FINAL

SPARC Data Quality and Interoperability Service (DQIS):

Scenario Planning Data Quality and Interoperability

June 27, 2012

Prepared for the

University of Texas at Austin

by

Thom York, *Senior Project Manager*

Contents

Document Changes from Draft to Final

Consistent use of GIS Steward / Site Manager instead of the term "user" throughout, except where the workflow being described applies to all DQIS users.

The Hardware Requirements section has been revised to accommodate additional disk space necessary for the ancillary zip file layer attachment function.

A diagram has been added on page 5 showing the relationship between SPARC, DQIS and the other components in the Sustainable Places Project.

The Ancillary Zip File Layer Attachment function has been added to the Shapefile Uploader section on page 12. It and some of its use cases-- including one relating to City Engine and 3D data exchange-- are described starting on page 8.

The function to trim incoming shapefiles to the Maximum Geographic Boundary (MGB) has been added to the Shapefile Uploader section on page 12.

Email notification for Site Managers of user uploads added to Shapefile Uploader section on page 12.

A SPARC-DQIS Enhancement Roadmap section has been added to keep track of possible future enhancements to the system.

Introduction

This document assumes familiarity with the SPARC schema specification document.

Sustainable Places Project requirements meet scenario planning challenges

The Capital Area Texas Sustainability (CATS) consortium faces a number of competing challenges in fulfilling the requirements of its Sustainable Places Project:

- It has requested scenario planning components from a number of different providers, some of which have not yet been fully defined.
- Each of these tools is unique, yet there is significant overlap in terms of their GIS inputs and outputs.
- Even for a single tool, scenario inputs vary dynamically with jurisdictional data availability, meaning that no single fixed schema can work for all possible scenarios.
- Multiple tools' results are to be consumed by a client-server application designed and maintained by the Texas Advanced Computing Center (TACC) and published on the web for easy access by stakeholders.
- All scenario planning tools are at the mercy of jurisdictional data quality.
- There are local and national data sources emerging (e.g. EPA Smart Location Database, RLIS, OpenStreetMap) that CATS wishes to leverage when they become more fully developed.
- The Sustainable Places Project is being funded through a HUD Sustainable Communities grant, which requires open access to developed tool components.
- With a new wave of tool components being created through this and other HUD grants, the CATS consortium recognizes the benefit of creating an open platform for data exchange, facilitating the generation of synergy between diverse tool ecosystems.

See the SPARC schema specification document for more details on the Sustainable Places Project requirements.

The Sustainable Places Analytical Resources Core (SPARC)

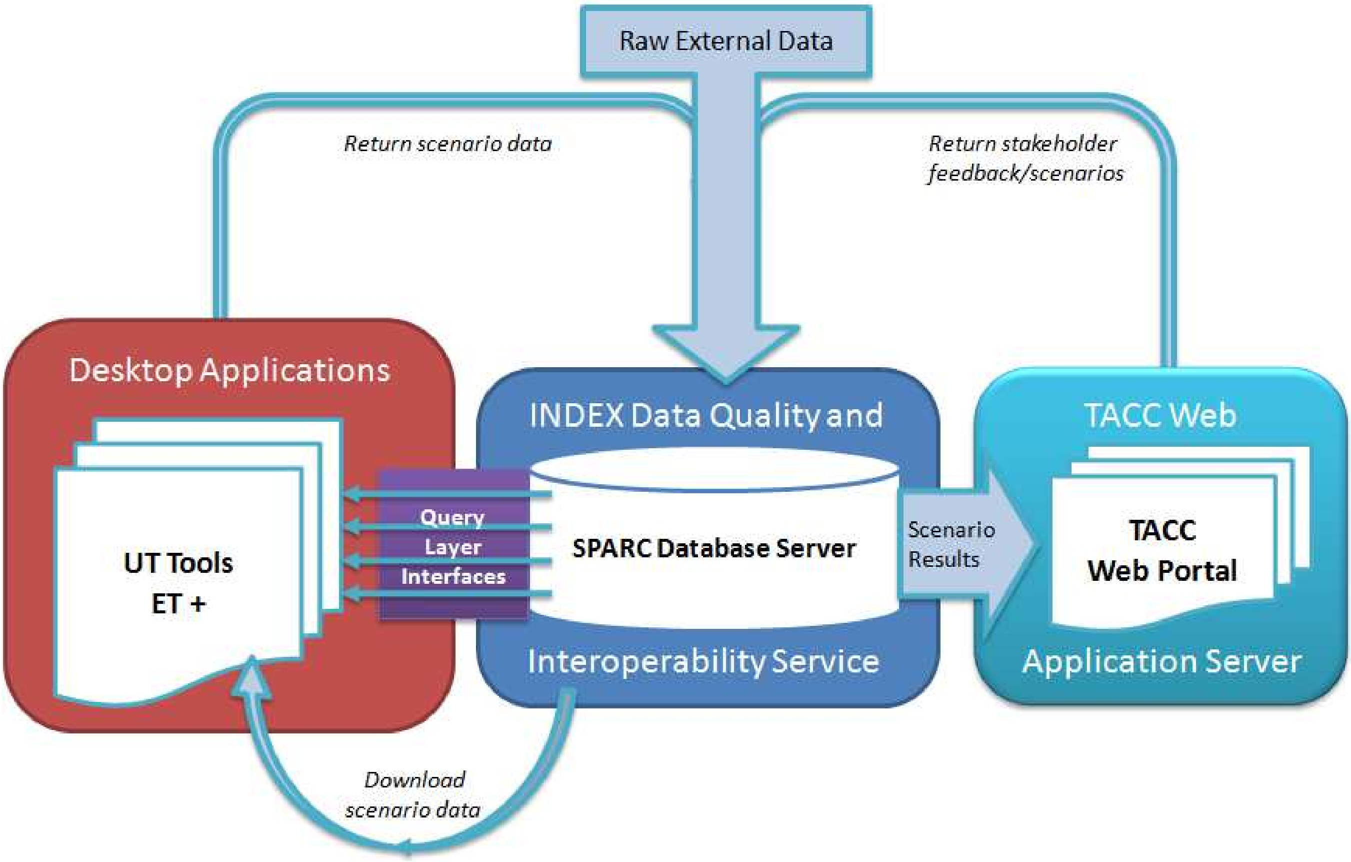
At the core of the solution to the Sustainable Places Project requirements is the SPARC data schema. Designed to be a foundational support for any enterprise client-server scenario planning application, it contains database objects that can-- with the help of the DQIS application described in this document-- support the following 3 use cases:

1. The GIS stewards at Diva Imaging upload raw external GIS data into SPARC. These data sets are cleaned and stored in a normalized, consistent form, then downloaded by Fregonese for analysis in ET+.
2. Fregonese uploads GIS data layers resulting from ET+ analysis to SPARC. CAMPO and UT staff download clean data to import into their respective models for further analysis.
3. TACC's web portal accesses clean SPARC warehouse data and publishes results in a form viewable by stakeholders using a web browser.

The SPARC schema itself is a set of scripts defining database storage objects and native-database triggers which perform the following functions:

- Transform uploaded raw GIS data into a standardized form.
- Generate and publish "query layer" view objects for each transformed standardized layer so that typical desktop GIS applications (ArcMap, QGIS) can add them to a map as they do any layer.
- Automatically clean individual geometries, consolidate duplicates and calculate/store many geometry properties required for scenario planning.

The following diagram shows these use cases as they apply to the Sustainable Places Project:



See the SPARC schema specification document for more details on the SPARC schema.

SPARC's Data Quality and Interoperability Server (DQIS)

While SPARC's trigger functions react automatically to actions such as table INSERT, UPDATE or DELETE events, the source of these events-- the GIS data itself-- comes from authenticated users supplying it via the DQIS application.

