pg_dumpall

You can dump the whole PostgreSQL cluster with pg_dumpall. That's *all* the databases and all the globals for a single cluster. From the command line on the server, I'd do something like this. (Mine's listening on port 5433, not on the default port.) You may or may not need the --clean option.

```
$ pg_dumpall -U postgres -h localhost -p 5433 --clean --file=dump.sql
```

This includes the globals--information about users and groups, tablespaces, and so on.

If I were going to backup a single database and *move it to a scratch server*, I'd dump the database with pg_dump, and dump the globals with either

- `pg_dumpall --globals-only`, or
- `pg_dumpall --roles-only` (if you only need roles)

like this.

```
$ pg_dump -U postgres -h localhost -p 5433 --clean --file=sandbox.sql sandbox
$ pg_dumpall -U postgres -h localhost -p 5433 --clean --globals-only --file=globals.sql
```

Outputs are just text files.

After you move these files to a different server, load the globals first, then the database dump.

```
$ psql -U postgres -h localhost -p 5433 < globals.sql
$ psql -U postgres -h localhost -p 5433 < sandbox.sql
```

> I thought pg_dumpall would at least backup foreign keys, but even that seems to be an 'option'. According to: http://www.postgresql.org/docs/9.1/static/app-pg-dumpall.html even with pg_dumpall I need to use a -o option to backup foreign keys

No, that reference says "Use this option if your *application* references the OID columns in some way (e.g., in a foreign key constraint). Otherwise, this option should not be used." (Emphasis added.) I think it's unlikely that your *application* references the OID columns. You don't need to use this option to "backup foreign keys". (Read the dump file in your editor or file viewer.)

http://www.postgresonline.com/special_feature.php?sf_name=postgresql90_pg_dumprestore_cheatsheet&outputformat=html

PostgreSQL 9.0 pg_dump, pg_dumpall, pg_restore Cheat Sheet

pg_dump, pg_dump_all, pg_restore are located in the bin folder of PostgreSQL and PgAdmin III installs.

pg_dump dumps a database as a text file or to other formats.

Manual: http://www.postgresql.org/docs/9.0/interactive/app-pgdump.html
Usage: pg_dump [OPTION]... [DBNAME]

pg_dumpall extracts a PostgreSQL database cluster into an SQL script file restorable with psql.

