

---

## Oracle error: io\_sequence\_next; ORA-02289: sequence does not exist

- **Article Type:** General
  - **Product:** Aleph
  - **Product Version:** 20, 21, 22, 23
- 

### Description:

In the pc\_server log, these series of errors display:

```
Oracle error: io_sequence_next
```

```
ORA-02289: sequence does not exist
```

```
Oracle error: io_znn_write
```

```
ORA-00001: unique constraint (xxx50.Znn_ID) violated
```

where "io\_znn\_write" is: io\_z309\_write, io\_z31\_write, io\_z45\_write, io\_z601\_write, or io\_z71\_write. And "xxx50" is your ADM library.

What do we do about this error?

### Resolution:

This error indicates that the last\_record\_sequence is not defined in the \$pw\_library.

This sequence is used by the five tables shown above as part of their record key (to make the key unique). (\*See more in Additional Information below.) The last\_record\_sequence is in the delivered ./aleph/tab/file\_list.PW \$pw\_library file\_list template and looks like this:

```
SEQ last_record_sequence
```

If you have a \$pw\_library file\_list, you should add the above line to it.

If your \$pw\_library file\_list is blank and you are using the default ./aleph/tab/file\_list.PW, it already has this line.

What you \*do\* need to do in either case is:

```
> dlib abcde <where "abcde" is your $pw_library>
```

```
> util a, 17, 9 ("Sequences")
```

```
> Enter sequence name to Create (all/sequence-name): all
```

To verify that the sequence has been created successfully, do:

```
> s+ abcde <where "abcde" is your $pw_library>
```

```
abcde@ALEPH20> select SEQUENCE_OWNER, SEQUENCE_NAME from all_sequences where  
SEQUENCE_OWNER like 'ABC%';
```

```
**** Hit return to continue ****
```

```
SEQUENCE_OWNER SEQUENCE_NAME
```

```
-----  
ABCDE LAST_PUBLISH_SEQ
```

```
ABCDE LAST_RECORD_SEQUENCE
```

```
ABCDE LAST_RESULT_SET
```

Note: The "Create/recreate all library sequences" is a standard step in the "How To Open a Library" procedures, but is sometimes overlooked.

---

## Additional Information

In each case (io\_z601, io\_z309, etc.) the io\_sequence\_next routine is used to populate the last 7 bytes of the 15-byte Znnn-SEQUENCE component of the record key -- with the first 8 bytes being the date in *yyyymmdd format*. Thus, the only thing which is important is that the sequence's last\_number be unique within that particular day. And if the last\_record\_sequence value is lost and the sequence needs to be recreated, it is fine for it to start at "1".

---

- **Article last edited:** 24-May-2016