The SPARC Data Quality and Interoperability Server (DQIS) is this application, an expansion of the originally envisioned desktop Data Quality Wizard into an enterprise solution to the challenges presented by the Sustainable Place Project. Its purpose is not only to improve the quality of the GIS data stored in SPARC, but also handle the role of central clearinghouse which multiple authorized users publish GIS data requirements and exchange consistent, predictable GIS datasets.

DQIS contains the following components (all interfaces are browser-based):

- "User Authentication," where user registration, approval and login access to a SPARC server is managed. Supports 3 user groups with different permission levels. Includes email notification to support registration workflow.
- A "Messaging Engine" that emails notifications when a workflow changes from one user to another.
- "Workflow Status Tracking," where user actions on SPARC objects are logged and made available in an object history table.
- A "Tool Template Engine" for saving/publishing layer definitions and re-using them in subsequent uploads, ensuring consistent, efficient workflow.

- The "Shapefile Staging Area," where raw uploaded GIS layers are stored, analyzed and reported on.
- "Intra-Layer Quality Assurance" model ensuring clean geometries and reporting on overlaps between geometries in the same layer.
- "Project-Scenario-Layer Manager," a three-tiered system for grouping multiple layers into a scenario, and multiple scenarios into a project.
- "Scenario Nesting," relating each polygon in a given scenario's LandUseCanvas PGLT to corresponding polygons in each of the scenario's ProjectBoundary PGLTs.
- A "Shapefile Download Facility" from which both raw layer data (attributed with quality notes) and clean layers can be exported from the server to the client.
- "Online Help" documents in HTML form describing intended SPARC DQIS workflows.

In addition to developing these DQIS application components, a development implementation will be created on an existing server for load and system integration testing purposes, and support will be given to TACC to assist them in building their server and installing a SPARC/DQIS implementation on it.

Assumptions/Objectives

DQIS Hardware Specifications

Cutting edge server hardware is required due to the extreme demands of processing GIS data. The DQIS web application is designed with the assumption of being implemented on a dedicated server separate from the SPARC database server.

The following are the DQIS application server's *minimum* recommended specifications:

- 64 bit architecture
- Quad-core CPUs
- 16 GB RAM
- Two 256 GB solid-state (SSD) hard drives configured in RAID 0 (striped)

DQIS Software Specifications

The DQIS server software stack specifications are as follows:

- Linux operating system, preferably Ubuntu 11.10 or higher
- Python 2.7
- Apache 2.2.20 prefork
- Django 1.3
- GEOS 3.3.3
- PROJ 4.7
- GDAL/OGR 1.8
- psycopg2 2.4.2
- Mapnik 0.7
- mod_wsgi 3.3

The PostgreSQL/PostGIS SPARC database resides on its own dedicated server instance-- see the SPARC schema specification document for more information on its software requirements.

Web-based client interface

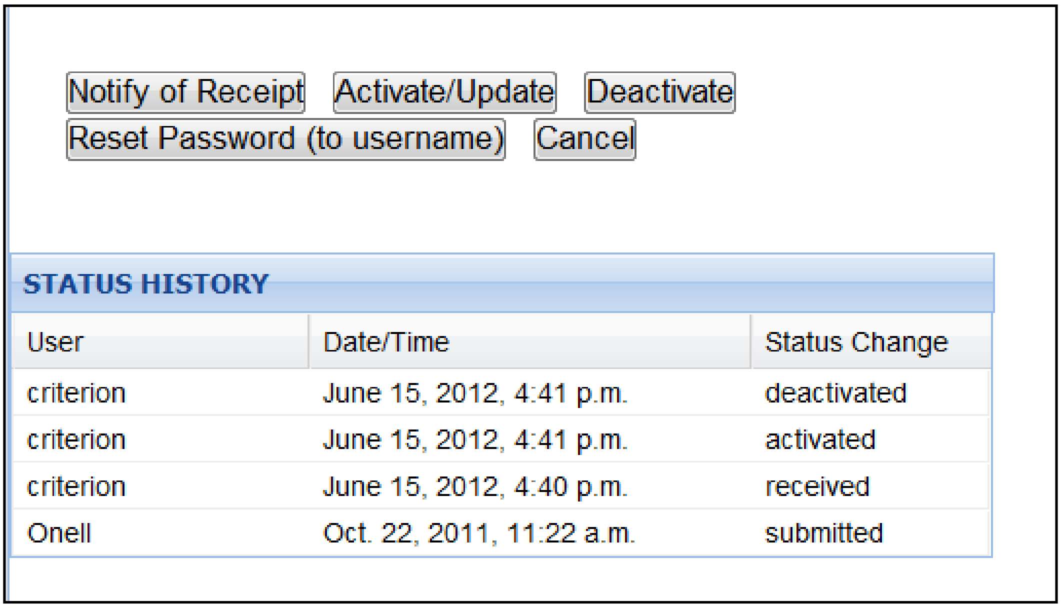
The DQIS application is web-based and accessed from any client computer with an internet connection. The following browsers are supported:

- Firefox
- Chrome
- Internet Explorer

The client-side software consists of HTML, JavaScript and ExtJS 3.2.

Workflow Status Tracking

DQIS's Workflow Status Tracking component is an audit trail mechanism for tracking user actions and internal processes occurring on other DQIS system components. These time-stamped entries are available for review on applicable components' Status History page:



STATUS HISTORY

User	Date/Time	Status Change
criterion	June 15, 2012, 4:41 p.m.	deactivated
criterion	June 15, 2012, 4:41 p.m.	activated
criterion	June 15, 2012, 4:40 p.m.	received
Onell	Oct. 22, 2011, 11:22 a.m.	submitted

In applicable component specification sections, there is a subsection listing the possible "status change" entries that can occur on the component.

Ancillary Zip File Layer Attachment

As discussed in the SPARC specification, the Ancillary Zip File Layer Attachment function exists in DQIS to allow GIS Stewards to "attach" additional data files to the shapefile being uploaded to SPARC, and make these available for download by another user.

The use cases where a GIS Steward user might want to attach a zipped set of files to a shapefile at the time of its upload include:

- Metadata files
- Raster aerial imagery files of the area covered by the uploaded layer
- LYR or another file type describing the rendering of the layer in a particular desktop GIS application
- Graphic files containing a fully realized static map of the uploaded layer created in another program
- Graphic files of graphs/charts produced through analysis of the uploaded layer
- For 3D Feature Footprint PGLT-classed layers, supporting files such as textures
- Excel or other spreadsheet data related to the uploaded layer

Generic Support for Procedural 3D Rendering Engines Like City Engine

The extent of SPARC's initial implementation is support for exchanging data that would allow desktop tools like City Engine to create 3D renderings *procedurally*. Procedural rendering involves the combination of the following 3 types of data that SPARC can store:

1. 2D vector geometries, such as building footprints or tree points or any of the "3D Feature Footprint" PGLT classes described in Appendix A.
2. Feature attributes loaded with the 2D geometries and stored in appropriate FieldDefs described in Appendix B.
3. City Engine can generate COLLADA formatted files with the .dae extension. COLLADA is a royalty-free XML schema that enables digital asset exchange within the interactive 3D industry. City Engine also makes use of texture map files. All such files can be zipped into a single file and "attached" (see previous section) to any layer that is being uploaded to SPARC.

Writing a script to transform the Collada xml-based 3D format into either of PostGIS' native TIN or Polyhedral surface 3D formats is currently out of scope. However, all tables inherited from 3D Feature Footprint PGLTs will have an optional PostGIS 3D column to support the eventual back-end implementation of such a conversion in the future.

