



Administering Your PostgreSQL Geodatabase

Michael Downey

Jim McAbee

Agenda

- **News since the last UC**
- **How do I ...**
 - **Configure Postgres to support geodatabases?**
 - **Create geodatabases?**
 - **Control access to my data?**
 - **Make sure that my data is safe?**
 - **Maintain good performance?**

PostgreSQL

A Free Open Source Option

- **Introduction to PostgreSQL**
 - <http://www.postgresql.org/>
 - **Open Source Enterprise level RDBMS**
 - **Free, distributed with bsd license**
 - **Supported by an active online development community**
- **Possible additional cost for administration guidance**
 - **3rd Quadrant, Enterprise DB, etc...**

PostgreSQL and ArcGIS Server for Amazon Web Services

- **Geodatabases in PostgreSQL included with ArcGIS for Server Amazon Machine Image (AMI)**
 - Includes optional PostgreSQL database cluster (Ubuntu Linux)
 - Amazon Relational Data Service (RDS) for PostgreSQL (Ubuntu Linux or Windows)
- **When ArcGIS Server Cloud Builder for AWS used to launch**
 - 2 databases in cluster: egdb and geodata
 - egdb – registered as a managed database
 - geodata – intended for use as a replicated geodatabase



How do I...?

**Common questions when working with PostgreSQL
databases and geodatabases**



How do I... configure PostgreSQL to support geodatabases?

How do I configure PostgreSQL to support geodatabases?

- **Install a supported version of PostgreSQL**
PostgreSQL database requirements for ArcGIS 10.3.x
- **PostgreSQL configuration parameters**
- **Enabling Connections to PostgreSQL**
- **PostgreSQL client libraries for ArcGIS**
- **ArcGIS Spatial Type libraries for PostgreSQL**



Supported versions at 10.3.x

- PostgreSQL 9.3.5 (64 bit) and PostGIS 2.1 (optional)
- PostgreSQL 9.2.9 (64 bit) and PostGIS 2.0 (optional)
- PostgreSQL 9.1.14 (64 bit) and PostGIS 2.0 (optional)
- Customers using 9.0.5 or earlier need to upgrade.

ArcGIS and Postgres Changes since last year

What's new at 10.3.x

Support for PostgreSQL 9.3.5

Support for PostGIS 2.1

New Geodatabase Administration Geoprocessing Tools

PostgreSQL Configuration Parameters

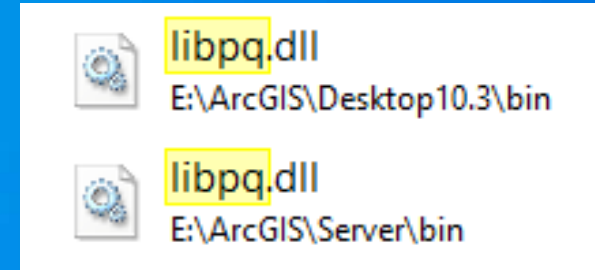
- **postgresql.conf**
 - most defaults ok, testing and monitoring should be done
- **Memory - #shared_buffers**
 - Set greater than minimum value of 128KB or 16KB times max_connections...
 - Windows – best 64MB to 512MB, little benefit to set higher, use OS cache
 - Linux – 25% of Physical Memory to start and as possible (no. of instances)
- **Connections**
 - #max_connections=100 (default)
 - one ArcGIS connection = multiple PostgreSQL connections
 - Geodatabase connections parameter (sde_server_config) – no longer relevant at 10.3

More PostgreSQL Initialization Parameters and Environment Variables

- **max_locks_per_transaction**
 - Most cases default value of 64 is sufficient
 - If loading several thousand datasets at once may need to be increased and possible OS shared memory configuration
- **cursor_tuple_fraction**
 - Set to 0.1 (10% of rows returned rapidly) by default
 - ArcGIS 10.2 and later use a setting of 1.0 (*set on the stream by ArcSDE*), data does not display until 100% has been returned, cursor_tuple_fraction environment variable set for session or system used to change, not postgresql.conf
- **st_geometry_output_format**
 - If performing sql queries (outside ArcGIS) against st_geometry, setting this environment variable (= TYPE) will cause the query to return st_geometry vs. WKT (well known text) which could result in improved performance.
- **Environment Variables (PG_HOME)**

PostgreSQL Client and enabling connections

- ArcGIS 10.3 and 10.3.x for Desktop, Server and Engine include the required PostgreSQL files (4 library files)



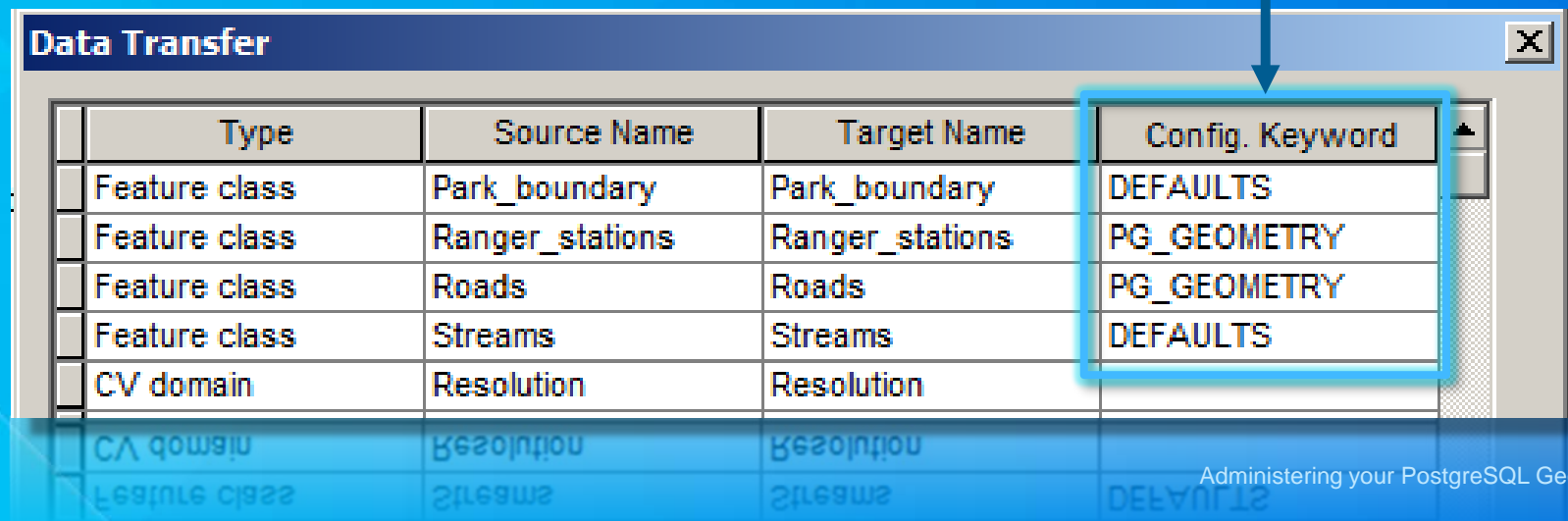
- Enable connections
pg_hba.conf

```
# TYPE DATABASE USER ADDRESS METHOD
# IPv4 local connections:
host all all 0.0.0.0/0 trust
host all all 127.0.0.1/32 md5
# IPv6 local connections:
host all all 0.0.0.0/0 trust
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#host replication postgres 127.0.0.1/32 md5
#host replication postgres ::1/128 md5
```

ST_Geometry: Default Geometry Datatype

- SQL 3 specification of user-defined data types (UDTs) – ISO and OGC compliant
- Provides structured query language (SQL) access to the geodatabase and database.
- Can be used in PostgreSQL databases that contain a geodatabase and those that do not.

