GIS Applications in Community Telecommunications
Seth Peery, Virginia Tech eCorridors
Created: 2004-07-19
Last Revision: *2004-08-30*

Contents

Table of Figures

Introduction

Geographic Information Systems (GIS) have widespread application to many of the problems and activities of communities. An emerging area of interest for communities is telecommunications, and as this guide will demonstrate, GIS can serve as a powerful tool to inform policy decisions in this realm. This guide is intended to serve as a reference for those who are new to GIS, as well as those with some prior GIS experience that want to learn more about how to leverage it towards problems relating to community telecommunications.

"Community Telecommunications" is a general term that refers to a wide range of activities that relate to the improvement of an area's capacity for data transmission, such as:

- Planning
    - Identifying concentrations of population
    - Locating potential subscriber premises
    - Classifying potential subscriber premises by levels of service required
    - Locating existing infrastructure
    - Locating rights-of-way
    - Identifying optimal routes for wired infrastructure and distribution points for wireless infrastructure
- Engineering
    - Detailed physical network design
        - Cable route layout
        - Switching and access locations
        - Outside plant equipment inventory
    - Cost estimation
    - Association of network equipment with network switching and access locations
- Management
    - Asset/Facilities management
    - Maintenance and expansion planning

What is GIS?

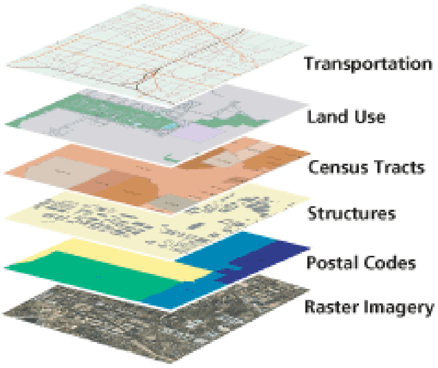


Figure 1: GIS Data Layers Example

GIS software systems link features on a map to descriptive information known as *attribute data*. This allows GIS uses to simultaneously leverage both the visual advantages of a map and the data storage and retrieval advantages of a relational database. The old adage that "a picture is worth a thousand words" is very relevant in this context, as many organizations have substantial amounts of data in tabular form that, when displayed on a map, become much easier to understand and analyze. GIS allows users to answer *spatial questions* of location, proximity, and geographic distribution. As shown in Figure 1[1], GIS represents multiple attributes of geographic space as a set of overlapping layers. Layers may be turned on or off, and new relationships between layers can be discovered, both through simple observation and through the application of advanced spatial analysis techniques. For more general information on GIS, visit http://www.gis.com.

Organization of the Document

This guide will familiarize its reader with some of the more immediately relevant aspects of GIS as applied to problems associated with telecommunications from the perspective of a government or public entity. First, an overview of the necessary hardware and software systems needed to support GIS within an organization will be presented. Next, the types of spatial data used in telecommunications-related GIS analysis, as well as some recommended sources for this data, will be discussed. The remaining portion of the guide will be devoted to a more detailed discussion of GIS techniques used in telecommunications-related analysis, including initial demographic assessments and physical network design.

Hardware and Software Specifications

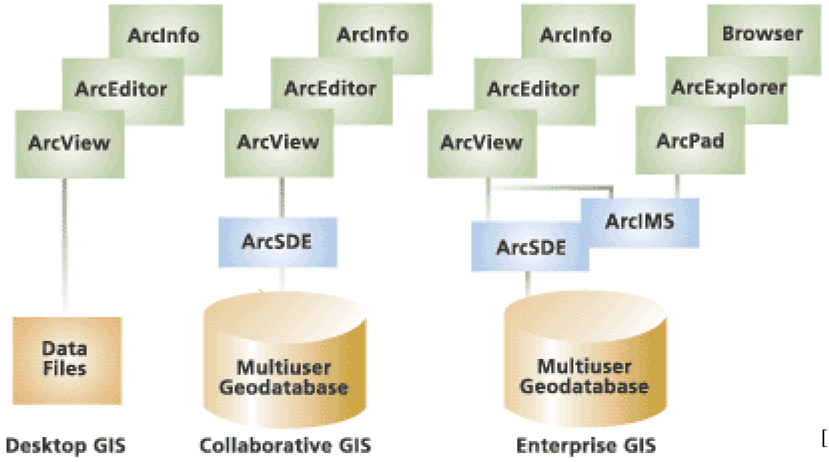
"What do I need, and how much will it cost?" This question is typically foremost on the minds of people new to GIS who are either contemplating or already charged with the task of integrating it into their organization. While this guide does not provide a single definitive answer, it is intended help the reader to more clearly frame the question by identifying the essential considerations and cost drivers, as well as provide some basic recommendations based on our own experience with GIS at Virginia Tech.

GIS Installation Types

First, it is important to realize that there are several ways that GIS can be deployed in an organization, and the most appropriate configuration will depend on an organization's size, budget, number of users, investments in existing data, and technical

[1] Figure 1 source: http://www.esri.com/software/arcgis/concepts/gis-data.html

expertise, as well as the nature of applications GIS is intended to be leveraged towards. The graphic below depicts the three basic ways GIS can be configured to serve the needs of an organization. Although it is taken from the product literature of the Environmental Systems Research Institute (ESRI)'s industry-leading ArcGIS software products, the concept is independent of any particular vendor. Specific software types will be discussed in more detail in the "Software" section of this guide, which appears shortly below.



[2]

Figure 2: GIS Configuration Options

The first configuration, "Desktop GIS," is typical of most local government GIS operations. In this arrangement, GIS software is installed on one or more workstations or laptops, and all the GIS data is stored on the local hard drive. This option is the least complex to implement, and is also the least expensive (both in terms of the software itself and the hardware and network infrastructure needed to support it). However, if multiple users need to share files, they must send them back and forth and keep their own local copies.

In the "Collaborative GIS" scenario, each user would have GIS software installed on their local machine, but date needed by multiple users would be stored on a central database server. In this example, an additional server application (ArcSDE) is needed to provide the interface between the Desktop GIS clients and the database server. The Collaborative GIS arrangement is suitable for organizations that have more intensive *internal* file-sharing demands than the average user, but do not have a need to deliver spatial data outside of their own LAN or WAN[3] environment. In order for Collaborative GIS to be deployed within an organization, it is an essential prerequisite that all users be connected to the central database server via a high-bandwidth network connection. In addition, there will be a need for staff to be on hand with the technical competencies to maintain the database server. Because of the additional costs associated with the database server, database software, database personnel, network infrastructure, etc., this option will be more expensive to implement than the Desktop GIS option.

[2] Figure 2 source: http://www.esri.com/software/arcgis/about/overview.html

[3] LAN – Local Area Network; WAN – Wide Area Network

The third configuration option, "Enterprise GIS," is suitable for organizations with many GIS users, intensive internal *and external* file-sharing needs, and the need to distribute geographic data in multiple formats, such as those required by Internet and mobile applications. This option will require the largest investment in hardware and software; however, organizations that require an enterprise GIS solution often already have database servers, web servers, and robust network infrastructure in place from which to run the GIS applications.

Hardware

The choice of a hardware platform for GIS will depend on many factors. As the previous section has demonstrated, there are different hardware needs associated with each type of GIS configuration. Desktop GIS is the foundation of the Collaborative GIS and Enterprise GIS configurations, and the other two may be thought of as adding additional functionality to the base system. For this base system, there are several rules of thumb to keep in mind:

Storage

As a rule, GIS data takes up a considerable amount of hard drive space. Datasets that cover a large geographic area, datasets with detailed attributes, and raster datasets, such as aerial imagery and elevation surfaces, are especially large. It is recommended that a GIS workstation be equipped with a large hard drive (60-80GB minimum), and/or multiple hard drives. Also, drives that run at a higher RPM (such as 7200 RPM drives) with shorter average seek times, will yield better performance when the system needs to store large amounts of data in virtual memory (swap space). The operating system's page file (swap file) should be set to at least twice the size of the system's RAM, and in some cases (e.g., when system RAM is less than 512 MB), it may be necessary to set the page file size even larger.

Memory

Due to the large size of the geographic datasets that GIS software applications typically process, the amount of physical memory (RAM) available to GIS software is usually the most significant determinant of overall system performance. It is recommended that an absolute minimum of 512MB RAM be installed in a GIS workstation, with 1024MB (1GB) being the preferred alternative. When purchasing a workstation or laptop for GIS, it is recommended that there be sufficient memory expansion slots to accommodate at least 1GB of RAM. Because a shortage of RAM will drastically reduce system performance, and may even cause the operating system to crash, it is recommended that other components of the system be scaled back before RAM if cost becomes an issue.

CPU

The clock speed of a GIS system's CPU should be as high as practical, but for most applications an increase in processor speed will have less of an impact on overall system performance than the amount of physical memory available. An exception to this rule of thumb occurs in cases where the workstation will frequently be used for computationally-intensive geoprocessing tasks, such as surface (raster) analysis, reprojecting large datasets, and 3D visualization. In situations where multiple users in a networked environment need to perform geoprocessing functions, it may be practical to designate a single workstation as a geoprocessing server, equip it with a faster CPU, and then configure the users' workstations to access this server instead of each of them having to perform all operations locally. At minimum, user workstation CPUs should have at least 1 GHz of processing power, with 2 GHz – 3 GHz and above being the preferred alternative.

Graphics

Because GIS is, by its nature, a graphically-intensive operation, user workstations should be equipped with OpenGL graphic cards and up-to-date display drivers. Graphic cards should be equipped with no less than 32MB of video RAM, and if the workstation is to be used for 3D visualization, up to 128 MB of video RAM may be needed for satisfactory performance. Monitors for desktop workstations should be at least 17" and capable of at least 1024x768 resolution in high color mode. Larger monitors, 19" – 21" in size, and capable of supporting up to 1600x1200 resolution in high color mode, will increase the system's ease of use. Wide-aspect-ratio monitors are also good alternatives. For laptops, the largest practical LCD display should be selected.

Removable Storage

For the purposes of exchanging and backing up large amounts of data, all GIS workstations should be equipped, at minimum, with a CD-RW drive, and preferably a DVD-R/DVD-RW drive. An external FireWire or USB hard drive is an additional accessory that makes the often overlooked necessity of performing regular backups much more convenient. Many laptops can be configured with the option of a removable modular second hard drive.

Peripherals

For printing maps, every GIS workstation should have access to some type of color printer. If a sufficiently large number of GIS users regularly need to produce a large number of paper maps, a color laser printer may be desirable. If large-format maps are required, it may be necessary to purchase a plotter. This, however, can substantially increase the total system cost – by $10,000 or more.

Network Infrastructure

If multiple users will need to share GIS data within an organization, having adequate local area network infrastructure is essential. If the

"Collaborative GIS" or "Enterprise GIS" configurations are used, a robust, reliable, high-bandwidth LAN is a crucial prerequisite.

Cost

A GIS workstation with the minimum specifications recommended by this guide is likely to cost around $1,500. However, the cost figure could rise to around $3,000 if the preferred recommendations for memory, CPU, etc. are followed. As a general rule, laptops will cost more than desktops with the same specifications. Because of the higher performance demands placed on a system running GIS, organizations should expect to spend more on a GIS workstation than on a regular non-GIS workstation. If

Software

There are several software products on the market today that have GIS functionality. ESRI, Autodesk, MapInfo, Microsoft, and many others have developed mapping applications that fill various niches. However, not all "mapping software" is created equal. ESRI's ArcGIS product is by far the most comprehensive, mature, robust, extendable, and scalable software solution. With a very large user community throughout the world, well-developed user support, and proven technology, ESRI ArcGIS is becoming the *de facto* standard for GIS software. As such, the examples in this guide were developed in an ArcGIS environment.

ArcGIS is a scalable desktop GIS software package. The current version is ArcGIS 9.0. Licenses may be purchased as single-use or floating. An organization has three licensing options to choose from:

Product	Description [4]	Cost
ArcView	provides extensive mapping, data use, and analysis along with simple editing and geoprocessing capabilities	$1,500 (single-use license)
ArcEditor	includes advanced editing for shapefiles and geodatabases in addition to the full functionality of ArcView	Variable [5]

[4] Source: ESRI ArcGIS Desktop Product Literature: http://www.esri.com/software/arcgis/about/desktop.html

[5] For the higher-end licensing options (ArcEditor and ArcInfo), ESRI does not publish cost figures. Price quotes depend on the number of licenses required, as well as the nature of the requesting organization.

ArcInfo	the full function, flagship GIS desktop. It extends the functionality of both ArcView and ArcEditor with advanced geoprocessing. It also includes the legacy applications for ArcInfo Workstation.	Variable[5]

Depending on the needs of the organization, it may be necessary to purchase specialized software extensions to the base system. This will be the case, for example, if raster-based surface analysis, nationwide street mapping, geostatistical analysis, or 3D analysis functionality is desired. Extensions range in price, but most cost $1,000 -$2,500.

Consideration should also be given to non-GIS software types. The computer operating system must be Microsoft Windows 2000 or XP – ArcGIS will not run on Windows 9x, and is currently not supported on the UNIX or Macintosh platforms. Other types of software may also be useful, including:

- A secure FTP client
- A personal firewall
- Antivirus software (a must)
- System backup software
- Standard office software

Data Considerations

Many sources of GIS information suggest that over the life-cycle of a GIS system, the largest cost driver is actually data, not hardware or software. This may or may not be true in all cases. If an organization needs to convert a large amount of paper-based or other legacy data, or if it is required to purchase large amounts of proprietary data, the share of the total system cost corresponding to data will be high. Much of this cost is associated with personnel time (in-house or consulting) devoted to performing the data conversion.

There are several kinds of data that will need to be acquired for a GIS system used for performing telecommunications-related analysis. The first of these is *basemap* data, which includes political boundaries, the locations of transportation infrastructure, place names, streams and water bodies, elevation surfaces, etc. Data of this type may be acquired from several sources, and often at little or no cost. Some sources include:

- The ESRI Data and Maps DVD that ships with ArcGIS, which contains most basic base map feature layers such as political boundaries, place names, and generalized roads.

Virginia has a negotiated Master Pricing Agreement with ESRI, and some localities have their own agreements. For more information, contact ESRI Sales at 1-800-447-9778.

ú The ESRI StreetMap USA extension, which provides more finely detailed nationwide coverage of roads, water bodies, airports, parks, and several other layers.

ú The US Geological Survey (USGS), which provides terrain/elevation models, low-resolution aerial imagery, digitized topographic maps, land cover maps, and many other products. Their primary distribution portal is located at http://seamless.usgs.gov.

ú The U.S. Census Bureau, which provides demographic data. Also, several third-party vendors have re-packaged this data into a more GIS-friendly format.

For a more detailed discussion of the public datasets that exist, an excellent reference guide is "GIS and Public Data" by Bruce Ralston, Thomson-Delmar Learning, 2004. Also, data sources may be located and accessed from GIS portal sites such as

http://www.geographynetwork.com
http://www.gis.com/jumpstation/
http://www.gisportal.com
http://data.geocomm.com/

The Virginia Economic Development Partnership (VEDP) provides several datasets that will be of interest to localities within the state of Virginia. VEDP offers, free of charge, datasets for schools, hospitals, colleges, emergency response stations, enterprise zones, and others – as well as USGS Digital Ortho Quarter-Quad (DOQQ) aerial imagery. Their website is http://gis.vedp.org/.

Local data, such as building footprints (structures), tax parcel information, street centerlines, and government facilities is often already available within planning, engineering, public works, and public safety/e-911 departments. The structures layer is especially useful for telecommunications planning, as it allows GIS analysis to be conducted at the level of the subscriber premise instead of the Census Block, street, or neighborhood level.

Another category of GIS data is that which includes features relating to telecommunications, such as cable and ductbank routes, rights-of-way, poles, wireless tower locations, and central offices. This data is typically not available from the usual free GIS data portals, and in most cases must be purchased from a third-party vendor or else created in-house. In many cases, the location of existing telecommunications plant is considered proprietary information by the companies that own it, and such data can typically only be obtained through nondisclosure agreements, in the rare cases where it can be obtained at all.

GIS in Initial Telecommunications Assessments

Typically, the first step for most communities in planning for telecommunications is some type of assessment. Assessments usually seek to determine the extent of existing telecommunications infrastructure and services, and the location of potential customers. If the assessment reveals a deficiency in supply relative to the estimated demand for telecommunications services, GIS can be used to design scenarios for bridging the gap.

Mapping Existing Infrastructure

Many community officials concerned with telecommunications availability within a region assume that a first step in understanding their situation is to map out the location of the physical infrastructure of their local telecommunications provider(s). This typically proves to be a futile effort, since as was previously mentioned, the telecommunications providers are nearly always reluctant to provide this type of information, which they view as competitively sensitive. As it turns out, however, this information is not nearly as useful as might initially be imagined. Traditional telecommunications providers (i.e., Local Exchange Telephone Companies and Cable TV Providers) are in the business of selling *services*, and their infrastructure networks are closed systems, wholly owned and controlled by the provider, that are simply a means to the end of delivering those services. Furthermore, the architecture of telecommunications networks frequently limits the number of points from which the network can be accessed, so knowing, for example, that a local-provider's fiber-optic cable passes an industrial site does not guarantee that that site can be connected or can receive broadband internet services. A host of other factors will determine this, and to simply concern oneself with the location of privately owned infrastructure is to ask the wrong spatial question.

Most communities embark upon telecommunications planning efforts in response to a perception of an unmet need, usually relating to the uneven availability of affordable advanced telecommunications services, or *broadband* [6]. The appropriate question to ask, then, is not "where is existing telecommunications infrastructure located?" but rather, "where are broadband services currently available?" There are several ways to provide partial answers to this question. Private telecommunications providers are far more likely to provide information relating to their service areas than their infrastructure, and conversations with their representatives can yield useful pieces of information that become even more valuable when mapped using GIS technology. If, for example, it is known that a central office facility is capable of providing DSL services,

[6] For the purposes of this guide, broadband is defined as "high-speed, always available, access to the Internet and Intranet. It should have sufficient downstream and upstream bandwidth per user to support the full range of converged data, voice, and video applications expected. Low end broadband today is in the megabit per second range and high end is gigabit per second range. Research needs are in the tens of gigabits per second range." (Definition proposed by John Nichols, Virginia Tech CNS)

Identifying Demand

Mapping Block-Level Population Densities

A useful first step in identifying areas of broadband demand is developing an understanding the local population distribution. Using Census Block-level demographic data, it is relatively easy to create a map of population density. Mapping population patterns at the block level – the smallest unit of Census geography – can yield insights into the general location of potential subscribers, and can help to ensure that isolated pockets of population are not left out of the planning process. The figure below is an example population density map for Montgomery County, VA, and the inset shows a zoomed-in view of the downtown Blacksburg area, near the Virginia Tech campus. As this example illustrates, the census blocks vary in size, but in more densely populated areas the level of detail increases.

