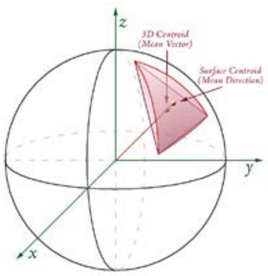
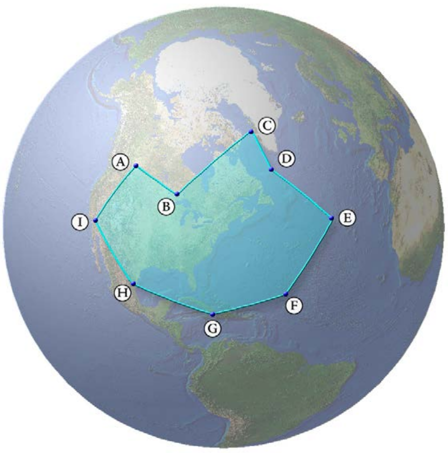
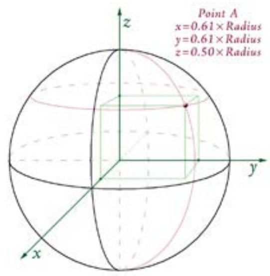
Tools for Graphics and Shapes







Last updated 1 June 2011

Jeff Jenness

Jenness Enterprises

About the Author

Jeff Jenness is an independent GIS consultant specializing in developing analytical applications for a wide variety of topics, although he most enjoys ecological and wildlife-related projects. He spent 16 years as a wildlife biologist with the USFS Rocky Mountain Research Station in Flagstaff, Arizona, mostly working on Mexican spotted owl research. Since starting his consulting business in 2000, he has worked with universities, businesses and governmental agencies around the world, including a long-term contract with the United Nations Food and Agriculture Organization (FAO) for which he relocated to Rome, Italy for 3 months. His free ArcView tools have been downloaded from his website and the ESRI ArcScripts site over 250,000 times.

Acknowledgements

Partially funded by the US Geological Survey, Astrogeology Team (http://astrogeology.usgs.gov/).

NAME: Tools for Graphics and Shapes

Last modified: June 1, 2011

TOPICS: Sphere, Vincenty, Geodesic, Great Circle, Spherical Triangle, Polygon, Centroid, Center of Mass, Vector, Area, Haversine

AUTHOR: Jeff Jenness
Wildlife Biologist, GIS Analyst
Jenness Enterprises
3020 N. Schevene Blvd.
Flagstaff, AZ 86004 USA

Email: jeffj@jennessent.com
Web Site: http://www.jennessent.com)
Phone: 1-928-607-4638

Description: This extension includes a large suite of tools for calculating geometric attributes of vector features and for selecting and naming graphics. All tools are available at the ArcView license level.

Among many other functions, this extension offers tools for calculating the true area and centroid of polygons as they lay on the sphere, thereby avoiding errors caused by projection distortions. This extension also offers tools to calculate true lengths of polylines as they lie on the spheroid, using Vincenty's equations. This manual thoroughly explains all algorithms used to calculate geometric attributes on the sphere and spheroid.

This extension also includes a tool to create a new shapefile from within ArcMap, either by creating an empty shapefile or by converting graphics.

Output: Several tools produce point, multipoint, polyline or polygon shapefiles. Some geometry tools produce new shapefiles while others add new fields to attribute tables.

Requires: ArcGIS 9.1 or better, at any license level. The "Build Polygon" tool requires ArcView 9.2 or newer. This extension has not been tested with ArcGIS 9.0, and I am sure it will not work with ArcGIS 8.x.

Installing these tools in ArcGIS 10.x requires a different installation method than ArcGIS 9.x. See the Installation Instructions for details.

Revision History on p. 69.

Recommended Citation Format: For those who wish to cite this extension, the author recommends something similar to:

Jenness, J. 2011. Tools for Graphics and Shapes: Extension for ArcGIS. Jenness Enterprises. Available at: http://www.jennessent.com/arcgis/shapes_graphics.htm

Please let me know if you cite this extension in a publication (jeffj@jennessent.com). I will update the citation list to include any publications that I am told about.