Map Viewing Using Desktop Software

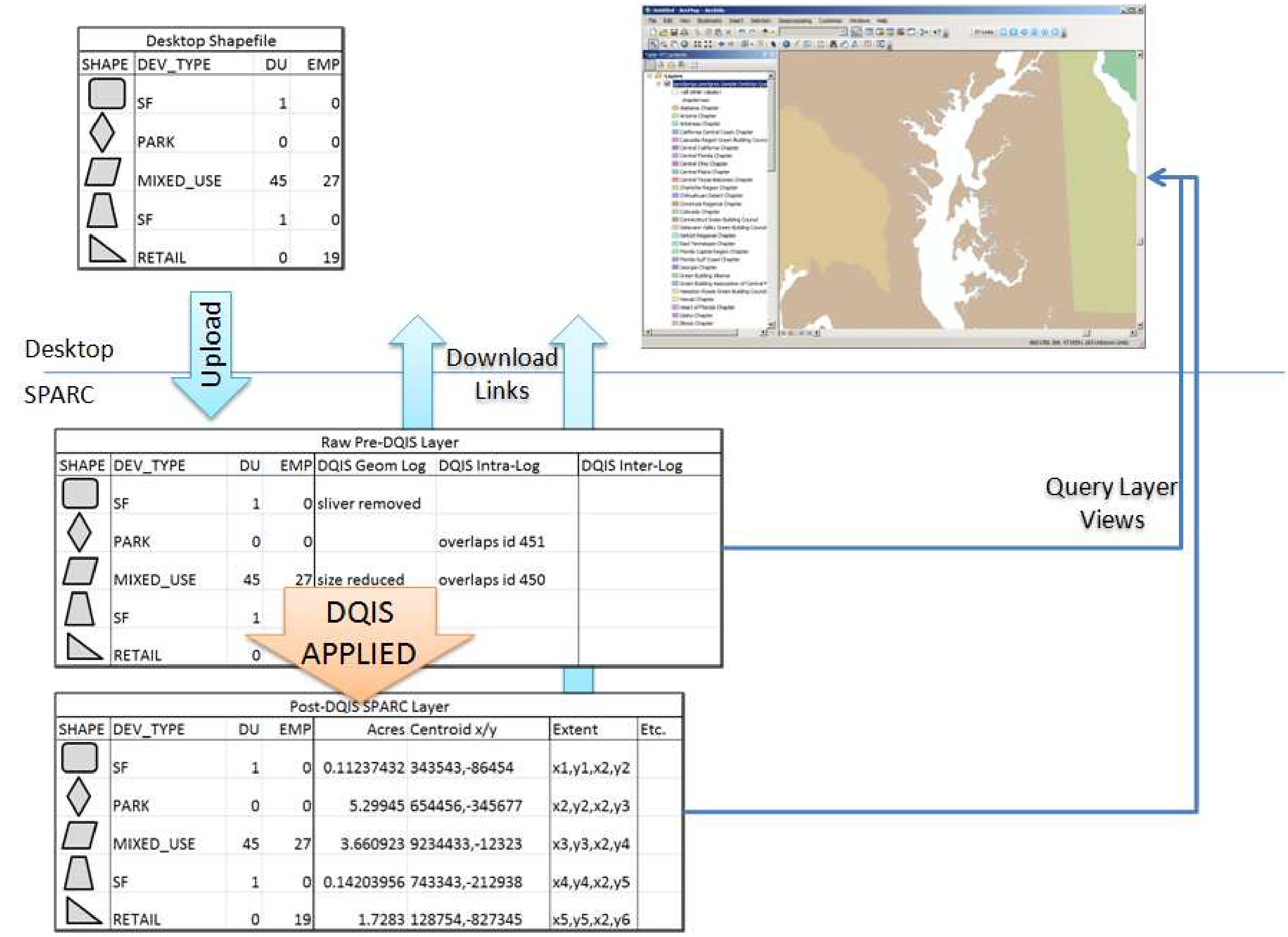
One of the significant benefits of deploying SPARC/DQIS as a client-server application is that TACC's web portal for results publishing and stakeholder engagement will be able to directly access the SPARC GIS data store, saving them from having to create a second database model to support their framework. When completed, the TACC web portal will provide a browser-based web map viewer of any desired SPARC layer.

An important parallel requirement is that SPARC and the DQIS service integrate with existing GIS Steward workflow. Therefore each layer uploaded into SPARC via DQIS is accessible through two Query Layer Views:

- A "before" view of the raw pre-DQIS layer, appended with log fields describing intra-layer quality assurance issues/actions.
- An "after" view of the new clean SPARC layer.

SPARC Query Layer Views can be added to desktop GIS applications like ArcMap (File > Add Data > Add Query Layer) and QGIS (Layer > Add PostGIS Layer) using authentication credentials supplied to GIS Stewards upon registration. These read-only layers can be rendered and presented directly as any local layer would be, or downloaded/exported to other formats (shapefile, geodatabase, etc) using tools available in these desktop applications.

The following diagram illustrates SPARC's desktop accessibility:



Desktop Shapefile

SHAPE	DEV_TYPE	DU	EMP
▢	SF	1	0
◇	PARK	0	0
▱	MIXED_USE	45	27
⏢	SF	1	0
◺	RETAIL	0	19

Raw Pre-DQIS Layer

SHAPE	DEV_TYPE	DU	EMP	DQIS Geom Log	DQIS Intra-Log	DQIS Inter-Log
▢	SF	1	0	sliver removed		
◇	PARK	0	0		overlaps id 451	
▱	MIXED_USE	45	27	size reduced	overlaps id 450	
⏢	SF	1				
◺	RETAIL	0				

Post-DQIS SPARC Layer

SHAPE	DEV_TYPE	DU	EMP	Acres	Centroid x/y	Extent	Etc.
▢	SF	1	0	0.11237432	343543,-86454	x1,y1,x2,y2	
◇	PARK	0	0	5.29945	654456,-345677	x2,y2,x2,y3	
▱	MIXED_USE	45	27	3.660923	9234433,-12323	x3,y3,x2,y4	
⏢	SF	1	0	0.14203956	743343,-212938	x4,y4,x2,y5	
◺	RETAIL	0	19	1.7283	128754,-827345	x5,y5,x2,y6	

See Appendix A for a guide to adding a layer from a cloud-based PostGIS database directly into ArcMap.

DQIS Component Specifications

User Authentication

Any enterprise application designed to be used by multiple users must have a user authentication system for defining user permissions, tracking activity and user communication. Permissions are governed by user role, of which there will initially be three:

- GIS Steward
- Site Managers
- Admins

It is assumed that all data uploaded to a particular SPARC implementation is available for viewing/downloading by all registered users. Greater privileges over the objects in SPARC-- be it a pre-DQIS raw layer, a post-DQIS SPARC layer, or a scenario/project-- belong to the GIS Steward user who creates them, as well as all Site Managers.

Workflow Interfaces

- Login page where user enters user name and password, which is validated against the database.
- Registration page where an unregistered user inputs basic registration information (name, username, password, email, organization, desired role) and submits the application to Site Managers for approval.
- Registration approval page where a Site Manager user reviews/approves registration requests.
- User Profile page where a GIS Steward user can update his or her registration information.
- Site Manager Profile Editor page where a Site Manager user can edit the registration information for any user, reset passwords and lock accounts.
- SPARC User List page that lists all registered users and their role. Sortable column headers.

Other Functions

- Email notification of pending registration request to Site Managers
- Email notification of approval to the approved GIS Steward user

Workflow Status Tracking

The following time-stamped status updates apply to user profiles:

- registration submitted
- registration approved
- registration rejected
- profile updated
- profile locked
- logged in
- uploaded shapefile
- applied DQIS transformation
- layer downloaded

Shapefile Uploader

Shapefiles are uploaded to SPARC from the GIS Steward's local file system. They are initially saved as raw pre-DQIS layers in SPARC. Each raw layer is accessible through an automatically-generated Query Layer View and can be downloaded as a shapefile to another user's local file system.

The DQIS service transforms a copy of a selected raw layer in creating a clean, standardized second layer within the SPARC sustainable geometry model (see next section). Therefore the original raw uploaded layer remains unchanged, except for the appending of several text attribute columns that are updated with DQIS process logs detailing data quality issues.