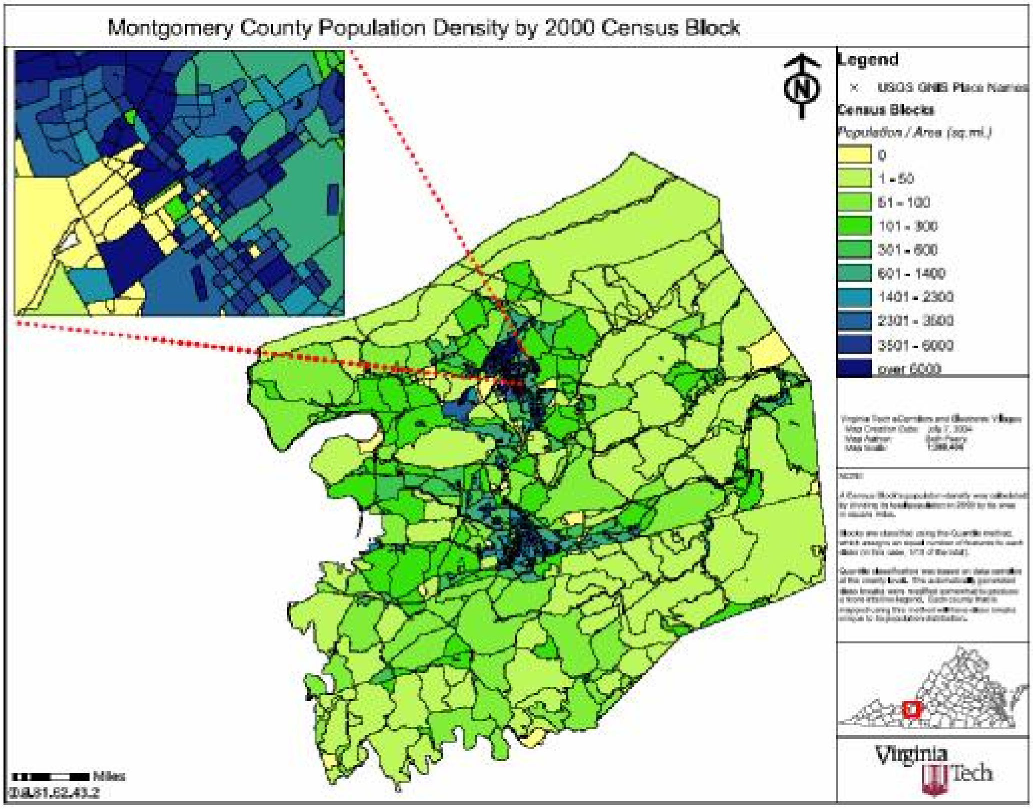
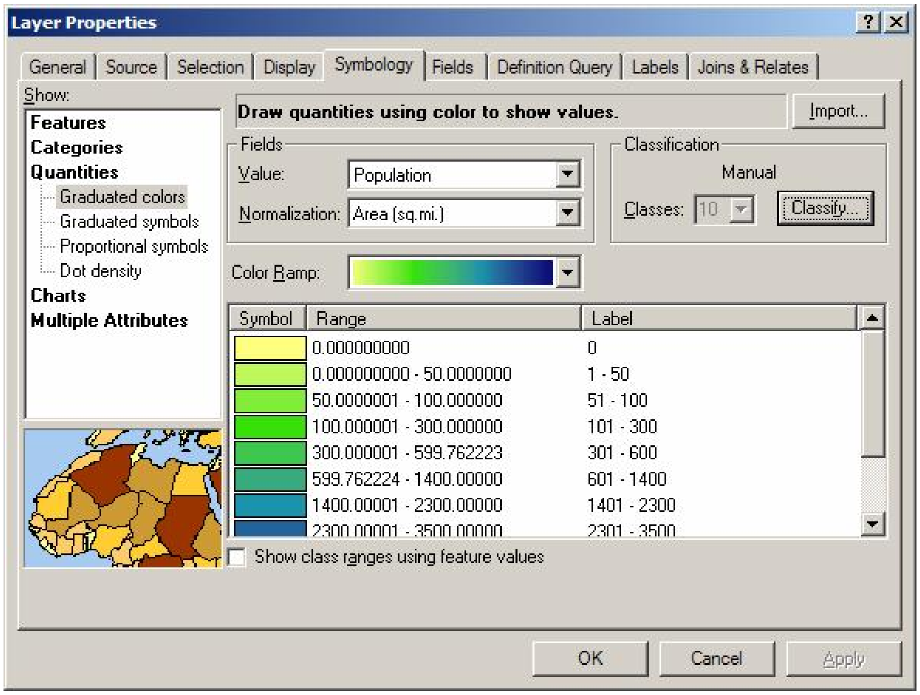


Figure 3: Example Population Density Map

This map was created using a census blocks shapefile prepared by a third-party vendor. However, Census data is in the public domain, and can be obtained from http://www.census.gov. There are two components needed for the population density map: the block boundary files, which require translation from the Census's native TIGER/LINE format (the book "GIS and Public Data" discusses a utility called TGR2SHP which performs this operation), and the census demographic information, which is can be obtained in tabular form in a variety of ways from the Census website. The third-party dataset was only used for convenience. The map symbology method is straightforward: the blocks were symbolized by the quantity “Population” and normalized (divided) by the area of the census block in square miles. [7] The blocks were then classified using the quantile (equal feature count per class) method. To display only the values for Montomgery County, a Definition Query was defined for the layer based on the FIPS code field in the Census Block shapefile.



Draw quantities using color to show values.

Fields

Value: Population

Normalization: Area (sq.mi.)

Classification

Manual

Classes: 10

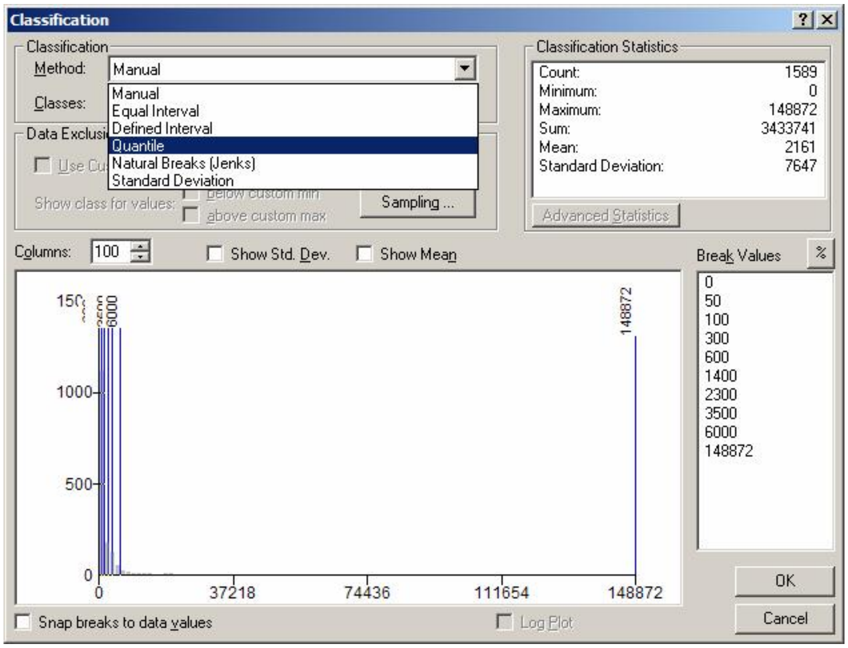
Color Ramp:

Symbol	Range	Label
🟨	0.000000000	0
🟩	0.000000000 - 50.0000000	1 - 50
🟩	50.0000001 - 100.000000	51 - 100
🟩	100.000001 - 300.000000	101 - 300
🟩	300.000001 - 599.762223	301 - 600
🟩	599.762224 - 1400.00000	601 - 1400
🟦	1400.00001 - 2300.00000	1401 - 2300
🟦	2300.00001 - 3500.00000	2301 - 3500

☐ Show class ranges using feature values

Figure 4: Census Blocks symbology settings

[7] The block area in square miles was calculated from the automatically generated SHAPE_AREA field in the geodatabase where the files were stored. In order for this to work, the data had to be projected into a coordinate system that used linear measurements (feet) instead of decimal degrees.



Classification

Method: Manual

Classes:

Classification Statistics

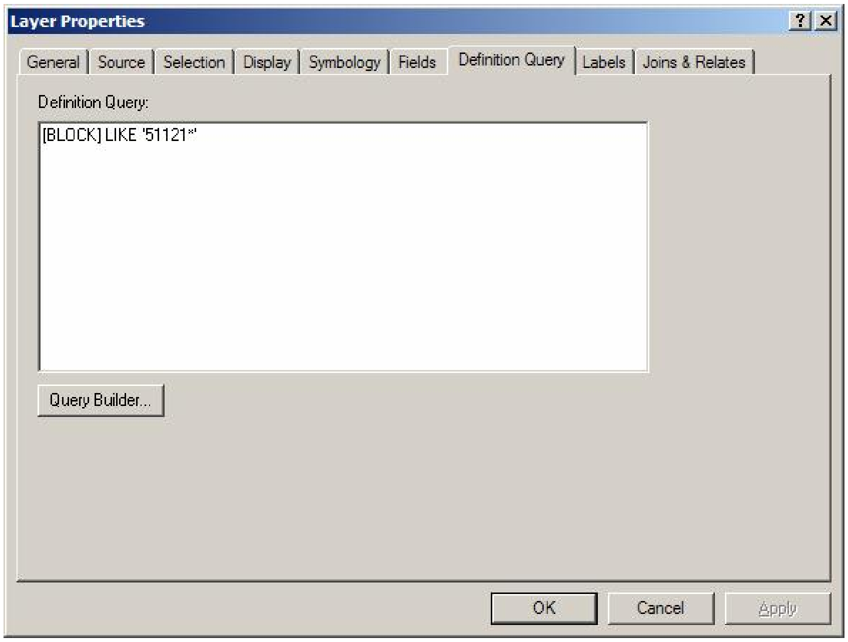
Count:	1589
Minimum:	0
Maximum:	148872
Sum:	3433741
Mean:	2161
Standard Deviation:	7647

Columns: 100

Break Values

0
50
100
300
600
1400
2300
3500
6000
148872

Figure 5: Census Blocks classification method



Definition Query:

[BLOCK] LIKE '51121*'

Figure 6: Definition Query for Montgomery County

Case Study: Methodology for Creating Business Point Maps

In the TOP Grant project, an analysis of business locations was conducted using a proprietary dataset called “Business Points,” which Virginia Tech purchased from MapInfo. The dataset contained point locations for businesses which had been obtained from a variety of sources such as yellow pages, telemarketing lists, and other publications. Each point was associated with attributes such as industry sector, employee count and annual sales. To map the data, it was necessary to first convert it from its native MapInfo format into Shapefile format. Some additional processing steps were also necessary, such as the addition of attribute fields for the full name of the industry sectors (for symbology purposes). Then, a thematic map was generated based on a “Multiple Attributes” classification method that assigned dot color based on industry sector and dot size based on employee count. This allowed concentrations of large businesses to be easily identified, and showed relationships between the location patterns of different business types.

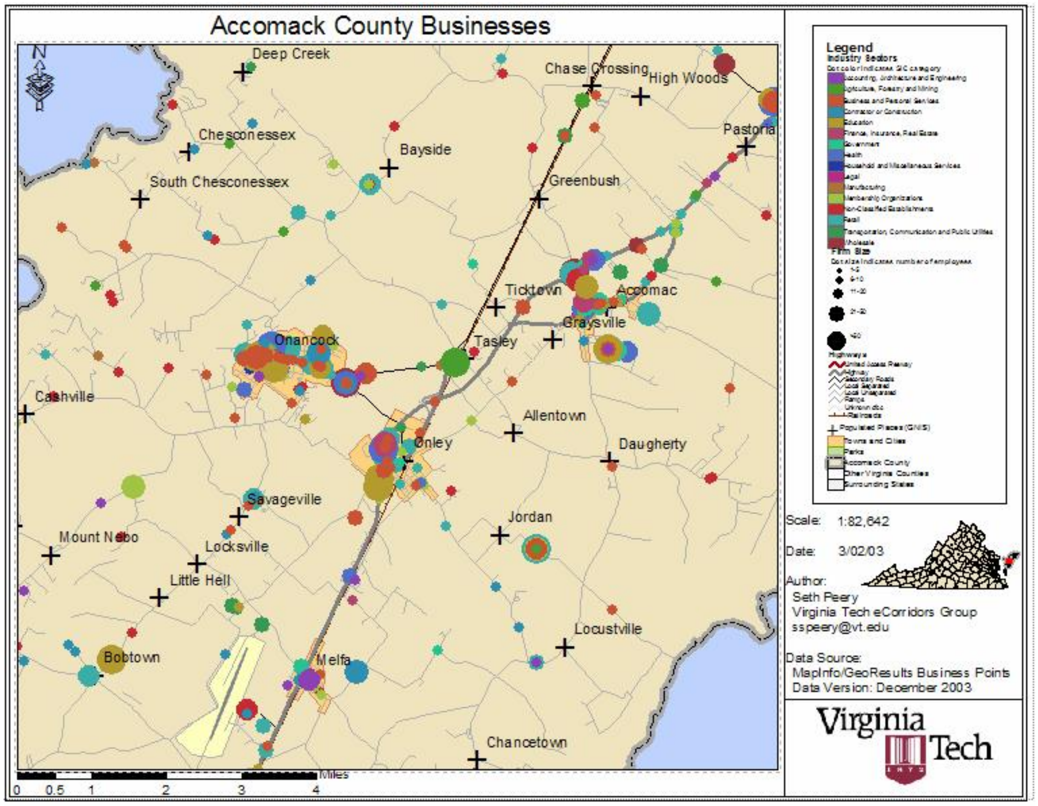
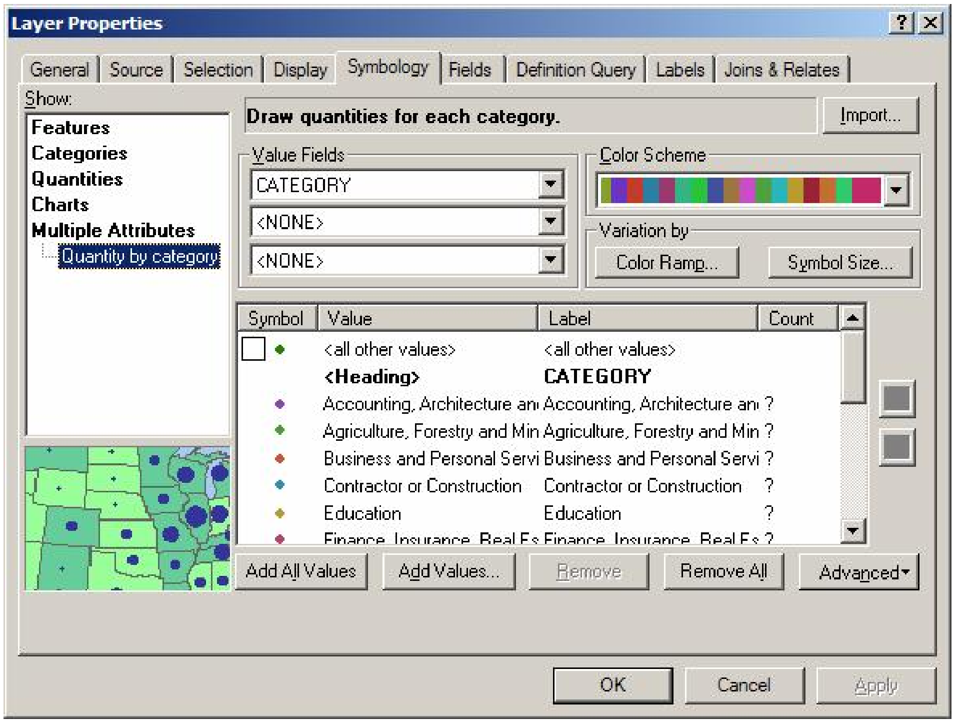


Figure 7: Example Business Point Map for Central Accomack County



Show:

- Features
- Categories
- Quantities
- Charts
- Multiple Attributes
  - Quantity by category

Draw quantities for each category.

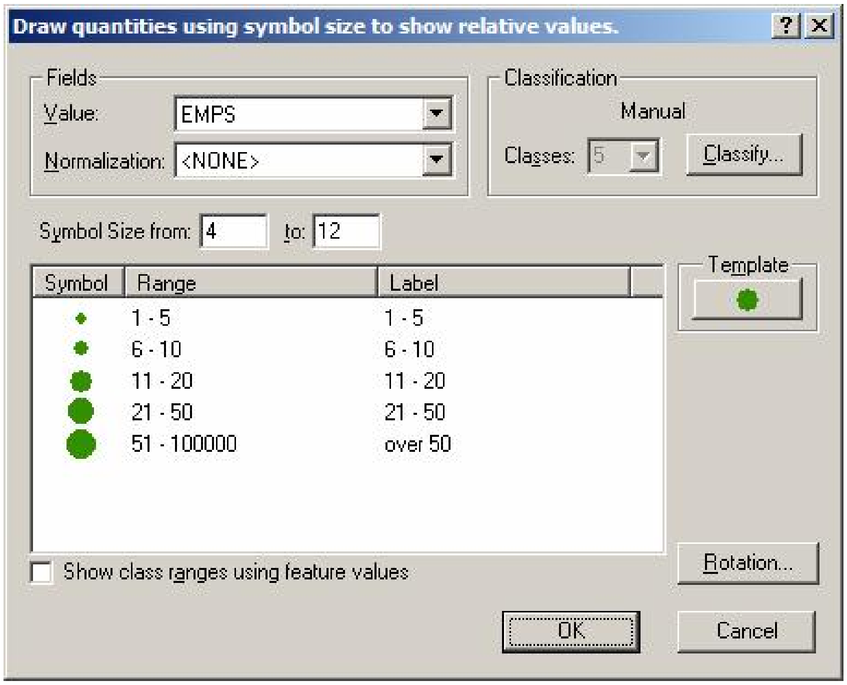
Value Fields: CATEGORY; <NONE>; <NONE>

Color Scheme

Variation by: Color Ramp... | Symbol Size...

Symbol	Value	Label	Count
□ ●	<all other values>	<all other values>	
	<Heading>	**CATEGORY**	
●	Accounting, Architecture an	Accounting, Architecture an	?
●	Agriculture, Forestry and Min	Agriculture, Forestry and Min	?
●	Business and Personal Servi	Business and Personal Servi	?
●	Contractor or Construction	Contractor or Construction	?
●	Education	Education	?
●	Finance, Insurance, Real Es	Finance, Insurance, Real Es	?

Figure 8: Multiple Attributes Symbology (colors)



Fields

Value: EMPS

Normalization: <NONE>

Classification: Manual

Classes: 5

Symbol Size from: 4 to: 12

Symbol	Range	Label
●	1 - 5	1 - 5
●	6 - 10	6 - 10
●	11 - 20	11 - 20
●	21 - 50	21 - 50
●	51 - 100000	over 50

□ Show class ranges using feature values

Figure 9: Multiple Attributes Symbology (dot size)

Using Parcel Data to Classify Structures

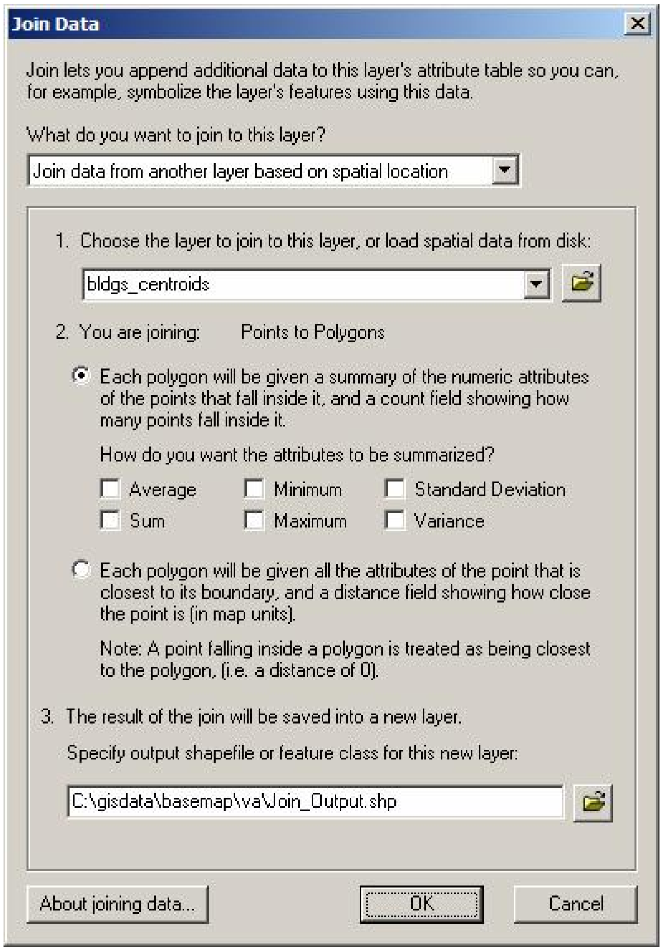
The Business Point dataset was useful, but it did contain errors; not all businesses were correctly located or attributed. More significantly, perhaps, its proprietary nature limits its applicability to community telecommunications planning. Because the location of businesses is an important piece of information as one uses GIS to estimate broadband demand, an alternate method was needed that could leverage data that local most governments already possessed.

As mentioned earlier, most county GIS, planning, engineering, or 911 departments have a GIS or CAD file that contains the building footprints. It was envisioned that a GIS-based process for the classification of buildings based on their use – residential, business, or government – could be as valuable (if not more so) than the proprietary business points dataset, since we would have information about actual buildings, instead of abstract points geocoded along roads with varying degrees of accuracy.

There are several ways to do this. Since local GIS files follow no uniform convention for the storage of attribute data, the exact process varies from jurisdiction to jurisdiction. In towns that have zoning ordinances, for example, the buildings may already be classified with their zoning code. In other places, however, the use of the building must be inferred. As it turns out, GIS is very good at associating the attributes of a feature with another feature that occurs at or near the same place (a process called *spatial joining*).

Commissioners of revenue typically keep a record of the current use of each land parcel in their jurisdiction for taxing purposes. By making the reasonable (though by no means flawless) assumption that a building has the same use as the land it sits on, we used a spatial join procedure to append the land use codes of a parcels layer to a structures layer. Initially, there were some errors in the calculation, because not all buildings are neatly situated on a single parcel of land. To remedy this problem, the centroids (polygon center points) were calculated for each building. The parcel attributes were then joined to the centroids, and the centroid attributes were joined to the buildings. This yielded much better results that seem to provide a better approximation of real-world business locations than the business points dataset. In addition, since each building in a community is coded with a use value, the locations of residential neighborhoods can be determined along with the location of businesses and government facilities, which was not possible with the business points.\

Though this method is by no means perfect, it does allow communities to leverage their own data resources towards meaningful broadband demand assessment. Because local structures files are not typically available to the public, it can give local officials an advantage over their counterparts in the private sector, and can yield “inside information” that can be valuable in discussions with telecommunications providers.



