10167.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-09-05 14:27:45 GMT (29 weeks, 3 days, 12 hours and 40 minutes ago)
Hello,

I like to open a discussion about Bacula project and some questions.

For now, I am a postgraduation in development software and need to do a work to
finish my course.
The central motivation is study the contribution process on open source and
free software. How the enthusiastic user can became a developer in some
project.

To prove my ideas and way I choice Bacula project to research and write my
work.

Basically, I want to implement a catalog backend to IBM DB2. In the same time I
want to write my work based on this experiences.

The motivation for the choice of IBM DB2 is not very especial. I need learning
DB2 and currently work with it.

So, before I spend more energy, what ours opinions about this?

Thanks a lot.

PS: Please, I am not requesting you to develop this backend. It's Just a
technical opinion

--
-------------------------------------------------------------
Joa~o Henrique Freitas - joaohf_at_gmail.com
Americana-SP-Brasil
BSD051283
LPI 1
http://paginas.terra.com.br/informatica/joaohf
http://www.livejournal.com/users/joaohf/
```

10176.txt

```
From: Arno Lehmann <al <at> its-lehmann.de>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-09-05 19:51:53 GMT (29 weeks, 3 days, 7 hours and 16 minutes ago)
Hi,

05.09.2007 16:27,, João Henrique Freitas wrote::
> Hello,
>
> I like to open a discussion about  Bacula project and some questions.
>
> For now, I am a postgraduation in development software and need to do a
> work to finish my course.
> The central motivation is study the contribution process on open source
> and free software. How the enthusiastic user can became a developer in
> some project.
>
> To prove my ideas and way I choice Bacula project to research and write
> my work.
>
> Basically, I want to implement a catalog backend to IBM DB2. In the same
> time I want to write my work based on this experiences.

Can we get that work, once it's written? [:-)]

> The motivation for the choice of IBM DB2 is not very especial. I need
> learning DB2 and currently work with it.
>
> So, before I spend more energy, what ours opinions about this?

I'd do the following:
- make sure you understand how Bacula works, at least to the extent
that you know what the catalog is needed for.
- read some of Baculas code to make sure you understand the way it's
written; this starts with coding style, and probably does not end when
it comes to understanding how the different parts of Bacula are
```

```
organized into classes or are represented by all the *cr structures.
- use the trace feature to see what Bacula does with the catalog.
- read the developers manual; there is some documentation of the
catalog schema there.
- read and understand the code of the different catalog backends.
- start with one of the existing catalog backends - SQLite, MySQL,
PostgreSQL - and start migrating it to DB2 SQL.
- don't hesitate to ask questions, but make sure you ask clever
questions. The developers who know the existing code best are really
helpful people, but they probably prefer coding, testing, writing
documentation over explaining things... personally, even if I were
totally stuck somewhere, I'd make sure that whenever I have things to
ask I can show some progress.

Actually, I think you've got a nice project there. Not really being a
developer, I thing getting another database to work as the catalog is
rather simple, i.e. the thing itself is defined and you don't have to
spend time thinking how to layout the database etc. The catalog
database schema is really not very sophisticated, it's mainly rather
simple SQL queries and data insertions. As a result, you can do some
valuable work, but still concentrate on the main points of your
project - understanding how open source projects work and learning
about DB2.

Good luck!

Arno

> Thanks a lot.
>
> PS: Please, I am not requesting you to develop this backend. It's Just a
> technical opinion

Also PS:
I don't think your intentions can be misunderstood this way. In fact I
think your post was a good example of a clever question [:-)]

Arno

>
> --
> ------------------------------------------------------------
> João Henrique Freitas - joaohf_at_gmail.com
> Americana-SP-Brasil
> BSD051283
> LPI 1
> http://paginas.terra.com.br/informatica/joaohf
> http://www.livejournal.com/users/joaohf/
>
>
> ------------------------------------------------------------------------
>
> ------------------------------------------------------------------------
```

10178.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-09-05 20:53:34 GMT (29 weeks, 3 days, 6 hours and 15 minutes ago)
Hello,

I second Arno's comments, but add one note of caution.  If I am not mistaken,
DB2 is a commercial product, so there may be licensing restrictions (both
with the GPL used by Bacula and with IBM) on using or referencing any DB2
client libraries that are needed in Bacula to interface to DB2.  This is
something that is important to understand from the beginning.  If you are
thinking of including your code in the Bacula project, you will need to read
the Developer's Guide, which is online on the web site and then fill out and
send in the FSFE FLA (fiduciary license agreement).  Please see the
Developer's guide for more details.

Best regards,

Kern

PS: A related project is to turn the current db drivers into shared objects so
that we can support all databases with the same binary but with different
```

```
shared objects, and if one is really clever, we could support multiple
different SQL engines at the same time.

On Wednesday 05 September 2007 16:27, João Henrique Freitas wrote:
> Hello,
>
> I like to open a discussion about  Bacula project and some questions.
>
> For now, I am a postgraduation in development software and need to do a
> work to finish my course.
> The central motivation is study the contribution process on open source and
> free software. How the enthusiastic user can became a developer in some
> project.
>
> To prove my ideas and way I choice Bacula project to research and write my
> work.
>
> Basically, I want to implement a catalog backend to IBM DB2. In the same
> time I want to write my work based on this experiences.
>
> The motivation for the choice of IBM DB2 is not very especial. I need
> learning DB2 and currently work with it.
>
> So, before I spend more energy, what ours opinions about this?
>
> Thanks a lot.
>
> PS: Please, I am not requesting you to develop this backend. It's Just a
> technical opinion
```

10189.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-09-06 12:59:19 GMT (29 weeks, 2 days, 14 hours and 9 minutes ago)
> > The motivation for the choice of IBM DB2 is not very especial. I
need
> > learning DB2 and currently work with it.
> >
> > So, before I spend more energy, what ours opinions about this?

Great idea. There is a lot of interest in DB/2 in the commercial world,
especially if you assume and test with z/OS DB/2 as the backend
database. The combination of Bacula on Linux for Z and the DB/2 on z/OS
is a very compelling argument for enterprise deployment.

If you don't have a z/OS system handy, talk to me offlist and I can put
you in touch with some people at IBM who can make development resources
available. Failing that, UDB on Linux for Z would be a good choice, and
there is a UDB Community Edition release for that platform (as well as
Intel and POWER).

> - read and understand the code of the different catalog backends.
> - start with one of the existing catalog backends - SQLite, MySQL,
> PostgreSQL - and start migrating it to DB2 SQL.

I did a little bit of looking at this a while back. Pay very close
attention to the Postgres back end; Postgres is very similar in behavior
and manner to DB/2 (both are picky about standards compliance, and have
a lot of the same quirks).