Workflow Interfaces

- Select External Shapefile page where GIS Steward targets a zipped shapefile on the local file system.
- Optionally select second zip file containing related data files on the local file system.
- Upload settings page, where user enters layer properties (e.g. name, notes).

Other Functions

- Desktop-to-server http transfer protocol implementation, for both shapefile zip as well as optional ancillary zip file layer attachment.
- Database import function to convert uploaded .shp, .dbf, .shx, etc. files into raw pre-DQIS layer inside SPARC.
- Trimming of pre-DQIS layer to SPARC implementation's Maximum Geographic Boundary (MGB).
- Optional server storage of ancillary zip file layer attachment and relating it to pre-DQIS layer.
- Creation of a Query Layer View of the raw pre-DQIS layer.
- Append DQIS log fields to raw pre-DQIS layer.
- Output layer as downloadable shapefile.
- Send notification email to initiating user if upload process exceeded specified duration and was performed asynchronously.
- Email notification sent to Site Managers notifying them of a user upload.

Workflow Status Tracking

The following time-stamped status updates apply to raw pre-DQIS layers:

- upload started
- upload completed
- DQIS transformation applied
- downloaded

DQIS Layer Transformation

Once a shapefile has been uploaded to a raw pre-DQIS layer, the user initiates the DQIS transformation process, converting the raw pre-DQIS layer into the SPARC sustainable geometry model. As it does so, the DQIS service logs the results of the transformation process to attributes previously appended to the raw pre-DQIS.

Among the tasks required to initiate a DQIS transformation, a GIS Steward selects the SPARC Primary Geometry Layer Type (PGLT) for the layer, and matches desired shapefile attributes to SPARC Field Definitions (FieldDefs). See the SPARC schema specification document for more details on this process.

In order to simplify GIS Steward workflow and promote consistency in all SPARC data sets, the particulars of a DQIS transformation can be saved as a Tool Template for later re-use on another shapefile.

Workflow Interfaces

- Tool Template List page, grouped by Tool, where a GIS Steward can optionally select a previously saved Shapefile Upload Instance.
- Transformation settings page, the completion of which can be partly or fully automated by the selected Tool Template. Included are controls/widgets to:
  - Select raw pre-DQIS layer.
  - Select destination SPARC PGLT.
  - Enter layer properties (e.g. name, notes).
  - Adjust Geometry-Level Quality Assurance thresholds/parameters (e.g. minimum polygon size)
  - Target raw data attributes for import.
  - Match each selected external data attribute to a SPARC FieldDef. Search feature minimizes likelihood of GIS Steward creating duplicate FieldDefs.
  - Enter the vintage (i.e. year) of the data in the selected raw data attribute.
- Add New FieldDef page where a missing FieldDef and its properties (data type, description, fieldname, split policy, merge policy) can be added by GIS Stewards. Site Managers have edit/delete permissions on existing FieldDefs. FieldDefs added by one user are available to all users.
- Save As page where DQIS transformation settings can be saved as a Tool Template.

Other Functions

- Application of Intra-Layer Quality Assurance (see next section).
- Creation of a Query Layer View of the clean standardized SPARC layer.
- Logging of quality assurance issues back to source raw layer.
- Transfer clean layer to SPARC Sustainable Geometry model.
- Output layer as downloadable shapefile.
- Send notification email to initiating user if transformation process exceeds specified duration and was performed asynchronously.

Workflow Status Tracking

The following time-stamped status updates apply to SPARC layers:

- DQIS transformation started
- (Intra-Layer Quality Assurance steps, see next section)
- DQIS transformation completed
- downloaded
- added to scenario (see Project-Scenario-Layer Management section)
- removed from scenario

Intra-Layer Quality Assurance

Every geometry in a raw layer undergoing DQIS transformation is subjected to a series of tests. A transformed, standardized version of the geometry is then saved into the SPARC sustainable geometry model and associated with the attributes uploaded with the feature in the selected FieldDefs . The source raw layer is retained, a relation to the clean, standardized SPARC geometry established, and raw layer log fields are updated with a description of which transformations occurred on each feature.

Workflow Interfaces

None. Server-side process initiated by DQIS Transformation component. See previous section and the SPARC schema specification document for more information.

Other Functions

Reproject all geometries

- To standard EPSG 3857 for seamless integration with Google Maps, OpenStreetMap, ArcGIS Online
- Also known as ESRI WKID 102100 Web Mercator Auxiliary Sphere

Repair bad geometries:

- Null geometry
- Incorrect ring ordering
- Self intersections
- Unclosed rings
- Empty parts: one part of multi-part is empty

Dissolve / simplify where possible

- round to closest unit (ft/m)
- eliminate redundant vertices
- makes indicator calculations faster

All features receive auto-generated scenario planning FieldDefs:

- centroid_x, centroid_y
- area_acres, area_sqft, area_sqmi
- area_ha, area_sqmt, area_sqkm
- perim_ft, perim_m
- vertex_count
- extent ranges: min_x, min_y, max_x, max_y

Polygon "duplication" detection and consolidation

- If duplicate found, it is discarded
- imported feature points to existing geometry
- Saves space, scalable
- Makes indicator calculations run faster
- Allows multiple scenarios sharing same features to compare attributes (indicators)

Eliminate unrealistically small artifact geometries

- minimum size a parameter of the Shapefile Upload Instance
- editable by GIS Steward

Polygon sliver elimination

- Typical check of area-to-perimeter ratio can detect some slivers
- BUT sliver could be part of a larger polygon therefore hard to detect using strictly area-to-perimeter ratio
- Therefore use "Tidal Buffering":
    - all polys are first negatively buffered
    - slivers disappear
    - then polys are positively buffered the same distance

    - explode multi-parts and re-apply process for eliminating unrealistically small geometries after each pass

Polygon overlap detection (optional)

- For specified PGLTs (e.g. LandUse Canvas, Planning Boundary), execute the following steps:
    - detect feature overlaps
    - report them in the appended log fields
    - deny conversion of the raw pre-DQIS layer to the SPARC sustainable geometry model

Workflow Status Tracking

The following time-stamped status updates apply to SPARC layers while undergoing SPARC transformation:

- reprojected
- repaired
- simplified
- fielddefs added
- slivers eliminated
- tiny polys removed
- overlaps detected
- no overlaps detected

Project-Scenario-Layer Management

The Project-Scenario-Layer Management component consists of a set of forms for editing the organizing a three-tiered system for grouping one or more layers into a scenario, and one or more scenarios into a project.

Workflow Interfaces

- Project-Scneario-Layer Explorer page that lists in hierarchical tree view format projects, a project's scenarios, and scenario's layers.
  - All items in list have edit links for Site Managers and GIS Stewards with ownership permissions.
  - Layers have associated download link.
- Layer Management page, where the user can:
  - edits layer properties
  - delete a SPARC layer's source raw pre-DQIS layer
  - delete a SPARC layer
  - transfer ownership so that another user can perform the above functions
- Scenario Management page, where the user can:
  - edits scenario properties
  - add a SPARC layer to the scenario
  - remove a SPARC layer from the scenario
  - move/copy scenario to another project
  - transfer ownership so that another user can perform the above functions
- Project Management page, where the user can:
  - edits scenario properties
  - add a SPARC layer to the scenario
  - remove a SPARC layer from the scenario
  - move/copy scenario to another project
  - transfer ownership so that another user can perform the above functions

Other Functions

- LandUse Canvas "Nesting", whereby a relationship is established between a scenario's LandUse Canvas PGLT and its Planning Boundary PGLTs:
  - When a Land Use Canvas layer is added to a scenario, DQIS dynamically creates a LandUseCanvas_Scenario table object
  - The table has a one-to-one relationship with the LandUse Canvas layer, and contains one column for every Planning Boundary layer in the scenario.
  - The LandUse Canvas geometry interior centroids (created as part of the SPARC sustainable geometry model) are intersected with each of the scenario's Planning Boundary layers, with the intersecting boundary geometry's id written to the LandUseCanvas_Scenario table.
  - A new LandUse Canvas Query Layer View is created containing both the layer's attributes and the id fields from the LandUseCanvas_Scenario table.

