

NPCA Geoserver Implementation Documentation

Contents

Prerequisites	1
Installation and Configuration	1
Troubleshooting	3
Appendix I – Explicit Projections in SQL Server 2008	4
Appendix II – SQL Server Secure User Configuration	6
Appendix III – Configuring a Datastore in Geoserver	7
Appendix IV – Creating a SQL Server Driven Layer in Geoserver	7
Appendix V – Creating a Layer Group in Geoserver	7

**Prepared by Piper McKinnon
For the NPCA
November 6, 2009
Updated July 20, 2010
Updated November 16, 2010**

NOTE: This documentation was written against the installation of Geoserver 2.0.2

Prerequisites:

This document assumes you have already installed and configured at least one instance of Tomcat and the ISAPI redirector on IIS. The instructions for these steps can be found in the document 'Installing Instances of Tomcat on IIS'.

Installation and Configuration:

- Download the latest stable Geoserver WAR from:
<http://geoserver.org/display/GEOS/Stable>
- Download the latest stable Geoserver SQL Server extension from the same location
- Download the latest JDBC driver from <http://msdn.microsoft.com/en-us/data/aa937724.aspx>
- **Install Geoserver**
 1. Go to Start → Administrative Tools → Services and stop the Tomcat service under which you plan to place the Geoserver instance
 2. Unpack the geoserver.war as with any other zip file and place the contents in [Tomcat Instance]\webapps\[Geoserver Instance Name Of Your Choosing]
 3. Go to `http://localhost:[Tomcat Instance PortNumber]/[Geoserver Instance Name]` to access the geoserver admin page (ie., if this a default installation, the url would be `http://localhost:8080/geoserver`). You have to log in; the default upon install is user admin, password geoserver.
*NOTE: to change the password, go to [Geoserver Instance name]\data\security\users.properties and change the line that says `admin=geoserver,ROLE_ADMINISTRATOR` to read `[YourNewUsername]=[yourNewPassword],ROLE_ADMINISTRATOR`
- **Install SQL Server Extension** (only required if using SQL Server as a data source)
 1. Unpack the `gt-jdbc-sqlserver-[release].jar` file and put it in geoservers WEB-INF\lib directory
 2. This file and the JDBC driver are required in order to use SQL Server as a GIS datasource in geoserver
- **Install JDBC Driver**
 1. Unpack `sqljdbc4.jar` and place it in geoserver WEB-INF\lib directory (the name of the jar may be different – at the time of this documentation,

sqljdbc.jar was outdated, so sqljdbc4 was used, there may be newer versions)

- **Configure Geoserver**

1. Create a workspace for your files – it is recommended to move the data directory out of the geoserver directory so your data is not affected when doing upgrades. Simply move the data folder and then change GEOSERVER_DATA_DIR parameter in WEB-INF/web.xml to point to the new path.
2. Configure Datastores, define layers, groups and styles using the Geoserver interface

- **Test**

1. First, use the geoserver 'Layer Preview' to check your layer configuration
2. If this works okay, you can try:
 - Adding your WMS in ArcCatalog
 - In ArcCatalog, go to GIS Servers → Add WMS Server and enter http://[your url]/[geoserver instance]/wms
 - Click 'Get Layers' – you should see your layers
 - If you get an error, there is a problem, likely in the Tomcat workers configuration or in the geoserver setup
 - Creating an html page with an open layers map or other mapping api that calls one of your (or one of the example) wms layers
 - You can easily do this by viewing one of the layers as an open layers preview, right-click, view source, and copy the contents into a text editor. Alter the path to your wms layer so it points to your url instead of the localhost. Test on an off-server browser

Troubleshooting

- **Error when attempting to retrieve get capabilities document**
 - Make sure there are no layer groups without layers and bounds. This throws an exception that causes the entire WMS to fail

Appendix I – Explicit Projections in SQL Server 2008

Setting explicit projection on SQL 2008 geometric datatype:

Creating a table with a spatial datatype is easy, you can use the SHPtoSQL tool in SQL Server 2008 Spatial Tools (<http://www.sharpgis.net/page/SQL-Server-2008-Spatial-Tools.aspx>), or any other open source tool that will do the same kind of thing. The catch is that for the data to work in GeoServer, the SRID must be set explicitly. To do this, create your spatial table as normal from the conversion software, then in SQL Server Management Studio, run an update query using the STSrid function to force the EPSG code to match that of the original shapefile you loaded. For example:

```
UPDATE tbl_UTMgeom  
SET geom.STSrid = 26917
```