I have a lot of DB/2 types around here; I can probably help you with
queries.
```

10911.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-11-30 18:19:17 GMT (17 weeks, 1 day, 8 hours and 49 minutes ago)
Hello,

First, sorry for the long time without reply.

I don't forgot this ideas. But now I have time to spend it and learn more about
```

```
Bacula devel.

About one week back, I started some experiences about DB2 API and C/C++. Read
the src/cats/cats.h, src/cats/postgresql.c and various related files. I
understood what I need to do and how to do it.

Ah, I convert the SQL script to DB2 too.

About license GPL and DB2:

IBM DB2 is a proprietary product and Bacula is free software how they can stay
togethers? I will use API DB2 (CLI ODBC). In DB2 license, there not have any
limit about developer and distributed the work.

For example, if I compiling bacula with DB2, I will has DB2 client and
developers files installed. Or, if I distribute a rpm package of Bacula
(previous compiling with DB2 support) the server need have a DB2 client
installed.


What I want say is: I can developing with DB2 API and distribute my work with
GPL license and this work use a proprietary API to connect a DB2 database
server. Somebody has experience about this situation between interaction GPL
and proprietary APIs ?

My intend is to only develop the backend to DB2 on Bacula and source code (GPL)
necessary to it.

Follow a text found in DB2 License IBM Data Server Driver for ODBC and CLI

Redistribution Information

If You have developed an application that is dependent upon the files or
modules listed below or located in the directory named below, You may
distribute these files or modules, subject to the following terms:
1) The files or modules must be in object code.
2) You will indemnify IBM or third parties that provide IBM products ("Third
Parties") from and against any third party claim arising out of the use or
distribution of Your application.
3) You may not use the same path name as the original files/modules.
4) You may not use IBM's or Third Parties' names or trademarks in connection
with the marketing of Your applications without IBM's or Third Parties' prior
written consent.
5) IBM or Third Parties provide copies of these files or modules "AS IS," i.e.,
You are responsible for all technical assistance for Your application.
6) In Your license agreement with the recipient, You will notify the recipient
that these files or modules may not be 1) used for any purpose other than to
enable the application, 2) copied (except for backup purposes), 3) further
distributed, or 4) reverse assembled, reverse compiled, or otherwise
translated.

Unix/Linux:
db2trc
db2ldcfg
db2lddrg
db2level
db2cli.ini
08501252.cnv
12520850.cnv
IBM00850.ucs
IBM01252.ucs
ReadMe.txt
libicclib.so
libcrypto.so.0.9.7
db2admh.mo
db2adm.mo
db2clia1.lst
db2clias.lst
db2clih.mo
db2cli.mo
db2clit.mo
db2clp.mo
db2diag.mo
db2sqlh.mo
db2sql.mo
libdb2.a
libdb2.sl
```

```
libdb2.so
libdb2.so.1
IBMOSauthclient.a
IBMOSauthclient.sl
IBMOSauthclient.so
IBMOSauthclient.so.1



On Sep 6, 2007 10:59 AM, David Boyes <dboyes_<at>_sinenomine.net> wrote:
     > > The motivation for the choice of IBM DB2 is not very especial. I
     need
     > > learning DB2 and currently work with it.
     > >
     > > So, before I spend more energy, what ours opinions about this?

     Great idea. There is a lot of interest in DB/2 in the commercial
     world,
     especially if you assume and test with z/OS DB/2 as the backend
     database. The combination of Bacula on Linux for Z and the DB/2 on z/
     OS
     is a very compelling argument for enterprise deployment.

     If you don't have a z/OS system handy, talk to me offlist and I can
     put
     you in touch with some people at IBM who can make development
     resources
     available. Failing that, UDB on Linux for Z would be a good choice,
     and
     there is a UDB Community Edition release for that platform (as well
     as
     Intel and POWER).

     > - read and understand the code of the different catalog backends.
     > - start with one of the existing catalog backends - SQLite, MySQL,
     > PostgreSQL - and start migrating it to DB2 SQL.

     I did a little bit of looking at this a while back. Pay very close
     attention to the Postgres back end; Postgres is very similar in
     behavior
     and manner to DB/2 (both are picky about standards compliance, and
     have
     a lot of the same quirks).

     I have a lot of DB/2 types around here; I can probably help you with
     queries.


     ------------------------------------------------------------------
     ----
```

10913.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-11-30 20:39:41 GMT (17 weeks, 1 day, 6 hours and 28 minutes ago)
On Friday 30 November 2007 19:19, João Henrique Freitas wrote:
> Hello,
>
> First, sorry for the long time without reply.
>
> I don't forgot this ideas. But now I have time to spend it and learn more
> about Bacula devel.

Nice.

>
> About one week back, I started some experiences about DB2 API and C/C++.
> Read the src/cats/cats.h, src/cats/postgresql.c and various related files.
> I understood what I need to do and how to do it.

Nice.

>
> Ah, I convert the SQL script to DB2 too.
>
> About license GPL and DB2:
```

>
> IBM DB2 is a proprietary product and Bacula is free software how they can
> stay togethers? I will use API DB2 (CLI ODBC). In DB2 license, there not
> have any limit about developer and distributed the work.
>
> For example, if I compiling bacula with DB2, I will has DB2 client and
> developers files installed. Or, if I distribute a rpm package of Bacula
> (previous compiling with DB2 support) the server need have a DB2 client
> installed.
>
>
> What I want say is: I can developing with DB2 API and distribute my work
> with GPL license and this work use a proprietary API to connect a DB2
> database server. Somebody has experience about this situation between
> interaction GPL and proprietary APIs ?
>
> My intend is to only develop the backend to DB2 on Bacula and source code
> (GPL) necessary to it.

Yes, as you have noted, there *are* license problems with Bacula (GPLv2)
interfacing to proprietary software. There are a few solutions though, but
I'll need to discuss this a bit with you offlist to get a good idea of what
you are planning to do so in order to know if it is possible and to decide
exactly how to do it.

(By the way, I just last night, not having seen your email, discussed these
kinds of problems with someone in the Free Software Foundation ...).

>
> Follow a text found in DB2 License IBM Data Server Driver for ODBC and CLI
>
> Redistribution Information
>
> If You have developed an application that is dependent upon the files or
> modules listed below or located in the directory named below, You may
> distribute these files or modules, subject to the following terms:
> 1) The files or modules must be in object code.

The above is very poor English coming from IBM. I have no idea what it really
means. It either means that you may distribute the source files providing
they are also in the object code, or it may mean that you may distribute the
results of using those files in object form only. I think you will need to
get this point clarified. Some of the files are obviously already in binary
form, but there are other ones which are obviously not.

> 2) You will indemnify IBM or third parties that provide IBM products
> ("Third Parties") from and against any third party claim arising out of the
> use or distribution of Your application.
> 3) You may not use the same path name as the original files/modules.

What does the above mean -- I cannot understand why such a restriction would
be needed.

> 4) You may not use IBM's or Third Parties' names or trademarks in
> connection with the marketing of Your applications without IBM's or Third
> Parties' prior written consent.

Well, the above is a bit absurd. It apparently says that you cannot use the
name DB2 in "marketing" Bacula, which means that it is useless to develop a
DB2 interface since we cannot say that we have a DB2 interface.

> 5) IBM or Third Parties provide copies of these files or modules "AS IS,"
> i.e., You are responsible for all technical assistance for Your
> application. 6) In Your license agreement with the recipient, You will
> notify the recipient that these files or modules may not be 1) used for any
> purpose other than to enable the application, 2) copied (except for backup
> purposes), 3) further distributed, or 4) reverse assembled, reverse
> compiled, or otherwise translated.
>
> Unix/Linux:
> db2trc
> db2ldcfg
> db2lddrg
> db2level
> db2cli.ini
> 08501252.cnv
> 12520850.cnv

```
> IBM00850.ucs
> IBM01252.ucs
> ReadMe.txt
> libicclib.so
> libcrypto.so.0.9.7
> db2admh.mo
> db2adm.mo
> db2clia1.lst
> db2clias.lst
> db2clih.mo
> db2cli.mo
> db2clit.mo
> db2clp.mo
> db2diag.mo
> db2sqlh.mo
> db2sql.mo
> libdb2.a
> libdb2.sl
> libdb2.so
> libdb2.so.1
> IBMOSauthclient.a
> IBMOSauthclient.sl
> IBMOSauthclient.so
> IBMOSauthclient.so.1

I think we need to understand the above a bit better before continuing.  If
you are writing something that work *only* on an IBM mainframe, then it is
not very interesting to the Bacula community and you might not get approval
from us to use Bacula with the above proprietary code.  On the other hand, if
you are writing a Generic DB2 Bacula driver that would work on any platform
that has DB2, then it would be quite interesting.