Workflow Status Tracking

None.

Online Help

Documents in HTML format describing intended SPARC DQIS workflows.

Workflow Interfaces

- Shapefile Uploading Help page, where the Shapefile Uploader component's workflow is described.
- DQIS Layer Transformation Help page where the DQIS Layer Transformation and Intra-Layer Quality Assurance workflows are described.
- Project-Scenario-Layer Management Help page, where the Project-Scenario-Layer Management component's workflow is described.

Other Functions

None.

Workflow Status Tracking

None.

SPARC-Related Sustainable Places Project Deliverables

SQL Development

In July 2012, following approval of the finalized MetaSchema document, PostgreSQL-compliant SQL scripts will be developed that can be executed on a server to generate table objects, query views, and triggers/functions designed to manage some of the dynamic elements of the MetaSchema.

Additionally, initial data sets for some of the table objects will be created and incorporated into the MetaSchema-generation scripts.

Test SPARC / DQIS Implementation

In August 2012, a limited test implementation of both the SPARC database and the DQIS application will be deployed on a Linux development server hosted by Criterion. User acceptance testing will be permitted on the development server.

Production SPARC Database Server Implementation

In September 2012, utilizing a dedicated Linux server instance provided by TACC, a PostgreSQL/PostGIS database stack will be built and the SPARC schema installed.

Production DQIS Application Server Implementation

In September 2012, utilizing a second Linux server instance provided by TACC, a Python/Django/Apache/WSGI/OGR software stack will be built and the DQIS application installed.

SPARC-DQIS Enhancement Roadmap

Further User Management and Permissions Refinement

SPARC's DQIS client initially implements 3 user groups: System Administrators, Site Managers and GIS Stewards. However, DQIS' extensible implementation supports the addition of new groups of users as the workflow surrounding SPARC-DQIS is refined.

Further Intra-Layer Quality Assurance

SPARC's DQIS client initially implements overlap detection. However, DQIS' extensible implementation supports expansion into other areas of intra-layer quality such as:

- Custom toplogy creation whereby near coincident geometries are snapped according to user-specified parameters, thus eliminating some overlaps automatically.
- Further rules-based overlap reconciliation, whereby the contents of another field in the DQIS layer is used to determine which polygon "wins" in an overlap situation. For example, a user may provide a land-use type classification and rank them such that if an open space parcel overlaps a mixed-use parcel, the mixed use parcel will remain unchanged and the open space parcel is clipped to make way for it.
- Automatic filling of voids or gaps between polygon geometries in a layer. This could be performed in a "gridded generic fill" approach whereby all voids are filled; then the resulting multipart polygon is exploded and gridded to some size; then finally its attributes are assigned to the FieldDef defaults (or special void defaults).
- Automatic network way connection, whereby a line layer intended to become a transportation network can have nearly coincident endpoints of two different geometries snapped to one another automatically according to user-specified parameters.
- Automatic network intersection creation, whereby a line layer intended to become a transportation network can have overlapping line geometries split and made into network intersections (again, according to user-specified parameters).

Further Inter-Layer Quality Assurance

Inter-Layer Quality Assurance involves ensuring relational integrity between geometries in different layers. SPARC's DQIS client initially implements a "nesting" function relating each polygon in a given scenario's LandUseCanvas PGLT to corresponding polygons in each of the scenario's ProjectBoundary PGLTs. However, DQIS' extensible implementation supports expansion into other areas of intra-layer quality such as:

- The Hierarchical Aggregation Model (HAM) Workflow. A hierarchical aggregation process can be accomplished only if a strict workflow is followed. The following is one solution, consistent with the SPARC schema:
  1. Create new scenario
  2. Select any layers to be organizationally linked to this scenario
     – Nothing happens with Tranportation, Policy Focus, Infrastructure and Environment layers
  3. Select LandUse Canvas (LUC) layer to act as basis for scenario
     – RULE: Only Field Definitions (FDs) contained in the single LUC layer selected for the scenario are aggregated
  4. Select existing HAM or create new one:
     – Name, descr, etc

        - RULE: one of the following HAM nesting strategies must be chosen:
            - Let aggregation occur as-is (simple centroid-based)
            - Fishnet incoming layer by next-highest layer and divide each incoming attrib according to its defined split policy
        - Starting at top level of the HAM, add layer to it by selecting from list of Planning Boundary (PB) layers
            - List of available layers ordered by PBs with highest number of FDs matching the FDs of the selected LUC layer
        - If split HAM nesting strategy, execute splits in creating new layer
            - If shape of final (post-split) nesting feature exists, re-use it
                - Otherwise, new feature added to PlanningBoundaryPy
        - Voids between current HAM layer and next-highest HAM layer replaced with fishnetted single-part polys
        - All fields except those with FD matching an FD in the scenario LUC are discarded
    5. Repeat for each desired level of HAM
        - Must start at top level and move downward to avoid cascading splits
    6. Once all HAM layers defined/configured, LUC layer applied
        - Each FD in LUC layer is created in lowest HAM layer (if it does not already exist)
        - All component features in LUC that belong to same HAM feature have their FDs aggregated according to FD “Merge Policy”
        - Existing fields from HAM layer fields which do not intersect LUC layer are retained unchanged
    7. For parent of first/lowest level HAM layer, repeat aggregation into its parent
    8. All HAM layers are guaranteed to fully nest (since voids were filled during HAM creation)
        - therefore every level up should be able to receive aggregated values from a merging of LUC features (scenario) and existing unchanged fields (FDs in existing HAM layer that matched LUC FDs)

- Street rights-of-way to land-use canvas layer reconciliation. Using user-specified parameters, the streets and land-use canvas layers in a scenario could be reconciled to ensure integrity. For example, rights-of-way polygons buffered precisely to a street centerline geometry's "right-of-way width" FieldDef could be generated automatically and then transferred to the land-use canvas layer, where the intra-layer rules-based overlap reconciliation process would take over.
- Automatically generate and store transportation network ways and origin/destination point buffers at user-specified distances, for example 1/4 or 1/2 buffers around transit stops or grocery stores, or 1/4 mile buffers along all street centerlines of a certain class such as collector or arterial.
- Automatically generate network origin/destination network access point objects.