Table of Contents

Installing Tools for Graphics and Shapes

For ArcGIS 9.x

First close ArcGIS if it is open. The tools do not install properly if ArcGIS is running during the installation.

Install the *Tools for Graphics and Shapes* extension by double-clicking on the file **GraphicsShapes.exe** (available at http://www.jennessent.com/downloads/GraphicsShapes.zip, or from a link on the page http://www.jennessent.com/arcgis/shapes_graphics.htm) and following the instructions. The installation routine will register the GraphicsShapes.dll with all the required ArcMap components.

The default install folder for the extension is named “Graphics_Shapes” and is located inside the folder “Program Files\Jennessent”. This folder will also include some additional files and this manual.

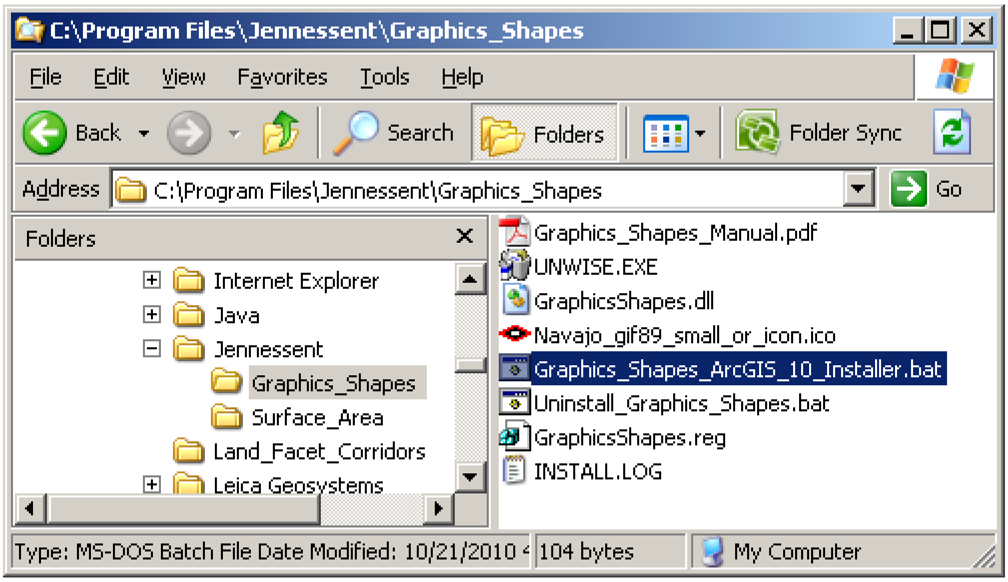
For ArcGIS 10.0

Note: This function will only work if you have ArcGIS 10 installed.

1. **First close ArcGIS if it is open.** The tools do not install properly if ArcGIS is running during the installation.

2. Install the *Tools for Graphics and Shapes* extension by double-clicking on the file **GraphicsShapes_10.exe** (available at http://www.jennessent.com/downloads/GraphicsShapes_10.zip, or from a link on the page http://www.jennessent.com/arcgis/shapes_graphics.htm) and following the instructions. This installation routine will install the GraphicsShapes.dll and several ancillary files on your hard drive, but will **not** register the tools with ArcGIS.

3. Use Windows Explorer to open your installation folder. If you used the default values, then this folder will be located at “Program Files\Jennessent\Graphics_Shapes\”. This folder will also include some additional files and this manual.



4. **For Windows XP:** Double-click the file “Graphics_Shapes_ArcGIS_10_Installer.bat” to register all the tools with ArcGIS 10.0.

   For Windows Vista or Windows 7: Right-click the file “Graphics_Shapes_ArcGIS_10_Installer.bat” and click “Run as Administrator” to register all the tools with ArcGIS 10.0.

If the registration is successful, then you should see a “Registration Succeeded” notice.

