

CITY OF SUFFOLK, VIRGINIA



UPDATED GIS DATABASE DESIGN:

Geodatabase Model

Prepared by:
Michael Baker Jr., Inc.
Virginia Beach, Virginia

Baker

TABLE OF CONTENTS

A. GEODATABASE OVERVIEW2

B. TYPES OF GEODATABASES3

C. DESIGNING THE GEODATABASE.....3

D. IMPLEMENTING THE GEODATABASE.....4

E. FUTURE CONSIDERATIONS5

F. GEODATABASE DESIGN GUIDELINES6

1. Model User’s View of Data6

2. Select Geographic Representation and Match to Geodatabase Model8

3. Organize Geodatabase Structure..... 11

4. UML Model of Geodatabase Features 13

APPENDIX 1 - MIGRATION OF DATA TO GEODATABASE..... 25

APPENDIX 2 – ARCTOOLBOX MIGRATION WIZARD 26

APPENDIX 3 – ARCCATALOG MIGRATION TOOL 27

APPENDIX 4 - REFERENCES 28

A. GEODATABASE OVERVIEW

ESRI recently released its new ArcGIS software that is based on a geodatabase data model, or “geographic database”. Successful implementation of a geodatabase with ArcGIS 8 requires the development of a good data model design. How the data is stored in the database, the applications and uses of the data, and the client and server hardware configurations are all key factors to a successful GIS database and system. Designing a geodatabase is a critical process that requires the necessary planning and revision to reach a design that meets an organization’s requirements.

The purpose of this document is to update the Data Dictionary document delivered to the city on July 5, 2000 to a geodatabase design that will support the implementation of ESRI’s geodatabase for the City. In the Data Dictionary, every feature and attribute to be captured and populated during the GIS Implementation phase is listed. This updated database design document describes the steps necessary to migrate these features and attributes from a coverage format to geodatabase format. It is important to note that the zoning, lot, and parcel data layers listed in the Data Dictionary will actually be captured in a geodatabase format and not migrated from a coverage format.

The geodatabase model is similar to the coverage model outlined in the Data Dictionary delivered to Suffolk in that it supports topologically integrated data and is essentially a generic data model for representing geographic information. Whereas coverages maintained graphic features and attributes in separate but related files, geodatabases maintain graphic features and attributes in a true relational database management system (RDBMS) such as an Oracle or SQL Server database. The eventual City geodatabase will contain the exact same features and attributes as listed in the original Data Dictionary, only using a geodatabase format.

Deleted:

Objects called Feature Classes and tables constitute the main components of geodatabases. Feature classes store geographic features represented as points, lines, or polygons, and their attributes in tables of a RDBMS. Feature classes can be organized into a collection of logically grouped feature classes (Feature Datasets) or they can exist independently in the geodatabase. RDBMS tables can also store non-geographic information representing real-world objects such as an Assessor’s tax bill having no graphic representation in the ArcGIS System. Both geographic features and non-graphic data are stored in rows of a relational database table.

Feature Datasets are, in essence, collections of feature classes with the same spatial coordinates. Feature datasets are similar to coverages in that they can contain data layers represented by different geometry (i.e., points, arcs, polygons) much like coverages can store an Arc Attribute Table(AAT) and Polygon Attribute Table(PAT) to represent a line and polygon features respectively. The real advantage and difference of the feature dataset is the ability to store multiple data layers of the same geometry type (i.e., line, polygon) in the same feature dataset. For example, Figure 6 depicts the City of Suffolk Geodatabase Landbase. In this Landbase, the Built Feature Dataset is designed to contain multiple data layers or Feature Classes with the same line geometry type (i.e., Building Lines and Structure Lines). The final geodatabase implemented for the City of Suffolk and outlined in this document will take advantage of this functionality and group feature classes together in feature datasets based on thematic similarity.

This geodatabase model includes an object-oriented model for defining the properties, behavior, and relationships of vector data and enforcing validation and integrity of data served by the system. For example, a vector data layer representing building footprints could be programmed so that no one building footprint would be able to cross another building footprint. A geodatabase also supports additional object types, including network features and annotation. Simple to very sophisticated GIS data models can be designed through a geodatabase. Simple data models represent points, lines, polygons, annotation, and associated attributes as simple feature classes, similar to ESRI® shapefiles. More

advanced data models include support for complex networks (i.e., utility), topology, advanced features such as dimensions, relationships among feature classes, and other object-oriented features.

The geodatabase can store vector and raster data (imagery), triangulated irregular networks (TINs) to represent surfaces, and geocoding data for address location. The flexibility of the geodatabase data model offers the City of Suffolk a scalable GIS platform with the ability to grow from a small GIS (ie. Limited number of users, maintainers and data access users) into a large, distributed GIS environment (i.e., unlimited users, editors, and data access). This scalability will allow future growth and access to the GIS database by additional City departments without the added cost of data migration or translation. The data model can be incorporated into legacy systems, utilize spatial and non-spatial data for analysis and to spatially enable the organization’s data, and serve it directly to the Internet and/or an Intranet.

The geodatabase model is also customizable using ArcObjects to develop organization specific tools with COM based languages like Visual Basic, and Visual C++. ESRI developed the geodatabase to incorporate the current Windows IT standards. This includes functionality like drag and drop, “wizard” driven menus, Windows type interfaces, and integration with the leading RDBMS. By adopting the Microsoft Windows ‘look and feel’, the ArcGIS 8 interface is intuitive to users of Microsoft products. The fact that the geodatabase is incorporated into the existing database environment establishes the ideal relationship to make spatial information available to legacy processes.

Deleted:

B. TYPES OF GEODATABASES

There are two broad categories of geodatabases:

- Personal geodatabases
- Multiuser (enterprise) geodatabases managed using ArcSDE™

Personal geodatabases support many readers, but only one user can edit a data layer at a time. These Personal geodatabases are stored in a Microsoft Access database. Users can create, maintain, and use personal geodatabases with ArcGIS™ without needing any additional software.

In contrast to Personal Geodatabases, Multiuser (Enterprise) geodatabases can be read and edited by multiple users at the same time. Enterprise geodatabases require the use of a Relational Database Management System (RDBMS) such as Oracle, SQL Server, Informix, or IBM DB2. Multiuser geodatabases can be read with any ArcGIS product (ArcView®, ArcInfo™, or ArcEditor™) but require the use of middleware software (ArcSDE) and either ArcEditor or ArcInfo for editing and database schema management of spatial data in a RDBMS.

C. DESIGNING THE GEODATABASE

To effectively implement a GIS using the geodatabase approach, a solid database design must be put in place. A database design must include data that will benefit the organization the most and identify what data can and will be stored. The database design process must take into account current business processes, the project goals and the ultimate plan for streamlining these existing functions. Without proper database design planning, an organization may develop a GIS database with inherent flaws such as missing data, duplicated data, or extraneous data.

Deleted:

Deleted:

From a broad perspective, a database design provides a snapshot of the current state of an organization’s data infrastructure. The database design also begins to identify the layout and state of an organization’s database in the future. The design is usually an iterative process, and should be flexible enough to accommodate modifications to the design at any time. As the design progresses, further detail can be added, including data definitions and additional data features.

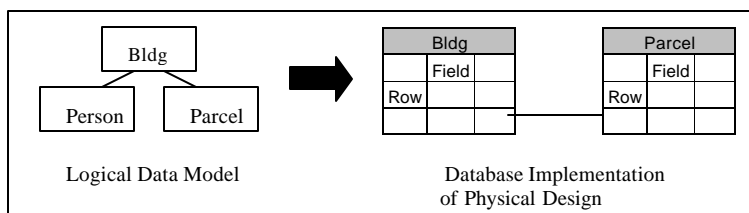
Designing a geodatabase is an essential process that requires planning and revision until a final data model is developed that meets the requirements of the City and performs well. Successfully implementing a geodatabase for the City involves a two-step process:

1. Conceptual/Logical Design
2. Physical Design

This process is outlined in *Modeling our World* by Michael Zeiler (1999), an official ESRI guide to geodatabase design (See Appendix 4 – References). The Conceptual/Logical Design involves organizing the data sources, the data to be included in the GIS into logical sets of features and determining the data types, and the attributes needed to build a simple model of these that will support the City’s functions. The Conceptual/Logical Design illustrates the database organization and structure in tables and Unified Modeling Language (UML) diagrams that define the features and geographic representation of the required ArcInfo datasets and relationships. It is also shows the user’s view of the data within a database environment.