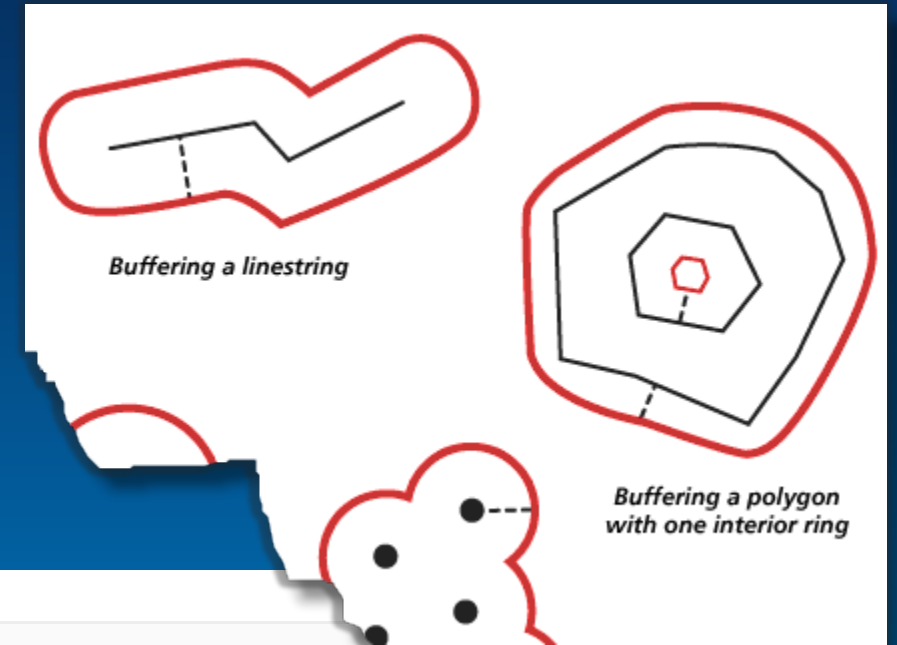
SDE.SDE_DBTUNE
table for storing keywords
and associated parameters



Type	Source Name	Target Name	Config. Keyword
Feature class	Park_boundary	Park_boundary	DEFAULTS
Feature class	Ranger_stations	Ranger_stations	PG_GEOMETRY
Feature class	Roads	Roads	PG_GEOMETRY
Feature class	Streams	Streams	DEFAULTS
CV domain	Resolution	Resolution	

Spatial Types and Functions

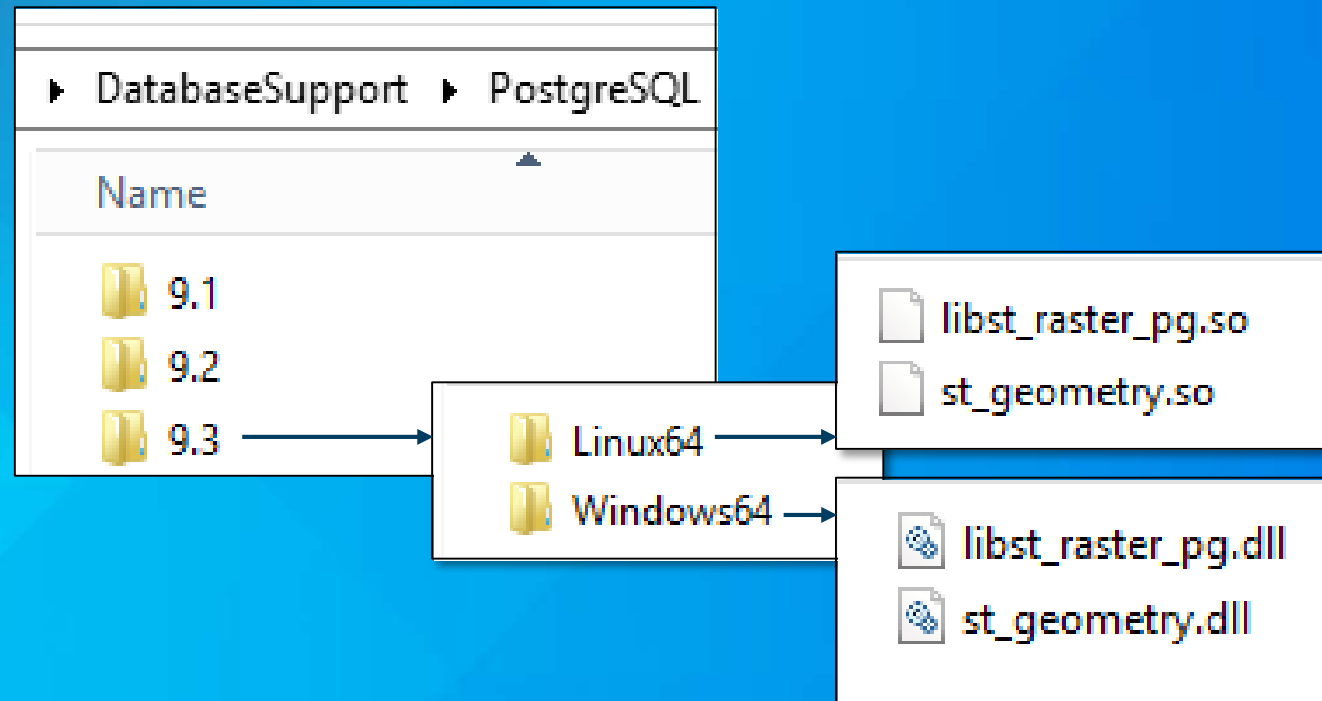
- Creation of Features through SQL
- Analysis through SQL
- Geodatabase behavior not supported through SQL



```
SELECT sa.name AS sensitive_area, hs.name AS hazardous_sites  
  
FROM sensitive_areas sa, hazardous_sites hs  
  
WHERE st_overlaps(sa.zone, st_buffer(hs.location, .01) = 't');
```

Configuring PostgreSQL for ST_Geometry spatial type

- Copy st_geometry library for PostgreSQL version to PostgreSQL lib directory
- Necessary for everything except a non-Geodatabase with PostGIS.



Datatype Mapping

- PostgreSQL supports almost 100 datatypes
- ArcGIS has 8
- Some PostgreSQL datatypes are mapped to one ArcGIS datatype
- Some datatypes are not supported
 - Error: “invalid column datatype”
 - Documented in online help

PostgreSQL

ArcGIS data types	PostgreSQL data types created	Other PostgreSQL data types that can be viewed	Notes
BLOB	BYTEA		
DATE	TIMESTAMP WITHOUT ZONE	TIMESTAMP	
DOUBLE	NUMERIC(p)	BIG SERIAL, DOUBLE PRECISION	The precision
FLOAT	NUMERIC(p)		The precision
GEOMETRY	ST_GEOMETRY or GEOMETRY		In a geodatabase created in the ArcGIS environment, the ST_Geometry type is the ST_Point type created in ArcGIS. To use ST_Geometry in a geodatabase, you must create a subtype. To use the PostgreSQL type, see the PostgreSQL documentation.
GLOBAL ID	VARCHAR(38)		Only supported
GUID	VARCHAR(38)		

GLOBAL ID	VARCHAR(38)		Only supported
GLOBAL ID	VARCHAR(38)		Only supported



How do I... create geodatabases?