Note: For the concerned or curious, the batch file *Graphics_Shapes_ArcGIS_10_Installer.bat* contains the following single line of text:

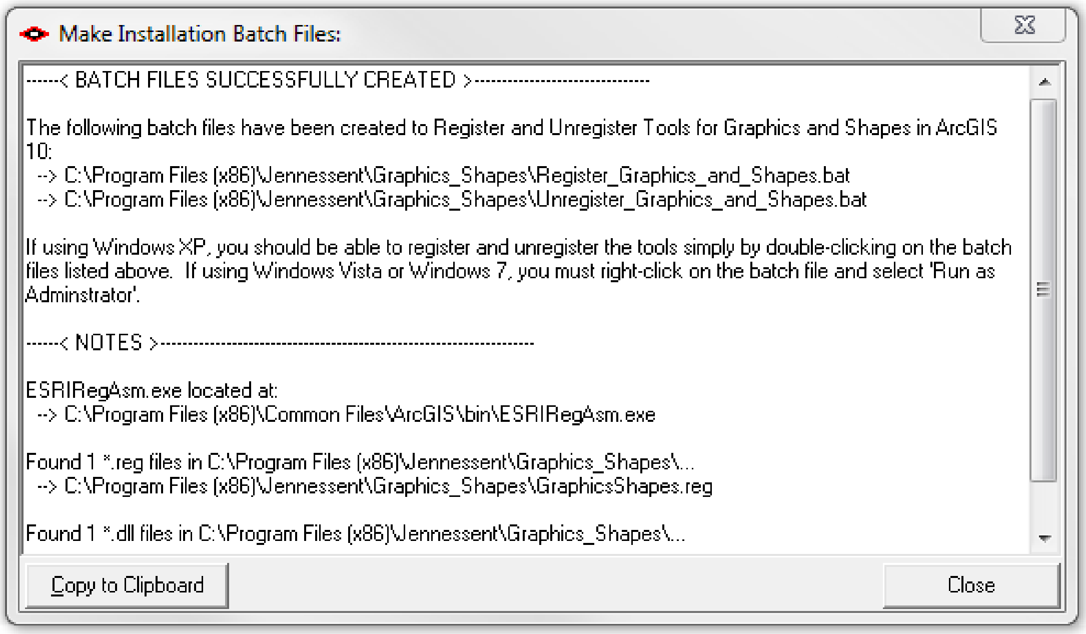
```
"%CommonProgramFiles%\ArcGIS\bin\ESRIRegAsm.exe" /p:Desktop
"GraphicsShapes.dll" /f:"GraphicsShapes.reg"
```

It directs the ESRI installer *ESRIRegAsm* to register the extension DLL *GraphicsShapes.dll* within ArcGIS, using GUID and Class ID values from the registry file *GraphicsShapes.reg* (also located in your installation directory). Both *Graphics_Shapes_ArcGIS_10_Installer.bat* and *GraphicsShapes.reg* may be opened and viewed using standard text editors such as Notepad or WordPad.

5. **Alternative Method if you do not get the “Registration Succeeded” message:** If the method above does not work, the reason is probably due to the “%CommonProgramFiles” environmental variable pointing to the wrong location, and/or Windows Vista or Windows 7 Security settings. The fix is to use a batch file that includes the full pathnames to “ESRIRegAsm.exe” and to the extension DLL and REG files. You may edit the BAT file yourself, or you may use the tool **Make_Batch_Files.exe** (located in your installation folder) to create new registration and unregistration batch files that are properly formatted to your system.

   If using Windows XP: Simply double-click on the file **Make_Batch_Files.exe** to create the new batch files.

   If using Windows Vista or Windows 7: Right-click on the file **Make_Batch_Files.exe** and click “Run as Administrator” to create the new batch files.

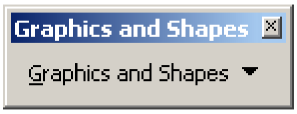


   Repeat Step 4 above to register the tools in ArcGIS 10, but this time use the new BAT file *Register_Graphics_and_Shapes.bat.*

