

p_manage_31: Deadlock Errors

- **Article Type:** General
- **Product:** Aleph
- **Product Version:** 17.01

Description:

Every so often when I run the Authorities update (p_manage_31), I will see Deadlock errors in the Oracle alert log for aleph17 on our production server. I ran the job yesterday morning, and the following errors are in the alert log:

Wed Oct 17 08:48:51 2007

ORA-00060: Deadlock detected. More info in file /exlibris/app/oracle/admin/aleph17/udump/aleph17_ora_27267.trc.

Wed Oct 17 08:48:54 2007

ORA-00060: Deadlock detected. More info in file /exlibris/app/oracle/admin/aleph17/udump/aleph17_ora_13715.trc.

Wed Oct 17 08:48:54 2007

ORA-00060: Deadlock detected. More info in file /exlibris/app/oracle/admin/aleph17/udump/aleph17_ora_27267.trc.

Wed Oct 17 08:48:57 2007

ORA-00060: Deadlock detected. More info in file /exlibris/app/oracle/admin/aleph17/udump/aleph17_ora_13715.trc.

Wed Oct 17 08:49:55 2007

A snippet from one of the trace logs is below:

*** 2007-10-17 08:48:51.584

*** SESSION ID:(142.122) 2007-10-17 08:48:51.575

DEADLOCK DETECTED (ORA-00060)

The following deadlock is not an ORACLE error. It is a deadlock due to user error in the design of an application or from issuing incorrect ad-hoc SQL. The following information may aid in determining the deadlock:

Current SQL Statement:

```
update SYS10.z01 set z01_rec_key = :r1:i1,z01_acc_sequence = :r2:i2,z01_hash = :r3:i3,z01_aut_tag = :r4:i4,z01_rec_key_4 = :r5:i5,z01_acc_sequence_see = :r6:i6,z01_number_of_doc = :r7:i7,z01_cataloger_name = :r8:i8,z01_cataloger_level = :r9:i9,z01_open_date = :r10:i10,z01_update_date = :r11:i11,z01_cataloger_library = :r12:i12,z01_non_filing_char = :r13:i13,z01_update_doc = :r14:i14,z01_update_z0102 = :r15:i15,z01_ref_type = :r16:i16,z01_normalized_text = :r17:i17,z01_display_text = :r18:i18 where z01_acc_sequence = :v1
```

End of information on OTHER waiting sessions.

Current SQL statement for this session:

```
update SYS10.z01 set z01_rec_key = :r1:i1,z01_acc_sequence = :r2:i2,z01_hash = :r3:i3,z01_aut_tag = :r4:i4,z01_rec_key_4 = :r5:i5,z01_acc_sequence_see = :r6:i6,z01_number_of_doc = :r7:i7,z01_cataloger_name = :r8:i8,z01_cataloger_level = :r9:i9,z01_open_date = :r10:i10,z01_update_date = :r11:i11,z01_cataloger_library = :r12:i12,z01_non_filing_char = :r13:i13,z01_update_doc = :r14:i14,z01_update_z0102 = :r15:i15,z01_ref_type = :r16:i16,z01_normalized_text = :r17:i17,z01_display_text = :r18:i18 where z01_acc_sequence = :v1
```

=====

Resolution:

Is ue_01 running when this problem occurs? Looking at \$aleph_proc/p_manage_31, I find that p_manage_31 does **not** stop ue_01. Perhaps it should. We have seen other such cases -- for example, when p_manage_50 is run in the xxx50 library and ue_01 is running there.

We suggest that you stop ue_01 in the xxx10 library before running p_manage_31 there.

(And that you stop ue_01 in the ADM library before running p_manage_50 there.)

- **Article last edited:** 10/8/2013