

Using PostgreSQL for a Domino 6 RDBMS Backend

Keith C. Perry, MS E.E.
PMHCC IS Department
kperry@pmhcc.org
netadmin@vcsn.com

31st December 2003

Abstract

This document discusses the steps needed to configure a Domino Server to communicate with a PostgreSQL server. The integration of a PostgreSQL database server will allow for the development of more complex Domino applications as well as better scalability and data integrity.

Contents

1 Prologue	2
1.1 Motivation	2
1.2 Requirements	2
1.3 Domino Upgrade Concerns	3
1.4 Test Environment	3
2 Workstation Setup	3
2.1 Domino Clients	3
2.2 PostgreSQL Client	3
3 PostgreSQL Setup	3
3.1 Server Configuration	4
3.2 Database Preparation	4
4 Domino Setup	4
4.1 ODBC	4
4.2 DECS	4
5 Methods	5
5.1 DCRs	5
5.2 Virtual Fields	6
6 Epilogue	6

1 Prologue

Domino is good for a number of things. However, for applications that call for your data to be stored in a relational fashion, it falls short.

Fortunately, Domino does have the ability to talk to external databases. There is a number of ways to do this based on the functionality you are trying to achieve. We will discuss two methods in this document. In either case, the integration you do will not only help you expand the complexity of your Domino applications but it will also protect your investment in Domino products.

As companies are becoming smarter about the software they choose to bring into their enterprise, one of the spaces that can no longer be ignored is the so called realm for Open Source Software¹. The low cost of ownership of OSS makes them attractive especially since in many cases they are as feature rich as similar commercial products. PostgreSQL ORDBMS² is very high on that list. Touted as “the most advanced Open Source database system in the world”³, PostgreSQL is feature rich with a very active user/development community.

In choosing a RDBMS for Domino, my goal find a cost effectively solution that was not going be major financial commitment and also something that could be used company-wide regardless of the application layer specifications. In short, PostgreSQL is the only product that was open enough and cost effective enough⁴ to fit the environment.

1.1 Motivation

I am much more of a Linux/PostgreSQL person than I am a Windows/Domino person. An authority on Domino I am not. What is documented here are the steps I took to complete a feasibility project where the goal was to have an RDBMS available to a Domino server.

The project was born out of a desire to expand Domino development to include more robust applications. It did not take very long to realize that this meant bringing an RDBMS in the environment. Furthermore the investment into Domino was significant and had been quite stable over the years for email and document management.

In addition to considering other products, I proposed to also consider PostgreSQL since I had already deployed several of applications with it. The only other system in production was Progress. MS-SQL and Oracle were also considerations.

The actual testing of the database integration to Domino is going to happen overtime because any current Domino application would have be re-written to take full advantage of the RDBMS server. In the most basic configuration, you could have Domino simply store each database’s document pages in the RDBMS. This is a trivial function but that one feature would add much better data integrity and reporting capabilities to the application environment. So the real challenge (and thus the real test) is developing applications that will leverage the power of a RDBMS with the features of Domino. That task however is outside the scope of this document.

1.2 Requirements

This document covers Domino 6.x releases and a modern⁵ version of PostgreSQL. You will need install the DECS product (included with Domino 6 releases) on your Domino server so that you can use external resources⁶. You will need to install the PostgreSQL ODBC driver⁷ so that the Domino server will be able use a PostgreSQL data source.

For testing, I recommend you install the admin & designer clients in addition to the regular notes client. This will allow you to try out different integration methods. You’ll probably want a DB administrative package on the workstation so you don’t have to keep running over to the PostgreSQL server to view the tables.

¹For more information on what **O**pen **S**ource **S**oftware is please visit <http://www.osdn.com>.

²**O**bject **R**elational **D**atabase **M**anagement **S**ystem

³see <http://advocacy.postgresql.org>

⁴PostgreSQL uses a BSD style license. You can review it at <http://www.postgresql.org/license.html>

⁵PostgreSQL 7 versions and beyond

⁶From more information on integrating Domino into the enterprise, (with DCRs, DECS or LEI) visit the forum at <http://www-10.lotus.com/ldd/eiforum.nsf> and see the Enterprise Integration White paper at <http://www-10.lotus.com/ldd/notesua.nsf/White%20Papers?OpenView>

⁷Obtainable from <http://gborg.postgresql.org/project/psqlodbc/projdisplay.php>

1.3 Domino Upgrade Concerns

The current environment was Domino 5.x so to do this testing, I needed to upgrade. You may be in a similar situation and if you are, you should definitely create a test environment for yourself. As with any upgrade there are quite possibly going to be “gotchas” and with Domino, you have to be completely committed to moving forward since your databases *will* be upgraded the first time you run the new version. You might also be forced to do upgrades to the OS and other lesser known components of your system as well. **Plan your work accordingly.**

1.4 Test Environment

Below is the software test configuration. Hardware specs are not being included as that relates more to tuning and “sizing” the servers to your environment (which is outside the scope of this document). Once you are confident in how to integrate Domino and PostgreSQL, tuning the environment to your needs will be a step along the road to production.