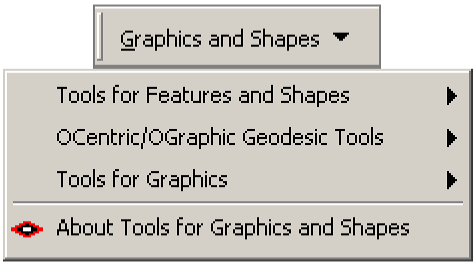
Viewing the Tools

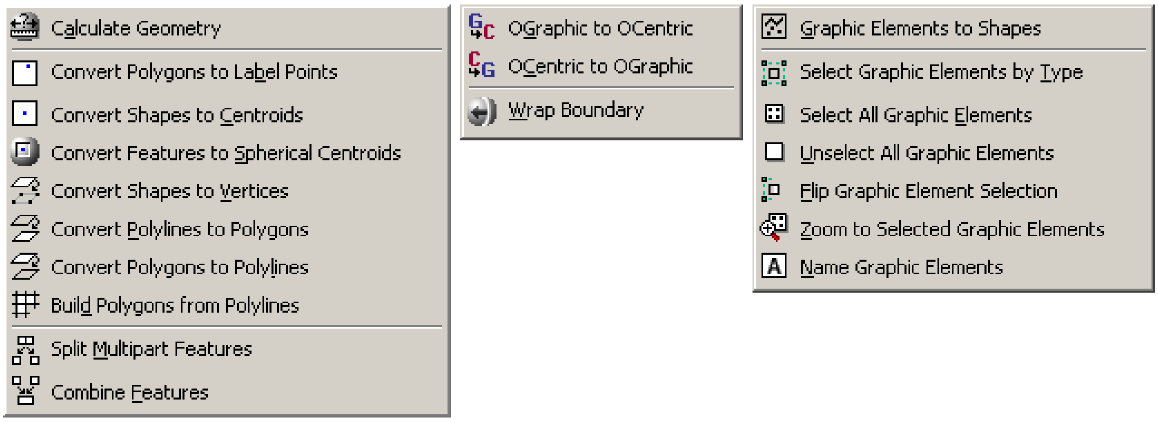
These tools are installed as an extension in ArcMap, but it is a type of extension that is automatically loaded. You will not see this extension in the "Extensions" dialog available in the ArcGIS "Tools" menu. It is not dependent on any other extensions or any ArcGIS license level.

After installing the extension, you should see the following new toolbar in your map (it may also be embedded in your standard ArcMap toolbars, rather than as a standalone object):



Click the "Graphics and Shapes" menu located on this toolbar to see the available tools:

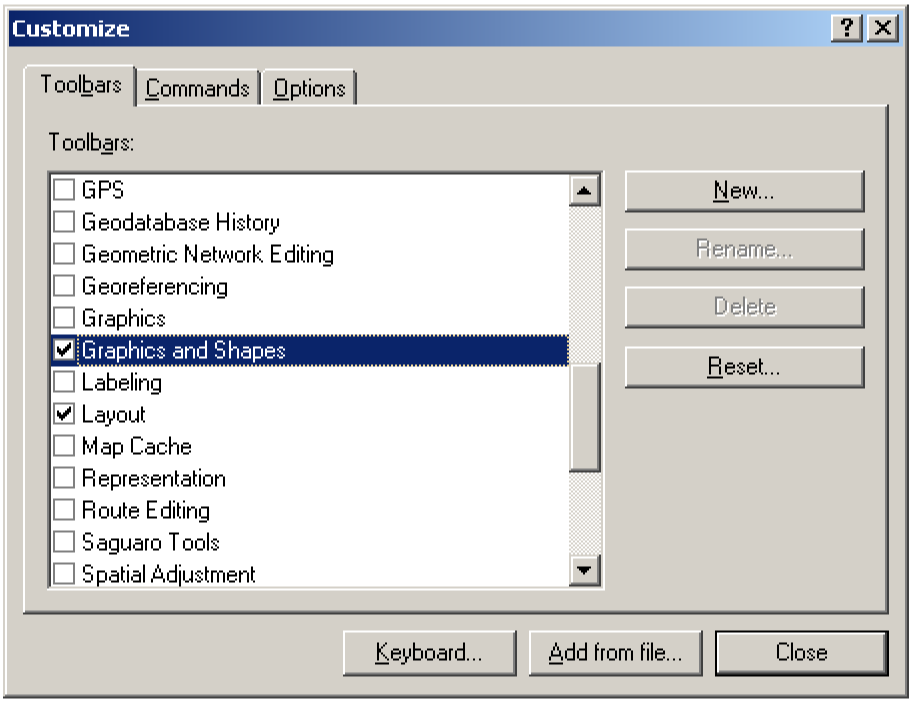




If you do not see this toolbar, then open your "Customize" tool by either:

1) Double-clicking on a blank part of the ArcMap toolbar, or
2) Clicking the "Tools" menu, then "Customize".