Best regards,

Kern

>
> On Sep 6, 2007 10:59 AM, David Boyes <dboyes <at> sinenomine.net> wrote:
> > > > The motivation for the choice of IBM DB2 is not very especial. I
> > >
> > need
> > >
> > > > learning DB2 and currently work with it.
> > > >
> > > > So, before I spend more energy, what ours opinions about this?
> > >
> > Great idea. There is a lot of interest in DB/2 in the commercial world,
> > especially if you assume and test with z/OS DB/2 as the backend
> > database. The combination of Bacula on Linux for Z and the DB/2 on z/OS
> > is a very compelling argument for enterprise deployment.
> >
> > If you don't have a z/OS system handy, talk to me offlist and I can put
> > you in touch with some people at IBM who can make development resources
> > available. Failing that, UDB on Linux for Z would be a good choice, and
> > there is a UDB Community Edition release for that platform (as well as
> > Intel and POWER).
> >
> > > > - read and understand the code of the different catalog backends.
> > > > - start with one of the existing catalog backends - SQLite, MySQL,
> > > > PostgreSQL - and start migrating it to DB2 SQL.
> > >
> > I did a little bit of looking at this a while back. Pay very close
> > attention to the Postgres back end; Postgres is very similar in behavior
> > and manner to DB/2 (both are picky about standards compliance, and have
> > a lot of the same quirks).
> >
> > I have a lot of DB/2 types around here; I can probably help you with
> > queries.
> >
> >
> > ------------------------------------------------------------------------
```

10914.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-11-30 21:16:32 GMT (17 weeks, 1 day, 5 hours and 52 minutes ago)
```

```
> > Redistribution Information
> >
> > If You have developed an application that is dependent upon the files or
> > modules listed below or located in the directory named below, You may
> > distribute these files or modules, subject to the following terms:
> > 1) The files or modules must be in object code.
>
> The above is very poor English coming from IBM.  I have no idea what it
> really
> means.

IANAL, however what this usually means when talking to IBM is that you can
compile and distribute modules
based on these source files, but you cannot distribute any source of the IBM-
supplied bits itself. If the
modules are distributed from IBM in binary form only, you cannot include them
in your package other than by
reference via dynamic linking.

> > 2) You will indemnify IBM or third parties that provide IBM products
> > ("Third Parties") from and against any third party claim arising out of
> the
> > use or distribution of Your application.

Usual "we didn't make it, not our fault" clause.

> > 3) You may not use the same path name as the original files/modules.
>
> What does the above mean -- I cannot understand why such a restriction
> would
> be needed.

It means you can't create something that replaces the IBM distributed files and
call it DB/2 UDB Client.
Also, on AIX, you're expected to register the pathname where your stuff goes
with IBM if you expect it to go
into /usr/lpp with all the other IBM-supplied stuff. They do this to minimize
conflicts between file
locations, and incidentally, to force their own developers to keep the files
for a single product in one
place (rather than scattering them around the filesystem like Microsoft or CA
is wont to do).

> > 4) You may not use IBM's or Third Parties' names or trademarks in
> > connection with the marketing of Your applications without IBM's or
> Third
> > Parties' prior written consent.
>
> Well, the above is a bit absurd.  It apparently says that you cannot use
> the
> name DB2 in "marketing" Bacula, which means that it is useless to develop
> a
> DB2 interface since we cannot say that we have a DB2 interface.

They have to have this in the agreement or THEY lose the ability to prosecute
if someone *does* hijack the
trademark. They have to enforce the trademark or lose the right to it entirely.
See Kleenex/Jello/Xerox
machine/etc.

There are a set of rules for using the DB/2 trademark published on the IBM WWW
site, and if you register as a one
of their developers, in exchange you get limited rights to reference the
trademarks according to the
rules. They're not onerous; it comes down to you just have to spell it
correctly and not cause them grief
about what you do with it.

> > 5) IBM or Third Parties provide copies of these files or modules "AS
> IS,"
> > i.e., You are responsible for all technical assistance for Your
> > application. 6) In Your license agreement with the recipient, You will
> > notify the recipient that these files or modules may not be 1) used for
> any
> > purpose other than to enable the application, 2) copied (except for
> backup
```

```
> > purposes), 3) further distributed, or 4) reverse assembled, reverse
> > compiled, or otherwise translated.

Not our fault, not our fault, not our fault, and if somebody misuses your
product to do something illegal,
then it's still your fault. Welcome to American IP law.

*gag*

> I think we need to understand the above a bit better before continuing.
> If
> you are writing something that work *only* on an IBM mainframe, then it is
> not very interesting to the Bacula community and you might not get
> approval
> from us to use Bacula with the above proprietary code.  On the other hand,
> if
> you are writing a Generic DB2 Bacula driver that would work on any
> platform
> that has DB2, then it would be quite interesting.

The DB/2 client code he's working with uses an IP socket connection to talk to
a DB/2 server. It doesn't care
what platform DB/2 actually runs on (albeit the client is supported on a subset
of the platforms that
Bacula runs on).
```

10920.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-01 11:28:17 GMT (17 weeks, 15 hours and 40 minutes ago)
On Friday 30 November 2007 22:16, David Boyes wrote:
> > > Redistribution Information
> > >
> > > If You have developed an application that is dependent upon the files
> > > or modules listed below or located in the directory named below, You
> > > may distribute these files or modules, subject to the following terms:
> > > 1) The files or modules must be in object code.
> > >
> > The above is very poor English coming from IBM.  I have no idea what it
> > really
> > means.
>
> IANAL, however what this usually means when talking to IBM is that you can
> compile and distribute modules based on these source files, but you cannot
> distribute any source of the IBM-supplied bits itself. If the modules are
> distributed from IBM in binary form only, you cannot include them in your
> package other than by reference via dynamic linking.
>
> > > 2) You will indemnify IBM or third parties that provide IBM products
> > > ("Third Parties") from and against any third party claim arising out of
> > >
> > the
> > >
> > > use or distribution of Your application.
>
> Usual "we didn't make it, not our fault" clause.
>
> > > 3) You may not use the same path name as the original files/modules.
> > >
> > What does the above mean -- I cannot understand why such a restriction
> > would
> > be needed.
>
> It means you can't create something that replaces the IBM distributed files
> and call it DB/2 UDB Client. Also, on AIX, you're expected to register the
> pathname where your stuff goes with IBM if you expect it to go into
> /usr/lpp with all the other IBM-supplied stuff. They do this to minimize
> conflicts between file locations, and incidentally, to force their own
> developers to keep the files for a single product in one place (rather than
> scattering them around the filesystem like Microsoft or CA is wont to do).
>
> > > 4) You may not use IBM's or Third Parties' names or trademarks in
> > > connection with the marketing of Your applications without IBM's or
> > >
> > Third
```

```
> >
> > > Parties' prior written consent.
> >
> > Well, the above is a bit absurd.  It apparently says that you cannot use
> > the
> > name DB2 in "marketing" Bacula, which means that it is useless to develop
> > a
> > DB2 interface since we cannot say that we have a DB2 interface.
>
> They have to have this in the agreement or THEY lose the ability to
> prosecute if someone *does* hijack the trademark. They have to enforce the
> trademark or lose the right to it entirely. See Kleenex/Jello/Xerox
> machine/etc.
>
> There are a set of rules for using the DB/2 trademark published on the IBM
> WWW site, and if you register as a one of their developers, in exchange you
> get limited rights to reference the trademarks according to the rules.
> They're not onerous; it comes down to you just have to spell it correctly
> and not cause them grief about what you do with it.
>
> > > 5) IBM or Third Parties provide copies of these files or modules "AS
> >
> > IS,"
> >
> > > i.e., You are responsible for all technical assistance for Your
> > > application. 6) In Your license agreement with the recipient, You will
> > > notify the recipient that these files or modules may not be 1) used for
> >
> > any
> >
> > > purpose other than to enable the application, 2) copied (except for
> >
> > backup
> >
> > > purposes), 3) further distributed, or 4) reverse assembled, reverse
> > > compiled, or otherwise translated.
>
> Not our fault, not our fault, not our fault, and if somebody misuses your
> product to do something illegal, then it's still your fault. Welcome to
> American IP law.
>
> *gag*

Well, all the above seem to have some solution or workaround, but it will take
a bit of organization.

>
> > I think we need to understand the above a bit better before continuing.
> > If
> > you are writing something that work *only* on an IBM mainframe, then it
> > is not very interesting to the Bacula community and you might not get
> > approval
> > from us to use Bacula with the above proprietary code.  On the other
> > hand, if
> > you are writing a Generic DB2 Bacula driver that would work on any
> > platform
> > that has DB2, then it would be quite interesting.
>
> The DB/2 client code he's working with uses an IP socket connection to talk
> to a DB/2 server. It doesn't care what platform DB/2 actually runs on
> (albeit the client is supported on a subset of the platforms that Bacula
> runs on).
>

Well, linking Bacula to a shared object that is proprietary, or using header
files in Bacula which are proprietary would be clearly a violation of the
GPL, so be careful what you are getting yourselves into.  License violations
are not much fun for the person(s) violating the license -- especially
because by properly working with the project up front, you can avoid
problems.