Join lets you append additional data to this layer's attribute table so you can, for example, symbolize the layer's features using this data.

What do you want to join to this layer?

Join data from another layer based on spatial location

1. Choose the layer to join to this layer, or load spatial data from disk:

bldgs_centroids

2. You are joining: Points to Polygons

● Each polygon will be given a summary of the numeric attributes of the points that fall inside it, and a count field showing how many points fall inside it.

How do you want the attributes to be summarized?

□ Average □ Minimum □ Standard Deviation
□ Sum □ Maximum □ Variance

○ Each polygon will be given all the attributes of the point that is closest to its boundary, and a distance field showing how close the point is (in map units).

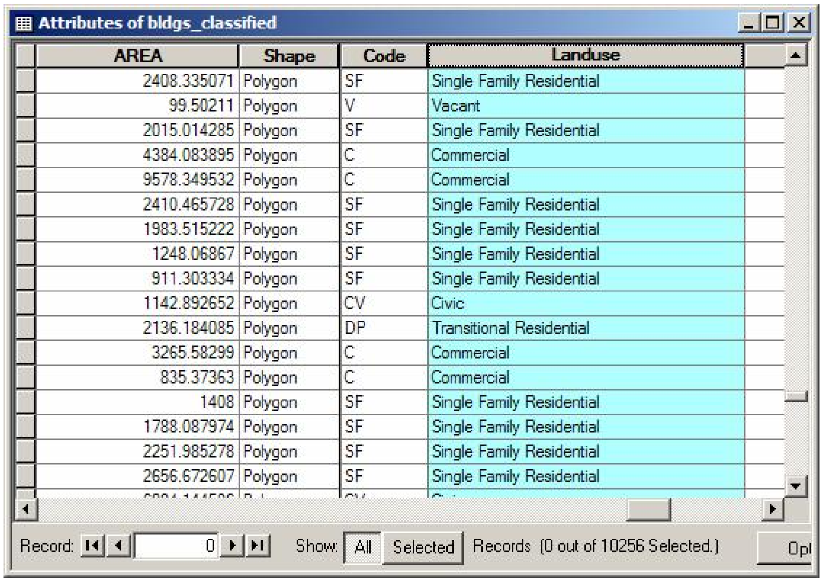
Note: A point falling inside a polygon is treated as being closest to the polygon, (i.e. a distance of 0).

3. The result of the join will be saved into a new layer.

Specify output shapefile or feature class for this new layer:

C:\gisdata\basemap\va\Join_Output.shp

Figure 10: Spatial Join in ArcMap



AREA	Shape	Code	Landuse
2408.335071	Polygon	SF	Single Family Residential
99.50211	Polygon	V	Vacant
2015.014285	Polygon	SF	Single Family Residential
4384.083895	Polygon	C	Commercial
9578.349532	Polygon	C	Commercial
2410.465728	Polygon	SF	Single Family Residential
1983.515222	Polygon	SF	Single Family Residential
1248.06867	Polygon	SF	Single Family Residential
911.303334	Polygon	SF	Single Family Residential
1142.892652	Polygon	CV	Civic
2136.184085	Polygon	DP	Transitional Residential
3265.58299	Polygon	C	Commercial
835.37363	Polygon	C	Commercial
1408	Polygon	SF	Single Family Residential
1788.087974	Polygon	SF	Single Family Residential
2251.985278	Polygon	SF	Single Family Residential
2656.672607	Polygon	SF	Single Family Residential

Record: 0 Show: All Selected Records (0 out of 10256 Selected.)

Figure 11: Landuse Attribute Joined to Buildings Layer

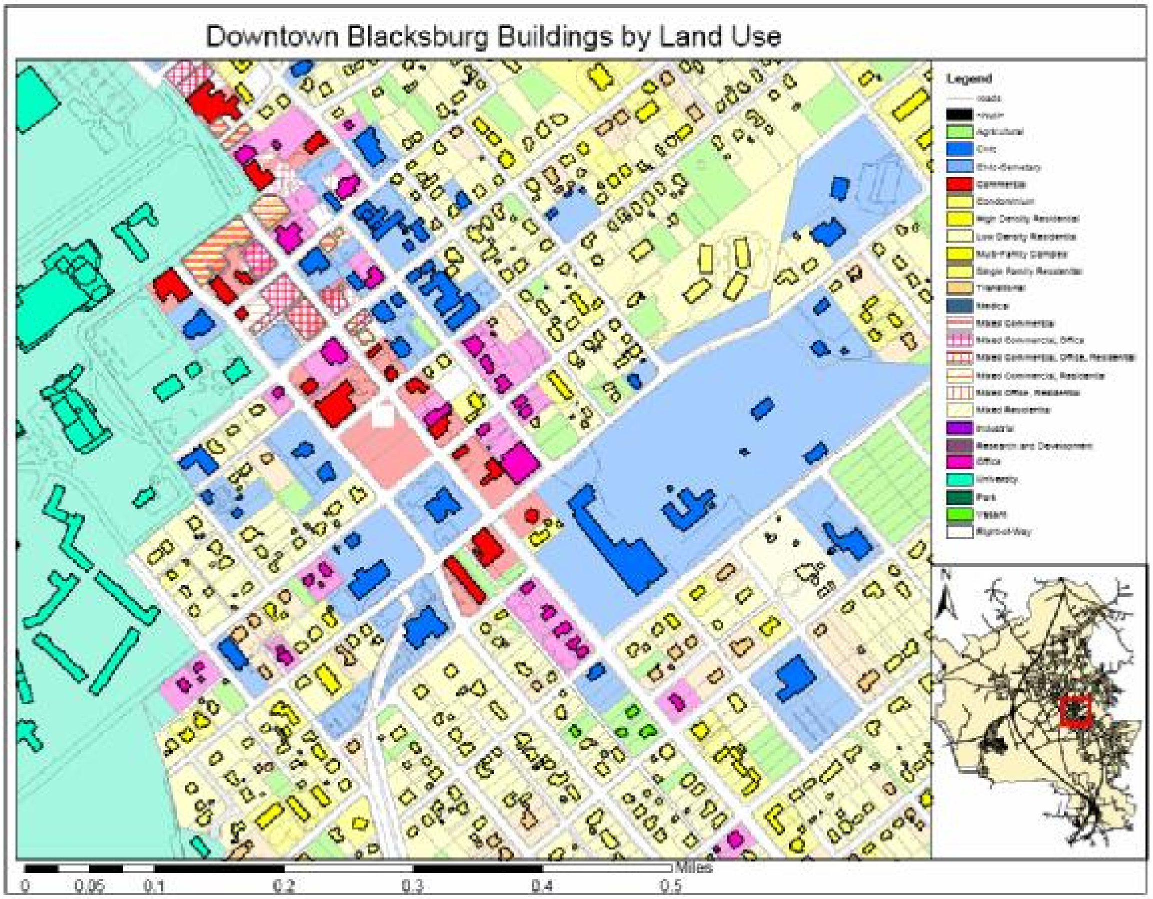


Figure 12: The Completed Classified Buildings Map

GIS in Fiber-Optic Network Design

Once a community's telecommunications needs are well understood, it may be deemed necessary to expand the existing infrastructure to support next-generation services. The task of designing and planning the layout of a telecommunications network may fall to existing telecommunications providers, third-party consultants, or the community itself. In any case, GIS offers the network planner a powerful solution for simultaneously managing the physical location of the infrastructure, the attribute data associated with infrastructure components, and the cost of such components in a single environment.

Designing a Fiber-Optic Geodatabase

The essential first step in network planning, and one that can save the most time later on if diligently pursued, is the construction of a flexible database that can capture the relevant pieces of information, allow for rapid design, and enable iterative updates with a minimal amount of effort. Because GIS software associates the attributes of a feature with its representation on the map, and stores this information in an industry-standard relational database format, it has some significant advantages over CAD software for the purposes of preliminary route design– though CAD has advantages of its own that should be kept in mind, particularly in more detailed engineering applications.

ESRI ArcGIS, the software platform we will be focusing on in this guide, developed a data storage format called the *geodatabase* in versions 8.0 and higher. In the older, perhaps more familiar, *shapefile* format, the map representation of a each geographic feature (its geometry) was stored in a .shp file and associated with a record in a separate DBASE (.dbf) file. Because the DBASE format has some technical limitations (e.g., flat-file table structure, limitations on the length of field names, etc.), ESRI adopted the relational database format for the storage of its attribute data, and incorporated the geometry into the database itself as a binary object. This allows the user much more flexibility, and enables advanced features such as subtyping, the use of attribute domains for validation, and the modeling of relationships between feature classes, to name a few.

There are two types of geodatabases. *Personal geodatabases*, which are stored in the Microsoft Access format, are intended for use on a single machine for datasets with under 250,000 features. For more demanding applications (i.e., large datasets or multi-user environments), the *enterprise geodatabase* format is more appropriate. Enterprise geodatabases allow data to be stored in a variety of industry-standard relational database formats, such as Oracle, through the use of an add-on called ArcSDE (Spatial Database Engine), A full description of the features of geodatabases is beyond the scope of this guide, but more information may be found in the ArcGIS software documentation or online at http://www.esri.com/software/arcgis/geodatabase/index.html.

First, we will present a brief overview of some technical design issues relating to fiber-optic network construction that guide how a fiber-optic network geodatabase should be structured. Then, we will discuss the process for building a geodatabase, including aspects such as its field structure, validation rules, and topology rules.

Background: Design Considerations for Fiber-Optic Networks

The Virginia Tech eCorridors report "Strategic Technology Infrastructure for Regional Competitiveness in the Network Economy", volume 4, offers an excellent and thorough discussion of the issues associated with fiber-optic network planning and design. This report forms the basis for much of what follows in this section of the guide. It may be downloaded from
http://www.ecorridors.vt.edu/research/papers/stircne/index_flash.shtml#download.

It is beyond the scope of this guide to address all the design considerations for fiber-optic network construction. For the purposes of the current discussion, we will focus primarily on those which relate to high-level network architecture and fiber route layout.

The fiber-optic network architecture we will be modeling in our geodatabase is similar to the example below:

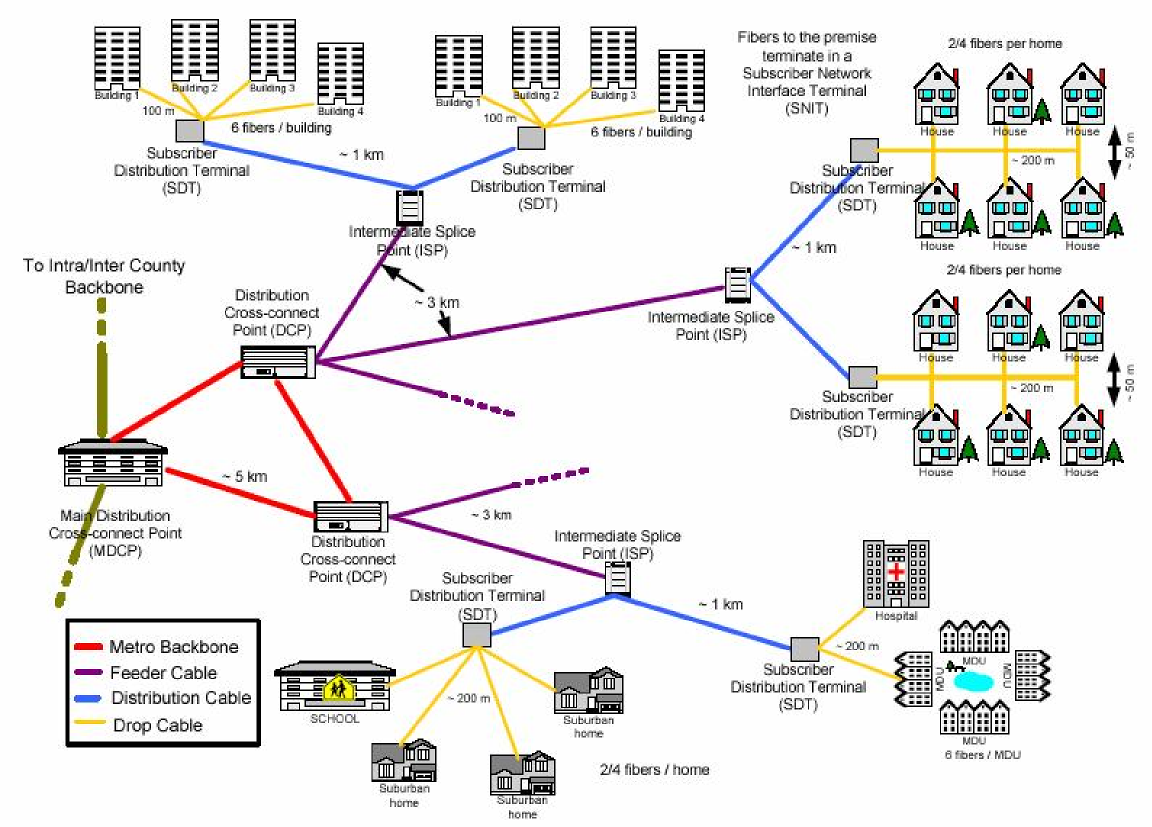
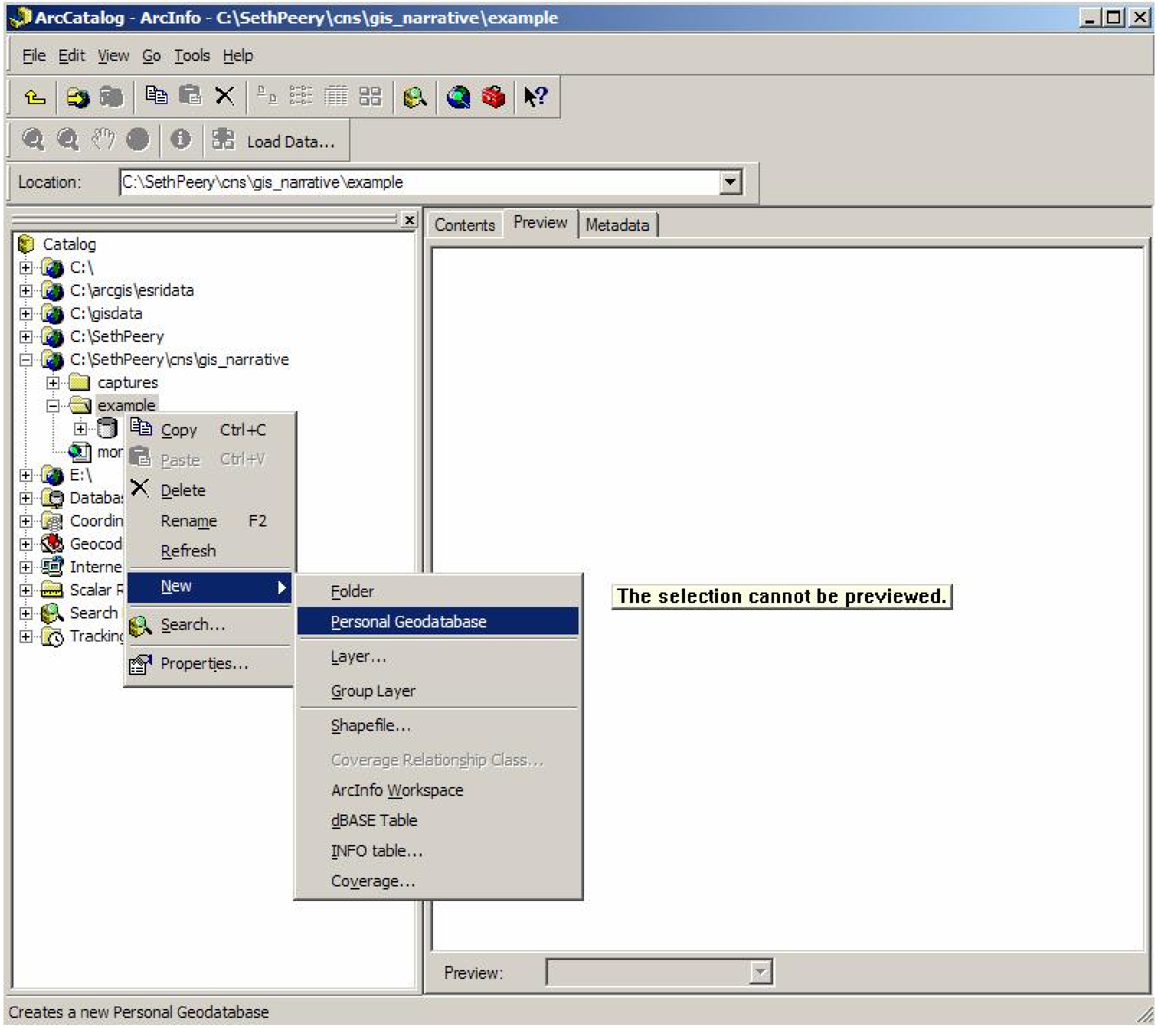

Building 1
Building 2
Building 3
Building 4
100 m
6 fibers / building
Subscriber Distribution Terminal (SDT)
~ 1 km
Building 1
Building 2
Building 3
Building 4
100 m
6 fibers / building
Subscriber Distribution Terminal (SDT)
Fibers to the premise terminate in a Subscriber Network Interface Terminal (SNIT)
2/4 fibers per home
House
House
House
Subscriber Distribution Terminal (SDT)
~ 200 m
~ 50 m
House
House
House
Intermediate Splice Point (ISP)
To Intra/Inter County Backbone
Distribution Cross-connect Point (DCP)
~ 3 km
~ 1 km
Intermediate Splice Point (ISP)
2/4 fibers per home
House
House
House
Subscriber Distribution Terminal (SDT)
~ 200 m
~ 50 m
House
House
House
~ 5 km
Main Distribution Cross-connect Point (MDCP)
Distribution Cross-connect Point (DCP)
~ 3 km
Intermediate Splice Point (ISP)
Subscriber Distribution Terminal (SDT)
~ 1 km
Hospital
~ 200 m
Subscriber Distribution Terminal (SDT)
MDU
MDU
MDU
MDU
6 fibers / MDU
SCHOOL
~ 200 m
Suburban home
Suburban home
Suburban home
2/4 fibers / home
Metro Backbone
Feeder Cable
Distribution Cable
Drop Cable

Figure 13: Generic Metro and Local Access Networks[8]

[8] Source: "Strategic Technology Infrastructure for Regional Competitiveness in the Network Economy", volume 4: Fiber-Optic Infrastructure Design Guide, p. 10.

Building the Database Structure

The first step in designing a fiber-optic geodatabase is to create the appropriate files. All data design and modeling tasks are carried out within the ArcCatalog application of ArcGIS. First, create a new folder to house the fiber-optic data. Right-click on that folder in the catalog tree and select New à Personal Geodatabase:



Location: C:\SethPeery\cns\gis_narrative\example

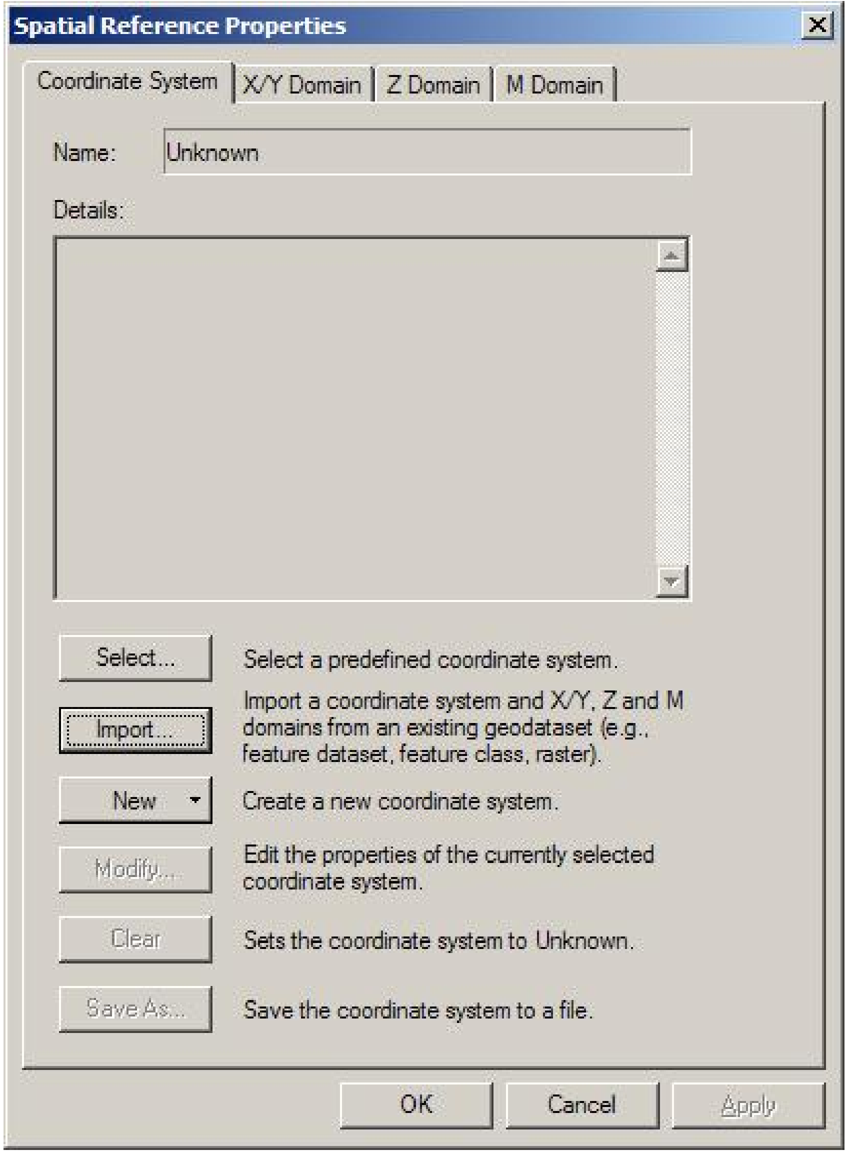
The selection cannot be previewed.