In the "Customize" dialog, click the "Toolbars" tab and check the box next to "Graphics and Shapes":



You should now see the Graphics and Shapes Tools toolbar.

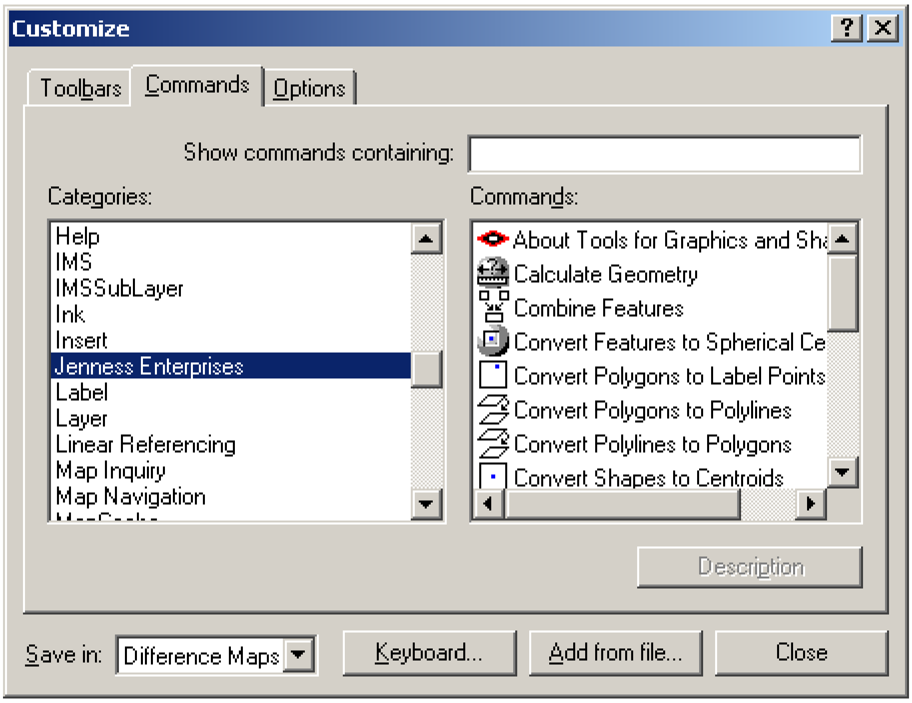
Copying and Adding Tools to Other Toolbars

Because of the way ArcGIS handles toolbars and command buttons, you may add any Graphics and Shapes Tools command buttons to any toolbar you wish. For example, if you would like to keep any of the tools available even when the Graphics and Shapes menu is closed, you may easily add those tools to any of the existing ArcGIS toolbars.

To do this, open your "Customize" tool by either:

1) Double-clicking on a blank part of the ArcMap toolbar, or
2) Clicking the "Tools" menu, then "Customize".

In the "Customize" dialog, click the "Commands" tab and scroll down to select "Jenness Enterprises":