Databases and Geodatabases

- A database is a PostgreSQL object
 - There can be many per PostgreSQL cluster (instance)
- A geodatabase is an ArcGIS construct hosted in a database
 - Administrative schema
 - One allowed in each database
- Options for creating geodatabases
 - Use a GP tool to create a new geodatabase from scratch
 - Use a GP tool to create a new geodatabase in an existing database



Database vs. Geodatabase

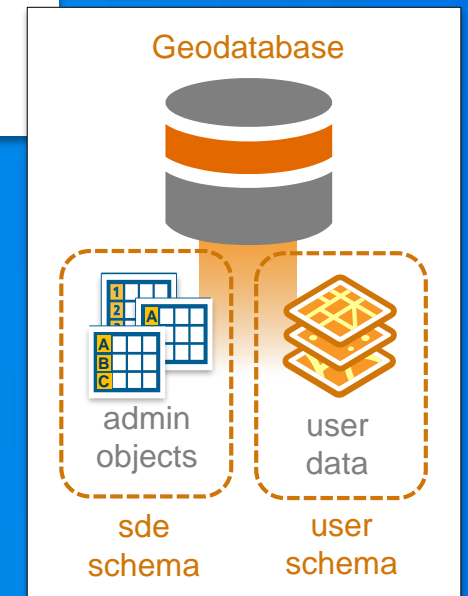
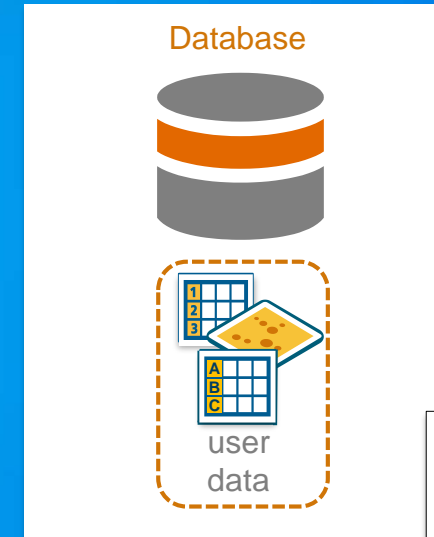
Behaviors, Complex Features, Versioning and Distributed Data

- **Database provides**

- Transaction Management
- Authorization/Security
- Data Management
- Spatial Types
- Backup

- **Enterprise Geodatabase is an Postgres Database with an Administrative Schema providing**

- Behaviors
- Complex Features (Topologies, Networks, Parcel Fabrics, etc...)
- Versioning (Long Transactions)
- Distributed Data
- Archiving



Demo



PostgreSQL Geodatabase Configuration

Michael Downey

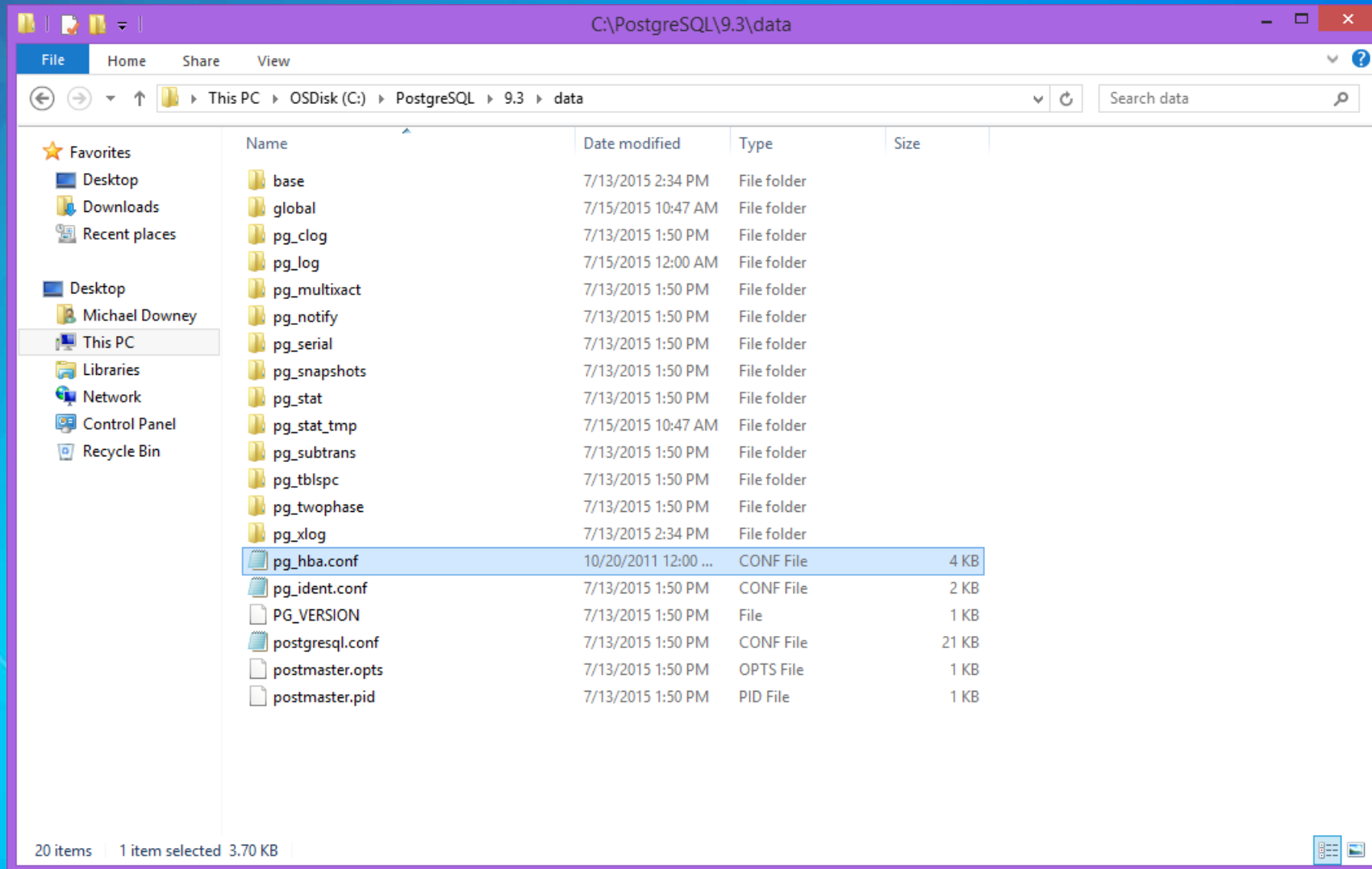
The screenshot shows the 'Geoprocessing' window with the 'Create Enterprise Geodatabase' task. The 'Parameters' tab is active, showing the following configuration:

- Database Platform: PostgreSQL
- Instance: michael81
- Database: production
- Database Administrator: postgres
- Database Administrator Password: [masked]
- Geodatabase Administrator Password: [masked]
- Tablespace Name: [empty]
- Authorization File: \\metro\Released\Authorization_Files\Version10.4\ArcGIS_Server\Advanced\Server_Ent_Adv.ecp

The 'Run' button is visible at the bottom right of the dialog. Below the dialog, a log window shows the execution results:

```
✓ Create Enterprise Geodatabase
Start Time: Mon Jul 13 14:34:23 2015
User has privileges required to create database objects.
Geodatabase admin user already exists.
Database created.
Validated authorization file.
User has required privileges for geodatabase setup.
XML support is enabled for the database instance.
Connected RDBMS instance is setup for Esri spatial type configuration.
Created geodatabase tables and stored procedures.
Finished creating geodatabase schema.
Succeeded at Mon Jul 13 14:34:51 2015 (Elapsed Time: 27.97 seconds)
```

PG_HBA file location



Our example is for local testing on our internal network.