Deleted:

The Physical Design is the implementation of the geodatabase design through one of the three processes outlined on the next page for “Creating a geodatabase”. This step basically involves the actual creation of the geodatabase tables from the abstract features defined in the Conceptual/Logical Design (See Figure below).



Ideally, the Physical Design involves the development of a geodatabase template that is a skeleton of the final geodatabase. Every feature dataset, feature class, and attributes will be defined and created in this template, so the actual data layers can be placed into the geodatabase. The pilot will be used as a mechanism to test the geodatabase design, refine it, and transition it into full-scale implementation. The pilot may be delivered in Personal Geodatabase, but as soon as the Notice to Proceed is given for full production, all deliveries will become part of an Enterprise Geodatabase once accepted. The Quality Control delivery will be submitted in Personal Geodatabase, but production deliveries will be made into an Enterprise Geodatabase Quality Control workspace. For proper design and use of the geodatabase, an accompanying Data Management Plan will be developed. It will review important issues the City must address such as the server configuration, users access, security, workflow, data location. This Plan will be included as part of the Updated Management Plan. The City will be moving forward with GIS server hardware and software purchase in the middle of the September 2001. A Baker Systems Analyst will have to spend some time with the City of Suffolk GIS Manager and the IT folks to coordinate the actual mechanics of the delivery process.

D. IMPLEMENTING THE GEODATABASE

Once the conceptual and logical geodatabase designs are developed, any of three methods below can be used to create the City’s geodatabase physical design:

- Migrating existing coverage/shapefile data into the geodatabase
- Creating a new geodatabase from scratch using ArcCatalog™
- Use Unified Modeling Language (UML) and Computer-Aided Software Engineering (CASE) tools

To create the City's geodatabase, Baker will employ a combination of the above steps. Since a coverage database design was created previously for the planimetric base mapping features, these features will be captured from the orthophotography using this coverage design. Baker staff will then migrate the Planimetric coverage data into the geodatabase (See Appendix 1 – Migration of Data to Geodatabase). There are two methods for loading coverage data into geodatabases:

- The first method uses a Graphical User Interface “wizard” within ArcToolbox to convert the coverages (See Appendix 2 – ArcToolbox Migration Wizard).
- The second method uses the Simple Data Loader “wizard” within ArcCatalog to load and create one feature class for each coverage feature captured from the orthophotography (See Appendix 3 – ArcCatalog™ Migration Wizard).

For the parcel and zoning data, this data will be captured directly in the geodatabase according to the geodatabase design outlined in this document.

In some cases, a user may not yet have any data that they want to load into a geodatabase, or the data they have only account for part of your database design. In this case, one can use the tools provided in ArcCatalog™ to create the schema for feature datasets, tables, geometric networks, and other items inside the database. ArcCatalog™ provides a complete set of tools for designing and managing items to store in the geodatabase. Through ArcCatalog™, a user can convert data stored in one of these formats to a geodatabase by importing it. A series of dialog boxes will guide the user through the conversion process. Once the user have become familiar with this process, more advanced batch data converters can be used to perform these operations more efficiently.

When converting data from one of these formats into the geodatabase, both the spatial and nonspatial component of each object will be translated. For example, when converting a coverage to a feature class, both the features (graphic) and attributes (non-graphic) are migrated into the geodatabase. The user can identify those attributes to be left out or renamed. Coverages of the same spatial extent can be imported into the same feature dataset. Some or all of a coverage based feature's geometry types (i.e., point, arc, polygon) can be imported as separate feature classes in an integrated feature dataset.

E. FUTURE CONSIDERATIONS

Once the data delivered by Baker during the pilot phase is accepted by the city as final, it will be necessary to migrate the personal geodatabase to an enterprise geodatabase. The two major components to this endeavor are:

- Loading the data
- Tuning the enterprise geodatabase

Loading the personal geodatabase into an enterprise geodatabase involves two processes. The first process is to migrate the feature classes from the personal geodatabase to an enterprise geodatabase. This is a straightforward process that doesn't require any further explanation. The second process involves appending subsequent data deliveries to the newly created enterprise geodatabase. ArcGIS 8.1 provides two methods for appending data to an existing geodatabase:

1. Object Loader in ArcMap
2. Simple Data Loader in ArcCatalog

The Object Loader method works well for appending data to simple feature classes and network feature classes in an enterprise geodatabase (See Appendix 4 – Object Data Loader in ArcMap). The one drawback of this method is the process can be very slow for appending large amounts of data to network

feature classes (i.e., Up to 8-12 seconds per feature). For appending large amounts data to network feature classes, ESRI recommends the use of the Simple Data Loader in ArcCatalog. Until the City of Suffolk develops a true utility network within the enterprise geodatabase, the Object Loader method will suffice.

A critical part of a well-performing geodatabase is the tuning of the database management system (DBMS) in which it is stored. This tuning, while not required for personal geodatabases, is critical for Enterprise Geodatabases. Baker will work with the City throughout the life of the project to assure the proper tuning techniques are applied to the ArcSDE Geodatabase.

The flexibility and control offered by the Geodatabase model should be a consideration for the development of a Citywide Enterprise GIS database, which will utilize the latest technology developed by ESRI. This architecture can be expanded and customized to suit the specific needs of the City. Through versioning control, the Geodatabase will evaluate any conflicts between versions before they are written back to the database. Versioning will be invaluable to an enterprise environment, when more than one end user may need to work on the same data set at the same time.

F. GEODATABASE DESIGN GUIDELINES

The following guidelines offer a basic process for designing an ArcInfo 8 geodatabase. These guidelines are based on information contained in *Modeling our World* by Michael Zeiler, 1999. The guidelines allow for a user to evaluate every piece of data being entered into the geodatabase and how the different data layers are associated. The output of the guidelines below form the Conceptual/Logical Design, which leads into the implementation of the Physical Design during the Pilot.

1. **Model User's View of Data**
2. **Select/Match Geographic Data with Geodatabase Model**
3. **Organize Geodatabase Structure**
4. **Unified Modeling Language (UML) Model of Geodatabase Features**

The above steps develop the conceptual model, model features based on the data needed for the organization's functions, and determine the spatial representation (i.e., point, line, polygon, image, surface) and non-spatial representation (i.e., attributes) of each feature. These steps also create the logical data model, which links the conceptual model with an ArcGIS 8 geodatabase.

1. Model User's View of Data

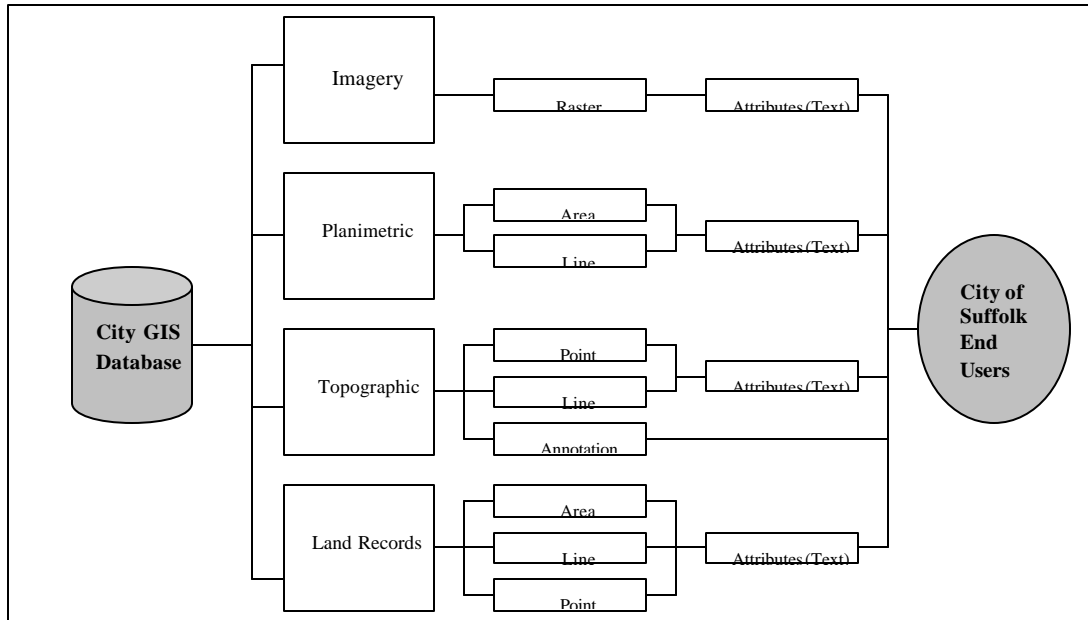
The foundation of this step is to identify data required to support the City's business functions and to organize this data into logical feature sets. In the Final Needs Assessment delivered to the City on February 10, 1999, Baker included several Workflow Diagrams. These diagrams were based on information gathered through interviews and general discussions with selected City staff and illustrate the flow of information throughout several City departments. Also identified with these diagrams are the data that would be needed to allow GIS to supplement these business functions. Based on this information and recent discussions with the City (i.e., Project kickoff meeting on June 28, 2001), the critical data layers required to support the City's business functions, especially with respect to GIS, have been identified (See Figure 1 – Data Required for GIS).