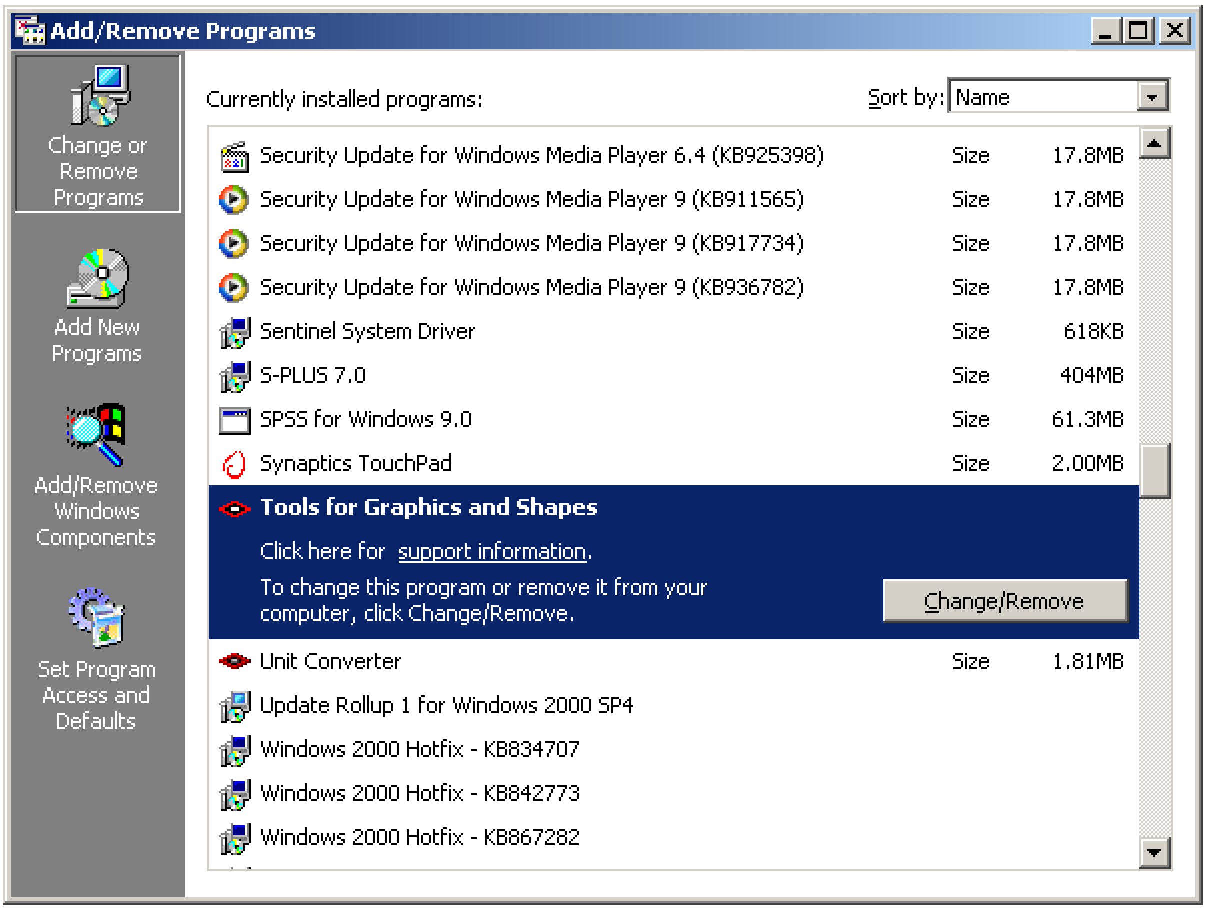


Finally, simply drag any of the commands out of the Customize dialog up into any of the existing ArcGIS toolbars.

Uninstalling Tools for Graphics and Shapes

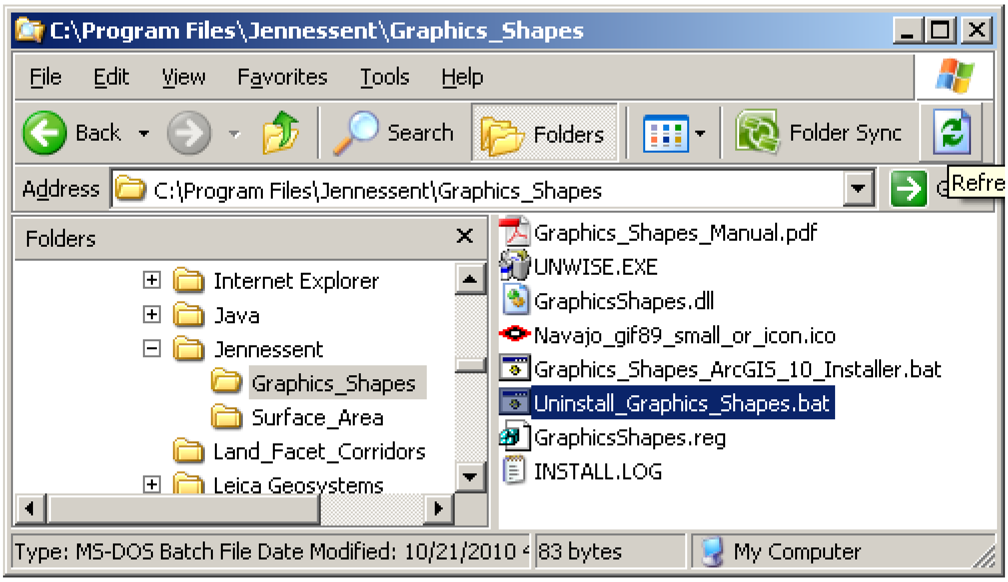
For ArcGIS 9.x.