- | # | TYPE | DATABASE | USER | CIDR-ADDRESS | METHOD |
|-----------------------------|-------|----------|------|--------------|--------|
| • # IPv4 local connections: | | | | | |
| • | host | all | all | 127.0.0.1/32 | trust |
| • | host | all | all | 10.0.0.0/8 | md5 |
| • # IPv6 local connections: | | | | | |
| • | #host | all | all | :::1/128 | trust |
| • | host | all | all | :::/0 | md5 |

PostgreSQL administrator connection in ArcGIS Pro

Database Connection ✕

Database Platform: PostgreSQL ▾

Instance: michael81

Authentication Type: Database authentication ▾

User Name: postgres

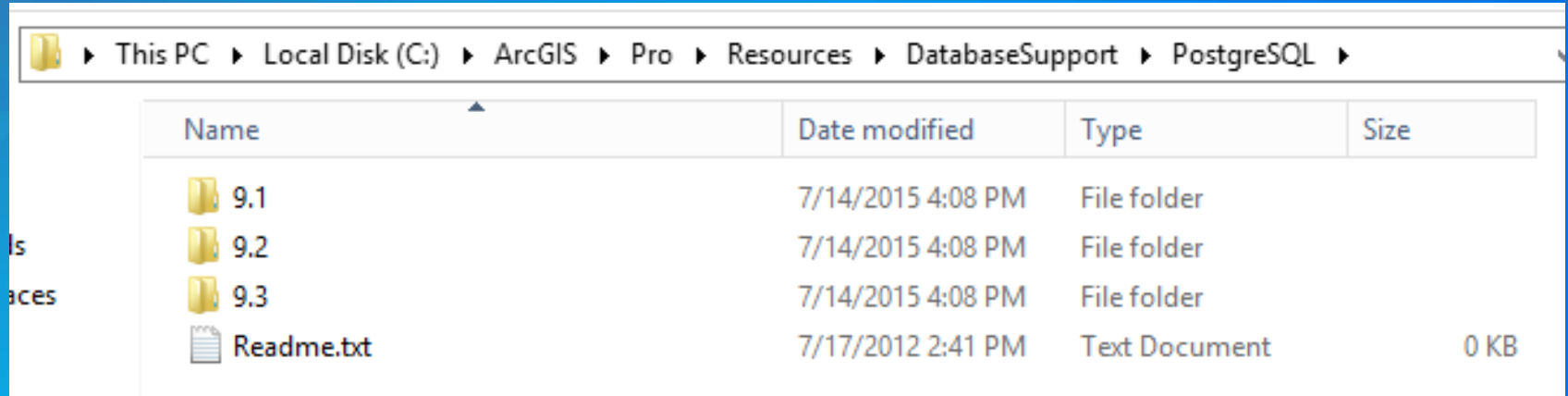
Password:

Save User/Password

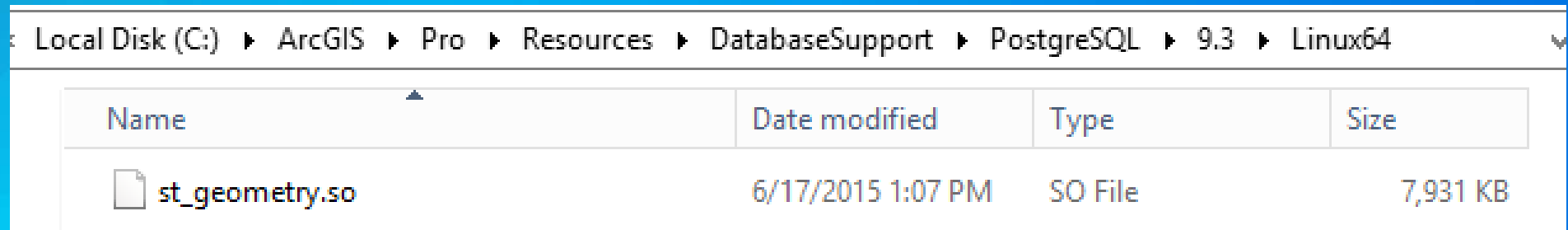
Database: production ▾

OK Cancel

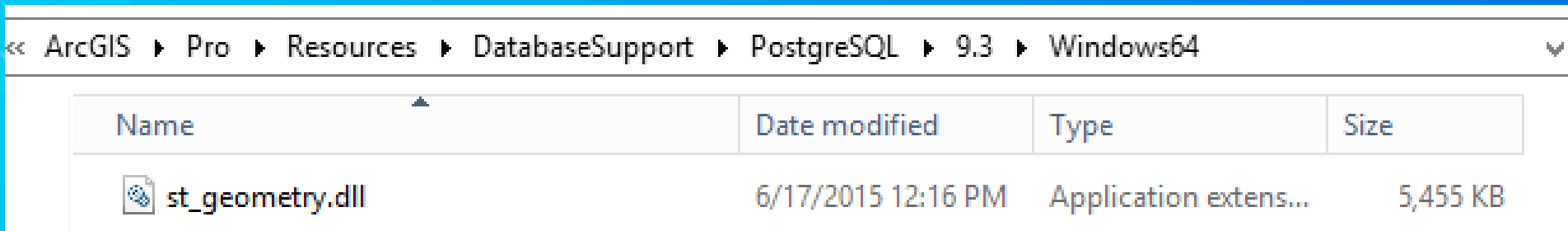
ST_Geometry source location



Name	Date modified	Type	Size
9.1	7/14/2015 4:08 PM	File folder	
9.2	7/14/2015 4:08 PM	File folder	
9.3	7/14/2015 4:08 PM	File folder	
Readme.txt	7/17/2012 2:41 PM	Text Document	0 KB