Using all GPLed code inside of Bacula and in any shared objects that Bacula
references, then using a socket to talk to a proprietary DB2 engine, may be
permitted under the GPL (I am not 100% sure as there seem to be some
differences of opinion).

```

10944.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-03 12:06:44 GMT (16 weeks, 5 days, 15 hours and 1 minute ago)

Ok,

Discussion about license is much serious and do not can have any mistakes.

I will search about license between DB2 and Bacula. When take some result, I
post here.

Kern, my intentions is to do this work to run in any platform not only for
mainframe.

Thanks

On Dec 1, 2007 9:28 AM, Kern Sibbald <kern_<at>_sibbald.com> wrote:
     On Friday 30 November 2007 22:16, David Boyes wrote:
     > > > Redistribution Information
     > > >
     > > > If You have developed an application that is dependent upon the
     files
     > > > or modules listed below or located in the directory named
     below, You
     > > > may distribute these files or modules, subject to the following
     terms:
     > > > 1) The files or modules must be in object code.
     > > >
     > > The above is very poor English coming from IBM. I have no idea
     what it
     > > really
     > > means.
     >
     > IANAL, however what this usually means when talking to IBM is that
     you can
     > compile and distribute modules based on these source files, but you
     cannot
     > distribute any source of the IBM-supplied bits itself. If the
     modules are
     > distributed from IBM in binary form only, you cannot include them
     in your
     > package other than by reference via dynamic linking.
     >
     > > > 2) You will indemnify IBM or third parties that provide IBM
     products
     > > > ("Third Parties") from and against any third party claim
     arising out of
     > > >
     > > the
     > > >
     > > > use or distribution of Your application.
     >
     > Usual "we didn't make it, not our fault" clause.
     >
     > > > 3) You may not use the same path name as the original files/
     modules.
     > > >
     > > What does the above mean -- I cannot understand why such a
     restriction
     > > would
     > > be needed.
     >
     > It means you can't create something that replaces the IBM
     distributed files
     > and call it DB/2 UDB Client. Also, on AIX, you're expected to
     register the
     > pathname where your stuff goes with IBM if you expect it to go into
     > /usr/lpp with all the other IBM-supplied stuff. They do this to
     minimize
     > conflicts between file locations, and incidentally, to force their
     own
     > developers to keep the files for a single product in one place
     (rather than
     > scattering them around the filesystem like Microsoft or CA is wont
     to do).
```

```
>
> > > 4) You may not use IBM's or Third Parties' names or trademarks
in
> > > connection with the marketing of Your applications without
IBM's or
> >
> > Third
> >
> > > Parties' prior written consent.
> >
> > Well, the above is a bit absurd. It apparently says that you
cannot use
> > the
> > name DB2 in "marketing" Bacula, which means that it is useless to
develop
> > a
> > DB2 interface since we cannot say that we have a DB2 interface.
>
> They have to have this in the agreement or THEY lose the ability to
> prosecute if someone *does* hijack the trademark. They have to
enforce the
> trademark or lose the right to it entirely. See Kleenex/Jello/Xerox
> machine/etc.
>
> There are a set of rules for using the DB/2 trademark published on
the IBM
> WWW site, and if you register as a one of their developers, in
exchange you
> get limited rights to reference the trademarks according to the
rules.
> They're not onerous; it comes down to you just have to spell it
correctly
> and not cause them grief about what you do with it.
>
> > > 5) IBM or Third Parties provide copies of these files or
modules "AS
> >
> > IS,"
> >
> > > i.e., You are responsible for all technical assistance for Your

> > > application. 6) In Your license agreement with the recipient,
You will
> > > notify the recipient that these files or modules may not be 1)
used for
> >
> > any
> >
> > > purpose other than to enable the application, 2) copied (except
for
> >
> > backup
> >
> > > purposes), 3) further distributed, or 4) reverse assembled,
reverse
> > > compiled, or otherwise translated.
>
> Not our fault, not our fault, not our fault, and if somebody
misuses your
> product to do something illegal, then it's still your fault.
Welcome to
> American IP law.
>
> *gag*

Well, all the above seem to have some solution or workaround, but it
will take
a bit of organization.

>
> > I think we need to understand the above a bit better before
continuing.
> > If
> > you are writing something that work *only* on an IBM mainframe,
then it
> > is not very interesting to the Bacula community and you might not
get
```

```
> > approval
> > from us to use Bacula with the above proprietary code. On the
other
> > hand, if
> > you are writing a Generic DB2 Bacula driver that would work on
any
> > platform
> > that has DB2, then it would be quite interesting.
>
> The DB/2 client code he's working with uses an IP socket connection
to talk
> to a DB/2 server. It doesn't care what platform DB/2 actually runs
on
> (albeit the client is supported on a subset of the platforms that
Bacula
> runs on).
>

Well, linking Bacula to a shared object that is proprietary, or using
header
files in Bacula which are proprietary would be clearly a violation of
the
GPL, so be careful what you are getting yourselves into. License
violations
are not much fun for the person(s) violating the license -
- especially
because by properly working with the project up front, you can avoid
problems.

Using all GPLed code inside of Bacula and in any shared objects that
Bacula
references, then using a socket to talk to a proprietary DB2 engine,
may be
```

10947.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-03 14:19:58 GMT (16 weeks, 5 days, 12 hours and 48 minutes ago)
On Monday 03 December 2007 13:06, João Henrique Freitas wrote:
> Ok,
>
> Discussion about license is much serious and do not can have any mistakes.

Yes, I agree that it is better not to make any mistakes, but what is more
important is your intention, which seems to me from what you wrote below to
be perfectly OK.

>
> I will search about license between DB2 and Bacula. When take some result,
> I post here.
>
> Kern, my intentions is to do this work to run in any platform not only for
> mainframe.

OK, that is good.  After that the most important thing is to understand
*exactly* what files Bacula will need to be able to build the Bacula  DB2
driver and what their license is.

The other critical issue is the availability of those files: that is can
anyone get them, or are they available only to certain people and under what
conditions.  At an absolute minimum, the Bacula project must be able to
possess all files needed to build Bacula.  If that is not the case, then it
would be pretty much impossible to include the DB2 driver.  The project also
needs to be able to distribute at least the binaries, which from the license
you listed seems to be OK.

The next thing that is not a requirement but is important is that any user who
wants to get those files to build the DB2 driver should have access to them.
This would permit anyone to build the driver and not just the project.  This
is not an absolute requirement, but without it, there is a serious conflict
with the spirit of the GPL.

Best regards,

Kern
```

```
>
> Thanks
>
> On Dec 1, 2007 9:28 AM, Kern Sibbald <kern <at> sibbald.com> wrote:
> > On Friday 30 November 2007 22:16, David Boyes wrote:
> > > > > > Redistribution Information
> > > > > >
> > > > > > If You have developed an application that is dependent upon the
> > >
> > files
> > >
> > > > > > or modules listed below or located in the directory named below,
> > > > > > You may distribute these files or modules, subject to the following
> > >
> > terms:
> > > > > > 1) The files or modules must be in object code.
> > > > >
> > > > The above is very poor English coming from IBM.  I have no idea what
> > >
> > it
> > >
> > > > really
> > > > means.
> > > >
> > > IANAL, however what this usually means when talking to IBM is that you
> > >
> > can
> > >
> > > compile and distribute modules based on these source files, but you
> > >
> > cannot
> > >
> > > distribute any source of the IBM-supplied bits itself. If the modules
> > >
> > are
> > >
> > > distributed from IBM in binary form only, you cannot include them in
> > >
> > your
> > >
> > > package other than by reference via dynamic linking.
> > >
> > > > > > 2) You will indemnify IBM or third parties that provide IBM
> > > > > > products ("Third Parties") from and against any third party claim
> > > > > > arising out
> > >
> > of
> > >
> > > > > the
> > > > >
> > > > > > use or distribution of Your application.
> > > >
> > > Usual "we didn't make it, not our fault" clause.
> > >
> > > > > > 3) You may not use the same path name as the original
> > > > > > files/modules.
> > > > >
> > > > What does the above mean -- I cannot understand why such a
> > > > restriction would
> > > > be needed.
> > >
> > > It means you can't create something that replaces the IBM distributed
> > >
> > files
> > >
> > > and call it DB/2 UDB Client. Also, on AIX, you're expected to register
> > >
> > the
> > >
> > > pathname where your stuff goes with IBM if you expect it to go into
> > > /usr/lpp with all the other IBM-supplied stuff. They do this to
> > > minimize conflicts between file locations, and incidentally, to force
> > > their own developers to keep the files for a single product in one
> > > place (rather
> > >
> > than
> > >
```

> > > scattering them around the filesystem like Microsoft or CA is wont to
> >
> > do).
> >
> > > > > > 4) You may not use IBM's or Third Parties' names or trademarks in
> > > > > > connection with the marketing of Your applications without IBM's or
> > > > > >
> > > > > Third
> > > > >
> > > > > > Parties' prior written consent.
> > > > >
> > > > > Well, the above is a bit absurd. It apparently says that you cannot
> >
> > use
> >
> > > > > the
> > > > > name DB2 in "marketing" Bacula, which means that it is useless to
> >
> > develop
> >
> > > > > a
> > > > > DB2 interface since we cannot say that we have a DB2 interface.
> > > >
> > > > They have to have this in the agreement or THEY lose the ability to
> > > > prosecute if someone *does* hijack the trademark. They have to enforce
> >
> > the
> >
> > > > trademark or lose the right to it entirely. See Kleenex/Jello/Xerox
> > > > machine/etc.
> > > >
> > > > There are a set of rules for using the DB/2 trademark published on the
> >
> > IBM
> >
> > > > WWW site, and if you register as a one of their developers, in exchange
> >
> > you
> >
> > > > get limited rights to reference the trademarks according to the rules.
> > > > They're not onerous; it comes down to you just have to spell it
> >
> > correctly
> >
> > > > and not cause them grief about what you do with it.
> > > >
> > > > > > 5) IBM or Third Parties provide copies of these files or modules
> > > > > > "AS
> > > > >
> > > > > IS,"
> > > > >
> > > > > > i.e., You are responsible for all technical assistance for Your
> > > > > > application. 6) In Your license agreement with the recipient, You
> >
> > will
> >
> > > > > > notify the recipient that these files or modules may not be 1) used
> >
> > for
> >
> > > > > any
> > > > >
> > > > > > purpose other than to enable the application, 2) copied (except for
> > > > >
> > > > > backup
> > > > >
> > > > > > purposes), 3) further distributed, or 4) reverse assembled, reverse
> > > > > > compiled, or otherwise translated.
> > > >
> > > > Not our fault, not our fault, not our fault, and if somebody misuses
> >
> > your
> >
> > > > product to do something illegal, then it's still your fault. Welcome to
> > > > American IP law.
> > > >
> > > > *gag*