Creates a new Personal Geodatabase

Figure 14: Creating a New Personal Geodatabase

Next, we need to create a "Feature Dataset" within the geodatabase to contain all our feature classes. Creating a feature dataset simply specifies a spatial reference (projection) that is automatically assigned to the files within it. This is necessary to ensure that all the feature classes in the geodatabase overlay properly. In addition, geometric networks, which we will need to create later on, require that all participating feature classes be contained in the same feature dataset to ensure proper topological associations are made.

To do this in ArcCatalog, right-click on the newly created geodatabase and select New **à** Feature Dataset from the menu that appears. You will be prompted to enter a name for the feature dataset, and assign it a coordinate system. Pressing the “Edit” button on this dialog results in the following screen:



Name: Unknown

Select a predefined coordinate system.

Import a coordinate system and X/Y, Z and M domains from an existing geodataset (e.g., feature dataset, feature class, raster).

Create a new coordinate system.

Edit the properties of the currently selected coordinate system.

Sets the coordinate system to Unknown.

Save the coordinate system to a file.

Figure 15: Setting a spatial reference for a feature dataset

A few general guidelines should be observed when setting the spatial reference for a feature dataset that will be storing fiber-optic network features.

First, DO NOT SET THE SPATIAL REFERENCE TO A GEOGRAPHIC COORDINATE SYSTEM. Since we will be using the geodatabase to measure the distances associated with fiber-optic cable routes, the data must be stored in a projected coordinate system. Projected coordinate systems use linear measurement units, such as feet or meters – geographic coordinate systems use angular units, such as degrees. Personal database feature classes automatically maintain a field called Shape_Length to store the length of a feature *in map units*. If the units are set to a geographic coordinate system, this field will have no meaning. The State Plane coordinate system for your area, with units of U.S. survey feet, is the best coordinate system to assign for this application.

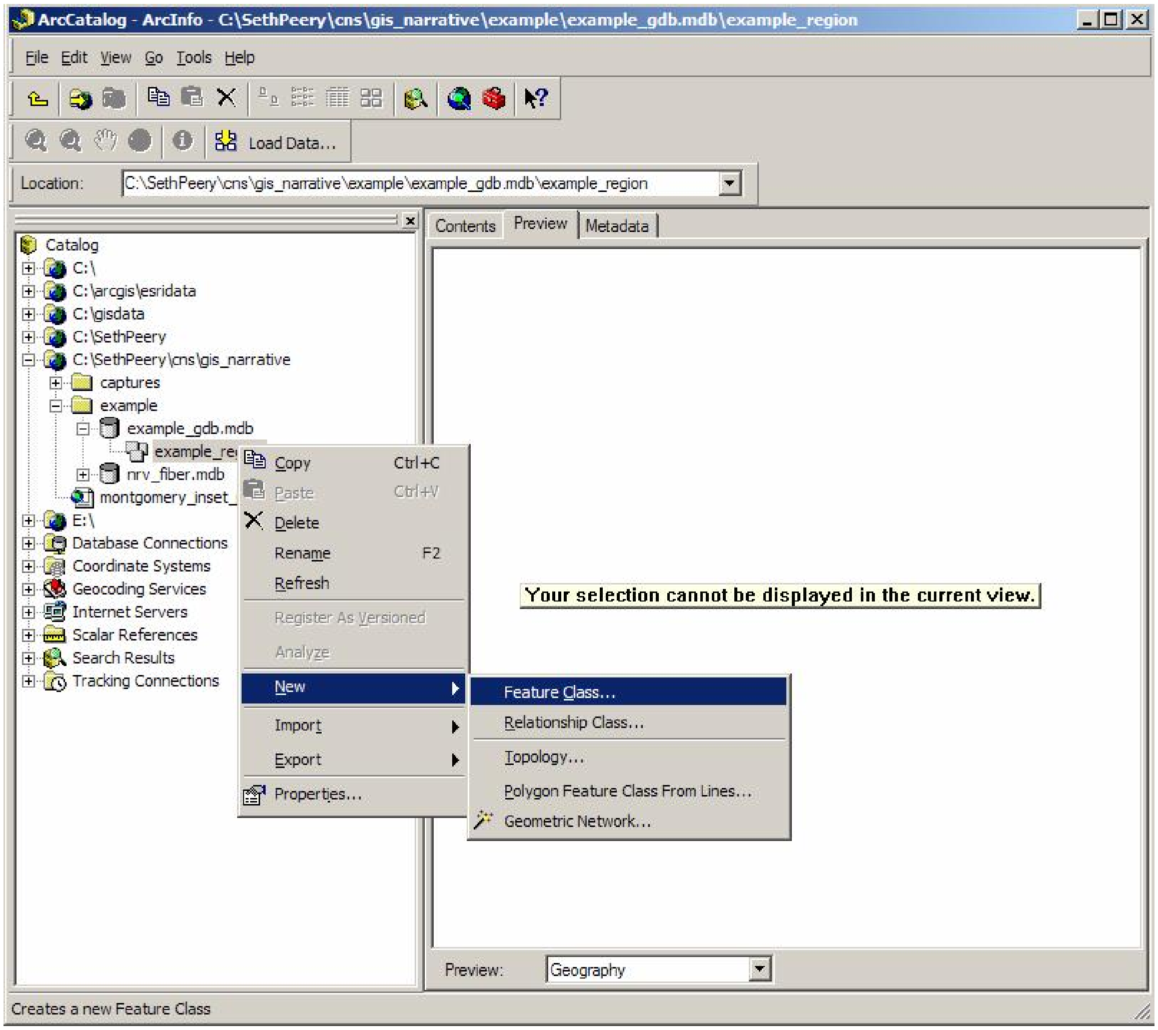
Second, the feature dataset requires that the bounding coordinates of the area of interest be defined, in addition to the coordinate system. It is rather troublesome to enter these manually. A far better alternative is to import the values from an existing file. If done properly, this will solve the coordinate system issue as well. First, determine the area of interest. If, for example, analysis is to take place at the county level, locate or create a shapefile containing only your county's outline. If it is not already in the projected coordinate system you wish to use for the fiber-optic network geodatabase, use the ArcToolbox Project Wizard for shapefiles to change its projection. Then, in the feature dataset spatial reference screen shown in Figure 15, select "Import". Navigate to the boundary shapefile and confirm that the correct coordinate system and X/Y/Z/M domain has been automatically inserted.

Once the personal geodatabase and feature dataset have been created, we are now ready to create feature classes to represent the relevant components of a fiber-optic network.

Defining the Information to be Stored

From the example in Figure 13, we can see that there are two fundamental types of geographic features that need to be modeled: Fiber-optic Cable Routes and Network Nodes. To accomplish this, we first need to create a feature class (analogous to a shapefile, but with enhanced features) for Fiber-optic Cable Routes and a feature class for Network Nodes.

We will begin by creating the Fiber feature class. In ArcCatalog, right click on the newly created feature dataset and select New **à** Feature Class:



ArcCatalog - ArcInfo - C:\SethPeery\cns\gis_narrative\example\example_gdb.mdb\example_region

Location: C:\SethPeery\cns\gis_narrative\example\example_gdb.mdb\example_region

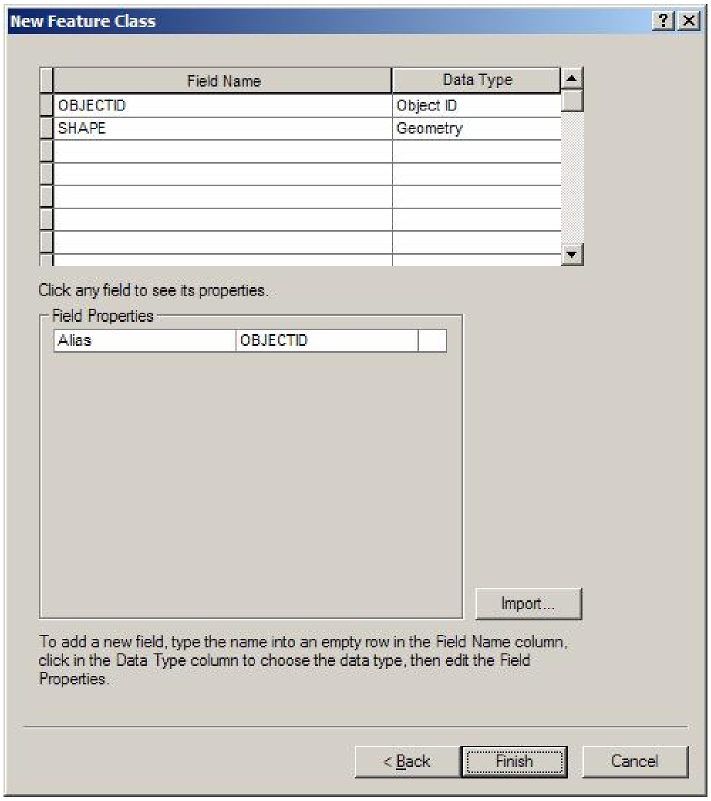
Your selection cannot be displayed in the current view.

Preview: Geography

Creates a new Feature Class

Figure 16: Creating a New Feature Class

In the wizard that pops up, you are asked to provide a name and alias name (optional) for the new feature class. Name the feature class Fiber. Go to the next screen and accept all default options until you get to the third screen, in which you are asked to define the fields for the feature class:



Field Name	Data Type
OBJECTID	Object ID
SHAPE	Geometry

Click any field to see its properties.

Field Properties

Alias	OBJECTID

To add a new field, type the name into an empty row in the Field Name column, click in the Data Type column to choose the data type, then edit the Field Properties.

Figure 17: Adding fields to a new feature class

At this point it's useful to think about what pieces of information we'd like to keep track of for each fiber route. The table below contains some suggested fields that we have used in past projects. Many of these will be referenced in subsequent sections of the guide. You may have others that are specific to your project, and they should by all means be included.

Field Name	Data Type	Field Properties	Purpose/Description
OBJECTID	Object ID		Created by the system
SHAPE	Geometry	Type=Line	Created by the system
Endpoint1	Text	Length=50	The network node this fiber route runs from
Endpoint2	Text	Length=50	The network node this fiber route runs to
Fiber_Type	Text	Length=50	Fiber type (e.g., single-mode, multi-mode)
Placement	Text	Length=50	Placement (e.g., aerial, direct buried)
Fiber_Count	Short Integer		Fiber count (e.g., 48, 60, 72, 144)
Owner	Text	Length=50	Who owns this fiber
Network_Layer	Long Integer		Layer this fiber belongs to (e.g., Backbone, Distribution, etc.)
Miles	Float	Precision=0, Scale=0	Length of the fiber route in miles (will be derived from Shape_Length)
Kilometers	Float	Precision=0, Scale=0	Length in kilometers (will be derived from Shape_Length)
Comments	Text	Length = 250	Comments

Once you've created the fiber feature class, repeat the process for the network nodes feature class. This feature class will represent all the point features in the fiber network, including end-user access points, switching locations, and even cable splices. The fields that should be associated with network nodes include:

Field Name	Data Type	Field Properties	Purpose/Description
OBJECTID	Object ID		Created by the system
SHAPE	Geometry	Type=Point	Created by the system
Name	Text	Length=50	Name of this network node
Cables_In	Short Integer		Number of fiber routes connected to this node
Address	Text	Length=50	Mailing/physical address of this node
Latitude	Text	Length=50	Latitude in decimal degrees
Longitude	Text	Length=50	Longitude in decimal degrees
Comments	Text	Length=250	Comments
JunctionLayer	Short Integer		Type (e.g., distribution switch, end-user access point)
Equipment	Text	Length=50	Equipment to be installed (or already installed) at this node

As a rule, time spent at the creation of the geodatabase thinking about what pieces of information need to be captured, how they need to be represented, what values they will be allowed to assume, and how additional information can be accommodated will yield a more closer approximation of reality and save a significant amount of time down the road.

Subtypes and Attribute Domains

Returning to the example in Figure 13, we can see that there are hierarchies within each of our feature classes, as follows:

Fiber-optic Cable Routes

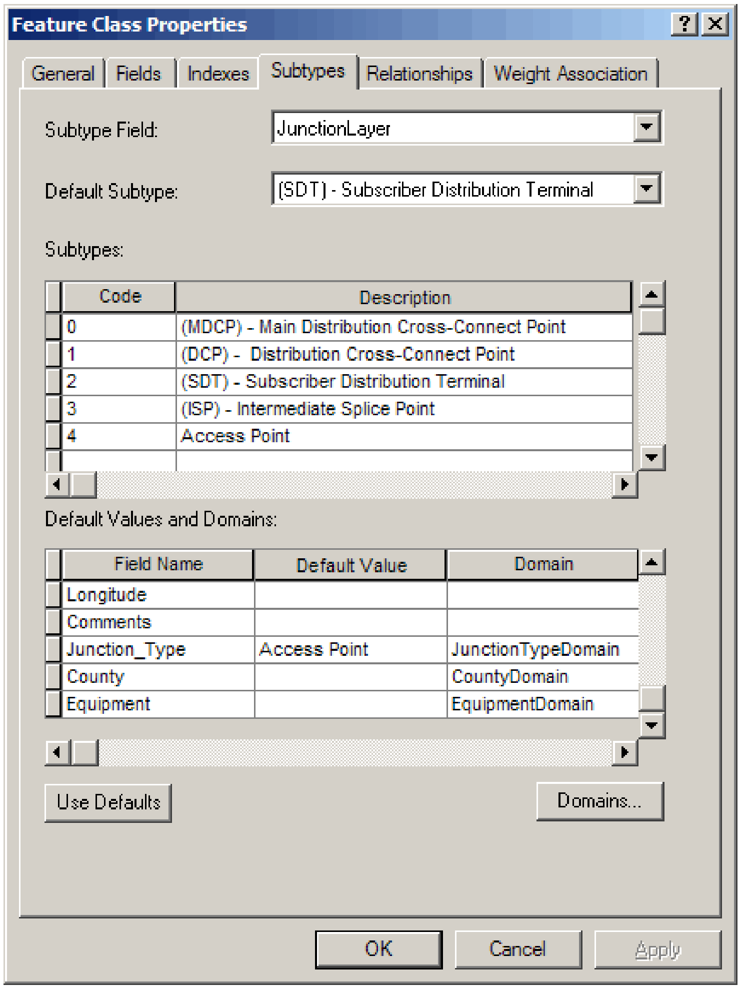
- Inter-County Backbone
    - Intra-County Backbone
        - Distribution
            - Local Access

Network Nodes:

- Main Distribution Cross-Connect Point
    - Distribution Cross-Connect Point
        - Intermediate Splice Point
            - Subscriber Distribution Terminal
                - Subscriber Network Interface Terminal

In the real world, we know that Inter-County Backbones connect MDCPs to one another, Intra-County Backbones connect MDCPs to DCPs, Distribution Fiber connects DCPs to ISPs or SDTs, and Local Access Fiber connects DCPs and SDTs to SNITs. We can model this behavior in the geodatabase by creating subtypes, building a geometric network and assigning different connectivity rules to each subtype. This section will deal with the first step of this process.

Creating subtypes is relatively easy. The first thing that is needed is an integer field to store the subtype code. In ArcCatalog, re-open the Properties dialog for the Network Nodes feature class. Click on the "Subtypes" tab. Select JunctionLayer as the subtype field, and fill in the codes and descriptions as follows.



Subtype Field: JunctionLayer

Default Subtype: (SDT) - Subscriber Distribution Terminal

Subtypes:

Code	Description
0	(MDCP) - Main Distribution Cross-Connect Point
1	(DCP) - Distribution Cross-Connect Point
2	(SDT) - Subscriber Distribution Terminal
3	(ISP) - Intermediate Splice Point
4	Access Point

Default Values and Domains:

Field Name	Default Value	Domain
Longitude		
Comments		
Junction_Type	Access Point	JunctionTypeDomain
County		CountyDomain
Equipment		EquipmentDomain

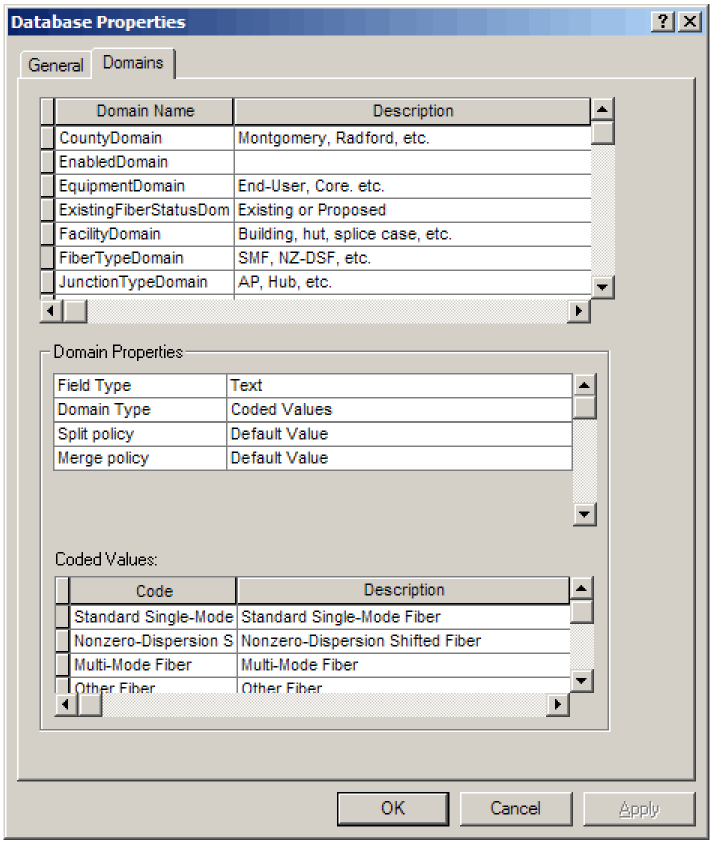
Figure 18: Creating subtypes for the Network Nodes feature class

Note that subtypes are stored in the attribute table as coded values. While working in the ArcGIS environment (ArcMap or ArcCatalog), the full descriptions will be displayed, but if the data is exported to another program, only the integer codes will be displayed. If it is anticipated that the data will need to be exported frequently, it may be useful to create a second field that stores the subtype name as text.

The subtype creation process can be repeated for the fiber feature class, using the subtypes outlined in the hierarchical bulleted list above.

There are other fields in the geodatabase that will take on a finite set of values, like fiber types and fiber placement methods. However, we don't want to use subtypes in this situation, since a feature class can only have one subtype – and we want to use the subtype field to regulate the connectivity rules we will define later on in the guide when we create a geometric network. In this instance, we use *Attribute Domains* to constrain the values of a field to the members of a pre-defined list. This builds a layer of robustness into the geodatabase by ensuring that all fields have allowable values with no typographical errors – but more importantly, it allows the user to select them from a drop-down list while editing instead of typing them in manually. This speeds up the process of initial network prototyping considerably.

Attribute domains are stored in the geodatabase, and are visible to all its feature classes. To use an attribute domain, the domain must first be defined, and all its allowable values enumerated, in the geodatabase properties dialog. Then, a field in a feature class of the same type as the attribute domain is "bound" to the attribute domain by associating it with the domain in its Field Properties. As with all data modeling and design tasks, this is accomplished within the ArcCatalog application.



Domain Name	Description
CountyDomain	Montgomery, Radford, etc.
EnabledDomain	
EquipmentDomain	End-User, Core. etc.
ExistingFiberStatusDom	Existing or Proposed
FacilityDomain	Building, hut, splice case, etc.
FiberTypeDomain	SMF, NZ-DSF, etc.
JunctionTypeDomain	AP, Hub, etc.