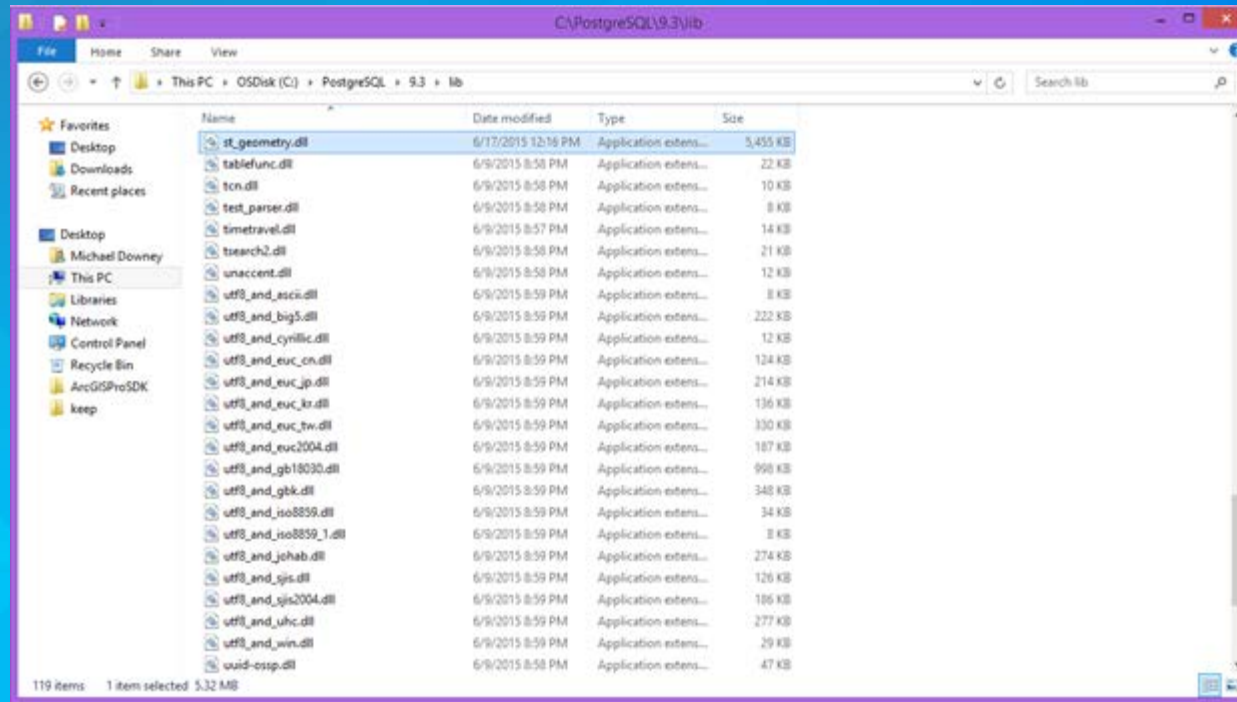


Name	Date modified	Type	Size
st_geometry.so	6/17/2015 1:07 PM	SO File	7,931 KB



Name	Date modified	Type	Size
st_geometry.dll	6/17/2015 12:16 PM	Application extens...	5,455 KB

ST_Geometry library destination folder



Create enterprise geodatabase geoprocessing tool

Geoprocessing

← Create Enterprise Geodatabase

Parameters | Environments

Database Platform
PostgreSQL

Instance
michael81

Database
production

Database Administrator
postgres

Database Administrator Password

Geodatabase Administrator Password

Tablespace Name

Authorization File
\\metro\Released\Authorization_Files\Version10.4\ArcGIS_Server\Advanced\Server_Ent_Adv.ecp

Run

✔ Create Enterprise Geodatabase

Create enterprise geodatabase geoprocessing tool results

The screenshot displays the 'Geoprocessing' window for the 'Create Enterprise Geodatabase' tool. The parameters are as follows:

- Database Platform: PostgreSQL
- Instance: michael81
- Database: production
- Database Administrator: postgres
- Database Administrator Password: [masked]
- Geodatabase Administrator Password: [masked]
- Tablespace Name: [empty]
- Authorization File: \\metro\Released\Authorization_Files\Version10.4\ArcGIS_Server\Advanced\Server_Ent_Adv.ecp

The execution log shows the following output:

```
✓ Create Enterprise Geodatabase
Start Time: Mon Jul 13 14:34:23 2015
User has privileges required to create database objects.
Geodatabase admin user already exists.
Database created.
Validated authorization file.
User has required privileges for geodatabase setup.
XML support is enabled for the database instance.
Connected RDBMS instance is setup for Esri spatial type configuration.
Created geodatabase tables and stored procedures.
Finished creating geodatabase schema.
Succeeded at Mon Jul 13 14:34:51 2015 (Elapsed Time: 27.97 seconds)
```

Create database user tool

The screenshot displays the 'Create Database User' tool within the Geoprocessing environment. The tool's parameters are as follows:

- Input Database Connection:** postgresql_michael81.sde
- Database User:** mytest1
- Database User Password:** masked with asterisks (*****)
- Role:** (empty field)

A 'Run' button is visible at the bottom right of the tool's parameter area. Below the tool, the execution log shows a successful completion:

```
✓ Create Database User
Start Time: Mon Jul 13 14:41:32 2015
Database user created.
Succeeded at Mon Jul 13 14:41:32 2015 (Elapsed Time: 0.31 seconds)
```



Points to remember

- **Upper- and mixed-case object names are not supported for Geodatabases in PostgreSQL.**
 - ArcGIS will convert them to lower case.
- **Use ArcGIS GP Tools to create and enable geodatabases**
 - Must connect as postgres superuser to create database
 - Python script example available in online help
 - SDE command lines tools deprecated at 10.3
- **More control over storage?**
 - Use PostgreSQL tools to create database first
- **Enable geodatabase tool**
 - Create a geodatabase in an existing database
- **Do not rename a database that contains a geodatabase**

Upper Case Database Identifiers

- PostgreSQL is case sensitive
 - It stores all of its object identifiers in lower case
 - Names of: Databases, Tables, Indexes, Column
 - SDE/GDB also stores all identifiers in lower case
 - User data can be in any case
 - To use identifiers in upper case, they need to be quoted
 - PgAdminIII quotes them automatically
 - ArcGIS does not look for quoted strings
 - Identifiers with upper case names not found

```
CREATE DATABASE "PGGDB"  
WITH ENCODING='UTF8'  
OWNER=postgres  
CONNECTION LIMIT=-1;
```

Upgrading a geodatabase

- All Application Server connections must be migrated to Direct Connect at 10.3.x
- Direct upgrades from 10, 10.1, 10.2, 10.2.1 or 10.2.2 to 10.3.x are supported if database at a supported release.
- Upgrading PostgreSQL
 - Must have same name and owner, set search_path variable to “\$user”, public, and sde.
- Update st_geometry (and optionally libst_raster) libraries to appropriate version.
- Permissions
- Order of operations when upgrading both gdb & sql server?
 - Recommended best practices