```
> >
> > Well, all the above seem to have some solution or workaround, but it will
> > take
> > a bit of organization.
> >
> > > > > I think we need to understand the above a bit better before
> > >
> > > continuing.
> > >
> > > > > If
> > > > > you are writing something that work *only* on an IBM mainframe, then
> > >
> > > it
> > >
> > > > > is not very interesting to the Bacula community and you might not get
> > > > > approval
> > > > > from us to use Bacula with the above proprietary code.  On the other
> > > > > hand, if
> > > > > you are writing a Generic DB2 Bacula driver that would work on any
> > > > > platform
> > > > > that has DB2, then it would be quite interesting.
> > > >
> > > > The DB/2 client code he's working with uses an IP socket connection to
> > >
> > > talk
> > >
> > > > > to a DB/2 server. It doesn't care what platform DB/2 actually runs on
> > > > > (albeit the client is supported on a subset of the platforms that
> > > > > Bacula runs on).
> > >
> > > > Well, linking Bacula to a shared object that is proprietary, or using
> > > > header
> > > > files in Bacula which are proprietary would be clearly a violation of the
> > > > GPL, so be careful what you are getting yourselves into.  License
> > > > violations
> > > > are not much fun for the person(s) violating the license -- especially
> > > > because by properly working with the project up front, you can avoid
> > > > problems.
> > >
> > > > Using all GPLed code inside of Bacula and in any shared objects that
> > > > Bacula
> > > > references, then using a socket to talk to a proprietary DB2 engine, may
> > > > be
> > > > permitted under the GPL (I am not 100% sure as there seem to be some
> > > > differences of opinion).
> > >
> > > > ------------------------------------------------------------------------
```

10948.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-03 19:19:12 GMT (16 weeks, 5 days, 7 hours and 49 minutes ago)

> OK, that is good.  After that the most important thing is to
understand
> *exactly* what files Bacula will need to be able to build the Bacula
DB2
> driver and what their license is.
> <snip>
> The other critical issue is the availability of those files: that is
can
> anyone get them, or are they available only to certain people and
under
> what
> conditions.

The files he needs are part of the DB/2 Client Development kit, which
are part of DB/2. If you legitimately have DB/2 UDB or DB/2 for z/OS,
you have them. If you don't, then you don't, and you can't get them any
other way. You'd also have to take into consideration what platforms
support DB/2 -- DB/2 isn't on all the platforms Bacula runs on, which
will cause problems.

Sounds like it might be smarter to implement a Bacula interface to the
Perl DBI:: package interface, and then the problem is permanently
solved, and not just for DB/2, but for just about any useful database
```

```
that currently exists. That would give us Oracle, Ingres, DB/2, Sybase,
etc w/o imposing other restrictions. There would be some restrictions on
what SQL statements can be fed to the DBI interface, but Bacula doesn't
do anything that fancy, so the restrictions would be fairly minor, IMHO.

> The next thing that is not a requirement but is important is that any
user
> who
> wants to get those files to build the DB2 driver should have access to
> them.

See above.
```

10949.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-03 21:00:21 GMT (16 weeks, 5 days, 6 hours and 8 minutes ago)
On Monday 03 December 2007 20:19, David Boyes wrote:
> > OK, that is good.  After that the most important thing is to
>
> understand
>
> > *exactly* what files Bacula will need to be able to build the Bacula
>
> DB2
>
> > driver and what their license is.
> > <snip>
> > The other critical issue is the availability of those files: that is
>
> can
>
> > anyone get them, or are they available only to certain people and
>
> under
>
> > what
> > conditions.
>
> The files he needs are part of the DB/2 Client Development kit, which
> are part of DB/2. If you legitimately have DB/2 UDB or DB/2 for z/OS,
> you have them. If you don't, then you don't, and you can't get them any
> other way. You'd also have to take into consideration what platforms
> support DB/2 -- DB/2 isn't on all the platforms Bacula runs on, which
> will cause problems.

As long as the project could get a copy of the necessary code, I don't see any
problem.  The fact that DB2 isn't on all the platforms that Bacula runs on is
not a problem -- what counts is that users have a choice and that the source
is open and that users are not restrained from building it on their systems.

>
> Sounds like it might be smarter to implement a Bacula interface to the
> Perl DBI:: package interface, and then the problem is permanently
> solved, and not just for DB/2, but for just about any useful database
> that currently exists. That would give us Oracle, Ingres, DB/2, Sybase,
> etc w/o imposing other restrictions. There would be some restrictions on
> what SQL statements can be fed to the DBI interface, but Bacula doesn't
> do anything that fancy, so the restrictions would be fairly minor, IMHO.

I am a bit skeptical about OBDC since all the good DBAs that I know tell me
that it doesn't really work as it should.  Judging from the problems we have
interfacing to SQLite, MySQL and PostgreSQL, I can understand what they are
saying.  Also Bacula is now considerably more sophisticated in its use of SQL
now that we have Batch Insert and bat.

>
> > The next thing that is not a requirement but is important is that any
>
> user
>
> > who
> > wants to get those files to build the DB2 driver should have access to
> > them.
>
```

```
> See above.
>

After thinking about it a bit, it seems to me that both Oracle and DB2 rather
freely provide developer's kits to pretty much any one who wants one -- that
would resolve the problem of availability of the interface code.  The only
other issue is the problem of incompatible licenses, which is much less of a
problem now that Bacula's code is "clean": i.e. written by me; code with BSD
3 clause license; or written by contributors that have signed the FSFE FLA.

Best regards,

Kern

```

10951.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-03 21:19:33 GMT (16 weeks, 5 days, 5 hours and 49 minutes ago)
> > Sounds like it might be smarter to implement a Bacula interface to
the
> > Perl DBI:: package interface, and then the problem is permanently
> > solved, and not just for DB/2, but for just about any useful
database
> > that currently exists. That would give us Oracle, Ingres, DB/2,
Sybase,
> > etc w/o imposing other restrictions. There would be some
restrictions on
> > what SQL statements can be fed to the DBI interface, but Bacula
doesn't
> > do anything that fancy, so the restrictions would be fairly minor,
IMHO.
>
> I am a bit skeptical about OBDC since all the good DBAs that I know
tell
> me
> that it doesn't really work as it should.

Perl DBI is not ODBC. It's a set of wrapper functions that allow
database-independent code to be written, with the actual database used
being selected at runtime by configuring the DBI interface code.
Database vendors supply drop-in back-ends (some open, some not), but no
code linkage occurs that is not open.

```

10953.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-03 21:29:07 GMT (16 weeks, 5 days, 5 hours and 39 minutes ago)
On Monday 03 December 2007 22:19, David Boyes wrote:
> > > Sounds like it might be smarter to implement a Bacula interface to
>
> the
>
> > > Perl DBI:: package interface, and then the problem is permanently
> > > solved, and not just for DB/2, but for just about any useful
>
> database
>
> > > that currently exists. That would give us Oracle, Ingres, DB/2,
>
> Sybase,
>
> > > etc w/o imposing other restrictions. There would be some
>
> restrictions on
>
> > > what SQL statements can be fed to the DBI interface, but Bacula
>
> doesn't
>
> > > do anything that fancy, so the restrictions would be fairly minor,
>
> IMHO.
```

```
>
> > I am a bit skeptical about OBDC since all the good DBAs that I know
>
> tell
>
> > me
> > that it doesn't really work as it should.
>
> Perl DBI is not ODBC. It's a set of wrapper functions that allow
> database-independent code to be written, with the actual database used
> being selected at runtime by configuring the DBI interface code.
> Database vendors supply drop-in back-ends (some open, some not), but no
> code linkage occurs that is not open.

