not trapped, the default action is to treat as string, which is fine, except ii.width returns 11, which is probably the binary width of the data. The string width varies according to Locale settings. On mine the output length is 28, but with a 4-digit year it would be 30.

Here's a sample value:
03-DEC-10 12.00.00.000000 AM

I added a case path and it corrected the problem:

```
 case SQLT_TIMESTAMP: // type 187
  ii.type = STRING_V;
  AddColumn(SQLT_STR, 30);
  break;
```

My Oracle client is 10.2.0.1 (dated 9/30/2007 8:09a) on Windows XP SP3.

Please add patch if you think it would help anyone else.
