



# **PostgreSQL Inc.**

## **Enterprise Replication Server eInstallation Guide**

---



<b>1 INTRODUCTION.....</b>	<b>3</b>
1.1 GENERAL.....	3
1.2 USEFUL INTERNET LINKS.....	3
<b>2 REQUIREMENTS.....</b>	<b>4</b>
<b>3 OVERVIEW.....</b>	<b>5</b>
<b>4 INSTALL THE ERSERVER SOFTWARE.....</b>	<b>6</b>
4.1 CREATE A UNIX ACCOUNT FOR ERSERVER.....	6
4.2 COMPILE AND INSTALL ERSERVER.....	7
4.3 ADD ERSERVER COMMANDS TO YOUR PATH ENVIRONMENT VARIABLE.....	7
<b>5 INSTALLATION OF ERSERVER.....</b>	<b>8</b>
5.1 ADD THE PL/PgSQL LANGUAGE TO YOUR MASTER DATABASE.....	8
5.2 INITIAL CREATION OF THE REPLICATION INFORMATION TABLES IN THE MASTER AND SLAVE DATABASES.....	8
<b>6 INSTALLATION OF ERSERVER ON ADDITIONAL SLAVE DATABASES.....</b>	<b>9</b>
6.1 CREATION OF THE REPLICATION INFORMATION TABLES IN THE SLAVE DATABASE(S).....	9
<b>7 ADDING TABLES TO THE REPLICATION LIST.....</b>	<b>10</b>
7.1 ADDING TABLES.....	10
7.2 THINGS TO REMEMBER.....	10
<b>8 SECURING THE ERSERVER INSTALLATION.....</b>	<b>11</b>
8.1 ENSURE THE REPLICATION.CFG FILE IS ONLY READABLE BY THE ERSERVER USER.....	11
8.2 ENSURE THE LOG FILES ARE ONLY READABLE BY THE ERSERVER USER.....	11
<b>9 STARTING AND STOPPING ERSERVER.....</b>	<b>12</b>
9.1 STARTING ERSERVER.....	12
9.2 STOPPING ERSERVER.....	12
<b>10 MONITORING ERSERVER.....</b>	<b>13</b>
10.1 THE LOG FILES.....	13
<b>11 TUNING ERSERVER INSTALLATIONS.....</b>	<b>14</b>
11.1 TUNING PostgreSQL.....	14
11.2 SIZING THE JAVA VIRTUAL MACHINE.....	14
11.3 DATA INTEGRITY ACROSS SERVER CRASHES ON SLAVE DATABASES.....	14
<b>12 FINAL CHECKLIST.....</b>	<b>15</b>
12.1 ENSURE DEBUG AND VERBOSE MODE ARE TURNED OFF IN THE REPLICATION.CFG FILE.....	15



# 1 INTRODUCTION

## 1.1 General

This document details the installation and configuration of PostgreSQL Inc.'s Enterprise Replication Server (eRServer). It is geared to an intended audience of System Installers, Unix Administrators and Security Managers / Officers within PostgreSQL, Inc. project and technical support teams.

## 1.2 Useful Internet Links

eRServer            <http://www.erserver.com>

PostgreSQL        <http://www.postgresql.org>

PostgreSQL, Inc. <http://www.pgsql.com>

Perl DBD::Pg      [http://www.perl.com/CPAN-local/modules/by-category/07\\_Database\\_Interfaces/DBD/](http://www.perl.com/CPAN-local/modules/by-category/07_Database_Interfaces/DBD/)



## 2 REQUIREMENTS

1. A supported Operating System
2. Sun JDK 1.2.x, or 1.3.1
3. Perl 5.005\_04 or greater
4. Perl DBI installed
5. Perl DBD::Pg installed\*
6. eRServer 1.2
7. GNU gcc 2.95.3 or greater
8. GNU make 3.79.1 or greater
9. PostgreSQL Superuser access to the master and slave database