These data layers will involve and support various business functions within several City departments including Planning, Public Utilities, Public Works, and the Assessor's Office. The logical data groups and associated data layers shown below are based on the feature coverages listed in the Data Dictionary and will be used to determine the layout of data in the City's geodatabase. A general schematic of the data layers to be maintained and distributed amongst the City's GIS users is also illustrated (See Figure 2 – User's View).

FIGURE 1 - DATA REQUIRED FOR GIS

LOGICAL DATA GROUPS	DATA LAYERS
Imagery	Scanned Parcel Maps
	Digital Orthophotos
Planimetric	Buildings
	Structures
	Transportation
	Vegetation
	Centerlines
	Recreation
	Parking
	Land Use
	Driveways
	Fences
Topographic	Railroads
	Airfields
	Hydrography
	Utilities
	Swamps/Marshes
	Bridges/Overpasses
	DTM
	Topography
Land Records	Survey Photo Control Points
	Photo Center Point
	Obscured Areas
	Parcels
	Lots
	City Boundary
	Zoning

FIGURE 2 – USER’S VIEW



2. Select Geographic Representation and Match to Geodatabase Model

In the previous Figures, all the data themes critical to the business functions of the City have been identified and illustrated as the City's end user views them. From this information, the appropriate geodatabase representation for these data themes, such as feature class conventions and ArcInfo geometry type, were determined (See Figure 3 – Geodatabase Representation for Data Layers). This includes determining the data type of the elements to be included (vectors, raster, text) and placing the logical data model objects into geodatabase elements.

All non-graphic (attribute) information must be matched to the geodatabase model as well. During the planimetric mapping process, all graphic information captured as coverages will be populated with attributes as specified in the Data Dictionary. These attributes will be stored in an INFO table. When the coverages are converted into feature classes in the geodatabase, the coverage attribute item types listed in the Data Dictionary will be matched and implemented to the new geodatabase model fields as detailed in Figure 4 – INFO Items to Geodatabase Fields. The INFO table items are mapped based on a combination of their type and their width. For example, an item of type I can map to a short integer, long integer, or double, depending on its width.

For a majority of the features outlined in the Data Dictionary, an item called CONV_TYPE will be populated in the INFO Feature Attribute Table (FAT) during the photogrammetric capture process. This item field will be populated from a list of integer codes representing feature descriptions as specified in the Data Dictionary. The list of available CONV_TYPE values will be controlled separately for each graphic element to eliminate the possibility of invalid data entry. For the ArcInfo coverage format, the only way to associate these feature descriptions with the coded value in each feature is through the use of a lookup table. From further discussions between Baker and the City's GIS Manager, it has been determined that the use of lookup tables would be cumbersome and possibly confusing to the end user. To eliminate the need to rely on ArcInfo Lookup Tables (LUT), a new function in ArcGIS called Attribute Domains will be used to populate the field called CONV_TYPE based on the values listed in the Data Dictionary.

Attribute domains are used to constrain the values allowed in any particular attribute feature class. Each feature class can have a set of attribute domains that apply to particular attribute fields of the feature class. These attribute domains can be shared across feature classes and tables in a geodatabase. There are two different types of attribute domains: range domains and coded value domains. For the purposes of the City's geodatabase, only coded value domains will be used. Each coded domain has a name, a description, and a specific attribute type to which it can apply. A coded value domain can apply to any type of attribute such as text, numeric, date, and so on. Coded value domains specify a valid set of numeric or string values for an attribute. As seen in Figure 5, the coded value domain includes both the actual code value that is stored in the database (i.e., C) and a more user-friendly description of what that value actually means (i.e., Conservation District). This will allow GIS users to color code features based on the CONV_TYPE field that contains the coded value or the <Feature Class> TYPE field that contains textual descriptions of the feature and references the coded values. When a City employee is maintaining a particular feature class, they will have the option to label the feature class based on the Code or the user-friendly description. When a user edits a field with a coded value domain, a dropdown list of all of the domain values appears. By only having the choice of using values from this pick list, this assures the feature is assigned a valid value.

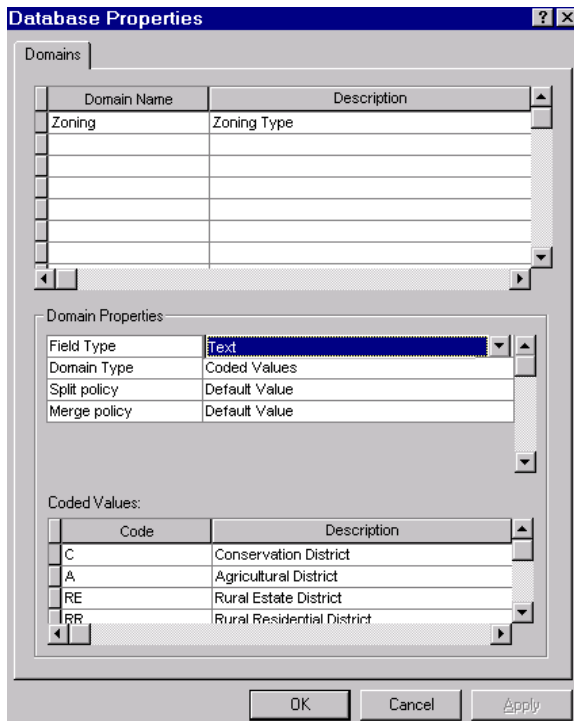
FIGURE 3 - GEODATABASE REPRESENTATION FOR DATA LAYERS

Data Layer	Spatial Type	Feature Class	ArcInfo Type	Feature Dataset	Geodatabase
Digital Images					
Orthophotos	image		raster	ORTHO IMAGES	LANDBASE
Scanned Parcel Maps	image		rater	PARCEL IMAGES	
Built Planimetrics					
Buildings	area	Building Areas	polygon feature	BUILT	LANDBASE
	line	Building Lines	line feature		
Structures	area	Structure Areas	polygon feature		
	line	Structure Lines	line feature		
Railroads	line	Railroad	line feature		
Land use	area	Landuse	polygon feature		
Fences	line	Fences	line feature		
Airfields	area	Airfields	polygon feature		
Recreation	area	Rec Areas	polygon feature		
	line	Rec Lines	line feature		
Utilities					
Utilities	line	Utility Lines	line feature	UTILITIES	LANDBASE
	point	Utility Points	point feature		
Street Planimetrics					
Transportation	area	Transportation Areas	polygon feature	STREET	LANDBASE
	line	Transportation Lines	line feature		
Centerlines	line	Centerlines	line feature		
Parking	area	Parking Areas	polygon feature		
	line	Parking Lines	line feature		
Driveways	area	Driveway Areas	polygon feature		
	line	Driveway Lines	line feature		
Bridges/Overpasses	area	Brdg_Over Areas	polygon feature		
	line	Brdg_Over Lines	line feature		
Natural Planimetrics					
Hydrography	area	Hydro Areas	polygon feature	NATURAL	LANDBASE
	line	Hydro Lines	line feature		
Swamps/Marshes	area	Swmp_Marsh	polygon feature		
Vegetation	area	Vegetation	polygon feature		
Topography					
Digital Terrain Model (DTM)	point	DTM	point feature	TOPOGRAPHY	LANDBASE
Topography	point	Spot Elevations	point feature		
	line	Contours	line feature		
	text	Elevation	annotation feature		
Survey Photo Control Points	point	Control Points	point feature		
Photo Center Point		Center Points	point feature		
Land Records					
Obscured Areas	area	Obscured	polygon feature	LAND RECORDS	LANDBASE
Parcel	area	Parcel	polygon feature		
Lots	area	Lot	polygon feature		
Zoning	area	Zoning	polygon feature		
City Boundary	area	Boundary	polygon feature		
Parcel Annotation	text	parcel_acres	annotation feature		
	text	parcel_gpin	annotation feature		
	text	parcel_length	annotation feature		
	text	parcel_taxnum	annotation feature		

FIGURE 4 – INFO ITEMS TO GEODATABASE FIELDS

INFO Item Type	Item Width	Geodatabase Field Type
B	4	Long integer
C	1-320	Text
D	8	Date
F	4	Float
F	8	Double
I	1-4	Short integer
I	5-9	Long integer
I	10-16	Double
N	1-9	Float
N	10-16	Double