Client and Geodatabase PostgreSQL Interoperability

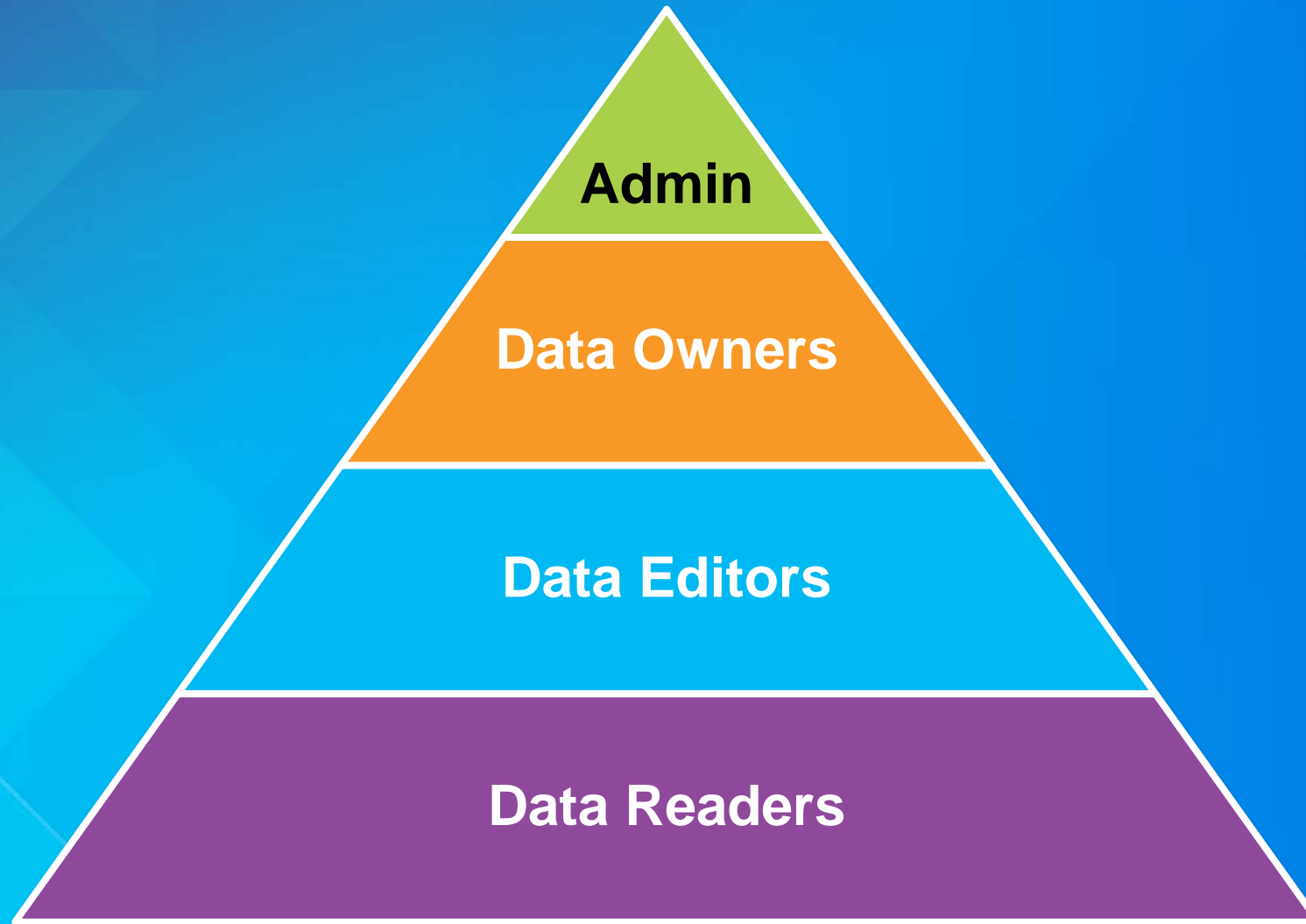
	10 geodatabase	10.1 geodatabase	10.2.x geodatabase	10.3.x geodatabase
10 client		 The DBMS must be PostgreSQL 9.0.5.	 The DBMS must be PostgreSQL 9.0.5.	
10.1 client			 The DBMS must be PostgreSQL 9.0.5 or 9.1.3.	 The DBMS must be PostgreSQL 9.1.3.
10.2.x client				
10.3.x client				



How do I... control access to my data?

Access to PostgreSQL objects are managed with permissions granted to logins (users) and roles

Limit Permissions for Most Users



PostgreSQL Principals

A role is an entity that can own database objects and have database privileges; a role can be considered a "user", a "group", or both depending on how it is used.

- **Authentication**

- Who is connecting?
- Controlled by `pg_hba.conf`, specific database user name or group

- **Login Roles/Users/Groups = Authorization**

- PostgreSQL Login Role = User
- What can this person do in the database?

- **Schemas = Containers**

- What are logical groups of database objects that should be managed as a whole?

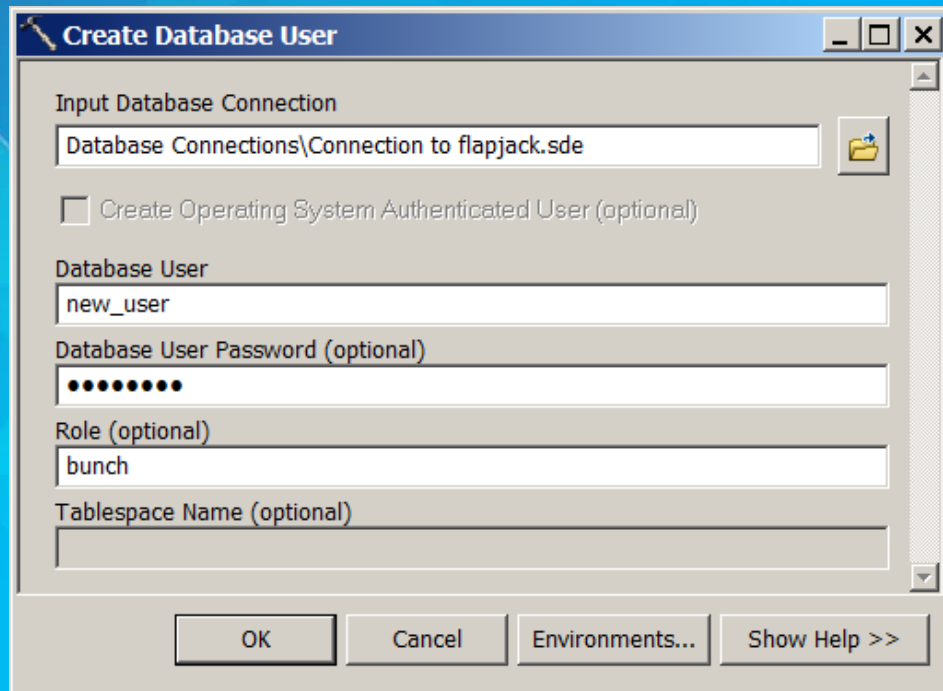
User-schema relationship

- **For users that create data, ArcGIS requires that user name = default schema name**
 - **Not a PostgreSQL rule**
- **Data readers & editors do not need a same-named schema**