Very interesting.  It sounds like something that would be well worth looking
at providing we can interface to it from C (or C++) as I imagine is the case.
```

10972.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-10 13:33:55 GMT (15 weeks, 5 days, 13 hours and 34 minutes ago)
Hello

I forgot the DB2, I agree that is not a good idea implement the driver direct
on Bacula code (license problems).

But we have a way (already proposed initially by Kern and now by David). We can
use the libdbi ( http://libdbi.sourceforge.net/) to do the work and perhaps the
project "Item 34: Commercial database support Origin: Russell Howe Date: 26
July 2006" listed on http://www.bacula.org/?page=projects, will be done.

I started some experiments with libdbi, following:

1. Test the framework libdbi. Understand how they work and limitations
2. Code in Bacula the drivers for libdbi interfacing with the databases
supported by libdbi
3. Create the a libdbi driver for DB2 (and others databases too)

The motivations to use libdbi, are:

LGPL license
Database-independent abstraction layer in C
Multiple databases

If anybody has any questions, please do it.

Important: this idea not discard the already interfacing drivers already coded
in Bacula, but one way to use proprietary databases and connectivity API in
Bacula.

Thanks


     >
     > Perl DBI is not ODBC. It's a set of wrapper functions that allow
     > database-independent code to be written, with the actual database
     used
     > being selected at runtime by configuring the DBI interface code.
     > Database vendors supply drop-in back-ends (some open, some not),
     but no
     > code linkage occurs that is not open.

     Very interesting. It sounds like something that would be well worth
     looking
     at providing we can interface to it from C (or C++) as I imagine is
     the case.
```

10975.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-10 14:45:00 GMT (15 weeks, 5 days, 12 hours and 23 minutes ago)
```

```
Hello João,

This is a very good way to do it.  Thanks.

Good luck.

Kern

On Monday 10 December 2007 14:33, João Henrique Freitas wrote:
> Hello
>
> I forgot the DB2, I agree that is not a good idea implement the driver
> direct on Bacula code (license problems).
>
> But we have a way (already proposed initially by Kern and now by David). We
> can use the libdbi (http://libdbi.sourceforge.net/) to do the work and
> perhaps the project "Item 34: Commercial database support Origin: Russell
> Howe Date: 26 July 2006" listed on http://www.bacula.org/?page=projects,
> will be done.
>
> I started some experiments with libdbi, following:
>
> 1. Test the framework libdbi. Understand how they work and limitations
> 2. Code in Bacula the drivers for libdbi interfacing with the databases
> supported by libdbi
> 3. Create the a libdbi driver for DB2 (and others databases too)
>
> The motivations to use libdbi, are:
>
> LGPL license
> Database-independent abstraction layer in C
> Multiple databases
>
> If anybody has any questions, please do it.
>
> Important: this idea not discard the already interfacing drivers already
> coded in Bacula, but one way to use proprietary databases and connectivity
> API in Bacula.
>
> Thanks
>
> > > Perl DBI is not ODBC. It's a set of wrapper functions that allow
> > > database-independent code to be written, with the actual database used
> > > being selected at runtime by configuring the DBI interface code.
> > > Database vendors supply drop-in back-ends (some open, some not), but no
> > > code linkage occurs that is not open.
> >
> > Very interesting.  It sounds like something that would be well worth
> > looking
> > at providing we can interface to it from C (or C++) as I imagine is the
> > case.
> >
> >
> >
> >
> >
> > ------------------------------------------------------------------------
```

10980.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2007-12-10 15:41:18 GMT (15 weeks, 5 days, 11 hours and 27 minutes ago)

> But we have a way (already proposed initially by Kern and now by David). We
can use the libdbi
> ( http://libdbi.sourceforge.net/) to do the work
Even better -- someone else already did the hard part..[8-)] Gluing this into
Bacula will be fairly simple, as the database interaction is already fairly
well abstracted into a C++ class.
> 1. Test the framework libdbi. Understand how they work and limitations
The major limitation is that if you want the SQL to work on multiple databases,
then you have to use the lowest common denominator language level and features.
There are places in Bacula where various things exploit "features" of the
currently supported databases. The Postgres code is the most rabidly standards-
compliant, so I'd start with that SQL and see if it just works out of the box
```

```
(ie, if you deliver the stuff written for Postgress to DB/2, I suspect it will
be more likely to work correctly than the MySQL code, which has a few local
features).

> 2. Code in Bacula the drivers for libdbi interfacing with the databases
supported by libdbi
See above. All the nasty parts are in the C++ class that abstracts the database
access.

> 3. Create the a libdbi driver for DB2 (and others databases too)
DB/2, Sybase and Oracle already have one, and they're officially supported by
the vendor. [8-)] Once you get the DBI layer in place, you can follow the
directions provided with the database to enable the DBI driver they provide.

> Important: this idea not discard the already interfacing drivers already
coded in Bacula, but one way to use
> proprietary databases and connectivity API in Bacula.

On the other hand, there are MySQL and Postgres DBI drivers, too. I don't know
about sqllite, but I wouldn't be heartbroken to see sqllite use fall off. It's
not like MySQL or Postgres are *that* big of a PITA to implement and manage. Or
someone could write a DBI driver for sqllite.
```

11126.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-01-11 12:39:09 GMT (11 weeks, 1 day, 14 hours and 29 minutes ago)
Hello,

I am like to report my status in libdbi interface to Bacula.

The libdbi drivers works in Bacula (configure, make, run).
Not regression tests had pass yet, because I need to do a job backup worked.

But I ran every comand in bconsole like "status dir", "list volumes"....

Basicaly I create and code in src/cats/dbi.c, change the definitions
in src/cats/cats.h and add a new option in dird_conf.h dbitype.

If someone want to see the code, please tell me. When the job backup
work. I report a new status here.

Thanks

On Dec 10, 2007 1:41 PM, David Boyes <dboyes <at> sinenomine.net> wrote:
>
>
>
>
> > But we have a way (already proposed initially by Kern and now by David).
> We can use the libdbi
> > ( http://libdbi.sourceforge.net/) to do the work
>
> Even better -- someone else already did the hard part..[8-)] Gluing this into
> Bacula will be fairly simple, as the database interaction is already fairly
> well abstracted into a C++ class.
>
>
> > 1. Test the framework libdbi. Understand how they work and limitations
>
> The major limitation is that if you want the SQL to work on multiple
> databases, then you have to use the lowest common denominator language level
> and features. There are places in Bacula where various things exploit
> "features" of the currently supported databases.  The Postgres code is the
> most rabidly standards-compliant, so I'd start with that SQL and see if it
> just works out of the box (ie, if you deliver the stuff written for
> Postgress to DB/2, I suspect it will be more likely to work correctly than
> the MySQL code, which has a few local features).
>
>
> > 2. Code in Bacula the drivers for libdbi interfacing with the databases
> supported by libdbi
>
> See above. All the nasty parts are in the C++ class that abstracts the
> database access.
>
```

```
>
> > 3. Create the a libdbi driver for DB2 (and others databases too)
>
> DB/2, Sybase and Oracle already have one, and they're officially supported
> by the vendor. [8-)] Once you get the DBI layer in place, you can follow the
> directions provided with the database to enable the DBI driver they provide.
>
>
> > Important: this idea not discard the already interfacing drivers already
> coded in Bacula, but one way to use
> > proprietary databases and connectivity API in Bacula.
>
> On the other hand, there are MySQL and Postgres DBI drivers, too. I don't
> know about sqllite, but I wouldn't be heartbroken to see sqllite use fall
> off. It's not like MySQL or Postgres are *that* big of a PITA to implement
> and manage. Or someone could write a DBI driver for sqllite.

--
-------------------------------------------------------------
João Henrique Freitas - joaohf_at_gmail.com
Campinas-SP-Brasil
BSD051283
LPI 1
http://joaohf.pbwiki.com
http://www.livejournal.com/users/joaohf/
```

11127.txt

```
From: Dan Langille <dan <at> langille.org>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-01-11 12:42:19 GMT (11 weeks, 1 day, 14 hours and 26 minutes ago)
João Henrique Freitas wrote:
> Hello,
>
> I am like to report my status in libdbi interface to Bacula.
>
> The libdbi drivers works in Bacula (configure, make, run).
> Not regression tests had pass yet, because I need to do a job backup worked.
>
> But I ran every comand in bconsole like "status dir", "list volumes"....
>
> Basicaly I create and code in src/cats/dbi.c, change the definitions
> in src/cats/cats.h and add a new option in dird_conf.h dbitype.
>
> If someone want to see the code, please tell me. When the job backup
> work. I report a new status here.

Congratulations.  :)

When you get the backup and restore working, run the regression tests
from the regress directory in the repository.

--
Dan Langille - http://www.langille.org/
BSDCan - The Technical BSD Conference: http://www.bsdcan.org/
PGCon  - The PostgreSQL Conference:    http://www.pgcon.org/

```

11128.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_catalog_backend_to_IBM_DB2
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-01-11 12:51:14 GMT (11 weeks, 1 day, 14 hours and 17 minutes ago)
Hello João,

On Friday 11 January 2008 13:39, João Henrique Freitas wrote:
> Hello,
>
> I am like to report my status in libdbi interface to Bacula.
>
> The libdbi drivers works in Bacula (configure, make, run).
> Not regression tests had pass yet, because I need to do a job backup
> worked.
>
> But I ran every comand in bconsole like "status dir", "list volumes"....