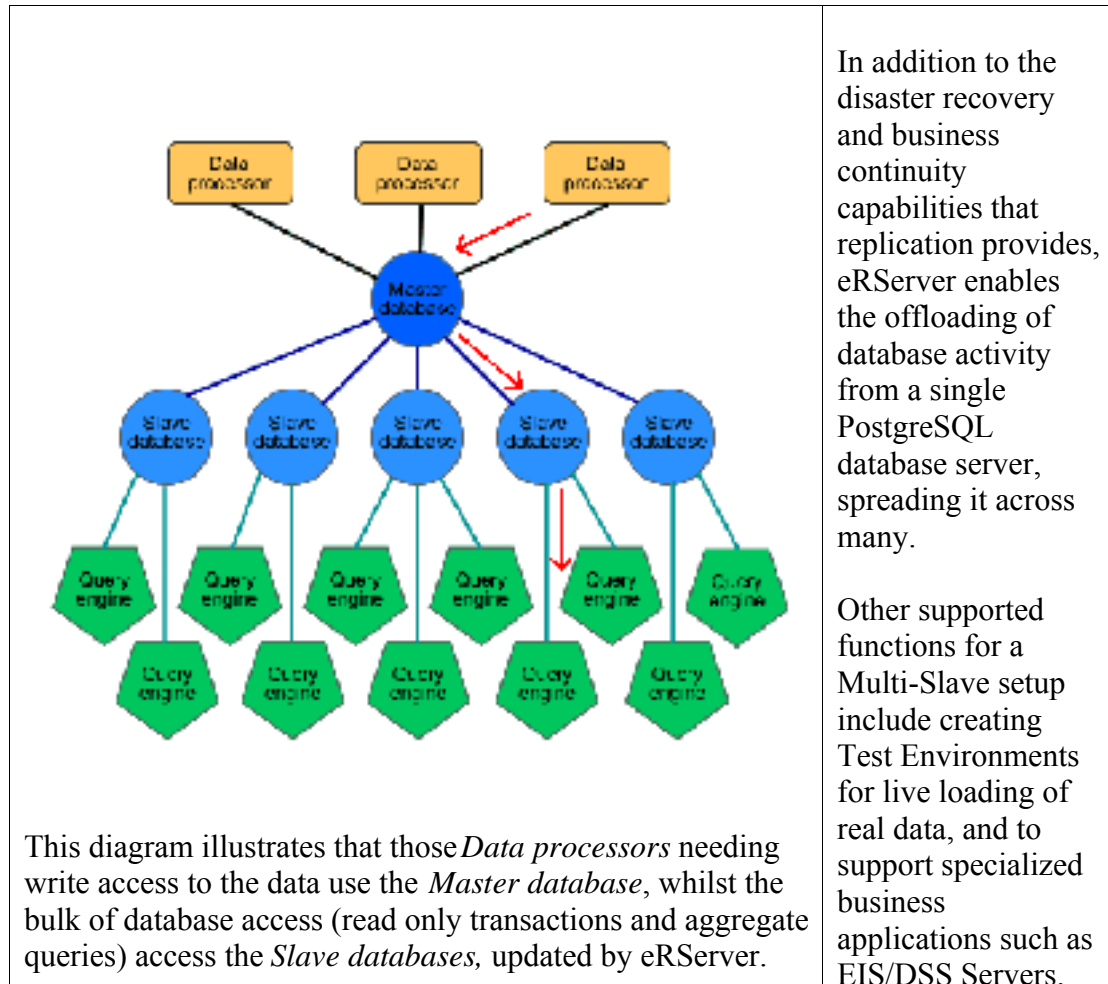
\* The Perl DBD::Pg libraries are available from:

[http://www.perl.com/CPAN-local/modules/by-category/07\\_Database\\_Interfaces/DBD/](http://www.perl.com/CPAN-local/modules/by-category/07_Database_Interfaces/DBD/)



### 3 OVERVIEW

This section gives an overview of eRServer in an n-tier environment.





## 4 INSTALL THE ERSERVER SOFTWARE

### 4.1 Create a unix account for eRServer

For maximum flexibility and security we recommend you create a unix "erserver" user and unix "erserver" group for the eRServer software to be controlled by.

For Solaris and Linux:

```
# groupadd erserver  
# useradd -g erserver
```

For FreeBSD:

```
# pw groupadd erserver  
# pw user add -m -n erserver
```



## 4.2 Compile and install eRServer

Extract the eRServer 1.2 tar file in a temporary installation directory.

```
$ tar xzf erserver12.tar.gz
```

Enter the extracted directory:

```
$ cd erserver
```

Configure the source code:

```
$ ./configure --with-pgincludes=/opt/pgsql
```

You need to tell the configuration script where PostgreSQL is installed by including the “--with-pgincludes” parameter. In this example, PostgreSQL was already installed in [/opt/pgsql](#). Optionally, if you want eRServer to install its files in a specific directory, you can tell it where by using the “--prefix” parameter in the above command.

For example, if you had wanted eRServer to install its files in the [/opt/erserver](#) directory instead of the default, you would have run this configure command instead:

```
$ ./configure --prefix=/opt/erserver --with-pgincludes=/opt/pgsql
```

Compile the configured source code:

```
$ make
```

Install the eRServer software in the target installation directory:

```
$ su  
# make install  
# exit  
$
```

## 4.3 Add eRServer commands to your PATH environment variable

It is important to add the eRServer commands in the bin subdirectory of your eRServer installation to the environment path for your erserver user. The method of doing this varies depending upon operating system and isn't covered in this manual.



## 5 INSTALLATION OF ERSERVER

### 5.1 Add the PL/pgSQL language to your master database

The PostgreSQL functions needed by eRServer are written in the procedural language PL/pgSQL.

You need to enable this language for each master database by using this command:

```
$ createlang plpgsql masterdb
```

An alternative method is to enable PL/pgSQL for the special *template1* database, and any subsequent databases (including the master database) created after this will automatically have it enabled.

```
$ createlang plpgsql template1
```

### 5.2 Initial creation of the Replication information tables in the master and slave databases

Each master and slave database uses several tables to hold and co-ordinate the replication. You should create these up by running the *ers\_setup* script, passing it the needed parameters.