Domain Properties

Field Type	Text
Domain Type	Coded Values
Split policy	Default Value
Merge policy	Default Value

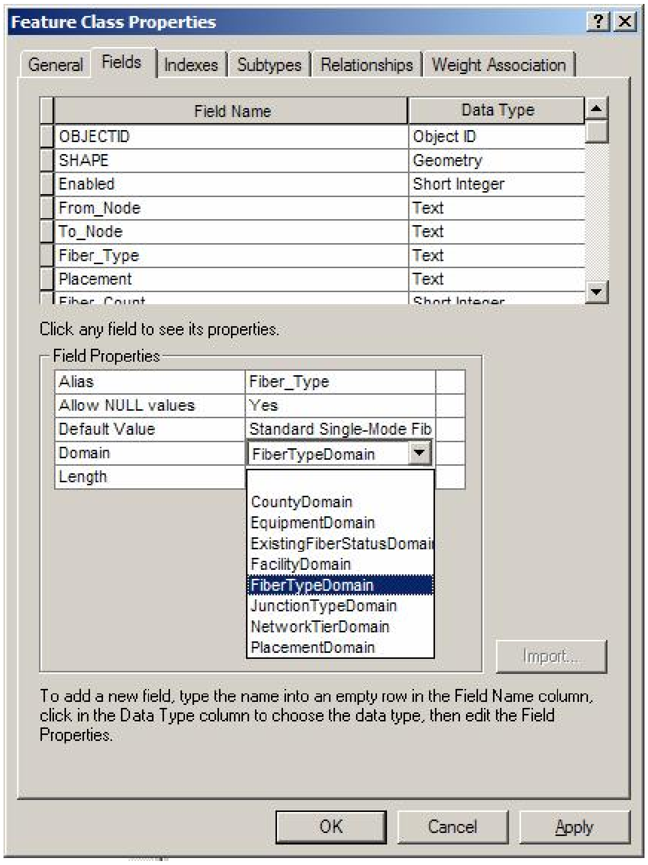
Coded Values:

Code	Description
Standard Single-Mode	Standard Single-Mode Fiber
Nonzero-Dispersion S	Nonzero-Dispersion Shifted Fiber
Multi-Mode Fiber	Multi-Mode Fiber
Other Fiber	Other Fiber

Figure 19: Assigning coded values to the Fiber Type Domain

To create a new attribute domain, enter its name in the next blank list entry in the "Domain Name" column. Optionally, enter a description. In the Domain Properties box, select a Field Type of "Text" and "Coded Values" as the Domain Type. Then, in the Coded Values box that appears at the bottom of the screen, enter the enumerated values of the domain. The Code is what will actually be entered in the attribute table, while the description is what will appear in the ArcGIS environment. For convenience, it is usually best to enter the same string into both fields.

Next, to bind a field in a feature class to the newly created domain, open its Properties dialog and click on the field to be bound. In the Domain entry in the Field Properties box, select the new domain from the drop-down list, as shown below:



Field Name	Data Type
OBJECTID	Object ID
SHAPE	Geometry
Enabled	Short Integer
From_Node	Text
To_Node	Text
Fiber_Type	Text
Placement	Text
Fiber_Count	Short Integer

Click any field to see its properties.

Field Properties

Alias	Fiber_Type
Allow NULL values	Yes
Default Value	Standard Single-Mode Fib
Domain	FiberTypeDomain
Length	

CountyDomain
EquipmentDomain
ExistingFiberStatusDomai
FacilityDomain
FiberTypeDomain
JunctionTypeDomain
NetworkTierDomain
PlacementDomain

To add a new field, type the name into an empty row in the Field Name column, click in the Data Type column to choose the data type, then edit the Field Properties.

Figure 20: Binding a field to an attribute domain

This process can be repeated for any field whose values are to be restricted to an enumerated set of values, and the same domain can be re-used for multiple feature classes in the same geodatabase.

Geometric Networks

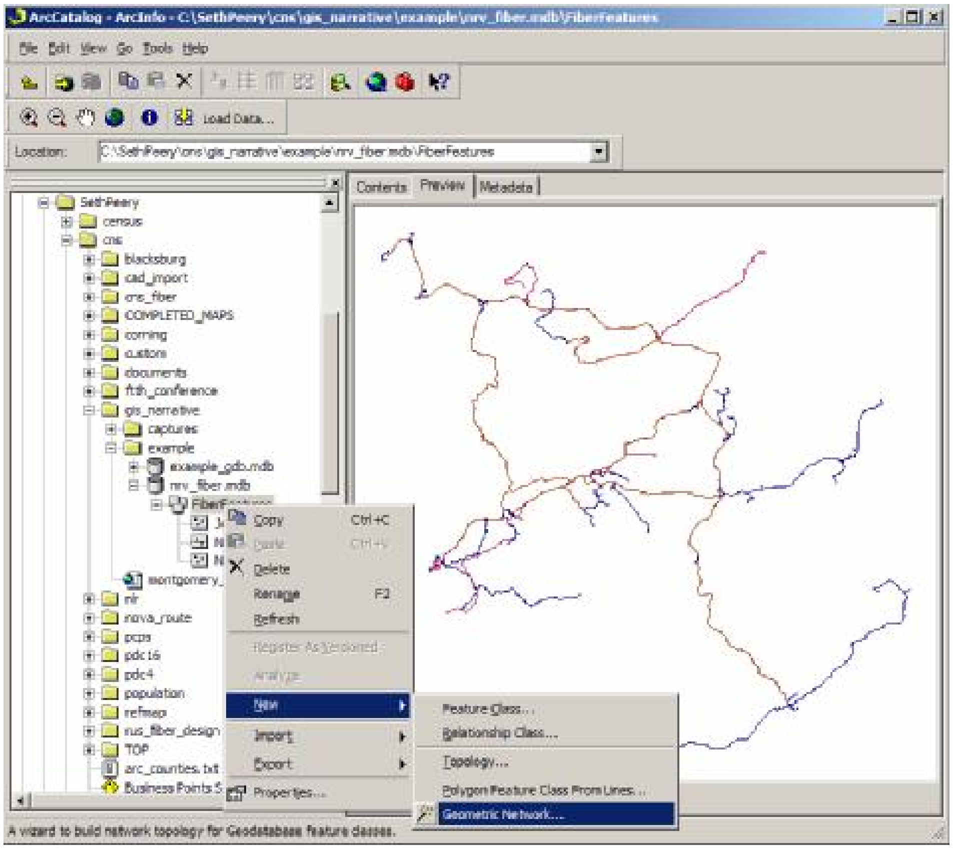
As we discussed earlier, the reason for creating subtypes of our fiber route and network node feature classes was to enable the modeling of different types of connectivity behavior. (e.g., Intra-county backbone connects MDCPs to DCPs). The geodatabase model allows us to do this using *Geometric Networks*. A geometric network is a collection of linear and point features ("edges" and junctions") that share topological relationships. Once a geometric network has been built, thus defining the relationships between the feature classes, ArcGIS maintains and enforces the relationships automatically. Furthermore, a feature class's participation in a geometric network imparts extra "intelligence" – in that we can look up the features to which it is connected, find paths through connected features, and perform other types of network analysis.

We will create a geometric network for our fiber-optic geodatabase to accomplish several purposes:

- Enforcing the topology rule that fiber routes always connect named endpoints
- Keeping track of which endpoints a fiber route connects
- Keeping track of how many fiber routes intersect a given endpoint
- Modeling the hierarchical relationships between routes and endpoints – e.g., backbone cables connect major switching/routing locations

Geometric networks are created in ArcCatalog. Only geodatabase feature classes (not shapefiles) can participate in a geometric network. Only feature classes that are contained in the same feature dataset can participate in a geometric network – which is one of the reasons we built a feature dataset in an earlier step.

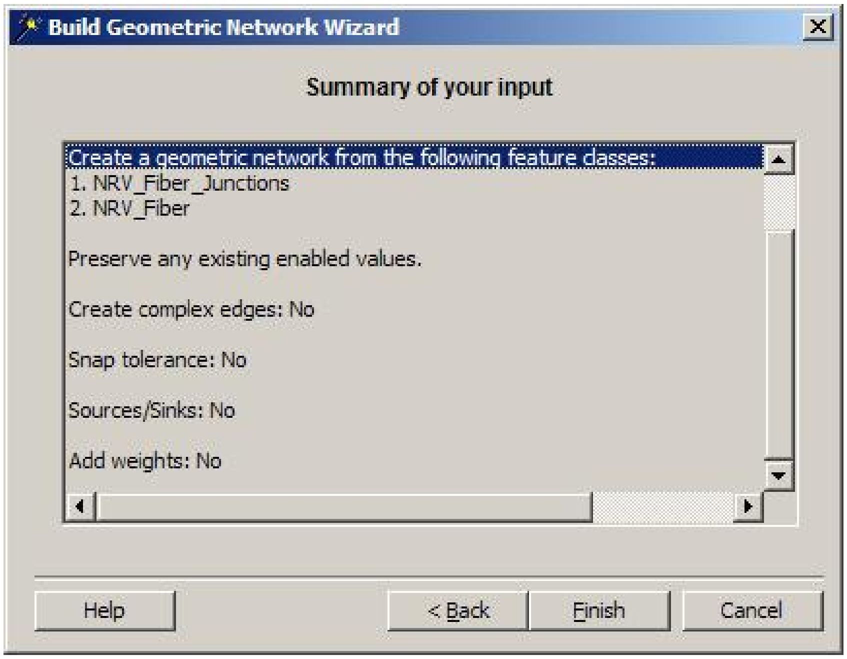
To create a geometric network, right-click on the Feature Dataset and select New **à** Geometric Network, as shown below:



A wizard to build network topology for Geodatabase feature classes.

Figure 21: Create a new geometric network

Click “Next” in the first dialog of the wizard that opens. Since we have already created feature classes for fiber routes and endpoints, we will “Build a Geometric Network from Existing Features”. Check the boxes beside the two feature classes you created, and click Next. Since the feature classes are empty, we can accept the default option in the dialog that follows: “Preserve Existing attribute values…”. Because we are modeling a point-to-point topology in which a fiber route has exactly two endpoints, we Select “No” when prompted to create complex edges in the network. Since there are no existing features in the feature classes, we can select “No” when asked if the features need to be snapped. Since we are not modeling a network with one-way flow (data flows across fiber networks in both directions simultaneously), we Select “No” when asked if the network contains sources or sinks. Since the path a data packet will take through a routed data network is not always known, we select “No” when asked if we want to assign weights to the network. A “Summary of your input” box is displayed, and should look something like the example below:



Create a geometric network from the following feature classes:
1. NRV_Fiber_Junctions
2. NRV_Fiber

Preserve any existing enabled values.

Create complex edges: No

Snap tolerance: No

Sources/Sinks: No

Add weights: No

Figure 22: Summary of input for network build wizard

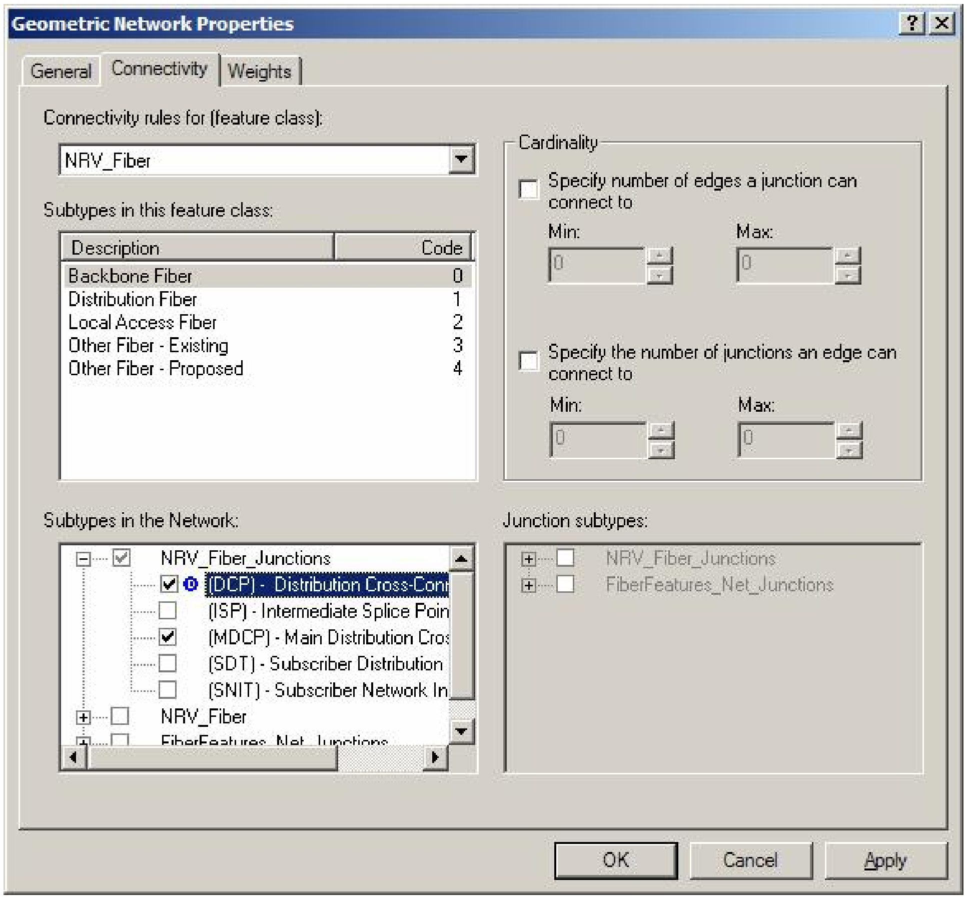
Click "Finish" to create the network. Going back to the catalog tree, you will notice that two new objects have been created in your feature dataset:

• FiberFeatures_Net	(a geometric network)
• FiberFeatures_Net_Junctions	(a point feature class)

The names may be different depending on the names you selected, the name of your feature dataset, etc. The geometric network object exists to store the topology relationships between its members. We will not be using the geometric network object in mapping applications – the only time it will be directly manipulated is in the modification of its properties, such as connectivity rules.

The point feature class that was automatically created is designed to store "Orphan Junctions" – junctions which are automatically created at the end of an edge when no "real" junction feature exists or is created at that location. Orphan junctions do not have the attributes of the junction feature class we created earlier, and the orphan junction file will be deleted if the network is deleted. So, it's best to ensure that few, if any, orphan junctions are created – and are dealt with appropriately as soon as possible.

To more accurately model the behavior of fiber networks in the real world, and to minimize the number of orphan junctions that are created in the editing process, we want to define *connectivity rules* that govern which subtypes of the fiber routes layer can connect to which subtypes of the junctions layer. The graphic below provides an example of how we might define the connectivity rules for the backbone fiber subtype:



Connectivity rules for (feature class):

NRV_Fiber

Subtypes in this feature class:

Description	Code
Backbone Fiber	0
Distribution Fiber	1
Local Access Fiber	2
Other Fiber - Existing	3
Other Fiber - Proposed	4

Cardinality

□ Specify number of edges a junction can connect to

Min: 0 Max: 0

□ Specify the number of junctions an edge can connect to

Min: 0 Max: 0

Subtypes in the Network:

- ☑ NRV_Fiber_Junctions
  - ☑ (DCP) - Distribution Cross-Conr
  - □ (ISP) - Intermediate Splice Poin
  - ☑ (MDCP) - Main Distribution Cros
  - □ (SDT) - Subscriber Distribution
  - □ (SNIT) - Subscriber Network In
- □ NRV_Fiber
- □ FiberFeatures_Net_Junctions

Junction subtypes:

- □ NRV_Fiber_Junctions
- □ FiberFeatures_Net_Junctions

Figure 23: Defining Connectivity Rules

Here, we have instructed the geometric network to allow connectivity between the edges in NRV_Fiber that are coded as backbone fiber (subtype code 0) and the junctions in NRV_Fiber_Junctions coded as DCP or MDCP. This implies that fiber backbones cannot connect directly to intermediate splice points or subscriber distribution terminals, which is a reasonable assumption – any site that has a backbone running into it is probably going to be a DCP. Note that the converse of this rule is not necessarily true – by defining this rule, we have not restricted any of the other fiber types' ability to connect to a DCP in the network. The next step, then, will be to define connectivity rules for each of the other fiber subtypes (e.g., that Local Access fiber connects SDTs to SNITs).

We can also define "default" junctions to be created if an edge does not terminate at an existing junction. By right-clicking on a junction subtype that is permitted to connect to a particular fiber subtype, the option appears to "Set as Default". When selected, a blue circle with a 'D' in it appears (see above). It is useful to set default junction subtypes, but if they exhibit undesired behavior later on (e.g., overwriting existing features), they can be disabled.

If connectivity rules are not defined for a geometric network, orphan junctions will be created at each end of each junction feature when it is digitized. To avoid this result in situations where detailed connectivity behaviors are not needed, allow each subtype of the edge feature class to connect to each subtype of the junction feature class.

Automation of Attribute Maintenance

In an earlier section, we defined a considerable number of attributes to be associated with our fiber and access point feature classes. It would be extremely time-consuming (as well as a potential source of error) to manually enter and keep track of these attribute values for each feature as the network grows in complexity. Fortunately, the GIS software has the internal ability to calculate many of these attributes automatically. However, it requires a bit of customization to get the system to perform such calculations and put the result where we want it.

The ArcGIS software system is built from a set of reusable software components called ArcObjects, which are based on Microsoft's Component Object Model (COM). By accessing and manipulating ArcObjects in custom applications and macros, users can leverage the existing application framework and extend its functionality to suit their needs. It is far, far, beyond the scope of this guide to give a thorough treatment of ArcObjects. Refer to the "Suggestions for Further Reading" for some excellent references on ArcObjects programming.

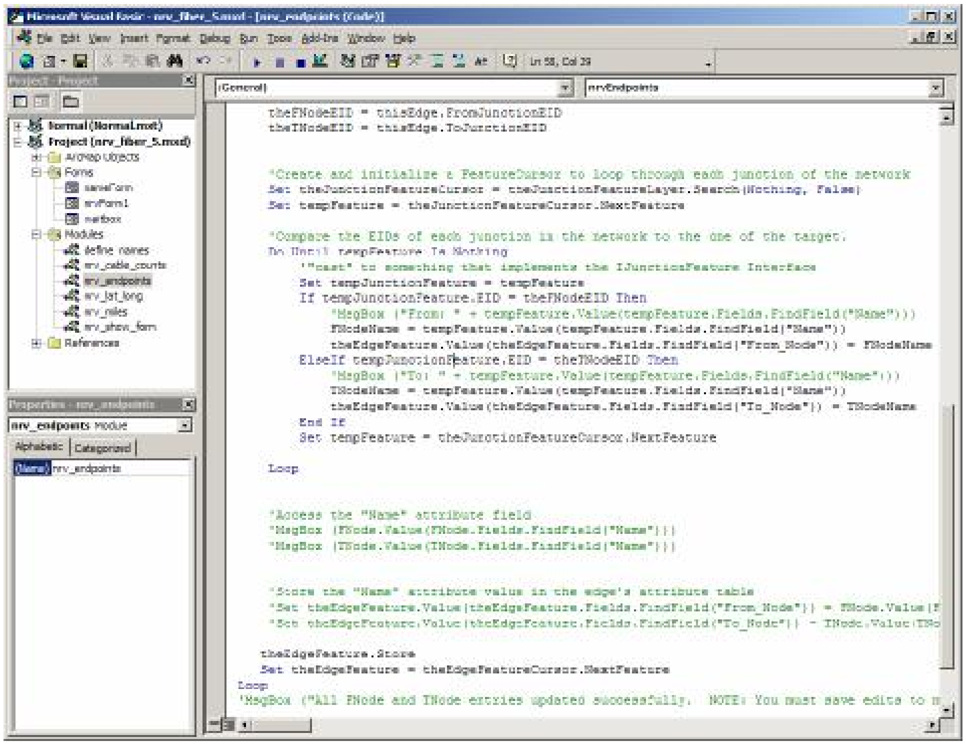
For the purposes of this guide, we can provide a few examples of how ArcObjects might be used to automate the task of keeping track of our attributes. First of all, we want to keep track of the latitude and longitude of each network access/switching point. Latitude and longitude measurements are in decimal degrees, but our map units are in U.S. survey feet. So we need a script that can read a point feature's location in map units, convert those units to decimal degrees, and write the output to that point feature's entry in the attribute table. As it turns out, this is not all that hard to do; see Appendix A for the solution.

Another task is to convert the Shape_Length field into other units besides its native U.S. survey feet. This is really just a convenience – we could certainly convert the values in Excel or some other program when we needed to do distance calculations. It's really nothing more than a field calculator operation, but since we're writing VBA modules already, we have something of an economy of scale.