```

```
Great.

>
> Basicaly I create and code in src/cats/dbi.c, change the definitions
> in src/cats/cats.h and add a new option in dird_conf.h dbitype.

OK.

>
> If someone want to see the code, please tell me. When the job backup
> work. I report a new status here.

Yes, please do send me the code.  It is important to see it early so that I
can avoid you running into problems later (programming style, Bacula specific
coding conventions, ...)

Best regards,

Kern

>
> Thanks
>
> On Dec 10, 2007 1:41 PM, David Boyes <dboyes <at> sinenomine.net> wrote:
> > > But we have a way (already proposed initially by Kern and now by
> > > David).
> >
> > We can use the libdbi
> >
> > > ( http://libdbi.sourceforge.net/) to do the work
> >
> > Even better -- someone else already did the hard part..[8-)] Gluing this
> > into Bacula will be fairly simple, as the database interaction is already
> > fairly well abstracted into a C++ class.
> >
> > > 1. Test the framework libdbi. Understand how they work and limitations
> >
> > The major limitation is that if you want the SQL to work on multiple
> > databases, then you have to use the lowest common denominator language
> > level and features. There are places in Bacula where various things
> > exploit "features" of the currently supported databases.  The Postgres
> > code is the most rabidly standards-compliant, so I'd start with that SQL
> > and see if it just works out of the box (ie, if you deliver the stuff
> > written for Postgress to DB/2, I suspect it will be more likely to work
> > correctly than the MySQL code, which has a few local features).
> >
> > > 2. Code in Bacula the drivers for libdbi interfacing with the databases
> >
> > supported by libdbi
> >
> > See above. All the nasty parts are in the C++ class that abstracts the
> > database access.
> >
> > > 3. Create the a libdbi driver for DB2 (and others databases too)
> >
> > DB/2, Sybase and Oracle already have one, and they're officially
> > supported by the vendor. [8-)] Once you get the DBI layer in place, you can
> > follow the directions provided with the database to enable the DBI driver
> > they provide.
> >
> > > Important: this idea not discard the already interfacing drivers
> > > already
> >
> > coded in Bacula, but one way to use
> >
> > > proprietary databases and connectivity API in Bacula.
> >
> > On the other hand, there are MySQL and Postgres DBI drivers, too. I don't
> > know about sqllite, but I wouldn't be heartbroken to see sqllite use fall
> > off. It's not like MySQL or Postgres are *that* big of a PITA to
> > implement and manage. Or someone could write a DBI driver for sqllite.
```

11241.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
```

```
Date: 2008-02-01 01:03:31 GMT (8 weeks, 2 days, 2 hours and 5 minutes ago)
Hello,

This is a status report of the work to implement libdbi in Bacula.

In my tests *All regression tests running*. Now I fix some things
after post the code here.

I have a design problem:

The libdbi API need know what dbms we want to use and load the correct
driver. For exemple mysql, pgsql, sqlite3, etc...

Actually, in mysql.c, postgresql.c and dbi.c the scope of funcion
db_init_database is:

 B_DB *
db_init_database(JCR *jcr, const char *db_name, const char *db_user,
const char *db_password,
                 const char *db_address, int db_port, const char *db_socket,
                 int mult_db_connections)

Is possible to add one more argument like char *db_type, this indicate
what driver libdbi will use?. So Bacula uses db_init_database  in
various places... and need to change them too.

OR

The argument char *db_name would be: mysql:bacula where mysql indicate
a driver to libdbi and bacula is the name of database.

Like this in bacula-dir.conf:

Catalog {
  Name = MyCatalog
  dbaddress = 127.0.0.1; dbport = 5432; dbname = "mysql:bacula"; user
= bacula; password = "bacula"
}

On db_init_database, we split db_name and everything will be done
right without change in others places of the code.

I don't want to change others places of Bacula. What is the best solution?

Thanks

--
------------------------------------------------------------
João Henrique Freitas - joaohf_at_gmail.com
Campinas-SP-Brasil
BSD051283
LPI 1
http://joaohf.pbwiki.com
http://www.livejournal.com/users/joaohf/
```

11242.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-01 08:09:25 GMT (8 weeks, 1 day, 18 hours and 59 minutes ago)
Hello,

I am really pleased to see that you are making progress.  Please don't forget
to email me a copy of the code, in its current state.  Doing so, can avoid
doing more work later as I will be able to give you early comments.

Concerning the design problem where you need to know the driver type:  I have
been planning to turn the drivers into shared objects at some point and so
what you are asking for is what will also be needed at that point.

I would be interested to hear comments from other people on this list, but
rather than overload an existing directive my inclination would be to create
a new directive named:

  DB Driver = "mysql" ...

or in your case it can be:
```

```
  DB Driver = "libdbi:mysql"  or "libdbi:postgres", ...

Note, once we have shared objects, in your case, two pieces of information are
needed by Bacula: 1. your driver name (libdbi or whatever you want), and 2.
the driver you are going to load.  Obviously, in the current situation where
the Bacula driver is compiled in, the "libdbi" is not currently needed, but
that won't hurt anything.

Then make a new db_ call which would be:

  db_load_driver(JCR *jcr, const char *db_driver);

At the moment, this would do nothing except save the name passed in the jcr
structure.  Then in your libdbi db_init_database() code, you will be able to
parse the name to find what driver to call.  Later, the db_load_driver() code
will be enhanced to actually load the appropriate shared object library ...

If everyone agrees on this, I'll be happy to add the necessary "core" code for
you.

Best regards,

Kern

On Friday 01 February 2008 02.03:31 João Henrique Freitas wrote:
> Hello,
>
> This is a status report of the work to implement libdbi in Bacula.
>
> In my tests *All regression tests running*. Now I fix some things
> after post the code here.
>
> I have a design problem:
>
> The libdbi API need know what dbms we want to use and load the correct
> driver. For exemple mysql, pgsql, sqlite3, etc...
>
> Actually, in mysql.c, postgresql.c and dbi.c the scope of funcion
> db_init_database is:
>
>  B_DB *
> db_init_database(JCR *jcr, const char *db_name, const char *db_user,
> const char *db_password,
>                  const char *db_address, int db_port, const char
> *db_socket, int mult_db_connections)
>
> Is possible to add one more argument like char *db_type, this indicate
> what driver libdbi will use?. So Bacula uses db_init_database  in
> various places... and need to change them too.
>
> OR
>
> The argument char *db_name would be: mysql:bacula where mysql indicate
> a driver to libdbi and bacula is the name of database.
>
> Like this in bacula-dir.conf:
>
> Catalog {
>   Name = MyCatalog
>   dbaddress = 127.0.0.1; dbport = 5432; dbname = "mysql:bacula"; user
> = bacula; password = "bacula"
> }
>
> On db_init_database, we split db_name and everything will be done
> right without change in others places of the code.
>
>
> I don't want to change others places of Bacula. What is the best solution?
>
> Thanks
```

11246.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
```

```
Date: 2008-02-01 15:44:48 GMT (8 weeks, 1 day, 11 hours and 23 minutes ago)

> Is possible to add one more argument like char *db_type, this indicate
> what driver libdbi will use?. So Bacula uses db_init_database  in
> various places... and need to change them too.

Why not add a /etc/bacula/dbms.conf file that the admin can edit to indicate
the installed database type
and read your database type information from there? I think it's unlikely that
there will be more than one
database type in an individual Bacula install. Then you can do the db_name_init
once and not have to touch
the rest of the code.

```

11247.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-01 15:52:37 GMT (8 weeks, 1 day, 11 hours and 16 minutes ago)
On Friday 01 February 2008 16.44:48 David Boyes wrote:
> > Is possible to add one more argument like char *db_type, this indicate
> > what driver libdbi will use?. So Bacula uses db_init_database  in
> > various places... and need to change them too.
>
> Why not add a /etc/bacula/dbms.conf file that the admin can edit to
> indicate the installed database type and read your database type
> information from there? I think it's unlikely that there will be more than
> one database type in an individual Bacula install. Then you can do the
> db_name_init once and not have to touch the rest of the code.

I have fewer problems with the idea of touching the code than with adding
another configuration file with possibly a different syntax. In fact, it
would probably take more code than what I am proposing to read the database
type from a separate file.

When the Bacula drivers become shared objects, Bacula will be capable of
working with multiple different database types simultaneously so any new
implementation should include that possibility.

Kern

```

11250.txt

```
From: John Stoffel <john <at> stoffel.org>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-01 21:26:03 GMT (8 weeks, 1 day, 5 hours and 42 minutes ago)
>>>>> "Kern" == Kern Sibbald <kern <at> sibbald.com> writes:

Kern> On Friday 01 February 2008 16.44:48 David Boyes wrote:
>> > Is possible to add one more argument like char *db_type, this indicate
>> > what driver libdbi will use?. So Bacula uses db_init_database  in
>> > various places... and need to change them too.
>>
>> Why not add a /etc/bacula/dbms.conf file that the admin can edit to
>> indicate the installed database type and read your database type
>> information from there? I think it's unlikely that there will be more than
>> one database type in an individual Bacula install. Then you can do the
>> db_name_init once and not have to touch the rest of the code.

Kern> I have fewer problems with the idea of touching the code than
Kern> with adding another configuration file with possibly a different
Kern> syntax. In fact, it would probably take more code than what I am
Kern> proposing to read the database type from a separate file.

I agree that making another seperate config file is a bad idea.

Kern> When the Bacula drivers become shared objects, Bacula will be
Kern> capable of working with multiple different database types
Kern> simultaneously so any new implementation should include that
Kern> possibility.

So why is this a good thing?  I've never understood the idea to have
different catalogs either, or what the design goal is.