Miscellaneous Enhancements

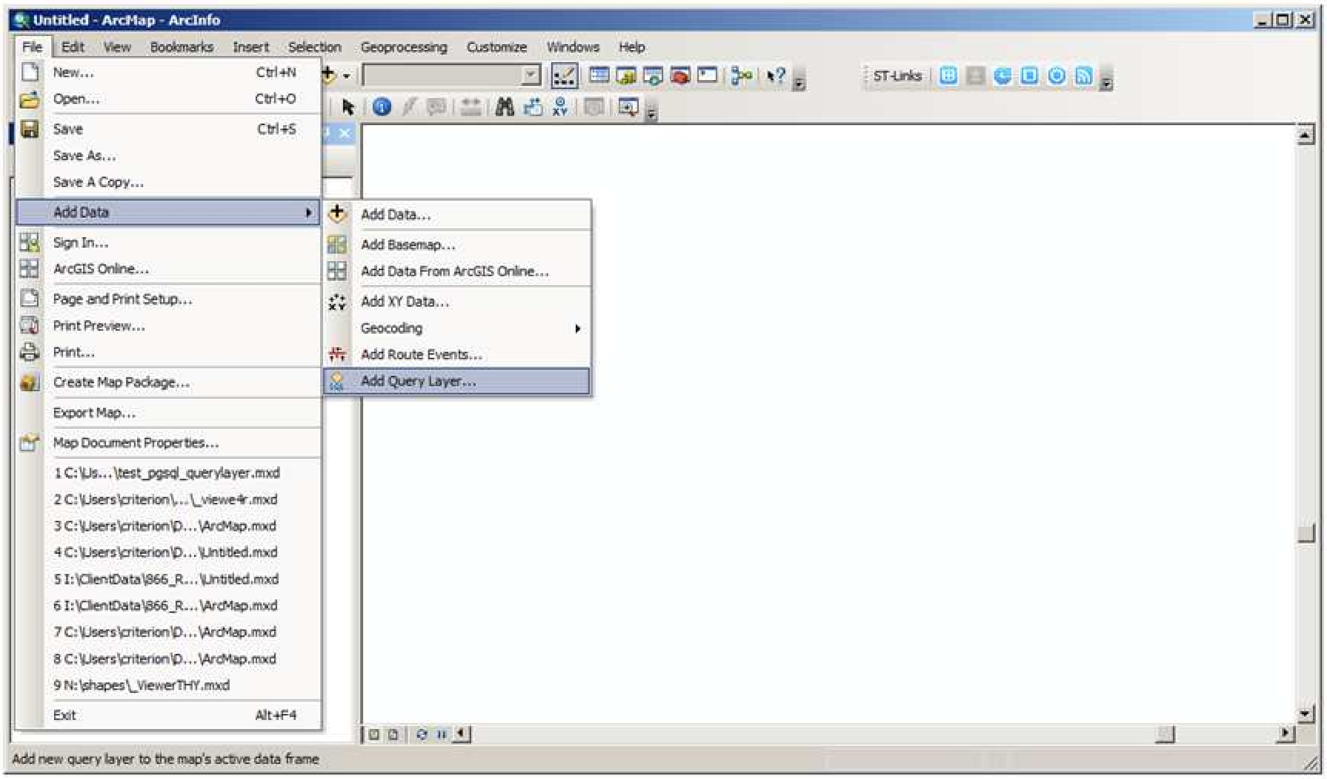
- Create an interface for the regular refreshing of Census Block Group FieldDef values from updated EPA Smart Location Database tables through use of a shared common numeric identify key such as GEOID10.

Appendix A: Add Read-only PostGIS Query Layer Views from Cloud to ArcMap

For more information on Query Layers and ArcGIS, see the following presentation from the ERSI Developer Summit 2011:

http://proceedings.esri.com/library/userconf/devsummit11/papers/demo/intro_to_query_layers_demo_theater.pdf

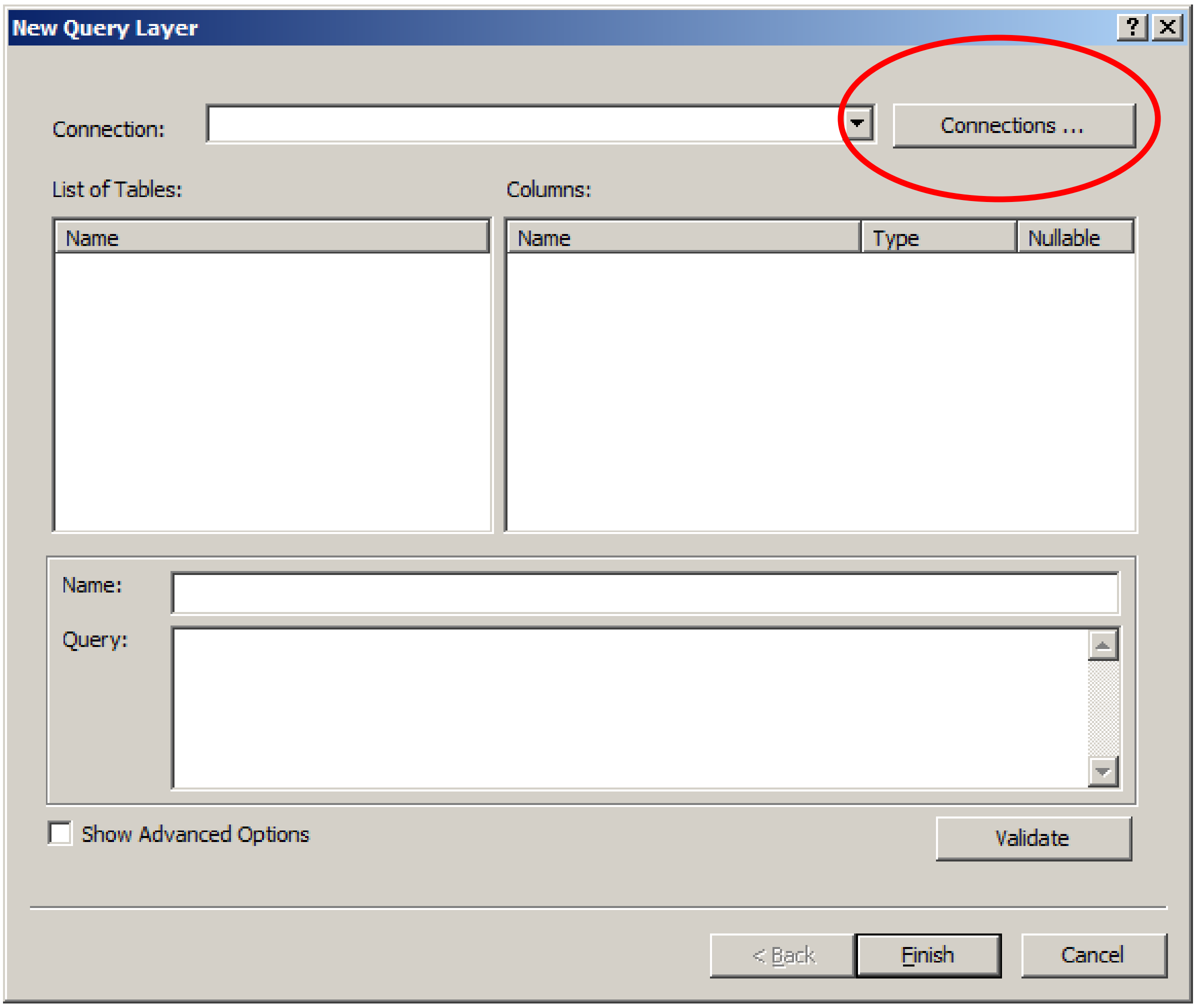
Select ArcMap's Add Data > Add Query Layer menu


Untitled - ArcMap - ArcInfo
File
Edit
View
Bookmarks
Insert
Selection
Geoprocessing
Customize
Windows
Help
ST-Links
New... Ctrl+N
Open... Ctrl+O
Save Ctrl+S
Save As...
Save A Copy...
Add Data
Sign In...
ArcGIS Online...
Page and Print Setup...
Print Preview...
Print...
Create Map Package...
Export Map...
Map Document Properties...
1 C:\Us...\test_pgsql_querylayer.mxd
2 C:\Users\criterion\..._viewe4r.mxd
3 C:\Users\criterion\D...\ArcMap.mxd
4 C:\Users\criterion\D...\Untitled.mxd
5 I:\ClientData\866_R...\Untitled.mxd
6 I:\ClientData\866_R...\ArcMap.mxd
7 C:\Users\criterion\D...\ArcMap.mxd
8 C:\Users\criterion\D...\ArcMap.mxd
9 N:\shapes_ViewerTHY.mxd
Exit Alt+F4
Add Data...
Add Basemap...
Add Data From ArcGIS Online...
Add XY Data...
Geocoding
Add Route Events...
Add Query Layer...