FIGURE 5 – ATTRIBUTE DOMAINS



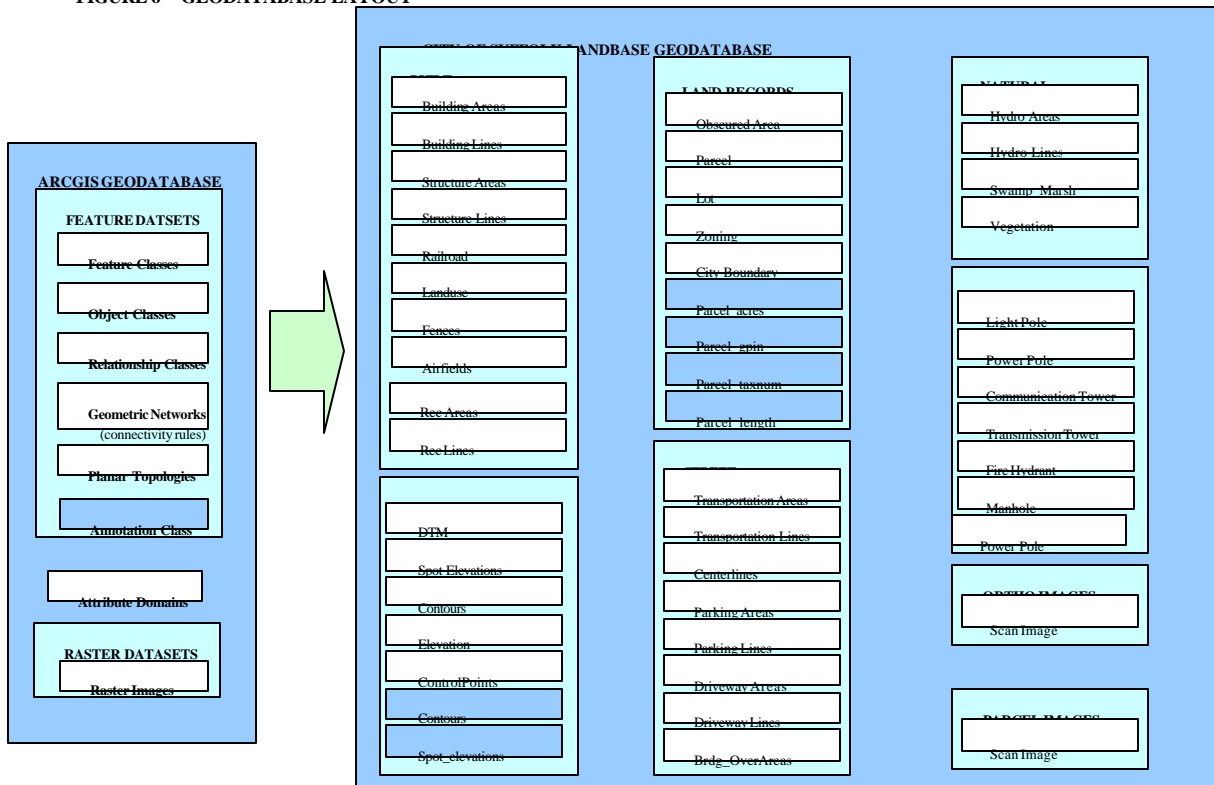
3. Organize Geodatabase Structure

One of the most critical steps to designing a geodatabase is determining the eventual layout for the geodatabase. The layout of the geodatabase really refers to the actual grouping of data layers or feature classes within feature datasets of the geodatabase. If too many feature classes are designated to be stored by one feature dataset, then the performance of the geodatabase will be degraded. ESRI has indicated that a feature dataset should have no more than roughly 12-15 feature classes.

While a feature dataset may contain an unlimited number of features without degrading performance, more than 12-15 feature classes may significantly slow the performance of the geodatabase. The main reason is because when editing a feature class, ArcGIS 8 must query every feature class in the related feature dataset. Another factor affecting the layout of feature classes in the geodatabase is figuring out how to group feature classes. In general, feature classes are grouped under like feature datasets based on thematic similarity, topological associations/shared geometry, networks, or departmental responsibility of data.

Based on the above information, the previously identified data layers were assigned to feature classes and grouped into logical feature datasets (See Figure 6 – Geodatabase Layout). Figure 4 illustrates the manner in which each feature class will reside in the final geodatabase for the City. It is important to remember this figure only shows the graphic features and not the related attribute information or any other non-graphic objects. The feature classes and their associated attributes are outlined under section **4. UML Model of Geodatabase Features**.

FIGURE 6 – GEODATABASE LAYOUT



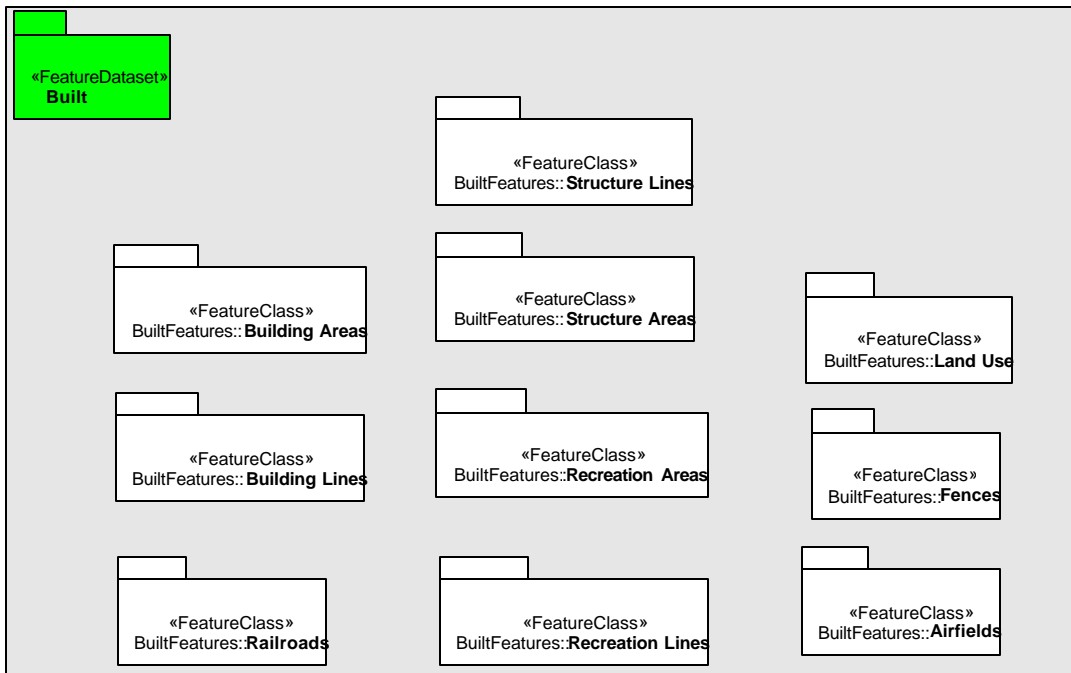
4. UML Model of Geodatabase Features

Unified Modeling Language (UML) is one of the tools ESRI has designated for use in designing and maintaining geodatabase models. In this section, UML was utilized to build the logical model of the City’s geodatabase, which includes workspace views of each feature dataset and simple diagrams of feature classes and their attributes. These UML models offer a view of conceptual framework or shell behind the organization of the City’s geodatabase. While UML was used to develop the conceptual framework of the geodatabase model, ArcCatalog will be used to implement this geodatabase model and create a physical geodatabase.