```

```
Should I be using different catalogs for each client?  Should I be
using different Pools for each client?

John

```

11251.txt

```
From: David Boyes <dboyes <at> sinenomine.net>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-01 23:03:13 GMT (8 weeks, 1 day, 4 hours and 5 minutes ago)
> Kern> When the Bacula drivers become shared objects, Bacula will be
> Kern> capable of working with multiple different database types
> Kern> simultaneously so any new implementation should include that
> Kern> possibility.

I wish you luck in supporting it -- you're going to need it.

I think the problems of using multiple databases for Bacula internals
and supporting multiple types of database clients are substantially
different, and if you want to support multiple types of databases for
the internals you'll either need to federate the disparate databases
into one view, or pick one for Bacula's internal use and stick with it.
I think there are serious integrity problems that you'll need to solve
if you want to use federated databases for Bacula internals (eg,
holographic table storage, and you'll need to deal with some really hard
failure scenarios that Just Aren't Worth It).

Reading a simple sequential file on daemon startup and storing the value
in a global variable for use by the various database routines passes the
KISS test in my book, but YMMV. Putting it a current config file is OK
too. In either case, get the value at startup, stuff it somewhere and
pass it behind the scenes.

> So why is this a good thing?  I've never understood the idea to have
> different catalogs either, or what the design goal is.

The original purpose of using the libdbi interface is to remove the
details of the database implementation from the core Bacula code, which
allows support of a much wider range of database engines without
recoding for each one. Libdbi supports postgres, mysql, DB/2, Oracle,
Sybase, kdb, etc, etc via vendor-supplied (and supported) plugins.

As far as different catalogs go, I've never seen a point in it. Others
may vary.

> Should I be using different catalogs for each client?

Not unless you have lots of free time to reconcile them in case of a
disaster.

> Should I be
> using different Pools for each client?

You can, but I wouldn't recommend it unless you have very odd retention
or audit requirements for different clients AND lots of spare time to
debug things.

```

11252.txt

```
From: Dan Langille <dan <at> langille.org>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-01 23:55:35 GMT (8 weeks, 1 day, 3 hours and 13 minutes ago)
David Boyes wrote:
>> Kern> When the Bacula drivers become shared objects, Bacula will be
>> Kern> capable of working with multiple different database types
>> Kern> simultaneously so any new implementation should include that
>> Kern> possibility.
>
> I wish you luck in supporting it -- you're going to need it.
>
> I think the problems of using multiple databases for Bacula internals
> and supporting multiple types of database clients are substantially
> different, and if you want to support multiple types of databases for
> the internals you'll either need to federate the disparate databases
```

```
> into one view, or pick one for Bacula's internal use and stick with it.
> I think there are serious integrity problems that you'll need to solve
> if you want to use federated databases for Bacula internals (eg,
> holographic table storage, and you'll need to deal with some really hard
> failure scenarios that Just Aren't Worth It).

I have no idea what you are talking about.  :)

Bacula now has the ability to use multiple Catalogs.  At present, all of
the Catalogs must be of the same type (e.g. PostgreSQL).  I read what Kern
said as allowing each Catalog to be a different database type (e.g. one of
PostgreSQL, another of MySQL).

A Catalog is a totally self-contained entity.  Data is not shared across
Catalogs.

--
Dan Langille - http://www.langille.org/
BSDCan - The Technical BSD Conference: http://www.bsdcan.org/
PGCon  - The PostgreSQL Conference:    http://www.pgcon.org/
```

11253.txt

```
From: Kern Sibbald <kern <at> sibbald.com>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-02 09:32:42 GMT (8 weeks, 17 hours and 36 minutes ago)
On Saturday 02 February 2008 00.03:13 David Boyes wrote:
> > Kern> When the Bacula drivers become shared objects, Bacula will be
> > Kern> capable of working with multiple different database types
> > Kern> simultaneously so any new implementation should include that
> > Kern> possibility.
>
> I wish you luck in supporting it -- you're going to need it.
>
> I think the problems of using multiple databases for Bacula internals
> and supporting multiple types of database clients are substantially
> different, and if you want to support multiple types of databases for
> the internals you'll either need to federate the disparate databases
> into one view, or pick one for Bacula's internal use and stick with it.
> I think there are serious integrity problems that you'll need to solve
> if you want to use federated databases for Bacula internals (eg,
> holographic table storage, and you'll need to deal with some really hard
> failure scenarios that Just Aren't Worth It).

Well, Bacula already supports multiple types of databases.  It has a single
view of such databases, and *very* little code varies from database to
database (unfortunately SQL is not standardized like C).  What Bacula
currently supports is multiple databases but of a single type per Bacula
binary.  In the future, it will evolve to multiple databases but of multiple
types that are supported.  The details support for the SQL servers is not in
Bacula's domain but rather what the SQL provider must support, so I don't see
any additional support requirements here other than to ensure that it is
clear in the job reports what database engine is being used ...

>
> Reading a simple sequential file on daemon startup and storing the value
> in a global variable for use by the various database routines passes the
> KISS test in my book, but YMMV. Putting it a current config file is OK
> too. In either case, get the value at startup, stuff it somewhere and
> pass it behind the scenes.

OK, for me it is much easier to put it in the current conf file.  It is one
define in a header table, one entry in a look up table and one entry to
release the allocated memory -- rather trivial. Reading a new file requires
lots of extra stuff ...

>
> > So why is this a good thing?  I've never understood the idea to have
> > different catalogs either, or what the design goal is.
>
> The original purpose of using the libdbi interface is to remove the
> details of the database implementation from the core Bacula code, which
> allows support of a much wider range of database engines without
> recoding for each one. Libdbi supports postgres, mysql, DB/2, Oracle,
> Sybase, kdb, etc, etc via vendor-supplied (and supported) plugins.
>
```

```
> As far as different catalogs go, I've never seen a point in it. Others
> may vary.

They are very useful for improving performance and for scaling while keeping a
single point of control (Director).

>
> > Should I be using different catalogs for each client?
>
> Not unless you have lots of free time to reconcile them in case of a
> disaster.
>
> > Should I be
> > using different Pools for each client?
>
> You can, but I wouldn't recommend it unless you have very odd retention
> or audit requirements for different clients AND lots of spare time to
> debug things.

Good advice.

Kern
```

11393.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-22 00:00:07 GMT (5 weeks, 2 days, 3 hours and 8 minutes ago)
Hello

Attached is the dbi patch for Bacula.

To use, are necessary libdbi-0.8.3 and libdbi-drivers-0.8.3 and Bacula
trunk code with ./configure --with-dbi

Last, configuring bacula-dir.conf with:

Catalog {
  Name = MyCatalog
  dbdriver = "dbi:postgresql"; dbport = 5432; dbaddress = 127.0.0.1
  dbname = regress; user = regress; password = ""
}

dbdriver can be: dbi:mysql and dbi:postgresql. In the future can be
all drivers supported by libdbi.

The following itens need some work or review:

1- Implement bath insert with dbi
2- Review the options ./configure --with-dbi
3- To solve/review the make install when ./configure --with-dbi.
Currently,the database scripts don't  is installed on system by make
install command.
4- Review in bscan patchs
5- Documentation to install, configure and compiling bacula and libdbi

Any question I am available.

Thanks
Attachment_(dbi.c): text/x-csrc, 27 KiB
Attachment_(trunk-dbi.patch): text/x-diff, 38 KiB
```

11400.txt

```
From: João Henrique Freitas <joaohf <at> gmail.com>
Subject: Re:_libdbi_backend_to_catalog_database
Newsgroups: gmane.comp.sysutils.backup.bacula.devel
Date: 2008-02-22 13:53:30 GMT (5 weeks, 1 day, 13 hours and 15 minutes ago)
Hello

Attached is the dbi patch for Bacula.

To use, are necessary libdbi-0.8.3 and libdbi-drivers-0.8.3 and Bacula
trunk code with ./configure --with-dbi

Last, configuring bacula-dir.conf with:
```