Add new query layer to the map's active data frame

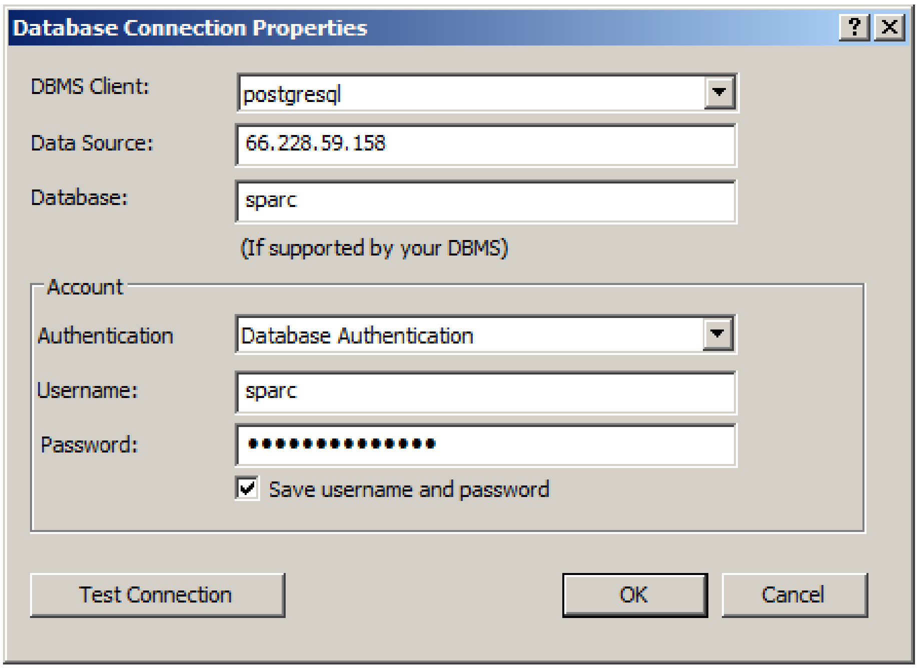
In addition to Postgresql/PostGIS data, the same process can be used to add GIS layers from SQL Server, Oracle, DB2 and Informix databases.

Choose a connection on the New Query Layer dialog



Once a connection is defined it is saved for future re-use.

Create a new connection on the Database Connection Properties dialog



Database Connection Properties

DBMS Client: postgresql

Data Source: 66.228.59.158

Database: sparc

(If supported by your DBMS)

Account

Authentication Database Authentication

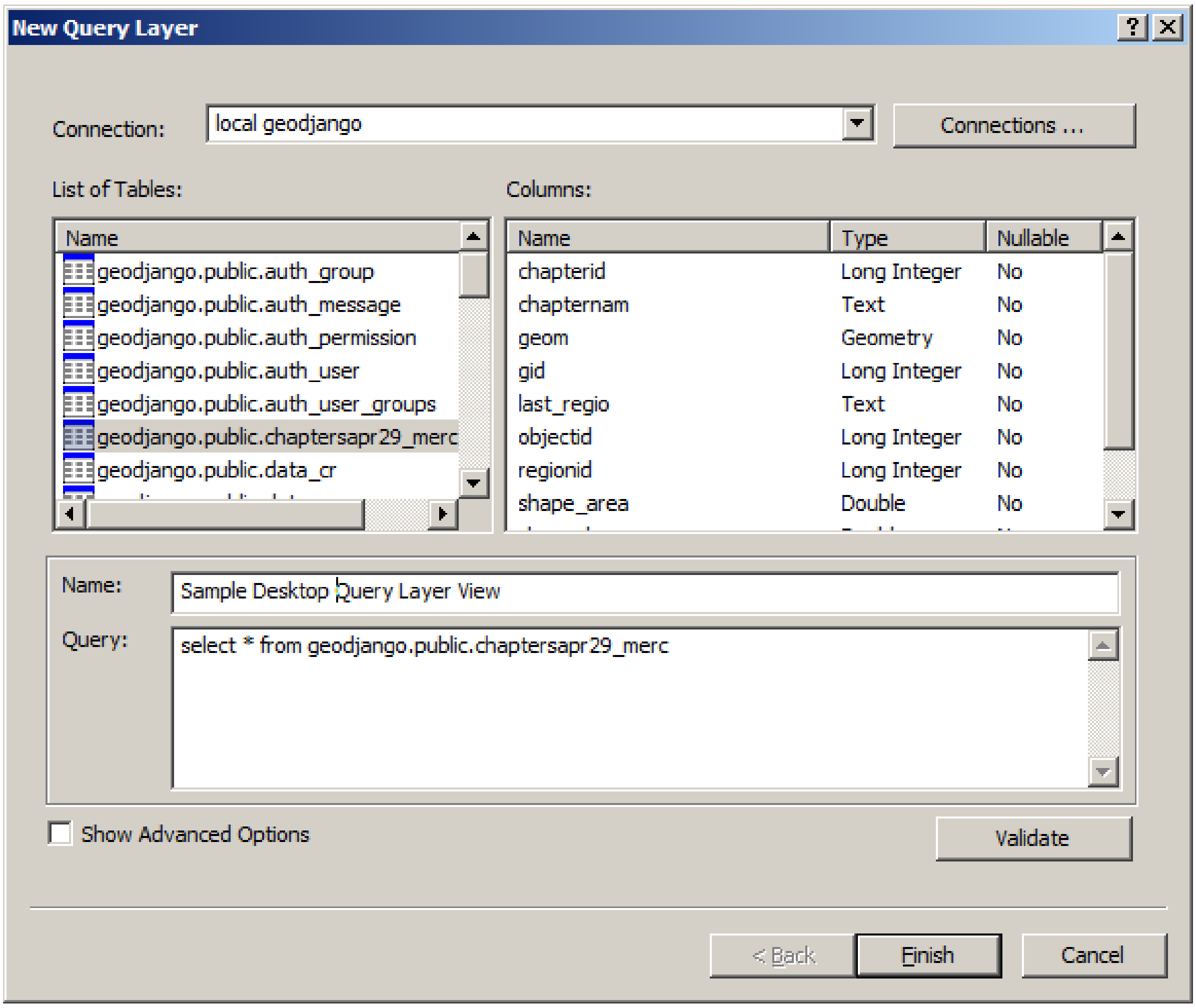
Username: sparc

Password: •••••••••••••••

☑ Save username and password

Requires database IP/name, Login/PW. These pieces of information will be sent via email to approved registered SPARC users.

Double-click the desired view, supply a name



Connection: local geodjango

List of Tables:

Name
geodjango.public.auth_group
geodjango.public.auth_message
geodjango.public.auth_permission
geodjango.public.auth_user
geodjango.public.auth_user_groups
geodjango.public.chaptersapr29_merc
geodjango.public.data_cr

Columns:

Name	Type	Nullable
chapterid	Long Integer	No
chapternam	Text	No
geom	Geometry	No
gid	Long Integer	No
last_regio	Text	No
objectid	Long Integer	No
regionid	Long Integer	No
shape_area	Double	No

