How I learned more than I ever wanted to know about DataProtector 8 Database Internals

Greg Baker (gregb@ifost.org.au)

August 30, 2013

Contents

1 Introduction

This document started life as the write up summary for customer who had a terribly, terribly corrupted internal database after upgrading to version 8. They had had numerous database problems with version 7 – performance and corruption issues – and an upgrade to version 8 did actually improve most things.

2 The Original Problem

The presenting symptoms were simply that the internal database backup would not run, even though all other backups were running wonderfully better.

The error message was

```
[Critical] From: OB2BAR_POSTGRES_BAR@XXXXX "DPIDB"  Time: 8/23/2013 3:01:29 PM
Check of the Internal Database consistency failed
```

Followed soon after by:

```
[Critical] From: BSM@XXXXXXXXXXXX "IDB Backup"  Time: 8/23/2013 3:02:05 PM
None of the Disk Agents completed successfully.
Session has failed.
```

When we tried to run an internal database backup without performing a consistency check, the backup would fail with the error message:

```
[Critical] From: BSM@XXXXXX "YYYYYY"  Time: 8/29/2013 11:03:36 AM
Cannot synchronize Media Management and Catalog database.
The session will terminate.
```

Which was also soon followed by:

```
[Major] From: OB2BAR_POSTGRES_BAR@XXXXXX "DPIDB"  Time: 8/29/2013 11:20:22 AM
Received ABORT request from BSM (ERR: Error starting backup/restore
(BMA cannot be started or similar).)
```

3 Stage 1 Fix

Running an `omnidbcheck -extended` reported errors in the -bf check as well as errors in media consistency. I was able to fix the former, and therefore re-enabled the internal database backup to run by fiddling around with `omnidbutil`.

```
 omnidbutil -remap_dcdir
 omnidbutil -fixmpos
 omnidbutil -remap_dcdir
```

I have no idea whether the remap commands are necessary, I have a sneaking suspicion that it actually did nothing.

4 Owners in upgrade-created IDB barlists

What transpired then is that my test internal database backup worked perfectly, but the customer's one (which had been created automatically by the upgrade process) still continue to fail. But it failed with a different error message. This time it started saying

```
[Critical] From: INET_thread_postgres_bar.exe@XXXXXXX "YYYYYYYYY"  Time: 8/29/2013 1:40:41
[70:24]  A system error occurred when starting the target script or an agent module.
The system error code reported is 1314 and the message resolves to '[1314] A required
privilege is not held by the client. '.
```

I opened up the bar list files and compared them and incrementally brought my bar list to match the upgrade generated bar list. It broke when I added the line for an owner; this makes sense: in DataProtector 8 it appears that an internal database backup can no longer have an owner. When I removed the owner line from the bar list the customers backups began to work correctly.

5 Hacking my way into PostgreSQL

At this point, running the `omnidbcheck -extended` command was reporting that there were problems with media consistency. Running `omnidbcheck -media_consistency -extended` discovered that there were several hundred tapes that it was complaining about.

Then came the fun part. I wanted to have a look at the internal database at a very low level, but I wanted to keep the database running and the DataProtector services running because there were large copy sessions going and the customer has barely have enough time to keep up with normal copy operations and it's a major pain for them if the copy gets interrupted.

I hacked the following file, which is the configuration file for authentication of users in postgresql:

`C:\ProgramData\OmniBack\server\db80\pg\pg_hba.conf`

Poking around at the following lines:

```
host postgres hpdp 127.0.0.1/32 sspi map=hpdpidb
host postgres hpdp ::1/128 sspi map=hpdpidb
```

I saw that sspi (meaning Kerberos authentication dropping down to NTLM if required) could be replaced. So I commented those lines out and replaced them by the following:

```
host all all 127.0.0.1/32 trust
host all all ::1/128 trust
```

Then I ran the following so that the changes would be picked up. In other circumstances I probably would have just stopped and started the DataProtector services.

`C:\Program Files\Omniback\idb\bin\pg_ctl reload -D C:\ProgramData\OmniBack\server\db80\p`

That meant that any connection to the database would workincluding the running sessions and the copy jobs that were due to start soon afterwards.

Incidentally, this pain only appears to be necessary for windows boxes. On a crash and burn Linux machine I could just run `/opt/omni/idb/bin/psql --host=/var/opt/omni/tmp --port=7112 -d hpdpidb`. But I digress.

6 Interlude

Anyway I could then finally connect to the database (hpdpidb) as the user hpdp:

`psql -U hpdp --port 7112 --host=127.0.0.1 -d hpdpidb`

But for reasons that elude me the hpdp user account doesn't default to the correct schema. So in order to do anything one needs to set the search path:

```
 set search_path to "$user",public,hpdpidb_app;
 show search_path;
```

Once that was working I used the `pg_dump` command to export the entire database out to an SQL file. Then I searched through that SQL file for the identifier codes which `omnidbcheck -media_consistency -extended` were saying were causing problems.

7 The media that didn't correspond to sessions or objects

It turned out that all the references were in the database table `dp_catalog_position_seqacc_med`. This table appears to contain positions where objects start being recorded in sequential media.

To get an understanding of what the problem was I found 2 very helpful views: `dp_medium_to_objects` and `dp_medium_to_sessions`.

So on a whim I decided to find out which sessions and which objects were causing grief. By chance I discovered that the 1st one I chose was not associated with any objects or sessions. I ran the following query:

```
select count(distinct medium_name)
   from dp_catalog_position_seqacc_med
  where medium_name not in
                    (select medium_id
                       from dp_medium_to_objects);
```

If all were good in life, this should return 0. As it turned out it returned a number in the hundreds. So very nervously, after much chasing around of referential integrity checks I ran:

```
delete from dp_catalog_position_seqacc_med
   where medium_name not in
  (select medium_id from dp_medium_to_objects);
```

I could just as easily have used sessions rather than objects. In fact I did check this afterwards, and confirmed that there were no media which had connections to object but no connection to a session.

This certainly made `omnidbcheck -media_consistency -extended` much happier and it reported far fewer media that it was concerned about.

8 Media that didn't have a name and sessions with no media

When I chased down the remaining media examples, they did correspond to sessions and objects (obviously, because I had already deleted any that didn't). Then I looked at the resulting sessions: and they had the bizarre property in the GUI that although they looked as though they had backed up correctly, when you went to the media tab these backups had magically managed to backup on to no tapes. Similarly, `omnidb -session` *XXXX* `-media` also reported no media.

So some more SQL later:

```
create temp table bad_media (medium_name varchar(1024));

insert into bad_media
   select distinct medium_name
     from dp_catalog_position_seqacc_med
     where medium_name not in
                      (select medium_id from dp_media);
```

This produced a long list whichmuch to my joyexactly matched the list from `omnidbcheck -media_consistency -extended`.

Just be really sure I ran:

```
select distinct name
  from dp_medium_to_sessions
 where medium_id in
           (select medium_name
              from bad_media)
order by name;
```

And looked at all the sessions. There was something wrong with the media tab in every single on of them. So I turned the output into a script running `omnidb -session` *XXXX* `-purge` and also deleted these mysterious media references with:

```
delete from dp_catalog_position_seqacc_med
 where medium_name in
       (select medium_name from bad_media);
```

After that, `omnidbcheck -media_consistency -extended` returned success rather than failure and omnidbcheck -extended also reported that all was well. And the internal database backup continued to work, and so did other backups.

Then I reverted the `pg_hba.conf` change because it was a multi-user box that other people could log in to.

9 Postscript

Greg Baker consults on this sort of thing. Email him `gregb@ifost.org.au` if you want to talk to him.