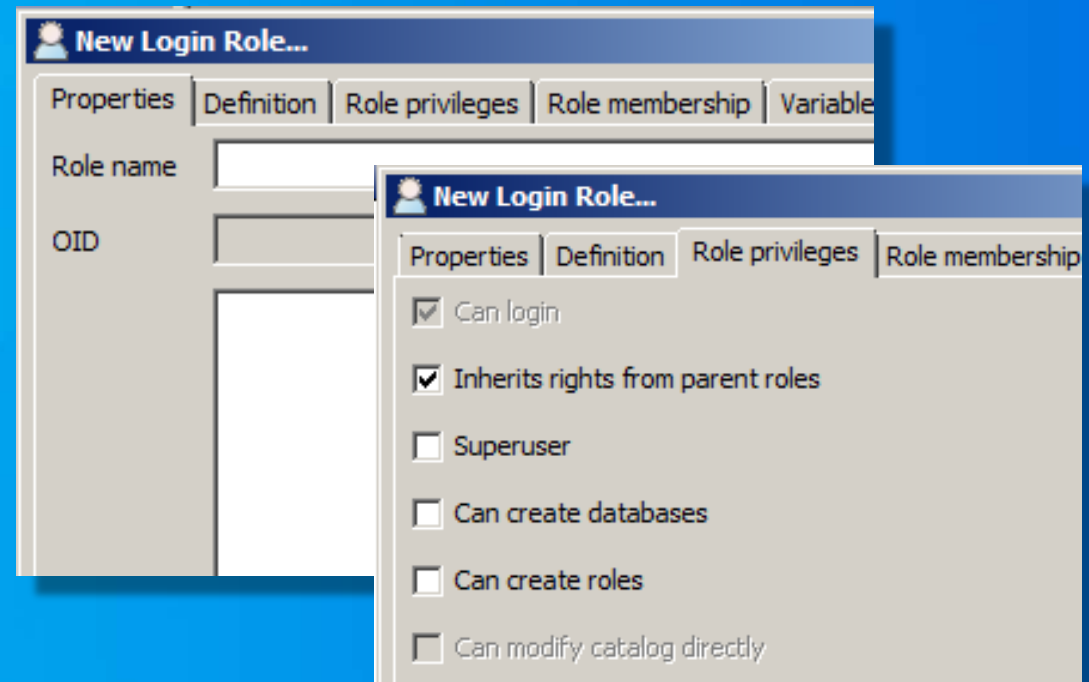
Database Users - Logins

- SDE Administrative user, Data Owners, Editors, Viewers
- login and schema must be same name for logins that will own objects in the geodatabase.
- You cannot create a schema for a group role.
- Can rerun tool to create a schema in a second database

ArcGIS Tools



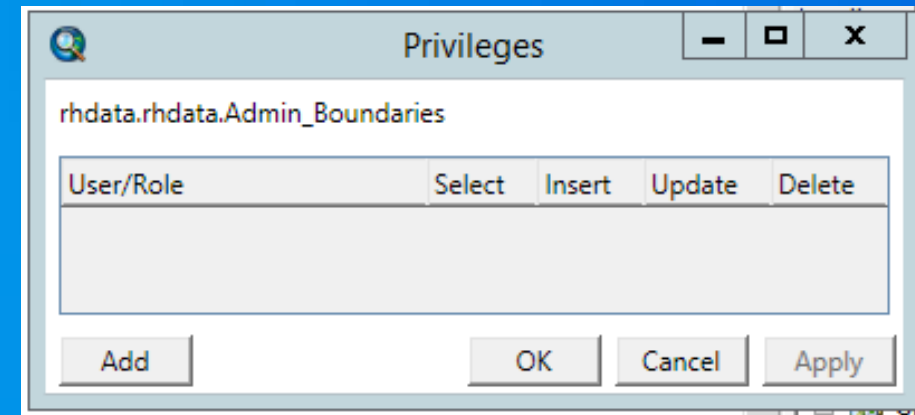
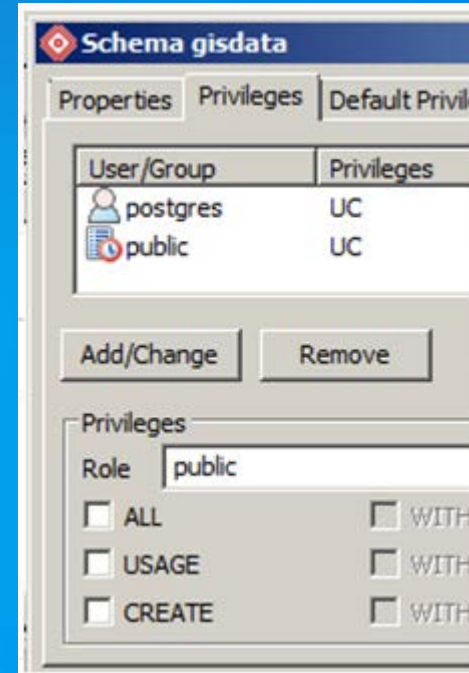
PostgreSQL Tools



PostgreSQL User Permissions

Login Roles (Users) and Group Roles (Groups)

- **Geodatabase Administrator (SDE)**
 - Authorization on SDE Schema
 - Usage on all other schemas
- **Data Owner/Creators**
 - Usage on SDE Schema, and any data schemas where access is required
 - Authorization on user's own schema
- **Data Editor**
 - Usage on SDE schema, and on data schemas to be edited
 - Grant editing privileges in Geodatabase through ArcGIS
- **Data Viewer**
 - Usage on SDE schema, and on data schemas to be viewed





**How do I...
make sure my data is safe?**

Backup

Backup – has it been tested?

Points to remember



Backups are the **only** way to reliably protect your data

1. Decide how much time you can afford to lose when disaster strikes and data must be restored
2. Create a restore plan that will achieve that goal
3. Create a backup plan that supports your restore plan
4. Implement your plan
5. Test your recovery plan regularly by using real backup media to restore to a system capable of being used in production

PostgreSQL Recovery Models

- **What are needs**
 - how often does data change?
 - how long can application(s) be down?
 - how fast does recovery need to be?
- **Weekly or nightly backups**
 - recovery to most recent backup – most common
- **Write-Ahead Logging (WAL)**
 - point-in-time recovery
 - must test thoroughly to understand resource requirements (e.g. disk i/o)

Backup – pg_dump

Geodatabases in PostgreSQL

- ▶ Set up a geodatabase
- ▶ Manage geodatabase access
- ▼ Maintain geodatabases

Geodatabase maintenance

PostgreSQL backups

Restore a geodatabase to PostgreSQL

Update database statistics

Geodatabase version administration

Rebuild system table indexes

Upgrade a geodatabase in PostgreSQL

PostgreSQL backups

There are several options for creating backups of PostgreSQL databases. However, for PostgreSQL databases used to store geodatabases, the recommended method is to use `pg_dump` to create an archive file, drop the database, re-create the database with the same name, then restore to the re-created database.

One advantage to running `pg_dump` to create a backup is it only blocks other operations that require an exclusive lock. However, any updates made to the database while `pg_dump` is running are not in the resultant file.

Archive file formats rebuild the database using `pg_restore`. Archive file formats can be used on different operating systems.

To create a backup of a geodatabase in PostgreSQL using `pg_dump`, execute the `pg_dump` command as a superuser from an MS-DOS or shell command prompt to create an archive file that can be used to re-create a database.

```
pg_dump -U <superuser_name> -F c <database_name> > <dump_file_name>
```

For example:

Backup Notes

- While there are several methods for creating backup – for PostgreSQL Geodatabases `pg_dump` to create an archive file is the recommended method.
- Recovery would be dropping database and recreating with same name.
- PostGIS has further backup recommendations – see PostGIS's documentation

Restore – pg_restore

- You must restore the public schema and its contents first. If you do not, some of your spatial data will not restore. Second, restore the sde schema. If you do not do this, spatial indexes will not be re-created on the feature classes.

Restore a geodatabase to PostgreSQL

To restore a database from an archive file created using the `pg_dump` command, use the `pg_restore` command. Be sure to test your backup and recovery models with test databases.

When you restore a geodatabase or a database with `ST_Geometry` installed, you must restore schemas in a specific order: first the public schema, then the sde schema, then the remaining schemas.

For general recovery instructions, such as syntax options, see your PostgreSQL documentation. Also, if you have PostGIS installed and are using Geometry storage, be sure to read the PostGIS documentation about creating backups and restoring databases. This procedure could vary depending on the version of PostGIS you are using.