A third task specific to our current application is to keep track of how many fiber routes connect to a particular junction or node, and store the current value in the attribute table. There is no specific built-in command that will do this for us. However, a ISimpleJunctionFeature (the data type for a network Junction) has the information we need as a built-in attribute called *edgeFeatureCount*. By creating a variable of type ISimpleJunctionFeature, accessing its *edgeFeatureCount* attribute, and writing the result to the attribute table, our problem is solved. See the Appendix for all the details.

The most complex attribute maintenance task is one that would seem at first to be quite simple – keeping track of which two network junctions a fiber route connects. Surprisingly, there is no built-in function to do this. Conceptually, the problem can be solved by accessing the *fromJunctionFeature* or *fromJunctionEID* attribute of an IEdgeFeature, accessing the "Name'' attribute of the feature that is returned (directly, in the former case, and indirectly, after iterating through the junction attribute table to find the matching feature in the latter), and writing the output to the attribute table of the edge feature class. The solution that uses EIDs can be found in Appendix A.

To use all these and any other bits of custom code within ArcMap, it is necessary to tell the system how it may find them. In most cases, custom programs are stored as *Modules* within the .mxd file that contains your map, although they may also be stored in modules within the Normal.mxt template – thereby making them visible to ALL maps you create. To create a custom module in ArcMap, go to Tools à Macros à Visual Basic Editor.

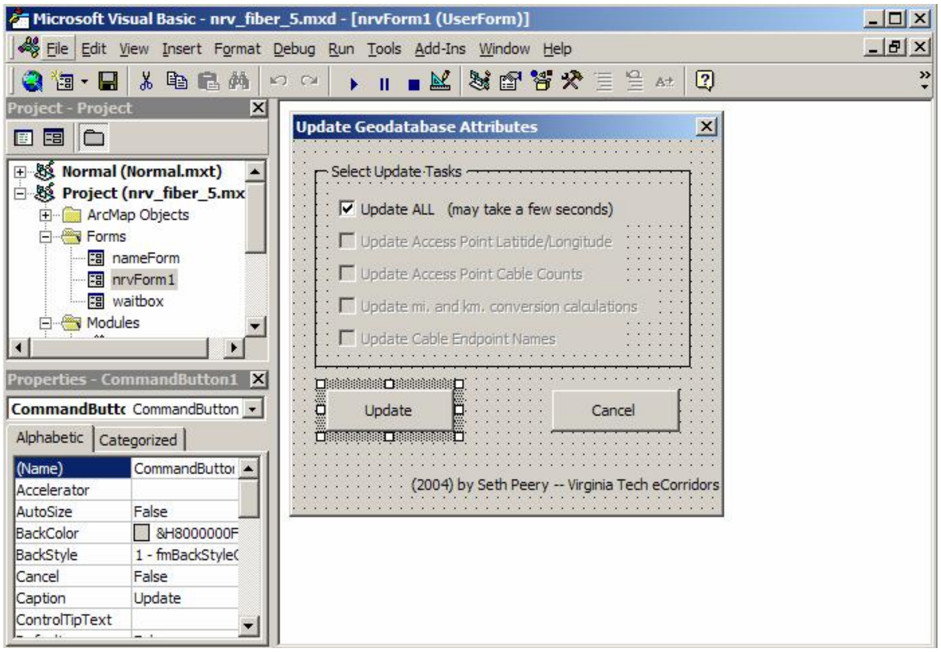


```
        theFNodeEID = thisEdge.FromJunctionEID
        theTNodeEID = thisEdge.ToJunctionEID


        'Create and initialize a FeatureCursor to loop through each junction of the network
        Set theJunctionFeatureCursor = theJunctionFeatureLayer.Search(Nothing, False)
        Set tempFeature = theJunctionFeatureCursor.NextFeature

        'Compare the EIDs of each junction in the network to the one of the target.
        Do Until tempFeature Is Nothing
            '"cast" to something that implements the IJunctionFeature Interface
            Set tempJunctionFeature = tempFeature
            If tempJunctionFeature.EID = theFNodeEID Then
                'MsgBox ("From: " + tempFeature.Value(tempFeature.Fields.FindField("Name")))
                FNodeName = tempFeature.Value(tempFeature.Fields.FindField("Name"))
                theEdgeFeature.Value(theEdgeFeature.Fields.FindField("From_Node")) = FNodeName
            ElseIf tempJunctionFeature.EID = theTNodeEID Then
                'MsgBox ("To: " + tempFeature.Value(tempFeature.Fields.FindField("Name")))
                TNodeName = tempFeature.Value(tempFeature.Fields.FindField("Name"))
                theEdgeFeature.Value(theEdgeFeature.Fields.FindField("To_Node")) = TNodeName
            End If
            Set tempFeature = theJunctionFeatureCursor.NextFeature

        Loop


        'Access the "Name" attribute field
        'MsgBox (FNode.Value(FNode.Fields.FindField("Name")))
        'MsgBox (TNode.Value(TNode.Fields.FindField("Name")))


        'Store the "Name" attribute value in the edge's attribute table
        'Set theEdgeFeature.Value(theEdgeFeature.Fields.FindField("From_Node")) = FNode.Value(F
        'Set theEdgeFeature.Value(theEdgeFeature.Fields.FindField("To_Node")) = TNode.Value(TNo

       theEdgeFeature.Store
       Set theEdgeFeature = theEdgeFeatureCursor.NextFeature
    Loop
    'MsgBox ("All FNode and TNode entries updated successfully.  NOTE: You must save edits to m
```

Figure 24: ArcMap Visual Basic Editor

In the project tree, right click on the root and select Insert à Module. A "Modules" folder will be created, as will a file "Module1". You can type or paste your program in there, and then run it using the "Run Sub/Userform" button (it looks like a "Play" button on a stereo.) You can also create a button in the interface using Tools à Customize à Commands à Macros. Simply drag the macro name to one of the standard toolbars, or create your own.

It is often helpful to have the ability to enter certain input parameters when a module executes. Or, you may have several modules you want to execute at once. In these cases, it's helpful to create *UserForms* – windows that appear for you to interact with in some way. For the fiber network, it was useful to create a form that allows the user to choose whether to run all the modules described on the previous page, or only some of them. To create a new UserForm, right click on the project tree and select Insert à Userform. A "Forms" folder will be created, as will a blank Userform ("UserForm1). You can then use the Toolbox to insert objects on the form (buttons, text boxes, etc.) and alter their properties and the code associated with them. Below is an example of the use of the VBA Editor to create a simple form:



(Name)	CommandButtoı
Accelerator	
AutoSize	False
BackColor	&H8000000F
BackStyle	1 - fmBackStyle(
Cancel	False
Caption	Update
ControlTipText	

Update Geodatabase Attributes

Select Update Tasks

☑ Update ALL (may take a few seconds)

□ Update Access Point Latitide/Longitude

□ Update Access Point Cable Counts

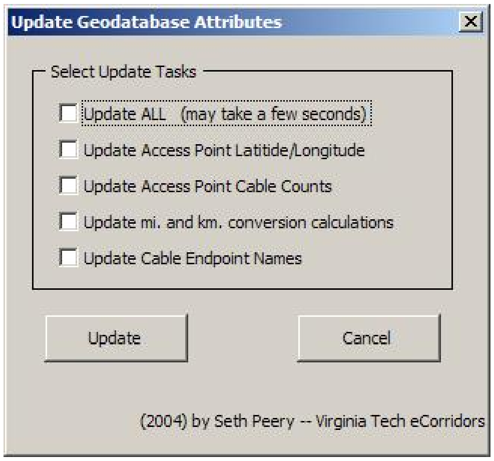
□ Update mi. and km. conversion calculations

□ Update Cable Endpoint Names

Update | Cancel

(2004) by Seth Peery -- Virginia Tech eCorridors

Figure 25: Creating a Simple Userform in VBA to launch ArcObjects modules



Update Geodatabase Attributes

Select Update Tasks

□ Update ALL (may take a few seconds)

□ Update Access Point Latitide/Longitude

□ Update Access Point Cable Counts

□ Update mi. and km. conversion calculations

□ Update Cable Endpoint Names

Update | Cancel

(2004) by Seth Peery -- Virginia Tech eCorridors

Figure 26: VBA UserForm as it appears in ArcMap

If this is your first exposure to ArcObjects programming (or any programming at all, for that matter), the preceding discussion may have been a bit overwhelming. The relevance of ArcObjects to fiber-optic network design is its ability to automate the updating of attribute values, so all you have to do is draw the features. All the information that was automatically inserted into the attribute tables by the custom macros can of course be entered manually, and this step of the process can be foregone entirely.

Editing a Fiber-Optic Geodatabase

At this point, we have progressed over 40 pages into the guide, and it might seem that we have little to show for it. We have, after all, not created any fiber features in our geodatabase. However, this is where things get really, really easy. The purpose of all the foregoing work was to make the editing process much more efficient. By creating a geodatabase, we can keep track of the length of features automatically, and can set up domains that allow us to select values from a drop-down list instead of typing them in. By creating a geometric network, we can automatically create access points when we terminate a fiber segment, and can keep track of which access points are connected by which fiber routes. By writing the macros to update the attributes, we eliminate the need to type in large amount of information as we're building a network. Initial layout of the network, therefore, is very fast – and subsequent changes and updates are easily accommodated.

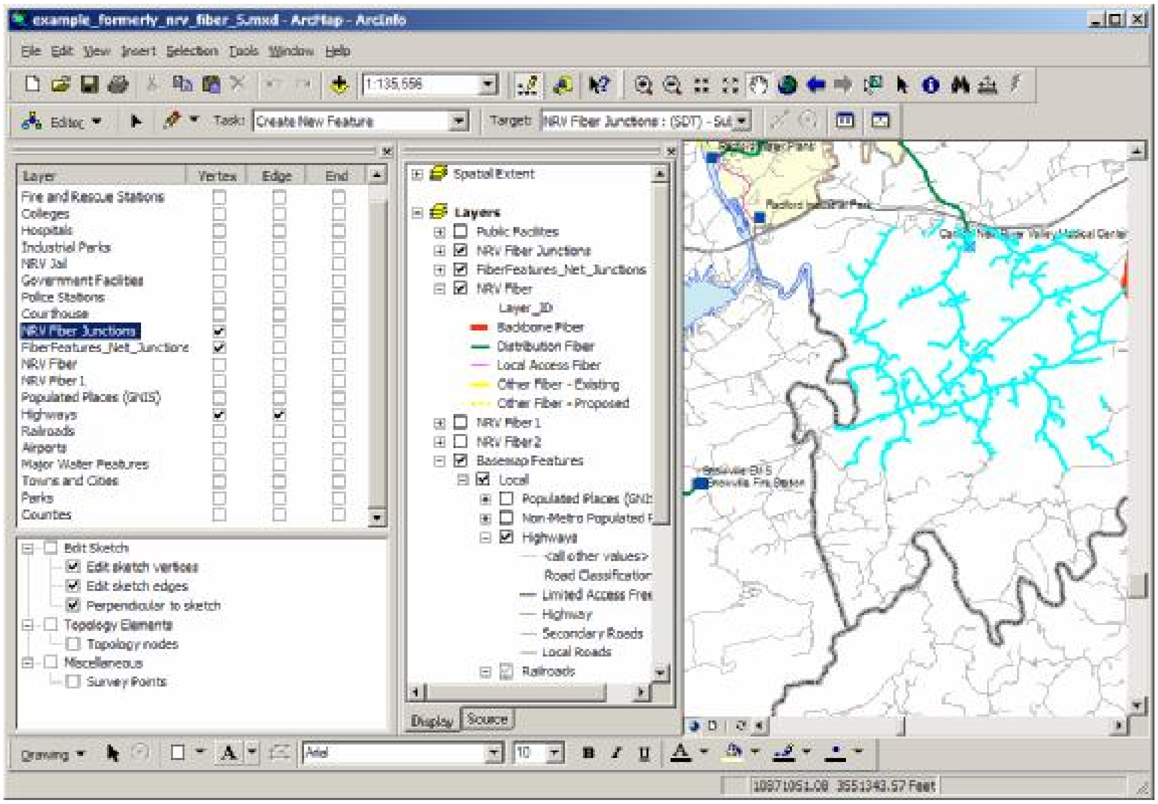
ArcGIS's built-in editing tools are more than adequate for fiber-optic network design. These tools allow you to snap newly created features precisely to existing features, follow the path of existing features (such as roads and rights-of-way), offset new features from existing features by a specified distance, as well as digitize new features freehandedly.

General Procedure for Creating and Editing Geodatabase Features:

1. Open ArcMap.
2. Load the two fiber-optic feature classes (routes and junctions)
3. Load any relevant basemap data (roads, facilities, boundaries, rights-of-way, etc.)
4. Start an Edit session.
    a. First, make sure the Editor toolbar is visible in ArcMap. If it is not, enable it under Tools **à** Customize **à** Toolbars **à** Editor.
    b. From the drop-down menu on the Editor toolbar, select "Start Editing".
    c. When asked which features to make available for editing, select the fiber-optic geodatabase.
5. Using the editing tools, create/move/delete/alter features within the geodatabase.
    a. Define the Snapping Environment and Selectable Layers, if necessary.
    b. Select the Edit task (e.g., "Create Feature", "Modify Feature", etc.)
    c. Select the Target layer and subtype (e.g., Fiber Junctions : (DCP)-Distribution Cross-connect Point)
    d. Perform the desired edits.
6. Run the macros that update the geodatabase attributes (if desired).
7. On the Editor toolbar, select "Save Edits".
8. On the Editor toolbar, select "Stop Editing".

Defining the Snapping Environment

ArcMap can snap features to the exact location of other features, which is very useful when using geometric networks. However, the snapping behavior needs to be carefully controlled. For the purposes of the examples (and, in general, most of the work we'll be doing with the fiber-optic geodatabases), the snapping environment is set to snap to the vertexes of the fiber-optic junction feature class, an the vertexes/edges/ends of the basemap layer in which we want to trace routes. Obviously, other layers can be snapped to – for example, you might have a point feature class that represents subscriber premises, and would want to snap fiber junctions to those locations. An example snapping environment configuration is shown below:



example_formerly_nrv_fiber_5.mxd - ArcMap - ArcInfo

Task: Create New Feature

Target: NRV Fiber Junctions : (SDT) - Sul

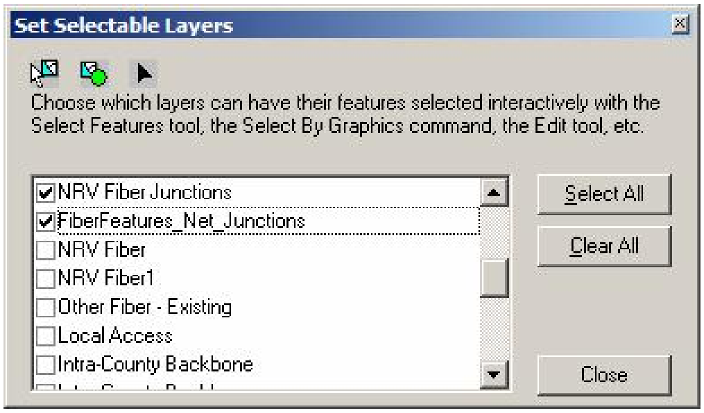
Layer	Vertex	Edge	End
Fire and Rescue Stations	□	□	□
Colleges	□	□	□
Hospitals	□	□	□
Industrial Parks	□	□	□
NRV Jail	□	□	□
Government Facilities	□	□	□
Police Stations	□	□	□
Courthouse	□	□	□
NRV Fiber Junctions	☑	□	□
FiberFeatures_Net_Junctions	☑	□	□
NRV Fiber	□	□	□
NRV Fiber 1	□	□	□
Populated Places (GNIS)	□	□	□
Highways	☑	☑	□
Railroads	□	□	□
Airports	□	□	□
Major Water Features	□	□	□
Towns and Cities	□	□	□
Parks	□	□	□
Counties	□	□	□

- □ Edit Sketch
  - ☑ Edit sketch vertices
  - ☑ Edit sketch edges
  - ☑ Perpendicular to sketch
- □ Topology Elements
  - □ Topology nodes
- □ Miscellaneous
  - □ Survey Points

Figure 27: Setting the Snapping Environment

Defining the Selectable Layers

During the editing process, it may be necessary to select certain features, but not others. For example, when editing the attributes of a fiber-optic network, one generally manually selects individual features, and thus, you'll probably want to allow all the fiber features to be selectable. However, when using the Trace tool, it is usually necessary to disable the selectable property for all layers except the layer to be traced. It is common practice to change the Selectable Layers properties several times within the same edit session. To do this, go to Selection **à** Set Selectable Layers. A window similar to the one below will appear. (Note: by default, all layers are selectable when added to the map.)



Choose which layers can have their features selected interactively with the Select Features tool, the Select By Graphics command, the Edit tool, etc.

☑ NRV Fiber Junctions
☑ FiberFeatures_Net_Junctions
□ NRV Fiber
□ NRV Fiber1
□ Other Fiber - Existing
□ Local Access
□ Intra-County Backbone

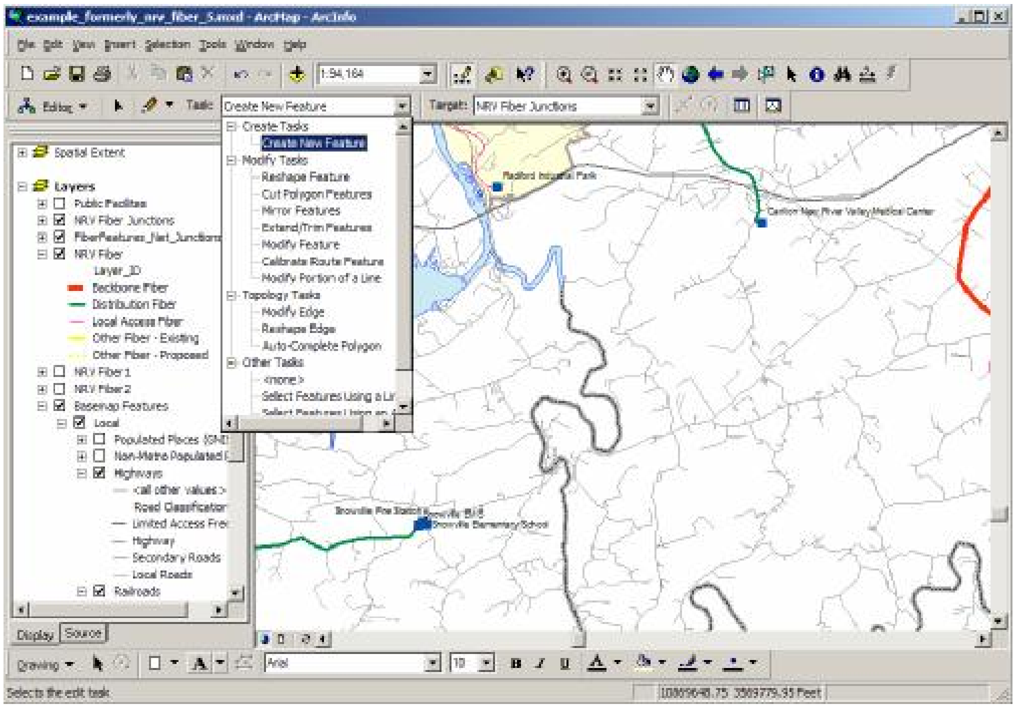
Figure 28: Setting the Selectable Layers

Creating New Fiber Features

Once the snapping environment and selectable layers have been defined, we're ready to start creating features. The time-saving benefits of all the detailed geodatabase definition work are realized in this step of the editing process.

Let's begin by creating a new junction feature. Hypothetically, imagine that this will represent a facility that houses a major backbone switch – or in the language of our conceptual network model, a Main Distribution Cross-connect Point (MDCP). To locate this feature on the map, we could snap it to the location of an existing feature, enter its coordinates from a known latitude and longitude, locate it manually by estimating its relative location to basemap features, etc. For demonstration purposes, we'll just draw it on the map at an arbitrary location.

To do this, we must first define the edit task and set the edit target layer. Both of these are accomplished through the use of drop-down menus on the Editor toolbar. The screen captures below illustrate this process. Note how the subtypes we defined in our geodatabase design process appear in the drop-down menu for Edit Targets – allowing us to specify which subtype of feature we want to create.