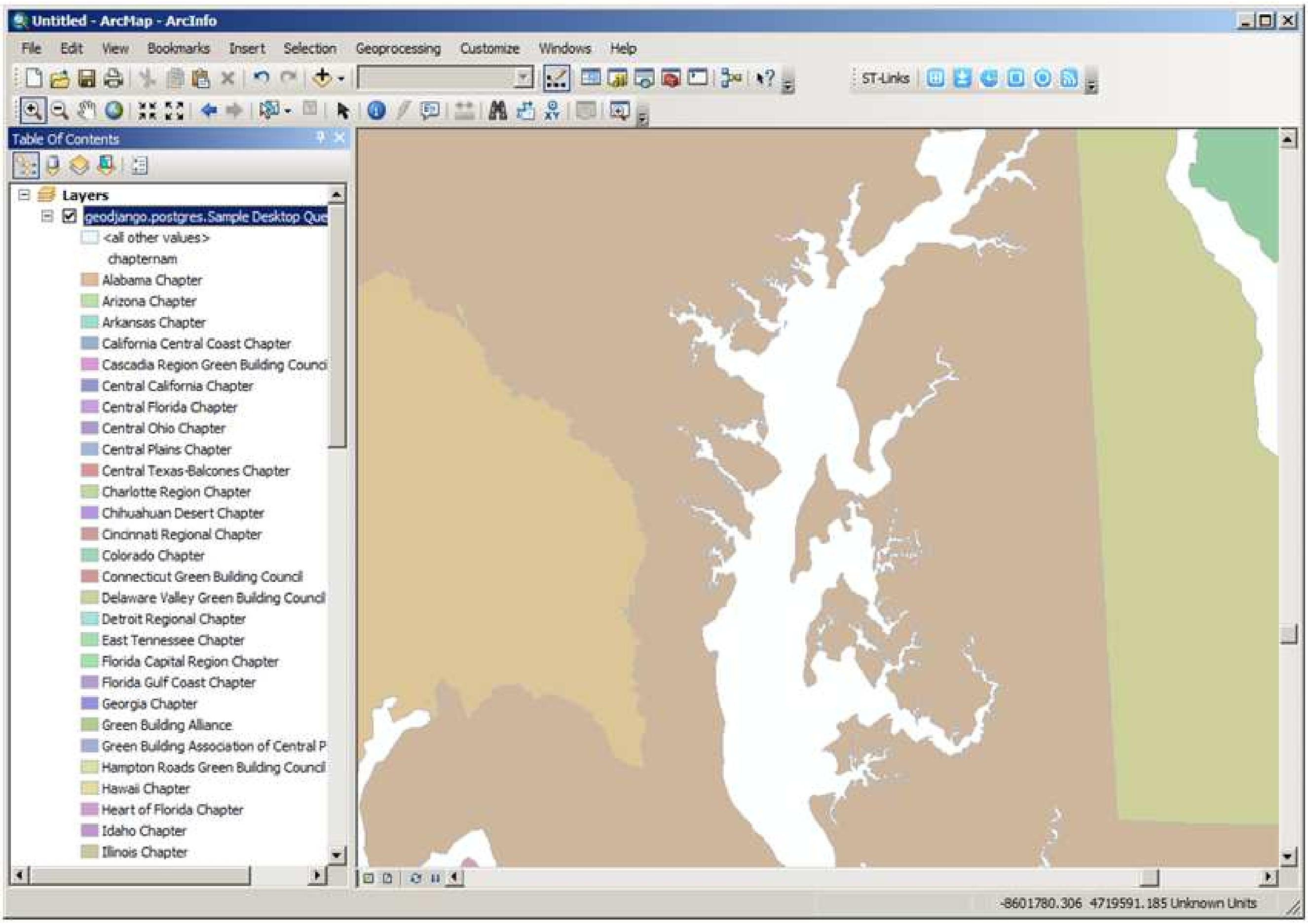
Name: Sample Desktop Query Layer View

Query: select * from geodjango.public.chaptersapr29_merc

□ Show Advanced Options

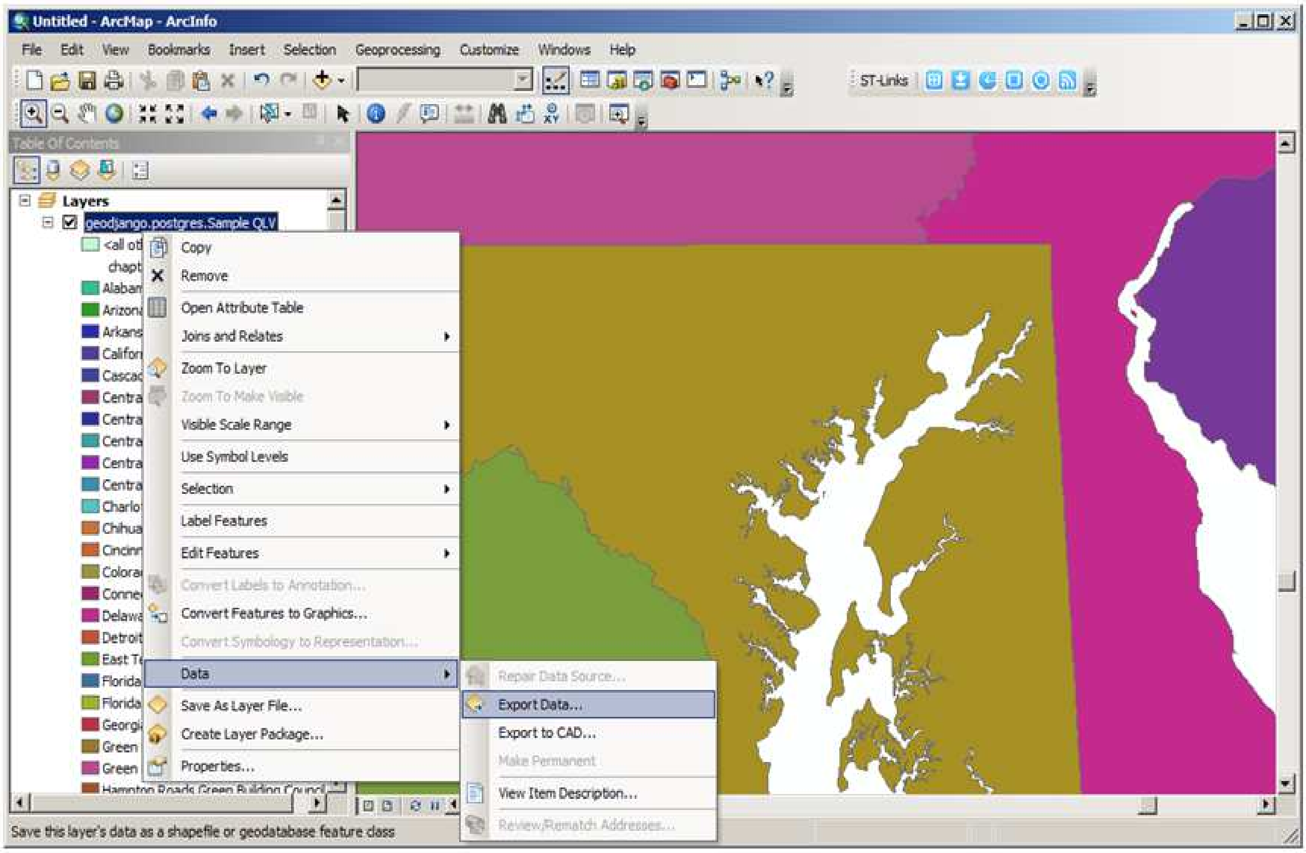
Double-clicking simply defaults the Query Layer SQL statement to "show me the entire layer with all attributes". Those familiar with SQL language can add filters and sort options to the SQL statement.

ArcMap: Layer is like any other read-only layer added from local source


Untitled - ArcMap - ArcInfo
File
Edit
View
Bookmarks
Insert
Selection
Geoprocessing
Customize
Windows
Help
ST-Links
Table Of Contents
Layers
geodjango.postgres.Sample Desktop Que
<all other values>
chapternam
Alabama Chapter
Arizona Chapter
Arkansas Chapter
California Central Coast Chapter
Cascadia Region Green Building Counci
Central California Chapter
Central Florida Chapter
Central Ohio Chapter
Central Plains Chapter
Central Texas-Balcones Chapter
Charlotte Region Chapter
Chihuahuan Desert Chapter
Cincinnati Regional Chapter
Colorado Chapter
Connecticut Green Building Council
Delaware Valley Green Building Council
Detroit Regional Chapter
East Tennessee Chapter
Florida Capital Region Chapter
Florida Gulf Coast Chapter
Georgia Chapter
Green Building Alliance
Green Building Association of Central P
Hampton Roads Green Building Council
Hawaii Chapter
Heart of Florida Chapter
Idaho Chapter
Illinois Chapter
-8601780.306 4719591.185 Unknown Units

Render the layer, combine it with other data sources in creating a multi-layer map, and print the layer just as you would any layer from a local data source.

Save local version of layer just as you would any other ArcMap layer



Save this layer's data as a shapefile or geodatabase feature class

Layer can be exported to Shapefile, File or Personal Geodatabase format.