1 | Database names must be unique on the PostgreSQL database cluster; therefore, if you are restoring to the same

<http://desktop.arcgis.com/en/desktop/latest/manage-data/gdbs-in-postgresql/restore-geodatabase-postgresql.htm>

Moving or Copying a PostgreSQL Database

- **Purposes**

- cloning to staging or development environments
- migrating to new VM environment or hardware

- **Methods**

- via ArcGIS to new database – copy/paste, export/import, etc...
- **pg_dump/pg_restore commands**
 - Restore PUBLIC schema first, then SDE, then data owner schemas
 - run as superuser
 - run ANALYZE after to update statistics
 - re-create Tablespaces
 - Text version of a table cannot be larger than 8GB if output to TAR
 - use `-o` option if object identifiers (OIDs) in user-defined data

- **Multiple PostgreSQL Instances on same Server**

Multiple Postgres instances on same server

- Create Instance

```
E:\PostgreSQL\9.2\installer\server>initcluster.vbs postgres postgres gisdata.101  
"E:\PostgreSQL\9.2" "E:\PostgreSQL\9.2\pgdata3" 5434 DEFAULT
```

Usage: initcluster.vbs <OSUsername> <SuperUsername> <Password> <Install dir> <Data dir> <Port> <Locale>

- Create Windows Service

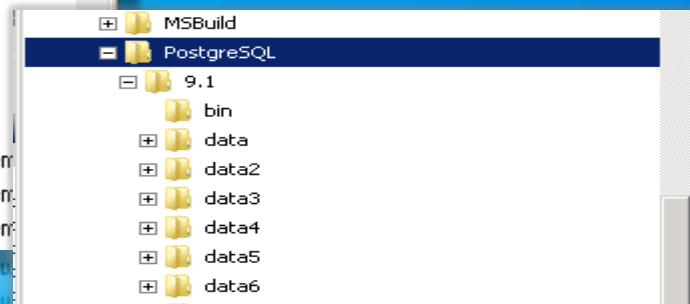
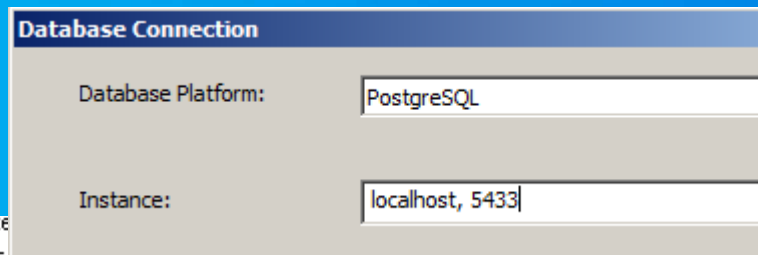
```
E:\PostgreSQL\9.2\installer\server>startupcfg.vbs 9.2 postgres gisdata.101  
"E:\PostgreSQL\9.2" "E:\PostgreSQL\9.2\pgdata3" postgresql-9.2-3
```

Usage: startupcfg.vbs <Major.Minor version> <Username> <Password> <Install dir> <Data dir> <ServerName>

- Environment Variables available

- PGDATABASE
- PGHOST
- PGPORT
- PGUSER

Service Name	Description	Status	Startup Type	Path
Portable Device En...	Enforces g...		Manual	Local System
postgresql-x64-9.1	Provides re...	Started	Automatic	.\postgres
postgresql-x64-9.1-2		Started	Automatic	.\postgres
postgresql-x64-9.1-3		Started	Automatic	.\postgres
postgresql-x64-9.1-4		Started	Automatic	.\postgres
postgresql-x64-9.1-5		Started	Automatic	.\postgres
postgresql-x64-9.1-6		Started	Automatic	.\postgres
Power	Manages p...	Started	Automatic	Local System
Print Spooler	Loads files ...	Started	Automatic	Local System
Problem Reports an...	This servic...		Manual	Local System





How do I... maintain good performance?

Monitoring: Why monitor?

- **Establish performance benchmarks to measure impacts:**
 - upgrades and patches
 - new applications or workflows
 - new server resources or deployment patterns
- **Assist in troubleshooting**
 - assist in isolating a problem when one takes place

How do I maintain good performance?

- **Standard maintenance**
 - **Reconcile/Post/Compress**
 - **Rebuild Indexes**
 - **Calculate Statistics**
- **Layer scale dependencies**

Performance Considerations

- **Excessive normalization**
 - Too many indexes
 - No optimizer hints, index use can not be forced
 - Need not worry about the Spatial Index
 - GIST index used, self correcting
- **Can change postgresql.conf initialization parameters**
- **Issue with long running ArcGIS edit sessions**
 - The larger the number of states
 - The larger the bloat in indexing belonging to the Feature Class
 - Key recommendation – save as often as possible as a best practice, combined with regular vacuum (improvements coming in next release)

PostgreSQL Geodatabase Performance

- Proper Maintenance

- Vacuum

- Have a specific plan – let it run automatically or run manually but this is what is going to keep your indexes usable if there are large amounts of editing.
 - removes dead tuples (rows)
 - Autovacuum – on by default, can do an analyze

- Statistics (Analyze)

- Statistics – table size, cardinality of joins, distribution of indexes, etc...
 - pg_stat_statements (create extension pg_stat_statements)
 - module provides a means for tracking execution statistics of all SQL statements executed by server.
 - shared_preload_libraries = "E:\\PostgreSQL\\9.2\\lib\\pg_stat_statements.dll"

- Indexes (Rebuild)

Monitoring and Performance tools

Many tools to assist in performance analysis during testing

ESRI System Monitor

PerfQA Analyzer 10.2.0.166

Help

Rendering Extents Script Log Fetch DBMS

Draw Events

Layer	Background	Geography	An
GreenValley.mxd			
buildings_point	-	0.00493	
street_arc	-	0.01752	
parks_polygon	-	0.00525	
All Layers	0.00047	-	
buildings_point	-	0.00527	
street_arc	-	0.01170	
parks_polygon	-	0.00289	
All Layers	0.00016	-	
buildings_point	-	0.00782	
street_arc	-	0.00821	
parks_polygon	-	0.00274	
All Layers	0.00016	-	

PerfQA Analyzer
ArcGIS Desktop

System Monitor:

Hosts Process ArcGIS DB Http RDP Amazon EXT

Alerting	Collecting	Type	Name	Agent
●	↑	System	ESLSRV15	ESLSRV12
●	↑	System	ESLSRV19	ESLSRV12

mxdperfstst

Item	At Scale	Layer Name	Refresh Time (sec)	Recommendations	Features	Vertices	Labeling	Geography Phase (sec)	Graphics Phase (sec)	Cursor Phase (sec)	DBMS CPU	DBMS LIO
18	8,000	Tax Lots	1.05	Simplify labeling, symbology: GraphicsPhase=.83;	2,226	33,872	True	.14	.83	.20	.08	6,396
19	8,000	Tax Lots Query Def	.13		1	26	False	.03	.02	.06	.03	3,204
20	8,000	TaxlotDenseLabel	1.84	Simplify labeling, symbology: GraphicsPhase=1.03; simplify geometry and/or set label scale; convert polygon to polyline: vertices fetched=200001; simplify geometry and/or set label scale: vertices fetched=200001;	1	200,001	True	.73	1.03	.95	.01	266
21	8,000	TaxlotDenseNoLabel	.53	simplify geometry: vertices fetched=200001;	1	200,001	False	.47	.02	.97	.00	140

Thank you...



Please fill out the session evaluation

Please note PostgreSQL Performance demo theater – Wednesday in room 6 at 3:30pm

Online – www.esri.com/ucsessionsurveys

Paper – pick up and put in drop box