The attributes for each feature class will follow the same design as contained in the original Data Dictionary. The only difference between the original coverage model and this new geodatabase model will be that some of the required default fields will be different. All tables and feature classes have a set of required fields that are necessary to record the state of any particular object in the table or feature class of the geodatabase. These required fields are automatically created when you create a new feature class or table, and cannot be deleted. The following fields are predefined for feature classes:

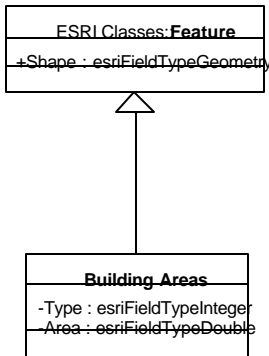
- **Object ID** - The ObjectID field is maintained by ArcGIS and guarantees a unique ID for each row in the table.
- **Shape** - Field that stores the geometry of the feature class
- **Shape_Area** - Field that tracks features’ areas for polygon feature classes
- **Shape_Length** - Field that tracks features’ perimeter for polygon feature classes and features’ line length for line feature classes

BUILT PLANIMETRIC FEATURE DATASET – WORKSPACE VIEW

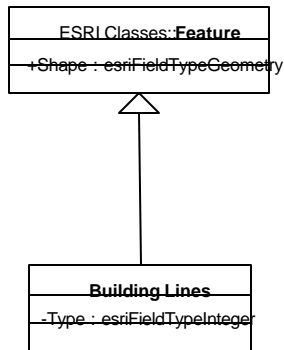


FEATURE CLASSES – BUILT DATASET

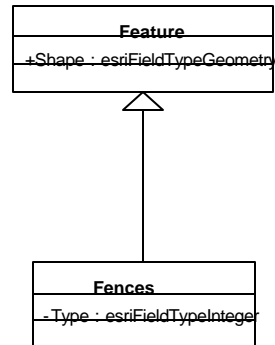
Building Footprint Areas



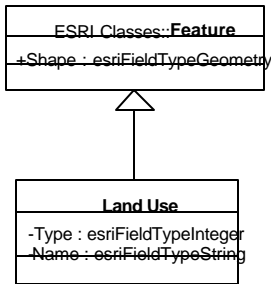
Building Footprint Lines



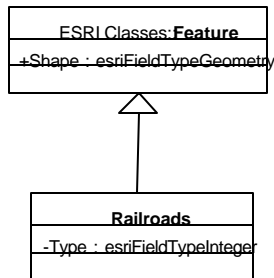
Fences



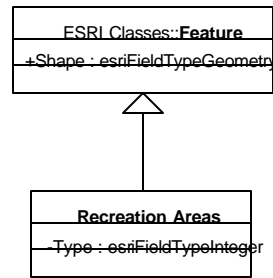
Land Use Areas



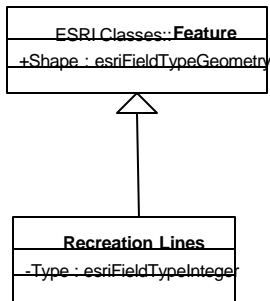
Railroads



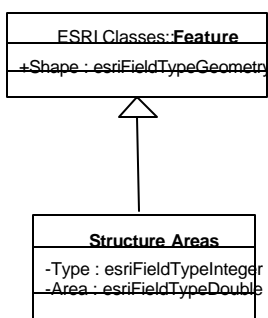
Recreation Areas



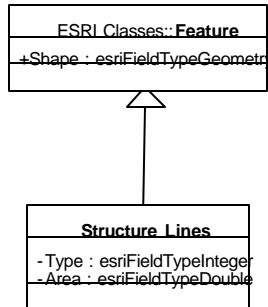
Recreation Lines



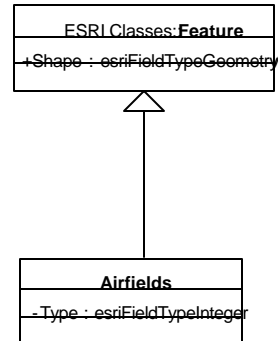
Structure Areas



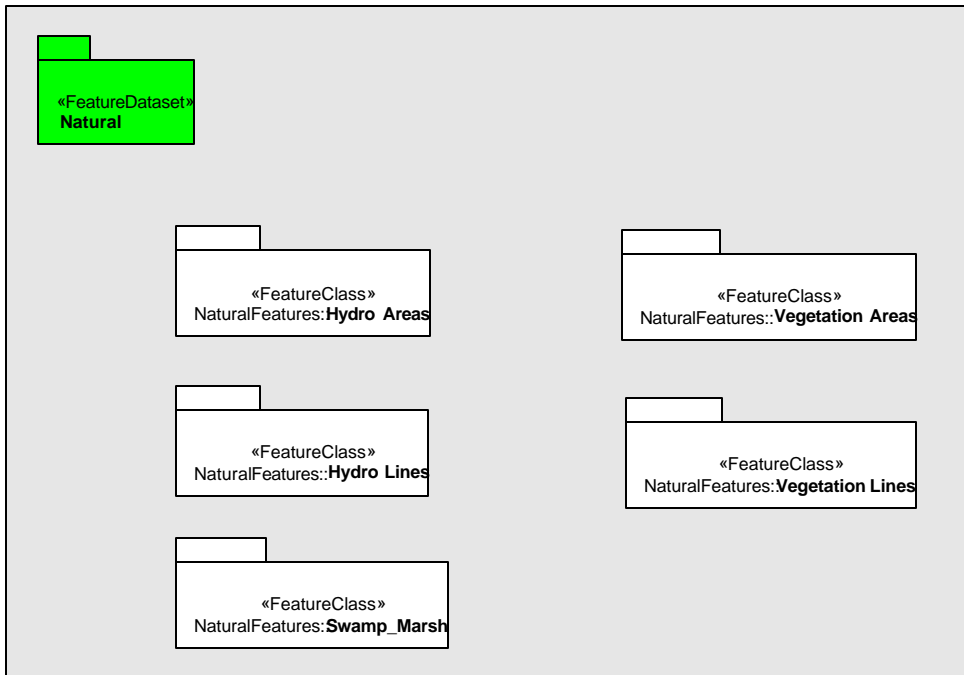
Structure Lines



Airfields

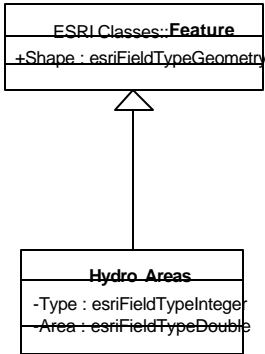


NATURAL PLANIMETRIC FEATURE DATASET – WORKSPACE VIEW

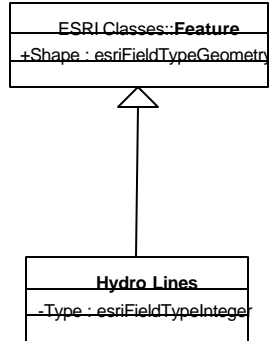


FEATURE CLASSES – NATURAL DATASET

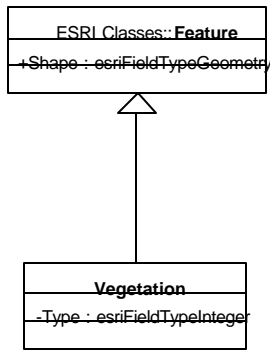
Hydrography Areas



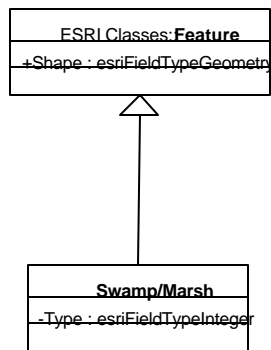
Hydrography Lines



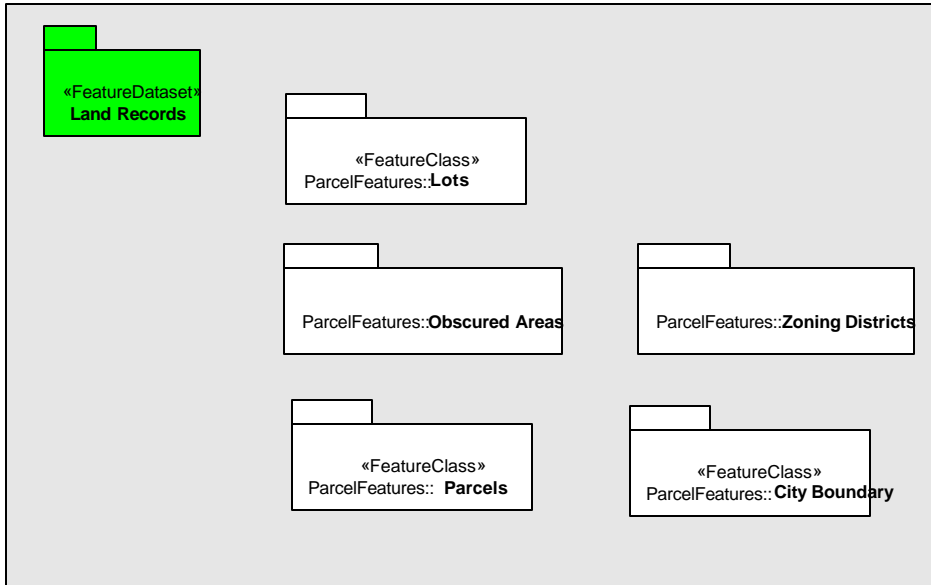
Vegetation



Swamp/Marsh Areas

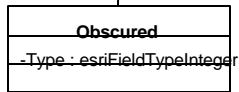
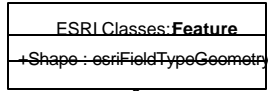