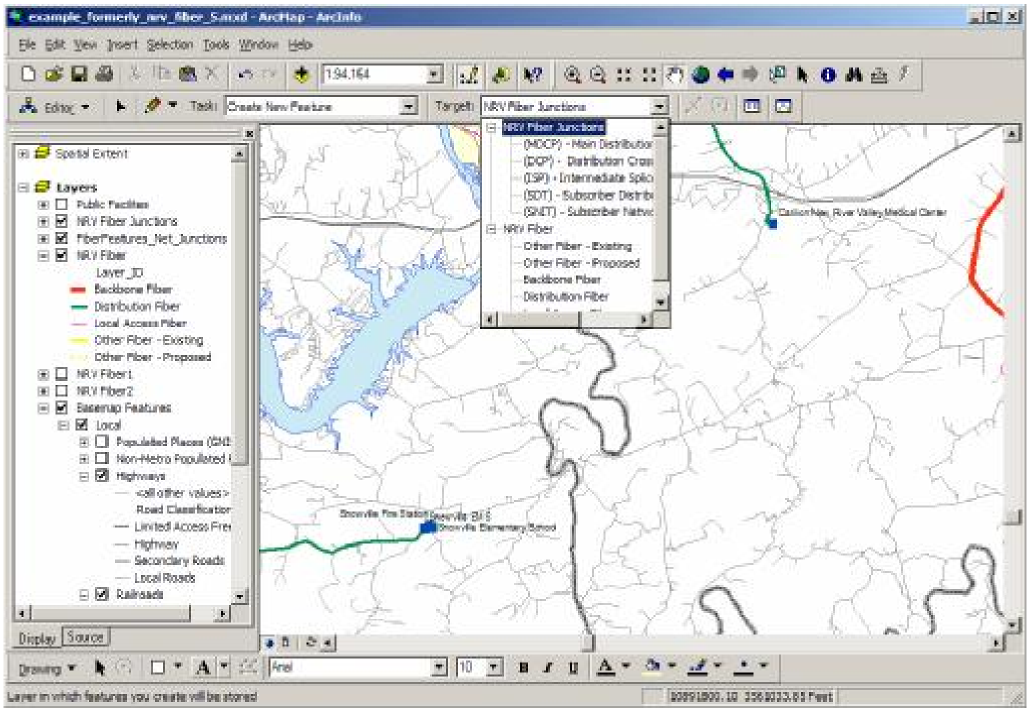


Task: Create New Feature

Target: NRV Fiber Junctions

Selects the edit task

Figure 29: Defining the Edit Task

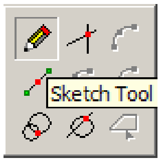


Task: Create New Feature

Target: NRV Fiber Junctions

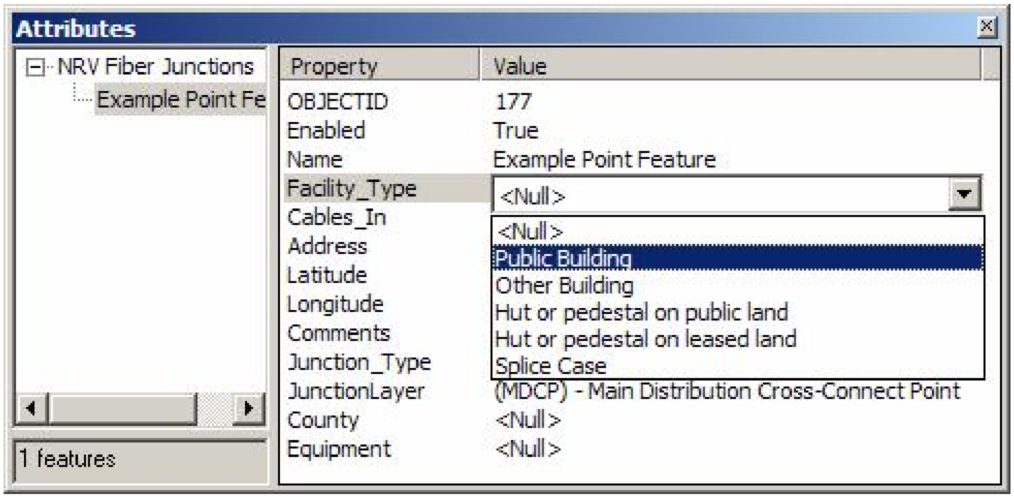
Layer in which features you create will be stored

Figure 30: Setting the Edit Target Layer and Subtype



Once the edit task and target layer/subtype have been defined, we activate the Sketch tool on the editor toolbar:

Then, we double-click on the map to create an MDCP junction feature at that location. Once the feature has been created, it is automatically selected. Now we want to define some its attributes. Click the Attribute Button on the Editor toolbar. A window similar to the one below should appear:



Attributes

- NRV Fiber Junctions
  - Example Point Fe

1 features

Property	Value
OBJECTID	177
Enabled	True
Name	Example Point Feature
Facility_Type	<Null>
Cables_In	
Address	
Latitude	
Longitude	
Comments	
Junction_Type	
JunctionLayer	(MDCP) - Main Distribution Cross-Connect Point
County	<Null>
Equipment	<Null>

<Null>
Public Building
Other Building
Hut or pedestal on public land
Hut or pedestal on leased land
Splice Case

Figure 31: Editing the Attributes of a junction feature class

The only information you actually need to *type* at this point is the Name of the feature, its address (if it exists) and any comments you may have. As the example indicates, the Facility_Type and Junction_Type field is bound to a geodatabase attribute domain, so it appears as a drop-down list as you edit. The subtype field, JunctionLayer, has this behavior as well – if you specified the wrong subtype as your edit target, you can always go back and change it. Other fields, like Cables_In, Latitude, and Longitude, can be populated through the use of ArcObjects macros, which were described in a previous section of this guide.

Next, we want to create a feature that represents a fiber route from the junction we created to some other location. There are many ways to do this, such as digitizing the line manually, digitizing it with snapping enabled, or (as we will demonstrate) tracing the exact path of existing features.

Using the Trace Tool

Fiber routes often follow other features, such as highways and utility rights-of-way. If construction of fiber in your project has been planned to follow existing features, you can use ArcMap’s Trace tool to make the task of laying out the network much easier. To use the Trace tool to create a fiber route:

1. Change the edit target to a fiber route layer/subtype (a linear feature).
2. Under Selection à “Set Selectable Layers”, de-select all layers except the one you want to trace.

3. Set the snapping environment to snap to the vertices of Fiber Junctions, and the vertices/edges/ends of roads.
4. Switch to the Select Features tool.
5. Select all the segments of existing features you wish to trace by drawing a box around them. This operation will most likely select more features than you want to trace, but that is not a problem.

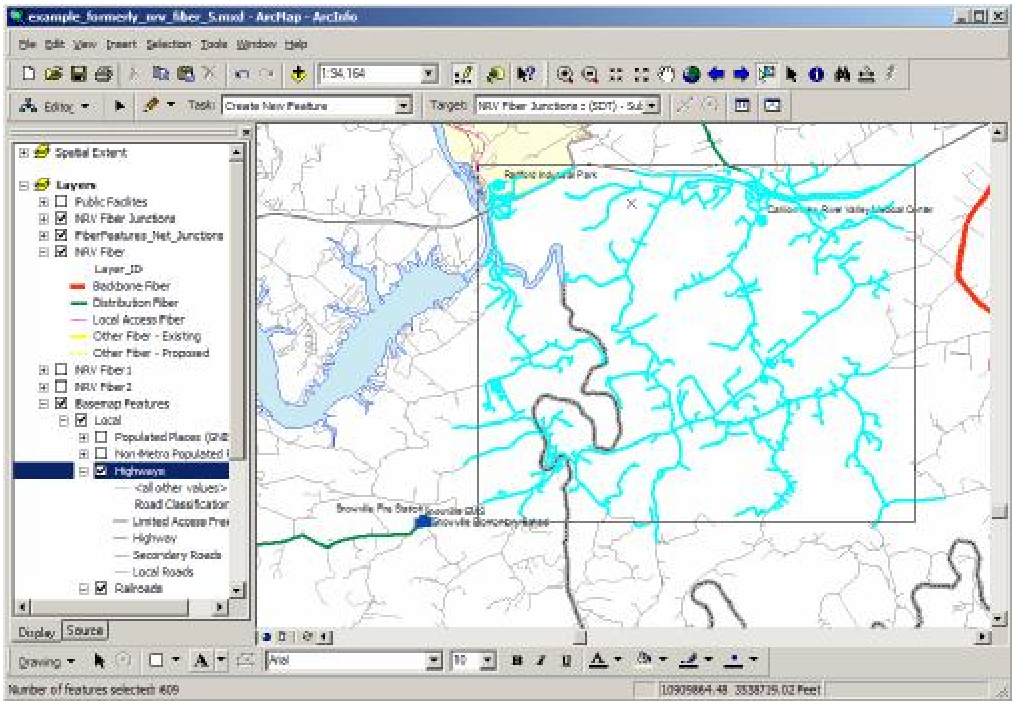


Figure 32: Selecting an area in which to trace features

6. Switch to the Sketch Tool.
7. Snap the end of the fiber route to the MDCP you created in the last step.
8. Snap the next vertex of the route to a point on the road to be traced (which should be selected).
9. Switch to the Trace tool.
10. Click again on the last vertex (the one you snapped to the road) to start the trace.

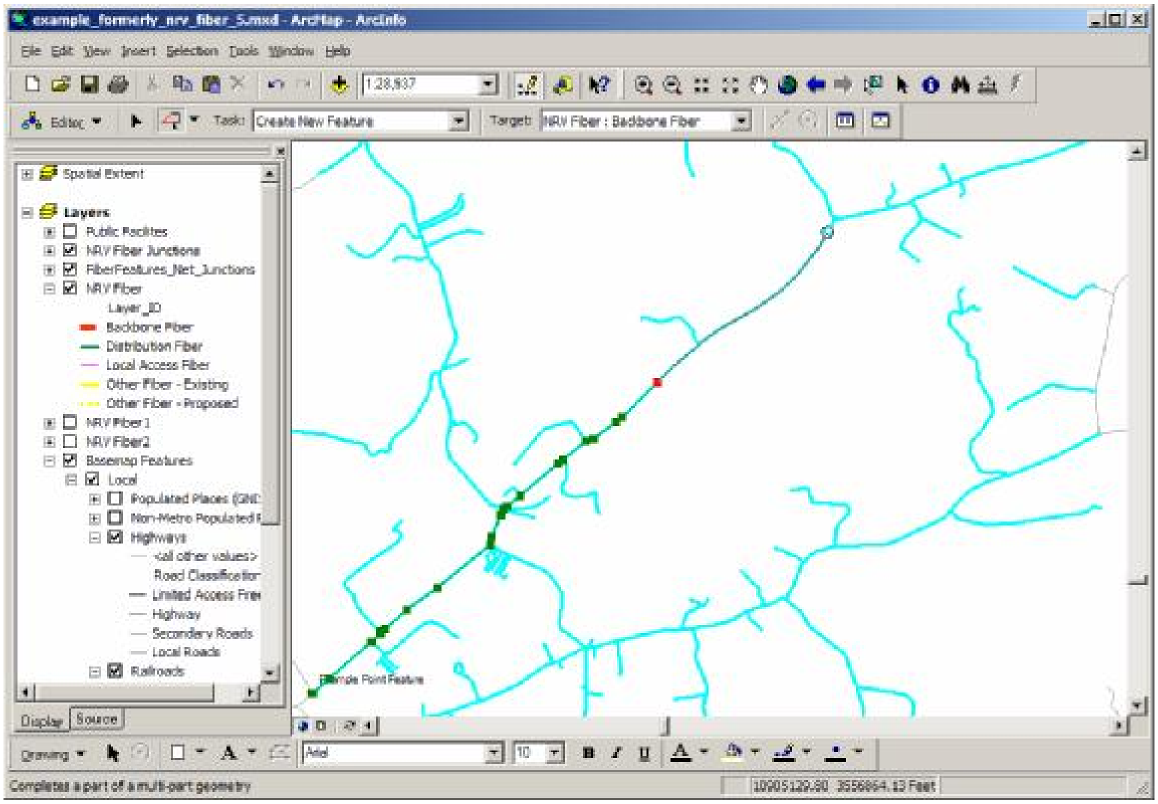
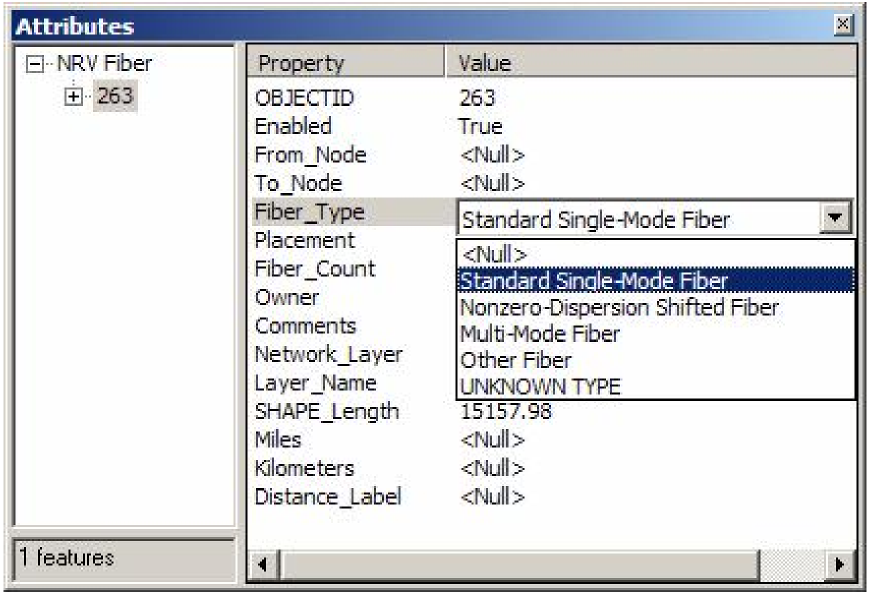


Figure 33: Using the Trace Tool

11. Without clicking the mouse, move it along the road until you get to where you want to go. If you've ever welded, it feels a bit like welding – you nudge the tool along little by little, taking care not to go too fast. You may need to back up if the tool takes off down a side road or otherwise goes the wrong way. Sometimes, you may encounter a situation where the tool cannot cross a particular intersection or feature. This is probably caused by a topological error in the basemap dataset. To fix it, do not terminate the sketch – rather, switch to the sketch tool and manually draw the line across the problem area. Then switch back to the trace tool and continue. If you don't do this, you may get to the end of your trace and receive an error message, which will abort the whole sketch, forcing you to start over.
12. When you finish the sketch, you should see an MDCP or DCP feature automatically appear at the end of it. (If this doesn't happen, verify that you have set the default junction property correctly in the geometric network connectivity rules.)
13. Set the attributes of the newly created junction feature in similar manner to the one we created in the previous step.

14. Set the attributes of the line feature, as shown below:



Property	Value
OBJECTID	263
Enabled	True
From_Node	<Null>
To_Node	<Null>
Fiber_Type	Standard Single-Mode Fiber
Placement	
Fiber_Count	
Owner	
Comments	
Network_Layer	
Layer_Name	
SHAPE_Length	15157.98
Miles	<Null>
Kilometers	<Null>
Distance_Label	<Null>

Figure 34: Setting the attributes for a run of fiber

Note that the Fiber_Type, Placement, and Network_Layer fields are all populated by drop-down lists. The SHAPE_Length field is maintained automatically by the system. The From_Node, To_Node, Miles, Kilometers, and Distance_Label can be populated through the use of macros.

As this brief example has shown, the editing process becomes much more streamlined when using geodatabases and geometric networks. Further optimization is certainly possible – this represents one way of doing it that is intended to serve as an example and a point of departure for future refinements.

Using the Geodatabase for Cost Estimation

Fiber Routes

Another advantage of the geodatabase data model is that it keeps track of the length of linear features (and the area and perimeter of polygon features) by default. When the feature class for fiber was created, a field called Shape_Length was automatically inserted into its attribute table. This field, maintained constantly by the system, stores the length of the feature in map units. Since we originally set the projection of our feature dataset (which was inherited by all feature classes contained within it) to a State Plane coordinate system that uses U.S. Survey Feet, the Shape_Length field in the fiber feature class stores the length, in feet, of each fiber route segment.

By applying a multiplier ($/mile, for example) to the length of a fiber route, we can get an approximate cost figure for initial planning purposes. This is only a ballpark estimate, but in the early stages of the planning process, a ballpark estimate may be all that is necessary – and GIS provides one of the fastest ways to arrive at such an estimate.

There are several ways to do cost estimation in GIS. One way is to create a “Cost” field in the attribute table of the fiber routes feature class, and use an ArcObjects macro to populate it. If, however, one is not comfortable with the programming necessary to accomplish this, the same thing can easily be done in a spreadsheet (for demonstration purposes, we will use Microsoft Excel).

Because geodatabases are physically stored in the Microsoft Access (.mdb) file format, their attribute tables can be imported seamlessly into Excel. Suppose we have a fiber route feature class with several routes already drawn and attributed. We want to create a report of the cost of each route segment, and the total project cost so far. We can do this very quickly in Excel by importing the attribute table using the Microsoft Query utility. (Note: Microsoft Query is not always installed with Excel by default. You may be prompted for your Microsoft Office installation CD the first time you try to use the feature.)

Follow these steps to quickly calculate approximate fiber route costs in Microsoft Excel:

1. Open Microsoft Excel with a new workbook.
2. Go to **Data à Import External Data à New Database Query.**

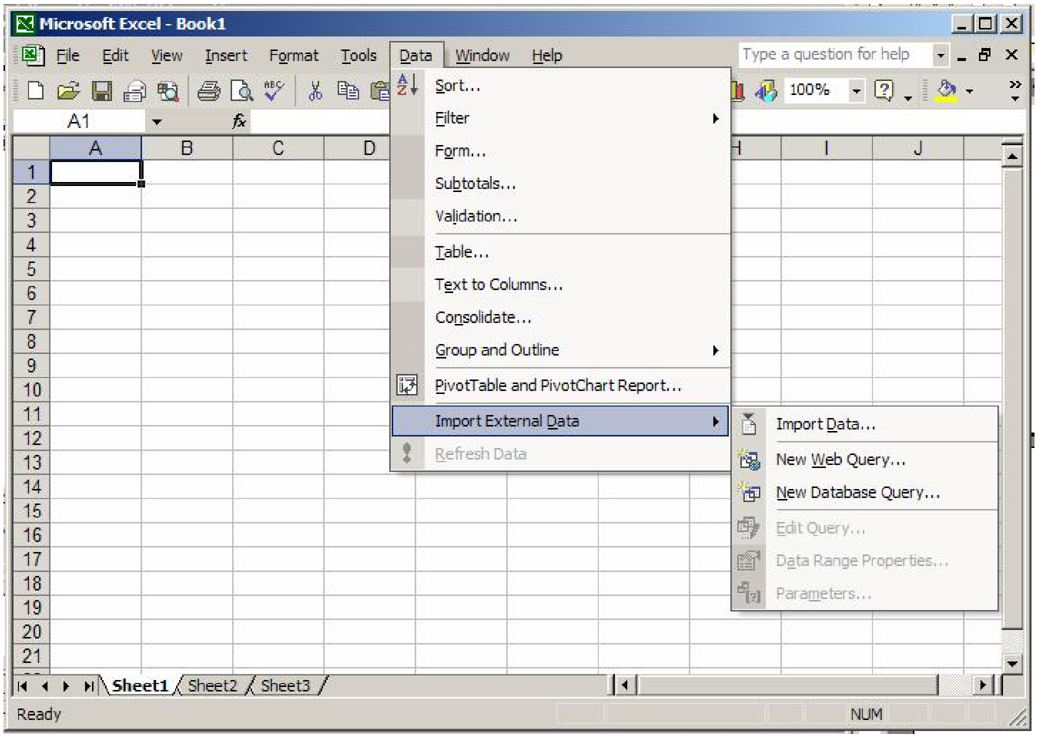
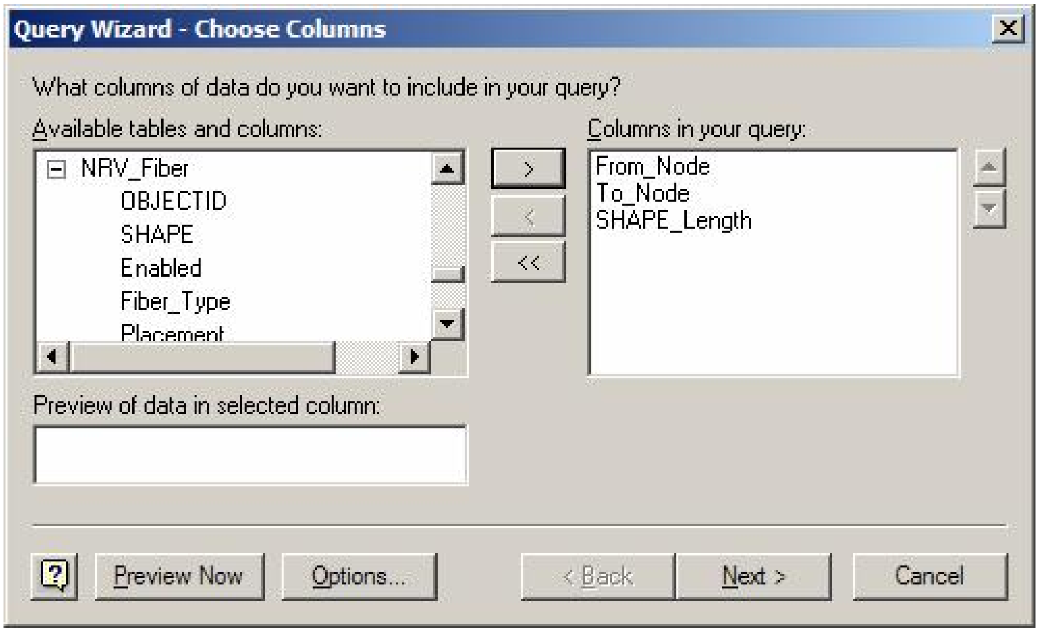


Figure 35: A new database query in Microsoft Excel

3. Under the “Databases” tab of the “Choose Data Source” window, select “MS Access Database”
4. A box should appear that says “Connecting to data source”. In the “Select Database” file open dialog that appears, navigate to the location of your geodatabase.
5. The next window that appears will be entitled “Query Wizard – Choose columns”. Ignore all the GDB_ tables – these are maintained by the system for storing database configuration information. You don’t want to edit these directly. Instead, look for the “Fiber” table (or whatever you named the feature class that stores fiber routes). Expand the ‘+’ beside the table to see all the fields.