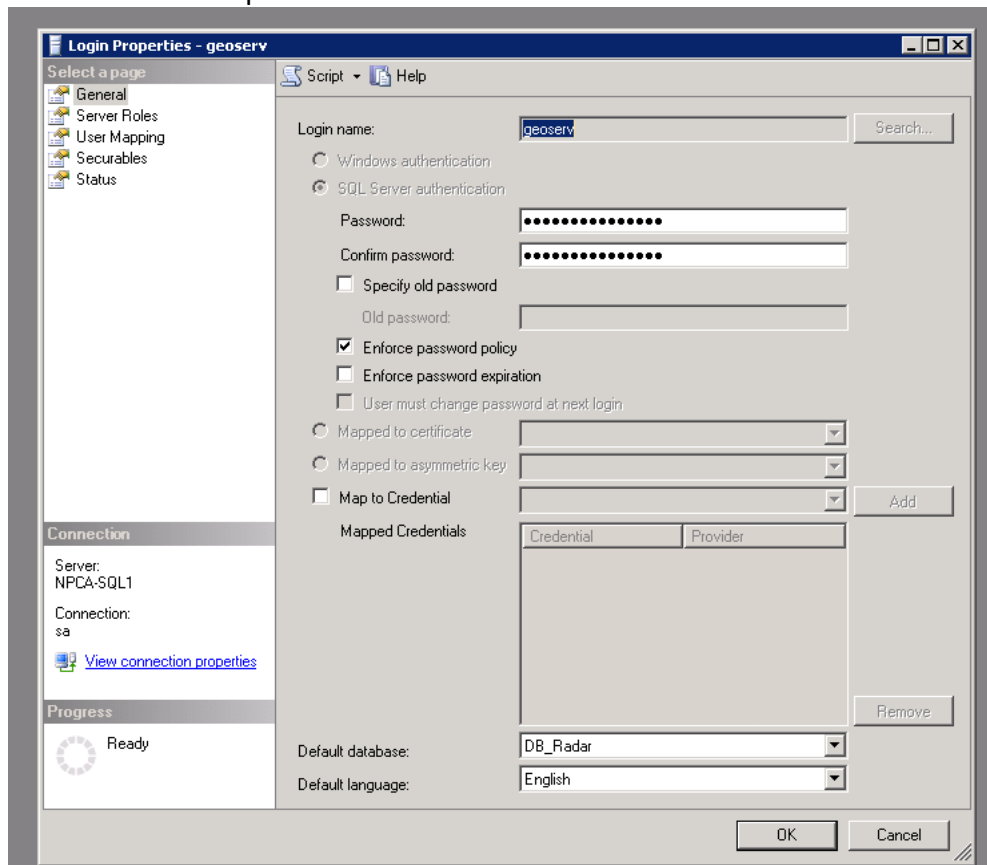
This updates the SRID setting of the geom column in the table tbl_UTMgeom to 26917 (NAD 83 UTM Zone 17N)

Appendix II – SQL Server Secure User Configuration

SQL Server database and user configuration for Geoserver:

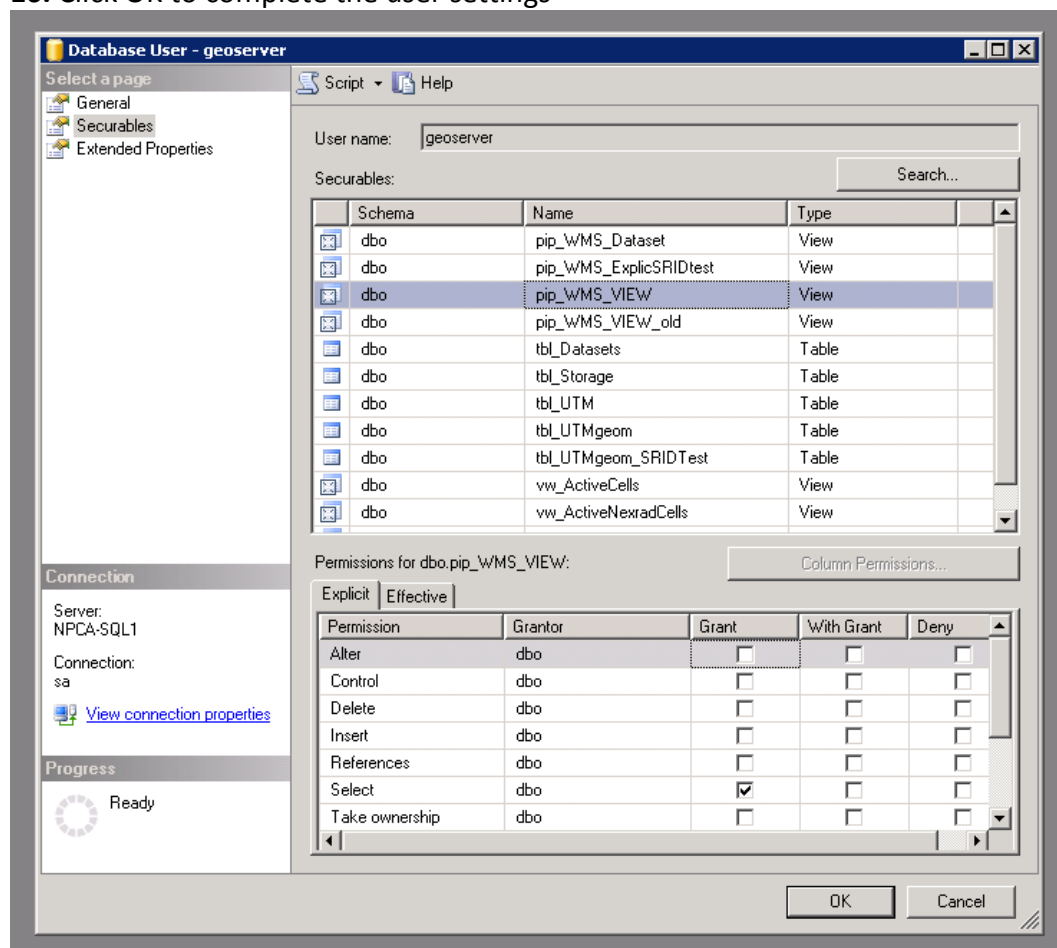
When using SQL Views as datastores in Geoserver, it's a good idea to have a SQL Server user account specific to that purpose with minimal security rights.