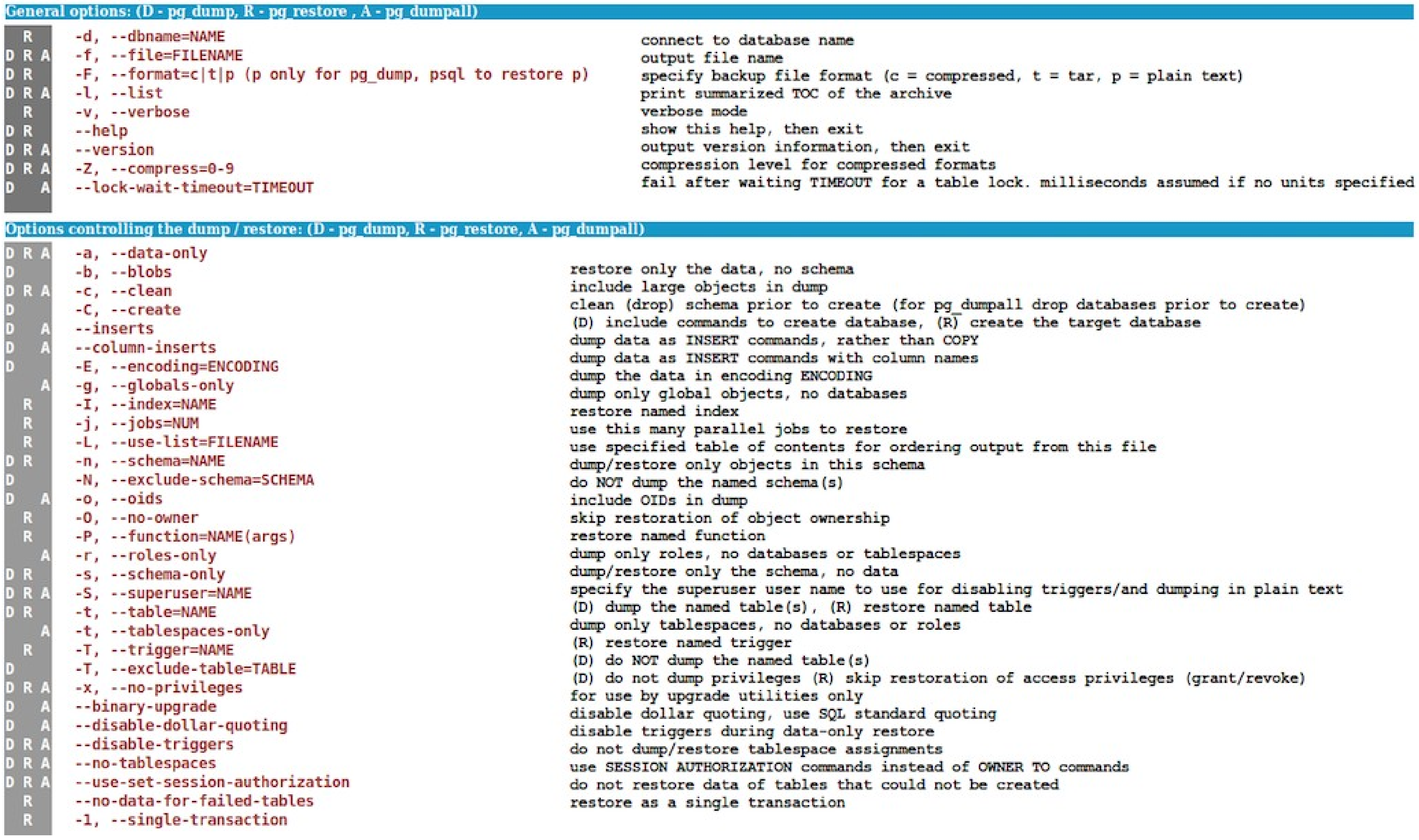
Manual: http://www.postgresql.org/docs/9.0/interactive/app-pg-dumpall.html
Usage: pg_dumpall [OPTION]...

pg_restore restores a PostgreSQL database from an archive created by pg_dump.

Manual: http://www.postgresql.org/docs/9.0/interactive/app-pgrestore.html
Usage: pg_restore [OPTION]... [FILE]

R -d, --dbname=NAME connect to database name

DRA -f, --file=FILENAME output file name



General options: (D - pg_dump, R - pg_restore , A - pg_dumpall)

	R		-d, --dbname=NAME	connect to database name
D	R	A	-f, --file=FILENAME	output file name
D	R		-F, --format=c|t|p (p only for pg_dump, psql to restore p)	specify backup file format (c = compressed, t = tar, p = plain text)
D	R	A	-l, --list	print summarized TOC of the archive
	R		-v, --verbose	verbose mode
D	R		--help	show this help, then exit
D	R	A	--version	output version information, then exit
D	R	A	-Z, --compress=0-9	compression level for compressed formats
D		A	--lock-wait-timeout=TIMEOUT	fail after waiting TIMEOUT for a table lock. milliseconds assumed if no units specified

Options controlling the dump / restore: (D - pg_dump, R - pg_restore, A - pg_dumpall)