There were 2 servers and 1 workstation used for administration in our environment.

- NT 4.0 (SP6a) / Domino 6.5 with DECS and PostgreSQL ODBC driver 07.02.0005 installed
- Linux 2.4.22 / PostgreSQL 7.3.4
- Win 98 / Domino Admin, Designer & Notes 6.0 clients + pgAccess (latest version)

2 Workstation Setup

This section assumes you have a working PC that has available resources for all your administrative and test clients.

2.1 Domino Clients

From your installation CDs, you should install the Notes, Designer and Admin clients. You will need all of these at some point during your testing.

2.2 PostgreSQL Client

You do not *need* a GUI style admin client for PostgreSQL. However, I do find that in some cases they are handy to have around. Ideally you should have a good understanding of both a GUI and the CLI (i.e. the psql program that comes with the software) style clients.

In this case I chose pgAccess as the workstation client since the workstation was Windows 98. It is written in TCL so that needed to be installed as well (it is freely available like Perl and some other languages). You should of course try out several different clients- especially the ones that are cross platform. In that regard, I have used pgAccess and pgAdmin III. At this time, pgAccess is a bit dated but pgAdmin III has very active development and I use that on Linux and other Windows OS⁸. Try out some clients for yourself and see what your mileage is⁸.

3 PostgreSQL Setup

This section assumes that you have a PostgreSQL server database up and running on a server already. If not, you should refer to the documentation for authoritative help.

⁸Put your favorite search engine to work and look around for some tools. Reading newsgroups would be a smart thing to do too. If you don't have time to do that, a good starting point for PostgreSQL admin tools is <http://gborg.postgresql.org/browse.php?83>

3.1 Server Configuration

Since we are going to be making connections from external servers, you need to enable the TCP/IP option in the postmaster process. So for example, a start up line for a Linux server's rc.local file might be:

```
su - postgres -c "/usr/local/pgsql/bin/pg_ctl -D /usr/local/pgsql/data start"
```

Note that you need to set "**tcPIP_socket = true**" in postgresql.conf file as well.

In order to connect to the database, you will need a userid and password. So create your PostgreSQL user in the normal fashion. In order for that user to connect, you would have to probably add a line like this to pg_hba.conf:

```
host all all 10.0.0.0 255.0.0.0 md5
```

What that line says is to allow any user to connect to any database coming from an SSL or non-SSL IP network of 10.0.0.0/8. The password will be authenticated in an md5 fashion. For other authorization options, please review the PostgreSQL Administration document.

You should restart your server to make sure it starts as expected.

3.2 Database Preparation

For Domino to access database, the user you created must have the appropriate privileges granted. You may even want to create databases owned by the Domino user instead of having to grant privileges. The decision should be based on what is most appropriate for your application environment. If possible, you should have the test environment to mimic the production environment as much as possible.

4 Domino Setup

This section assumes you have a working Domino 6 server and a workstation with the clients installed from section 1.2. If not, you should refer to the documentation for authoritative help. You may also need to refer to OS documentation for section 4.1.

4.1 ODBC

ODBC connectivity is a common way to link heterogeneous systems together. The windows operating system comes with some drivers but chances are, you do not have the driver for PostgreSQL. So the first thing you'll need to do is install the ODBC driver for PostgreSQL⁹ on the Domino server.

Next, you should go to the Control Panel and double-click the **ODBC** or **Data Source** icon. Open that up and choose the tab that says System DSN¹⁰ and then click **Add**. Find the PostgreSQL driver (its the plainest one- not the Beta or Unicode) and click **Finish**. Fill in the appropriate information for the PostgreSQL server and click **Save**. Make sure you include the database name. You should now see an entry for the server. Click **Ok** to exit.

Keep in mind you will have to do an ODBC connection for each logical database you want to connect to. That means that if you 5 databases on the same server under the same user name, you need to add a DSN for each one.

4.2 DECS

Find the notes.ini on the Domino server. You will need to append your **ServerTasks=** line with **DECS** and add or append the **EXTMGR_ADDINS=** line with **decsext**.

Start (or restart) the Domino server.

Watch the Domino server's console for messages about DECS. It should create the database for you from the DECS template file. Make sure you do not have one already in your Domino data directory. If you do, stop the server, delete the database and start the server again. This will get you a clean DECS database.

⁹see <http://gborg.postgresql.org/project/psqlodbc/projdisplay.php> for the latest driver information and links

¹⁰**Data Source Name**- this is the same name you will use in section 4.2 to identify the ODBC source to the DECS database.

On the workstation, open the Domino Admin client. Type <ctrl-O> to open a database. Switch to the Domino server and you choose DECS Administrator and click **Ok**.