LAND RECORD FEATURE DATASET – WORKSPACE VIEW

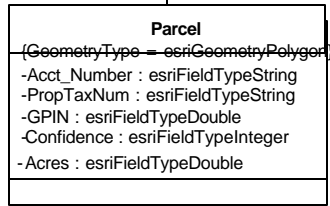
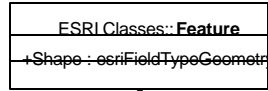


FEATURE CLASSES – LAND RECORD DATASET

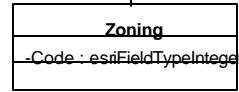
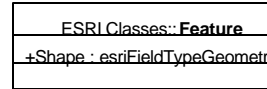
Obscured Areas



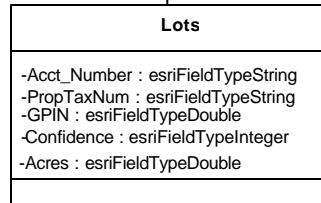
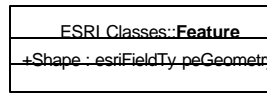
Parcels



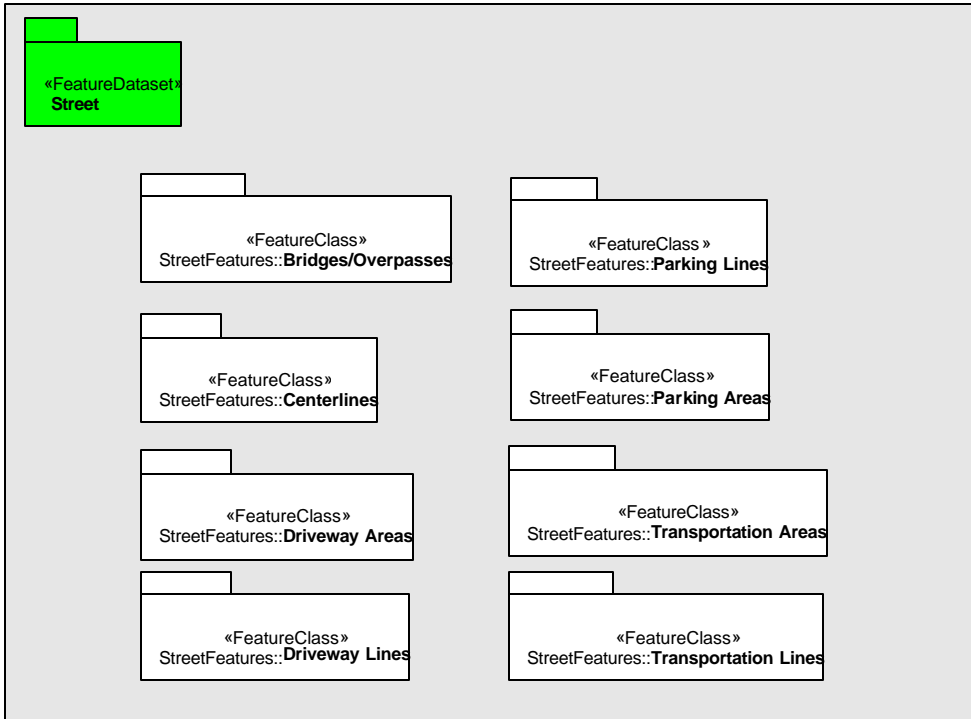
Zoning Districts



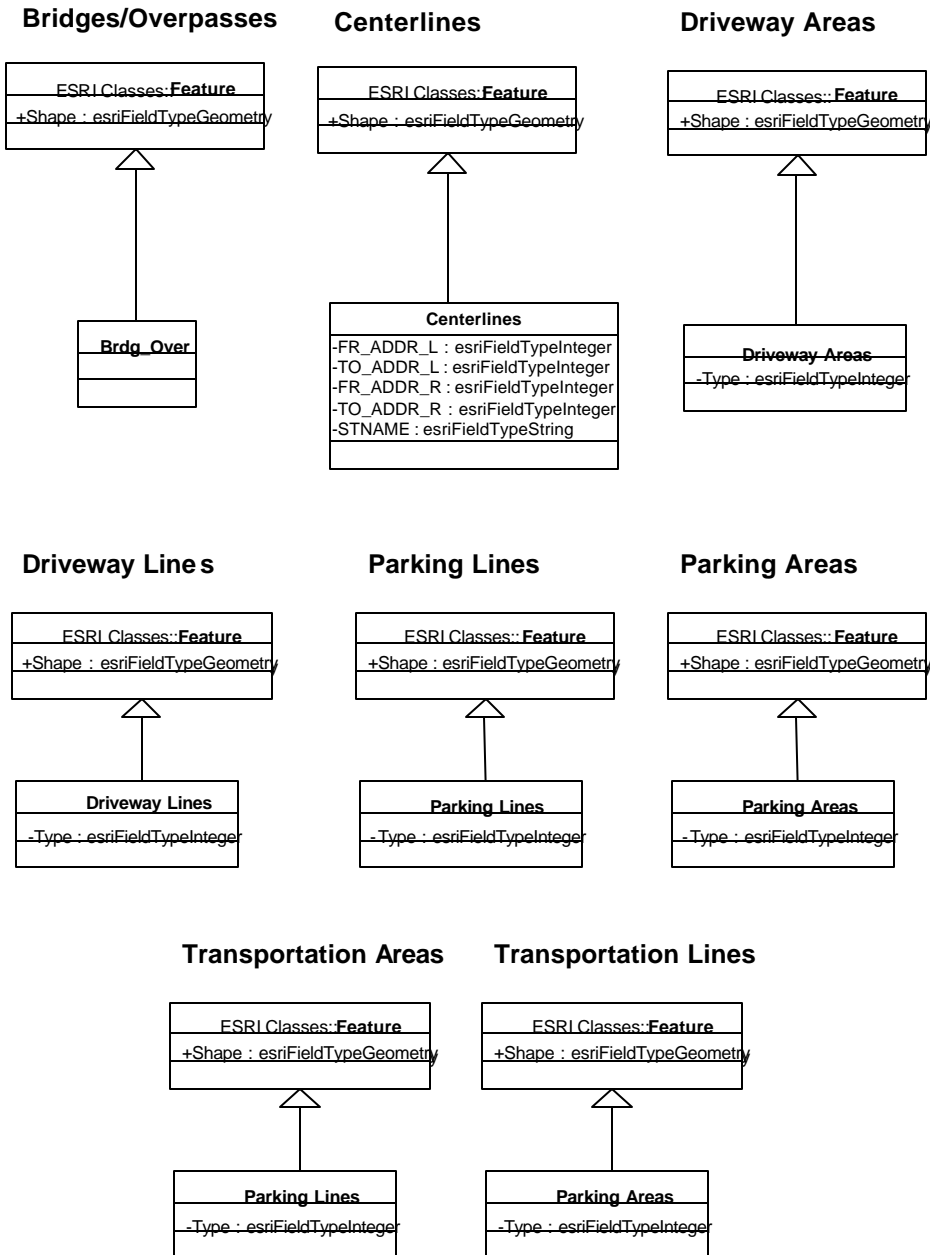
Lots



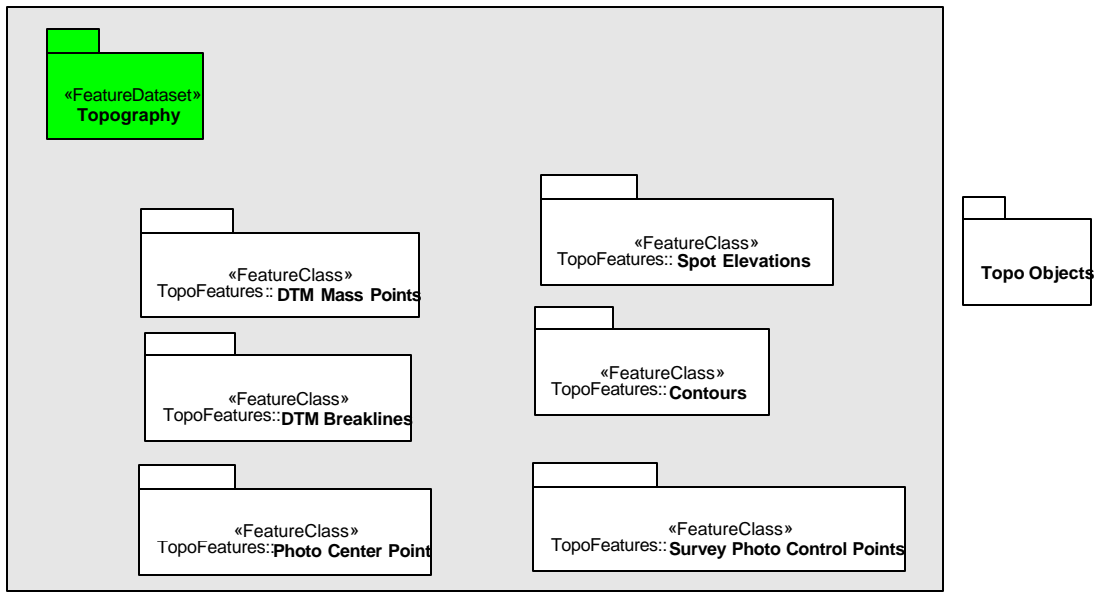
STREET PLANIMETRIC FEATURE DATASET – WORKSPACE VIEW