D	R	A	-a, --data-only	restore only the data, no schema
D			-b, --blobs	include large objects in dump
D	R	A	-c, --clean	clean (drop) schema prior to create (for pg_dumpall drop databases prior to create)
D			-C, --create	(D) include commands to create database, (R) create the target database
D		A	--inserts	dump data as INSERT commands, rather than COPY
D		A	--column-inserts	dump data as INSERT commands with column names
D			-E, --encoding=ENCODING	dump the data in encoding ENCODING
		A	-g, --globals-only	dump only global objects, no databases
	R		-I, --index=NAME	restore named index
	R		-j, --jobs=NUM	use this many parallel jobs to restore
	R		-L, --use-list=FILENAME	use specified table of contents for ordering output from this file
D	R		-n, --schema=NAME	dump/restore only objects in this schema
D			-N, --exclude-schema=SCHEMA	do NOT dump the named schema(s)
D		A	-o, --oids	include OIDs in dump
	R		-O, --no-owner	skip restoration of object ownership
	R		-P, --function=NAME(args)	restore named function
		A	-r, --roles-only	dump only roles, no databases or tablespaces
D	R		-s, --schema-only	dump/restore only the schema, no data
D	R	A	-S, --superuser=NAME	specify the superuser user name to use for disabling triggers/and dumping in plain text
D	R		-t, --table=NAME	(D) dump the named table(s), (R) restore named table
		A	-t, --tablespaces-only	dump only tablespaces, no databases or roles
	R		-T, --trigger=NAME	(R) restore named trigger
D			-T, --exclude-table=TABLE	(D) do NOT dump the named table(s)
D	R	A	-x, --no-privileges	(D) do not dump privileges (R) skip restoration of access privileges (grant/revoke)
D		A	--binary-upgrade	for use by upgrade utilities only
D		A	--disable-dollar-quoting	disable dollar quoting, use SQL standard quoting
D	R	A	--disable-triggers	disable triggers during data-only restore
D	R	A	--no-tablespaces	do not dump/restore tablespace assignments
D	R	A	--use-set-session-authorization	use SESSION AUTHORIZATION commands instead of OWNER TO commands
	R		--no-data-for-failed-tables	do not restore data of tables that could not be created
	R		-1, --single-transaction	restore as a single transaction

```
Connection options:

  -h, --host=HOSTNAME      database server host or socket directory
  -p, --port=PORT          database server port number
  -U, --username=NAME      connect as specified database user
  -w, --no-password        never prompt for password
  -W, --password           force password prompt (should happen automatically)
  -e, --exit-on-error      exit on error, default is to continue

If no input file name is supplied, then standard input is used.
```

pg_restore Example Use

restore whole database
```
pg_restore --host=localhost --dbname=db_to_restore_to --username=someuser
/path/to/somedb.backup
```

restore only the schema (no objects)
```
pg_restore --schema-only=someschema --dbname=db_to_restore_to --username=someuser
/path/to/somedb.backup
```

restore only a specifically named schema's data: note the schema has to exist before hand
```
pg_restore --schema=someschema --dbname=db_to_restore_to --username=someuser
/path/to/somedb.backup
```

Get a listing of items in backup file and pipe to text file (only works for tar and compressed formats)
```
pg_restore --list --file=C:/somedb_list.txt backupfilepath
```

pg_dump, pg_dumpall Example Use

dump database in compressed include blobs show progress
```
pg_dump -h someserver -p 5432 -U someuser -F c -b -v -f "/somepath/somedb.backup" somedb
```

dump database in utf8 encoding and wait a maximum of 1 minute for a lock
```
pg_dump -h someserver -p 5432 -U someuser -E UTF8 --lock-wait-timeout=6000 -F c -b -v -f
"/somepath/somedb.backup" somedb
```

dump all tables named roads in all schemas in compressed binary format
```
pg_dump -h someserver -p 5432 -U someuser -E UTF8 -t "*.roads" -F c -b -v -f
"/somepath/somedb.backup" somedb
```

backup pgagent schema of postgres db in plain text copy format, maintain oids
```
pg_dump -h someserver -p 5432 -U postgres -F p -o -v -n pgagent -f "C:/pgagent.sql" postgres
```

backup table roads in schema ma use column inserts rather than copy
```
pg_dump -h someserver -p 5432 -U postgres -F p -t "ma.roads" --column-inserts -f
"C:/ma.roads.sql" somedb
```

dump all databases - note pg_dumpall can only output to plain text
```
pg_dumpall -h someserver -p 5432 -U someuser -c -o -f "/somepath/alldbs.sql"
```

Restore a full database cluster backup generated with pg_dumpall
```
psql -h someserver -p 5432 -U postgres -f /somepath/alldbs.sql postgres
```