Query Wizard - Choose Columns

What columns of data do you want to include in your query?

Available tables and columns:

⊟ NRV_Fiber
OBJECTID
SHAPE
Enabled
Fiber_Type
Placement

Columns in your query:

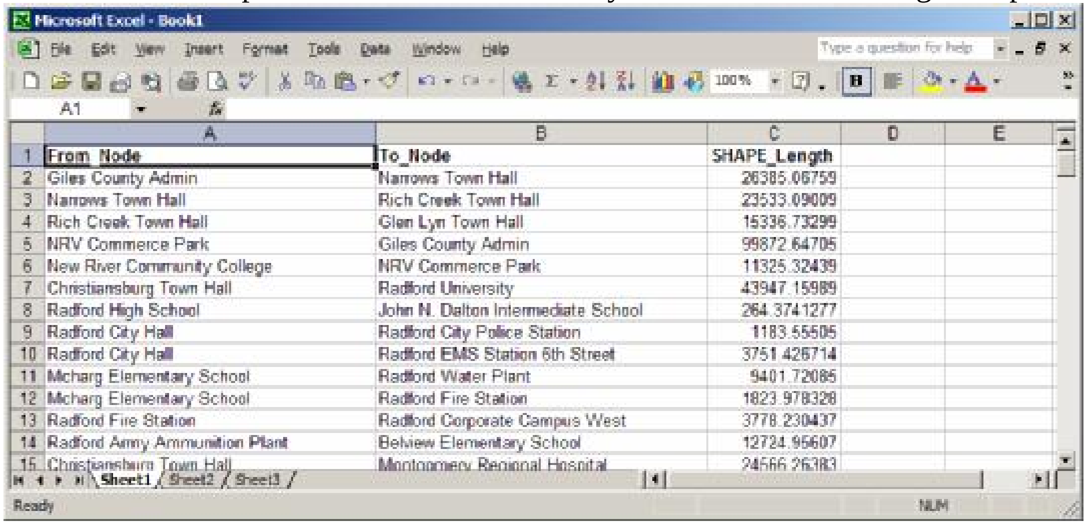
From_Node
To_Node
SHAPE_Length

Preview of data in selected column:

Figure 36: Query Wizard - Choose Columns

6. Select, at a minimum, the From_Node, To_Node, and SHAPE_Length fields for inclusion in the query. If you have filled in other fields, such as Owner, Fiber_Type, and Placement, you may include them as well.
7. In the next screen, you may choose to filter the data – define a query that excludes certain features. This is optional; for the purposes of this example, we will skip this step.
8. In the next screen, you may choose to sort the data. This is optional; for the purposes of this example, we will display records in the order in which they appear in the geodatabase (the default).
9. In the “Query Wizard – Finish” screen, select “Return Data to Microsoft Excel.”
10. In the “Import Data” box that appears, choose the location where the data will be returned. Assuming you have a blank worksheet, A1, the default, is fine.

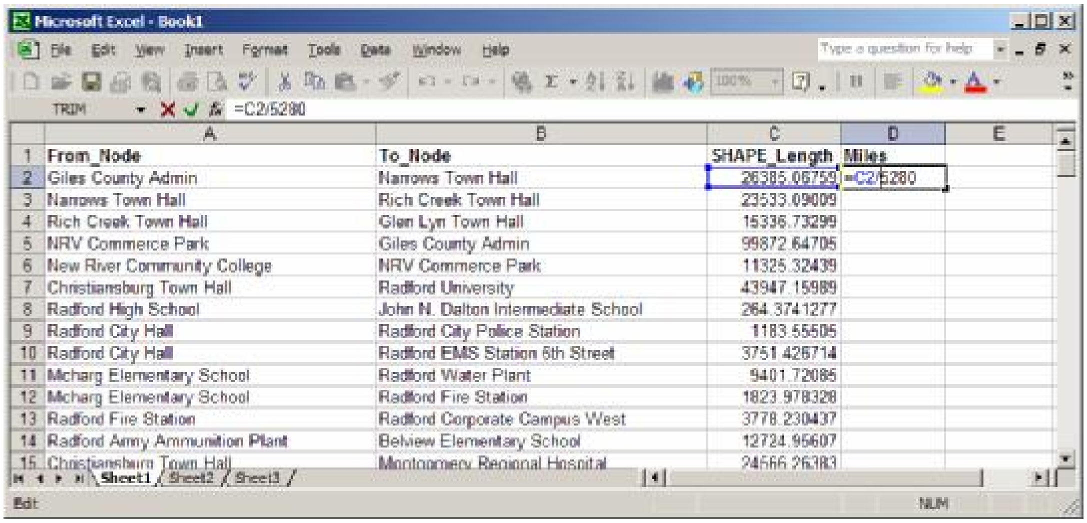
11. The results of this process should be structurally similar to the following example:



	A	B	C	D	E
1	From_Node	To_Node	SHAPE_Length		
2	Giles County Admin	Narrows Town Hall	26385.06759		
3	Narrows Town Hall	Rich Creek Town Hall	23533.09009		
4	Rich Creek Town Hall	Glen Lyn Town Hall	15336.73299		
5	NRV Commerce Park	Giles County Admin	99872.64705		
6	New River Community College	NRV Commerce Park	11325.32439		
7	Christiansburg Town Hall	Radford University	43947.15989		
8	Radford High School	John N. Dalton Intermediate School	264.3741277		
9	Radford City Hall	Radford City Police Station	1183.55505		
10	Radford City Hall	Radford EMS Station 6th Street	3751.426714		
11	Mcharg Elementary School	Radford Water Plant	9401.72085		
12	Mcharg Elementary School	Radford Fire Station	1823.978328		
13	Radford Fire Station	Radford Corporate Campus West	3778.230437		
14	Radford Army Ammunition Plant	Belview Elementary School	12724.95607		
15	Christiansburg Town Hall	Montgomery Regional Hospital	24566.26383		

Figure 37: Excel - Import Data Results

12. Now, since fiber costs are usually stated in miles, let’s create a column to the right of the Shape_Length field that stores the feature’s length in miles. Simply create a column called “Miles”, and enter the formula “=C2 / 5280” in the first row of data. Then use the fill handle to carry down the formula to the other rows. Optionally, format the cells to reduce the number of decimal places.



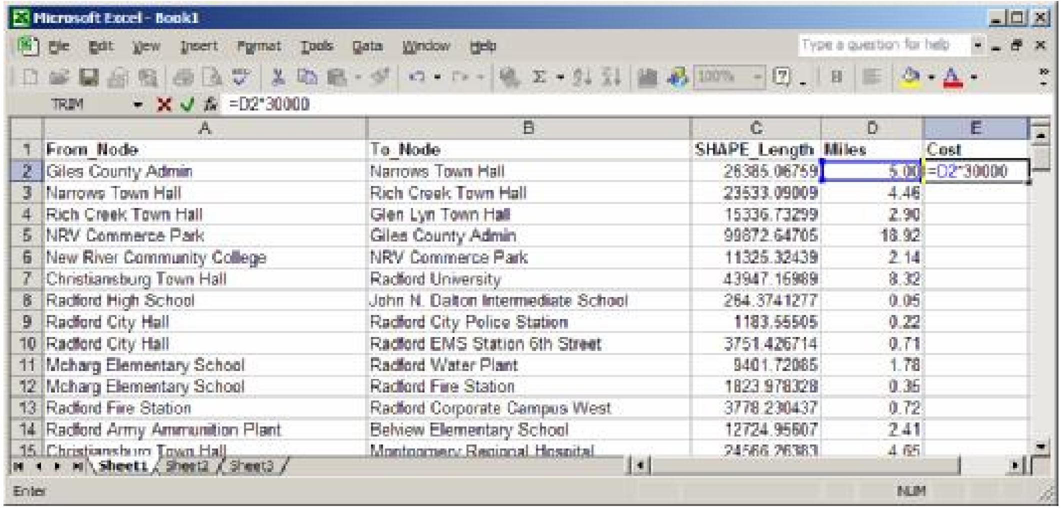
=C2/5280

	A	B	C	D	E
1	From_Node	To_Node	SHAPE_Length	Miles	
2	Giles County Admin	Narrows Town Hall	26385.06759	=C2/5280	
3	Narrows Town Hall	Rich Creek Town Hall	23533.09009		
4	Rich Creek Town Hall	Glen Lyn Town Hall	15336.73299		
5	NRV Commerce Park	Giles County Admin	99872.64705		
6	New River Community College	NRV Commerce Park	11325.32439		
7	Christiansburg Town Hall	Radford University	43947.15989		
8	Radford High School	John N. Dalton Intermediate School	264.3741277		
9	Radford City Hall	Radford City Police Station	1183.55505		
10	Radford City Hall	Radford EMS Station 6th Street	3751.426714		
11	Mcharg Elementary School	Radford Water Plant	9401.72085		
12	Mcharg Elementary School	Radford Fire Station	1823.978328		
13	Radford Fire Station	Radford Corporate Campus West	3778.230437		
14	Radford Army Ammunition Plant	Belview Elementary School	12724.95607		
15	Christiansburg Town Hall	Montgomery Regional Hospital	24566.26383		

Figure 38: Excel - Converting feet to miles

13. The next step can be handled in several ways. Fundamentally, we need to multiply the length of each fiber run by some number that serves as an appropriate cost multiplier for our project. In the past, Virginia Tech has used a multiplier of $30,000 per mile to cost out projects in the early stages. This number has built into it many factors such as materials and labor. It can be applied to every run of fiber in the project and result in a generally acceptable estimate. It will, as a rule, tend to overestimate costs of aerial construction and underestimate costs of buried construction; on a project with a mix of both construction types, these errors may cancel each other out to an extent. If you had a field that distinguished between construction types, you could set up an "IF" statement in an Excel formula that used one multiplier for aerial and a different one for buried. For simplicity, we will demonstrate the use of the single multiplier.

    In Excel, create a column "Cost" next to "Miles" and enter the formula "=D2 * 30000" in the first row of data. Then use the fill handle to carry down the formula to the other rows. Optionally, format the cells as currency.



TRIM =D2*30000

	A	B	C	D	E
1	From_Node	To_Node	SHAPE_Length	Miles	Cost
2	Giles County Admin	Narrows Town Hall	26385.06759	5.00	=D2*30000
3	Narrows Town Hall	Rich Creek Town Hall	23533.09009	4.46	
4	Rich Creek Town Hall	Glen Lyn Town Hall	15336.73299	2.90	
5	NRV Commerce Park	Giles County Admin	99872.64705	18.92	
6	New River Community College	NRV Commerce Park	11325.32439	2.14	
7	Christiansburg Town Hall	Radford University	43947.15989	8.32	
8	Radford High School	John N. Dalton Intermediate School	264.3741277	0.05	
9	Radford City Hall	Radford City Police Station	1183.55505	0.22	
10	Radford City Hall	Radford EMS Station 6th Street	3751.426714	0.71	
11	Mcharg Elementary School	Radford Water Plant	9401.72065	1.78	
12	Mcharg Elementary School	Radford Fire Station	1823.978328	0.35	
13	Radford Fire Station	Radford Corporate Campus West	3778.230437	0.72	
14	Radford Army Ammunition Plant	Belview Elementary School	12724.95607	2.41	
15	Christiansburg Town Hall	Montgomery Regional Hospital	24566.26383	4.65	

Figure 39: Calculating Fiber Cost

14. At the bottom of the table, create a row "TOTAL" and use the Excel =SUM() or Autosum function to calculate the total length and cost of the project.
15. As your network changes, you can either use the "Refresh Query" command in cell A1 or create a new query.

Suggestions for Further Reading

Booth, B. and Environmental Systems Research Institute (Redlands Calif.) (2002). Editing in ArcMap. Redlands, CA, Esri.

Virginia Geographic Information Network (VGIN) Library of Local Government GIS Resources. http://www.vgin.virginia.gov/VGIN_Library/LIBlocalgvt.cfm.

Virginia Tech eCorridors Program (2002). Strategic Technology Infrastructure for Regional Competitiveness in the Network Economy (vol.4: Fiber-Optic Infrastructure Design Guide.) http://www.ecorridors.vt.edu/research/papers/stircne/index_flash.shtml#download.

Zeiler, M. and Environmental Systems Research Institute (Redlands Calif.) (2001). Exploring ArcObjects: GIS by ESRI. Redlands, Calif., Esri.

Appendix A: ArcObjects Code Samples

```
Public Sub updateCableCounts()

' This calculates the number of fiber cables (not individual fibers)
' running into a node in the Fiber geometric network
' WORKS as of 2004-04-28

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

Dim pFeatureLayer As IFeatureLayer
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature
Dim thisJunction As ISimpleJunctionFeature


Set pFeatureLayer = GetLayerByName(pMxDoc.FocusMap, “<Fiber Junctions
Layer Name>”, True)

Set pFeatureCursor = pFeatureLayer.Search(Nothing, False)
Set pFeature = pFeatureCursor.NextFeature

Do Until pFeature Is Nothing
    Set thisJunction = pFeature
    Count = thisJunction.EdgeFeatureCount
    pFeature.Value(pFeature.Fields.FindField("Cables_In")) = Count
     pFeature.Store
    Set pFeature = pFeatureCursor.NextFeature
Loop

Set pFeature = Nothing
Set pPoint = Nothing
Set pFeatureCursor = Nothing
Set pFeatureLayer = Nothing
Set thisJunction = Nothing
'MsgBox ("All Cable Counts updated successfully.")

End Sub
```

```
Public Sub updateLatLong()

' This calculates the latitude and longitude of the fiber nodes
' (stored in StatePlane - VA South - 4502)
' and writes the result to the Latitude and Longitude fields
' WORKS as of 2004-04-27

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

Dim pFeatureLayer As IFeatureLayer
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature

Dim pInSpatialReference As ISpatialReference
Dim pOutSpatialReference As ISpatialReference
Dim pPCS As IProjectedCoordinateSystem
Dim pGCS As IGeographicCoordinateSystem
Dim pSpatialReferenceEnv As SpatialReferenceEnvironment
Dim pPoint As IPoint

Set pSpatialReferenceEnv = New SpatialReferenceEnvironment
Set pPCS =
pSpatialReferenceEnv.CreateProjectedCoordinateSystem(esriSRProjCS_NAD19
83SPCS_VASouthFT)
Set pInSpatialReference = pPCS
Set pGCS =
pSpatialReferenceEnv.CreateGeographicCoordinateSystem(esriSRGeoCS_NAD19
83)
Set pOutSpatialReference = pGCS

Set pFeatureLayer = GetLayerByName(pMxDoc.FocusMap, "“<Fiber Junctions
Layer Name>”, True)
Set pFeatureCursor = pFeatureLayer.Search(Nothing, False)
Set pFeature = pFeatureCursor.NextFeature

Do Until pFeature Is Nothing
    Set pPoint = pFeature.ShapeCopy
    Set pPoint.SpatialReference = pInSpatialReference
    pPoint.Project pOutSpatialReference

     pFeature.Value(pFeature.Fields.FindField("Latitude")) = pPoint.y
     pFeature.Value(pFeature.Fields.FindField("Longitude")) = pPoint.x
    pFeature.Store

    Set pFeature = pFeatureCursor.NextFeature

Loop

'MsgBox ("All lat/long fields updated successfully.")

Set pFeature = Nothing
Set pPoint = Nothing
Set pFeatureCursor = Nothing
Set pFeatureLayer = Nothing
Set pInSpatialReference = Nothing
Set pOutSpatialReference = Nothing
Set pPCS = Nothing
Set pGCS = Nothing
Set pSpatialReferenceEnv = Nothing
Set pPoint = Nothing


End Sub
```

```
Public Sub updateMiles()

' This just calculates miles and kilometers from SHAPE_Length


Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

Dim pFeatureLayer As IFeatureLayer
Dim pFeatureCursor As IFeatureCursor
Dim pFeature As IFeature

Dim miles, kilometers As Double


Set pFeatureLayer = GetLayerByName(pMxDoc.FocusMap, “<Fiber Routes
Layer Name>”, False)
True)
Set pFeatureCursor = pFeatureLayer.Search(Nothing, False)
Set pFeature = pFeatureCursor.NextFeature

Do Until pFeature Is Nothing

    miles = pFeature.Value(pFeature.Fields.FindField("SHAPE_Length")) /
5280
    pFeature.Value(pFeature.Fields.FindField("Miles")) = miles
    'MsgBox (miles)
    kilometers = (miles * 1.609347)
    'MsgBox (kilometers)
    pFeature.Value(pFeature.Fields.FindField("Kilometers")) =
kilometers
    pFeature.Value(pFeature.Fields.FindField("Distance_Label")) =
Format(miles, "##0.00")
    pFeature.Store
    Set pFeature = pFeatureCursor.NextFeature

Loop

'MsgBox ("All mile and Kilometer measurements updated successfully.")

Set pFeature = Nothing
Set pFeatureCursor = Nothing
Set pFeatureLayer = Nothing
```

```
End Sub
```

```
Public Sub updateEndpoints()

'WORKS as of 2004-05-21
'Appends the "Name" fields of the nodes corresponding to the from_node
'and to_node of a run of fiber.

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument

Dim theEdgeFeatureLayer As IFeatureLayer
Dim theEdgeFeatureCursor As IFeatureCursor
Dim theEdgeFeature As IFeature
Dim thisEdge As IEdgeFeature
Dim FNodeName, TNodeName As String

Dim theFNodeEID, theTNodeEID As Long
Dim tempJunctionFeature As ISimpleJunctionFeature
Dim tempFeature As IFeature
Dim theJunctionFeatureLayer As IFeatureLayer
Dim theJunctionFeatureCursor As IFeatureCursor

'Return pointers to the appropriate map layers
Set theEdgeFeatureLayer = GetLayerByName(pMxDoc.FocusMap, “<Fiber
Routes Layer Name>” False)
Set theJunctionFeatureLayer = GetLayerByName(pMxDoc.FocusMap, “<Fiber
Junctions Layer Name>”, False)

'Create and initialize a FeatureCursor to loop through each edge of the
network
Set theEdgeFeatureCursor = theEdgeFeatureLayer.Search(Nothing, False)
Set theEdgeFeature = theEdgeFeatureCursor.NextFeature

'For each edge in the network...
Do Until theEdgeFeature Is Nothing
    Set thisEdge = theEdgeFeature


    'get their unique EIDs

    theFNodeEID = thisEdge.FromJunctionEID
    theTNodeEID = thisEdge.ToJunctionEID


    'Create and initialize a FeatureCursor to loop through each
junction of the network
    Set theJunctionFeatureCursor =
theJunctionFeatureLayer.Search(Nothing, False)
    Set tempFeature = theJunctionFeatureCursor.NextFeature

    'Compare the EIDs of each junction in the network to the one of the
target.
    Do Until tempFeature Is Nothing
```

```
        '"cast" to something that implements the IJunctionFeature
Interface
        Set tempJunctionFeature = tempFeature
        If tempJunctionFeature.EID = theFNodeEID Then
            'MsgBox ("From: " +
tempFeature.Value(tempFeature.Fields.FindField("Name")))
            FNodeName =
tempFeature.Value(tempFeature.Fields.FindField("Name"))

theEdgeFeature.Value(theEdgeFeature.Fields.FindField("From_Node")) =
FNodeName
        ElseIf tempJunctionFeature.EID = theTNodeEID Then
            'MsgBox ("To: " +
tempFeature.Value(tempFeature.Fields.FindField("Name")))
            TNodeName =
tempFeature.Value(tempFeature.Fields.FindField("Name"))

theEdgeFeature.Value(theEdgeFeature.Fields.FindField("To_Node")) =
TNodeName
        End If
        Set tempFeature = theJunctionFeatureCursor.NextFeature

    Loop


   theEdgeFeature.Store
   Set theEdgeFeature = theEdgeFeatureCursor.NextFeature
Loop

End Sub
```