FEATURE CLASSES – STREET DATASET

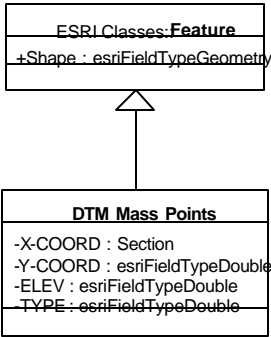


TOPOGRAPHIC PLANIMETRIC FEATURE DATASET – WORKSPACE VIEW

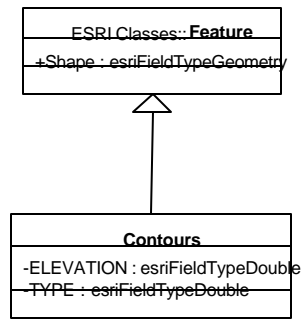


FEATURE CLASSES – TOPOGRAPHIC DATASET

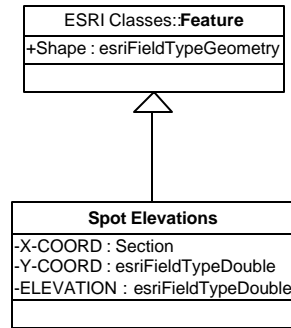
DTM Mass Points



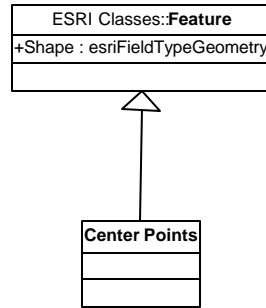
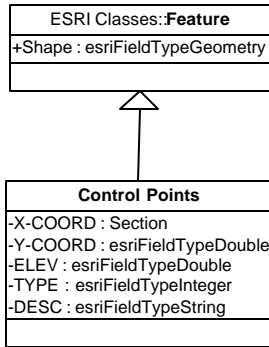
Contours



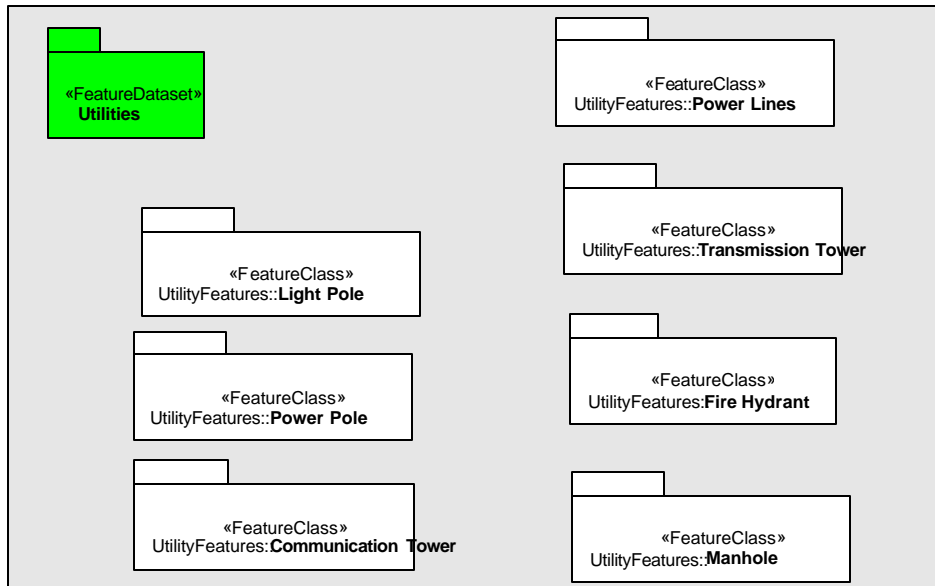
Spot Elevations



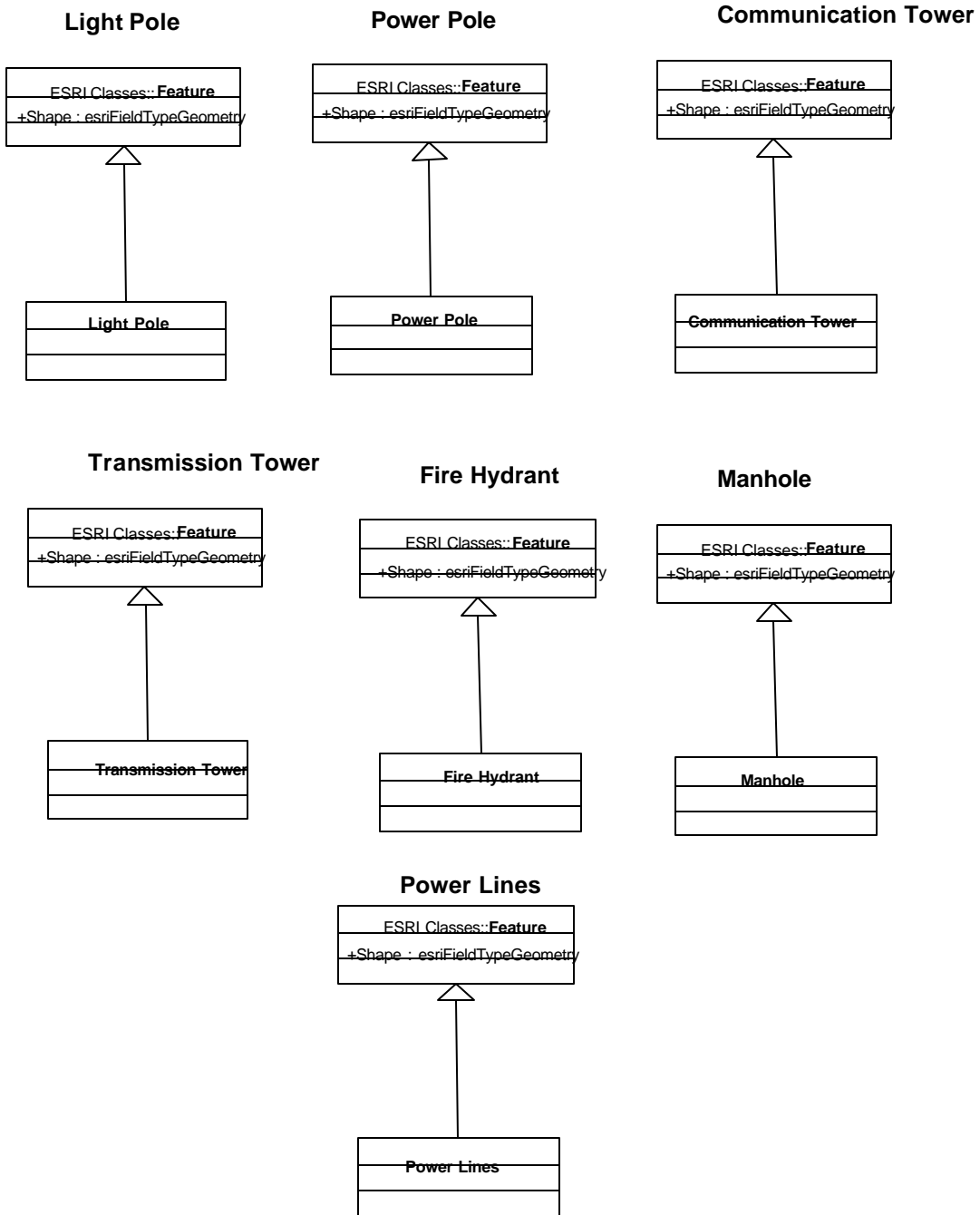
Survey Photo Control Points Photo Center Points