Parameter	Description
--masterserver	The name of the server that the master database is on.
--masterport	The TCP port number that the master database is on. The default is 5432.
--masterdb	The name of the master database.
--masteruser	The PostgreSQL username to connect to the master database with.
--masterpass	The password for the master database user.
--slaveserver	The name of the server that the first slave database is on.
--slaveport	The TCP port number that the first slave database is on. The default is 5432.
--slavedb	The name of the first slave database.
--slaveuser	The PostgreSQL username to connect to the first slave database with.
--slavepass	The password for the first slave database user.

For example:

```
$ ers_setup --masterserver=servername --masterdb=accounts --masteruser=someuser --  
masterpass=apasswd --slaveserver=otherserver --slavedb=erp --slaveuser=anotheruser --slavepass=foo
```



## 6 INSTALLATION OF ERSERVER ON ADDITIONAL SLAVE DATABASES

This needs to be done for each slave database.

### 6.1 Creation of the Replication information tables in the slave database(s)

Each slave database uses two tables for holding replication co-ordination data, created by running the *ers\_addslave* command and telling it how to connect to the slave database.

The connection information you give is also saved to eRServer's *sreplication.cfg* file, so the new slave database will be included in the replication process with the next restart of eRServer.

A point to note is that the owner of the slave database needs to have PostgreSQL super-user status.

```
$ ers_addslave --slaveserver=slavehost --slavedb=dbname --slaveuser=slaveuser --slavepass=slavepw
```



## 7 ADDING TABLES TO THE REPLICATION LIST

### 7.1 Adding tables

Adding tables to the replication list is done with the `ers_addtable` command. You tell it the name of the table you want added to the replication list, then it will automatically go through all of the master and slave servers that have been defined and add it to their lists.

```
$ ers_addtable --table=tablename
```

### 7.2 Things to remember

- ) Every table for replication must have an integer based column for co-ordinating changes. The eRServer installation scripts will add an `ers_uniq` column for this to all of the tables marked for replication. Do not change this *column* with your SQL queries.
- ) All the tables to be replicated should have the same definition on both the master and slave databases.
- ) Sequences in the master database aren't updated on the slave database(s).

If you have tables in the master database that use Sequences for number generation, the corresponding Sequences in the slave database(s) will remain unchanged regardless of how much data is replicated.

In a failover situation where a slave database needs to become the master database, the Sequences in the slave database have to be updated to current values before its operation as a master begins, otherwise duplicate sequence numbers will occur and insertions into the database will fail.

This is done through using the `setval()` and `max()` commands on the slave database at failover time, prior to it becoming a master.



## 8 SECURING THE ERSERVER INSTALLATION

### 8.1 Ensure the replication.cfg file is only readable by the erserver user

Because the replication.cfg file contains the passwords used to connect to your remote master and slave databases, you need to take care that it's only readable by the erserver user.

```
$ chmod 600 [erserver base installation directory]/etc/replication.cfg
```

### 8.2 Ensure the log files are only readable by the erserver user

Because sensitive information can sometimes be present in the eRServer log files, you need to take care that they too are only readable by the erserver user.

```
$ chmod 700 [erserver base installation directory]/log  
$ chmod 600 [erserver base installation directory]/log/*
```



## 9 STARTING AND STOPPING ERSERVER

### 9.1 Starting eRServer

eRServer requires the Sun JDK 1.2.x or 1.3.x, and is recommended to be run on the same server as the master database.

You start it as the *erserver* user, as follows:

```
$ ers_ctl start
```

### 9.2 Stopping eRServer

You stop eRServer as the *erserver* user, as follows:

```
$ ers_ctl stop
```



## 10 Monitoring eRServer

### 10.1 The log files

As it runs, eRServer will generate a log file called “replication.log” in the log sub-directory of your eRServer installation.

You can check this file to verify the operation of eRServer, and to watch for errors if you ever have problems.



## 11 TUNING ERSERVER INSTALLATIONS

### 11.1 Tuning PostgreSQL

You should run this SQL command at least once on PostgreSQL installations as a PostgreSQL superuser, otherwise performance may be sub-optimal.

```
UPDATE pg_class SET relpages=1000, reltuples=100000 WHERE relname like '_rserv_log__'
```

### 11.2 Sizing the Java Virtual Machine

In the `ers_ctl` script, there is a variable called `JVM_MAX` tells the Java Virtual Machine running eRServer the maximum amount of memory it should ever use. The default at installation time is 512MB, but can be changed if needed. One reference installation uses this set to 2000MB.

### 11.3 Data integrity across server crashes on slave databases

For slave databases that have no need of data integrity in the event of a crash, it may be worthwhile turning off the PostgreSQL `fsync()` call for improved disk performance. This setting is adjustable in the main PostgreSQL configuration file and is documented in the main PostgreSQL manuals.



## 12 FINAL CHECKLIST

### 12.1 Ensure debug and verbose mode are turned off in the replication. cfg file

Both of these modes have a performance hit when used, although the verbose mode is negligible.

An additional reason to turn them off in a production environment is because of the large amount of log files they generate.