Click **Connections** on the left hand side of the DECS Administrator then click **Add Connection** toward the top of the screen. Choose Connect to ODBC. Scroll down to where it says Data Source and enter a DSN you want to use that you setup in section 4.1. You don't need a userid id and password if it was supplied as part of the DSN. Scroll down again and now you can choose the table in the database you wish to link to. Click **Save & Close** at the top of the screen when you are done. *You will have add a connection for every table you want to link to.*

5 Methods

This section assumes have a domino database setup with the equivalent fields from your database tables. Furthermore, this section is being provided as a bridge point between the PostgreSQL server connection setup and the actual Domino development using external resources. You should consult your Domino resources for in depth information.

There are two ways to use and access data stored on the PostgreSQL server.

1. DCRs- these are setup in the Domino Designer client. It allows the database designer to make external field references (on the forms) to any DSN created in section 4.1 on the Domino server. You can also import the external records if necessary.
2. Virtual Fields Activity- these are setup using the DECS Administrator database by the Domino administrator. For each database form you can reference Domino server DSN by mapping the form fields to the table fields. Thus the external reference is transparent to the database designer Furthermore, you can "monitor" the activity of these fields in the DECS administrator. This method is more transparent than DCRs because the designer does not need to get involved with the external connections (a task that is better suited to the server administrator anyway). You can also import the external records if necessary.

Keep in mind that PostgreSQL is being used as the backend so even though these two methods may seem like we are synchronizing the Domino and PostgreSQL databases what we are really doing is importing table data and creating domino documents. From that point on Domino should be considered the interface. That means you should not make data changes via any other interface. For instance, deleting a record on PostgreSQL will cause an error on the Domino side (though with DECS, you can ignore the error. That is because the Domino key document will not have a reference to anything. Another way to say this is that in order to maintain the highest level of data integrity, information should be managed *entirely* from the Domino side¹¹. The PostgreSQL server can still be used for other query-only clients and reporting.

5.1 DCRs

To create DCRs, you must first go into the Designer client and open your Domino database. Click the **Recent Databases** icon from the vertical icon menu on the left. Click **Shared Resources** and then **Data Connections**. Click the **New Data Connection** at the top of window.

There are a number of fields here but you only *need* the following:

- Name - the name you want to use for the DCR connection
- Class - should be set to RDBMS
- Type - should be set to ODBC
- Data Source - should be set to the DSN you created in section 4.1.

¹¹The Domino LEI product *does* allow for true synchronization between both databases. LEI however, *does not* come with Domino 6 and thus was not tested.

At this point, you should be able to click **Browse Metadata**. If not, click on **Data Connection** at the top of the windows and select Database. Check the box at the bottom that says, Allow Connection to External Databases using DCRs. Close and re-open the window. You should now be able to browse the database table metadata from previous the Data Connection screen.

To use the use fields from the database, your Domino application developer would need to make the Domino form field external. It is at that point you link the external field to the PostgreSQL table field.

Once all your fields are links, you can click **Import external records** in the Data Connections screen. This will create the database documents on the Domino server. From this point on the document's form fields will be linked to the PostgreSQL table fields.

5.2 Virtual Fields

Virtual Fields are created from the DECS Database Administrator. After the database opens up, click **Add Activity**.

There are several sections here that you need to complete:

1. Identification - this is the name of your activity.
2. Domino Application - choose the domino database that is going to link to the PostgreSQL server. You will then choose the form that has the field you want to link.
3. Lotus Connection - choose the DSN you created in step 4.1. You will then choose the table that is being linked to.
4. Mapping - in this screen, you map the Domino database field to the PostgreSQL database table fields. This includes mapping any keys you have.
5. Event Monitoring - for the test case, we selected all events so anything could be monitored. In a production scenario this might not be appropriate. There are some other options in this section. As always, refer to your Domino documentation for more information.

At the top of that window, click **Save & Close**. This will take you back to the activity screen. Before we can start the activity (and thus synchronize the databases) Double-click on the activity you just created to get back to you configuration screen. At the top of the window, click **Initialize Keys Now**. This will take as the Domino documents are being created. When finished, you can go back to the activity screen and click **Start**. At this point, the databases will stay in sync and you can monitor the activity.

6 Epilogue

This document is designed to be a short guide and introduction to PostgreSQL for Domino application designers who are interested in developing more robust products. It is also designed to be a point of information for the PostgreSQL community that Domino is an area that we can interface with. I welcome your comments, feedback and questions.

There are many products available for you to develop your applications. If you search around long enough, you will come across the religious war between OSS and commercial products. As usual, the best solutions lie somewhere in the middle. The integration of PostgreSQL and Domino is a very good example of the benefits of implementing both products and methodologies. Additionally because PostgreSQL is OSS, monetarily it will be one of the cheaper ways to get your feet wet using an RDBMS.

Remember, always thoroughly research the products you choose and choose the right product for the job. Let industry standards guide you, not just brand names. Invest in the experience of people that are doing and have done because in the end, *that* is the person who is going to take the time to help you.