1. In the *SQL Management Studio Object Explorer*, go to Databases → Security, Right-click logins → New Login
2. Type a login name, choose SQL Server authentication and type a password
3. Uncheck *Enforce password expiration* and *User must change password at next login*
4. **Under *Default Database*, choose the database you wish to access through geoserver
5. Click OK to complete



6. In the *SQL Management Studio Object Explorer*, go to the database you wish to access from geoserver
7. Go to Security → Users, right click → New User
8. Type a user name of your choice (just used as a 'people friendly' reference)
9. Click the ... button next to login name and select the user that you created in the global logins

10. Just use dbo for the default schema
11. Click on *Securables* in the page menu at left and then click on the search button below user name
12. in the *Add Objects* dialog, click the *Specific objects* radio button and click OK
13. In the *Select Objects* dialog, click the *Object Types* button, and then select any of the **types** you would like to allow geoserver permission to access. It's best to only allow access to the tables/views you need as opposed to an entire database. Click OK, then click *Browse* in the select objects dialog.
14. Select each object to allow access to, click OK
15. Back in the securables page of the new database user dialog, select one of the objects in the list, then put a check in the *Grant* column next to *Select*. Do this for each table/ or view in the securables list.
16. Click OK to complete the user settings



Appendix III – Configuring a Datastore in Geoserver

Configuring a SQL Server Datastore in GeoServer:

1. Go to Start → All Programs → GeoServer → GeoServer Web Admin Page and log in
2. Go to Data → Stores → Add new Store
3. Select *Microsoft SQL Server* from the *Vector Data Stores* section
Configure the setting on the New Vector Data Source page as below:
Workspace: your namespace of choice (must configure a new namespace first if that's what you want)
Data Source Name: Name of your choice (human friendly)
enabled: check
host: name or IP of the database server (ie. NPCA-SQL1)
port: the port on which SQL Server allows connections (default is 1433)
database: Leave this blank! The current build of geoserver (2.0.2) has a bug that will cause the connection string to fail every time if the db is specified. This is why the master db setting in the user configuration of the database noted above is important in this setup.
schema: fine to leave blank
user: The user configured in the database as above
password: The password for the user above
max connections, min connections, fetch size and connection timeout can be left as default values, or customized.
5. Click *Save*

Appendix IV – Creating a SQL Server Driven Layer in Geoserver

Creating a SQL Server driven Layer:

1. Go to Data → Layers → Add a new resource
2. Find Store from which you want to create a layer in the dropdown (configured as Namespace:objectName)
3. Choose the table/view to use as the datasource
4. Configure the settings on the Data tab as below:
 - Name:** a name for the layer
 - Title:** a human friendly name for the layer
 - Abstract:** A description of the layer contents
 - Keywords:** keywords for the layer
 - Metadata Links:** Add a link to a webpage containing your metadata if available
 - Native SRS:** The EPSG (now governed by OGC) projection code for the layer (this should populate automatically if the geometry SRID is set explicitly in SQL Server)
 - Declared SRS:** Should also populate automatically,, but if not search for the SRS of your data
 - SRS handling:** Force declared SRS
 - Native Bounding Box:** Click 'Compute from data' to populate
 - Lat/Lon Bounding Box:** Compute from native bounds
4. Click *Save*
5. Go to *Layer Preview*, and find your layer in the list to preview it.

Appendix V – Creating a Layer Group in Geoserver

Creating a Layer Group:

1. Go to Data → Layer Groups
2. Click *Add new layer group*
3. Give the group a name
4. Configure the settings as below (the page is not laid out very logically):
 - Name:** a human friendly name for the group
 - Layers:** Add layers before dealing with the bounds. Click 'Add layer' and include all of the layers you want in the group
 - Bounds:** Now click 'Generate Bounds'. The bounds and the SRS code should be populated automatically
4. Click *Save*
5. WMS groups can also be previewed in the Demo → Map Preview section