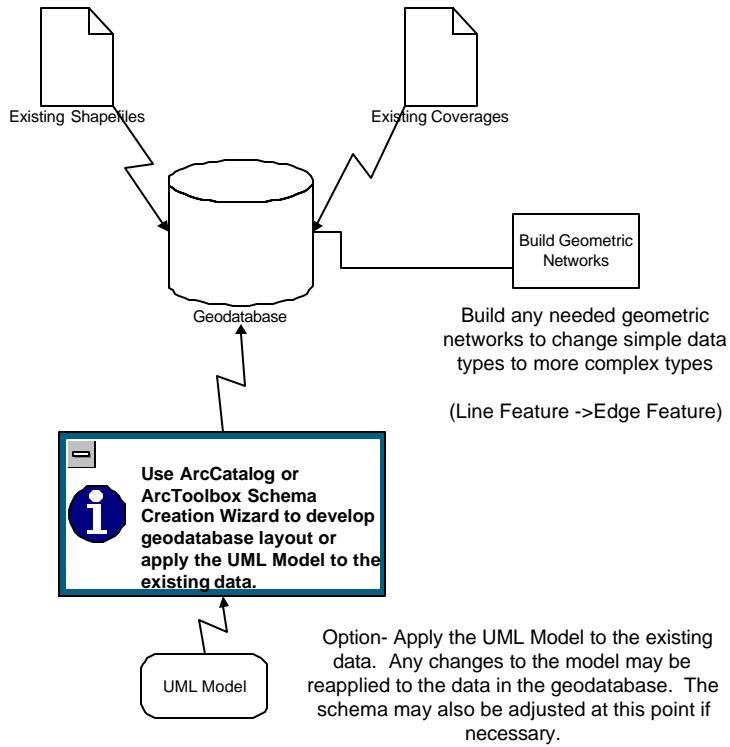
UTILITIES PLANIMETRIC FEATURE DATASET – WORKSPACE VIEW



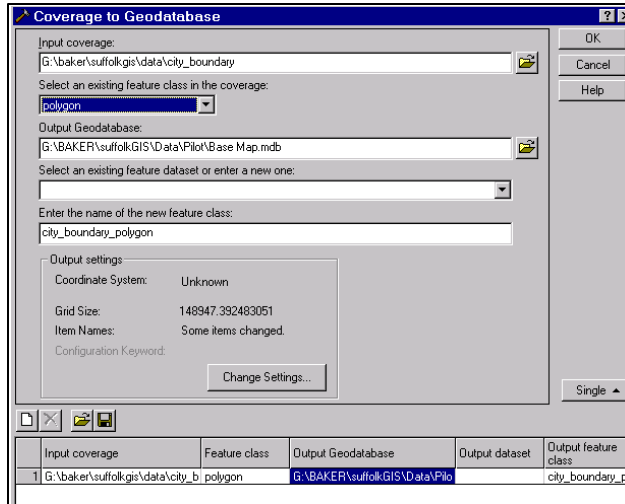
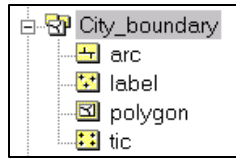
FEATURE CLASSES – UTILITIES DATASET



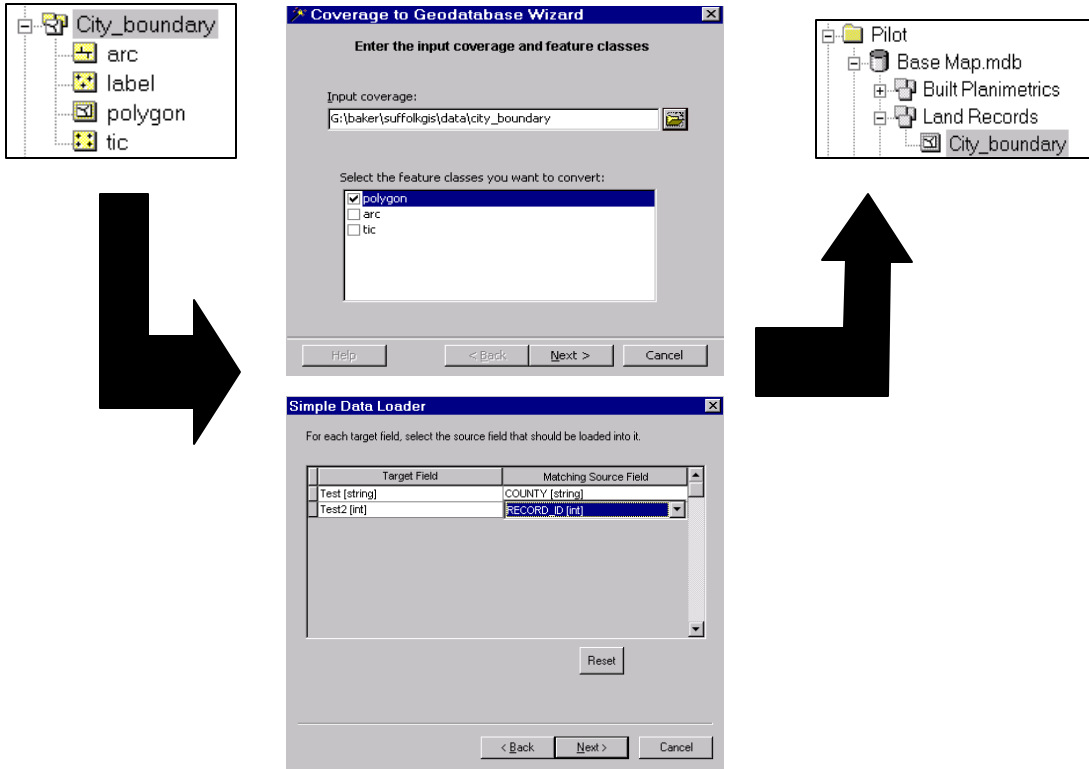
APPENDIX 1 - MIGRATION OF DATA TO GEODATABASE



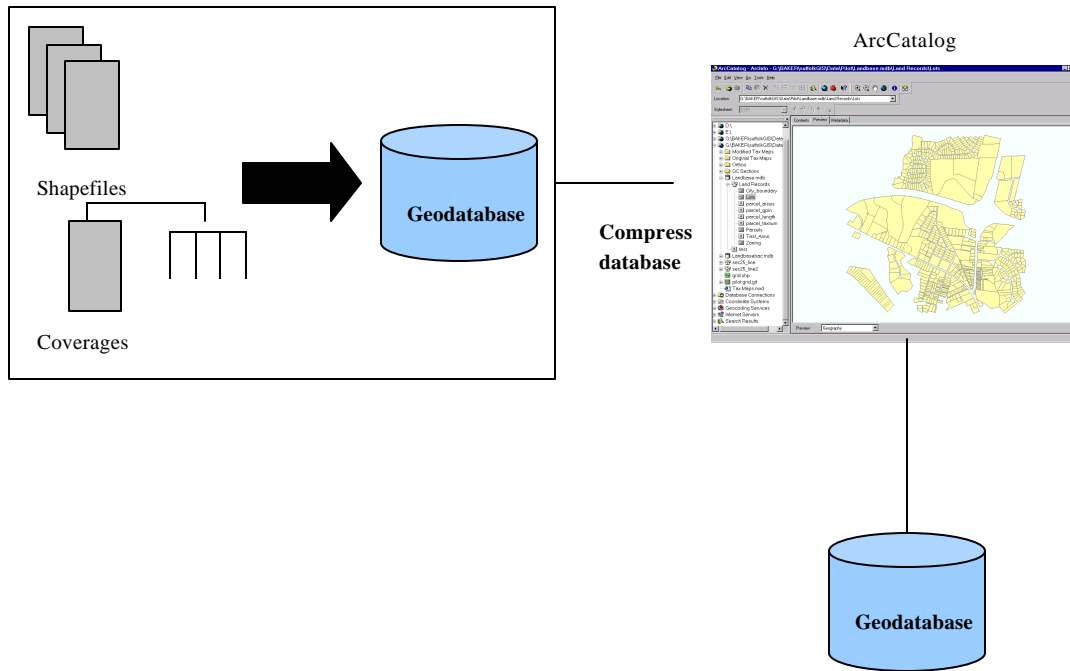
APPENDIX 2 – ARCTOOLBOX MIGRATION WIZARD



APPENDIX 3 – ARCCATALOG MIGRATION TOOL



APPENDIX 4 – OBJECT DATA LOADER IN ARCMAP



APPENDIX 5 - REFERENCES

Zeiler, Michael, (1999), Modeling our World, ESRI Press, Redlands California, 1999