1) Close ArcGIS if it is open.
2) Click the Start button.
3) Open your Control Panel.
4) Double-click “Add or Remove Programs”.
5) Scroll down to find and select “Tools for Graphics and Shapes”.
6) Click the “Remove” button and follow the directions.



For ArcGIS 10.0

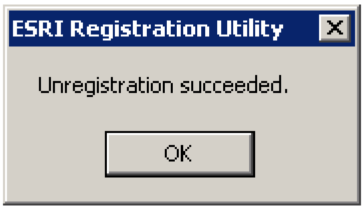
1) Close ArcGIS if it is open.
2) Use Windows Explorer to open your installation folder. If you used the default values, then this folder will be located at “Program Files\Jennessent\Graphics_Shapes\”. This folder will also include some additional files and this manual.



3) **For Windows XP:** Double-click the file *Uninstall_Graphics_Shapes.bat* to unregister all the tools with ArcGIS 10.0.

For Windows Vista and Windows 7: Right-click the file *Uninstall_Graphics_Shapes.bat* and select "Run as Administrator" to unregister all the tools with ArcGIS 10.0.

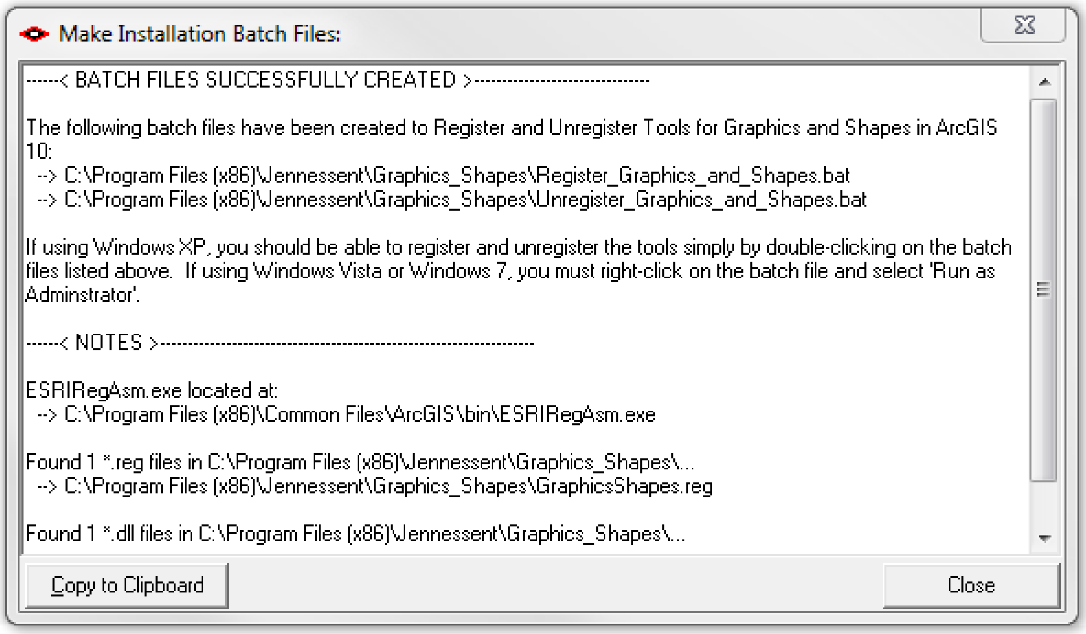
If the unregistration is successful, then you should see an "Unregistration Succeeded" notice.



4) **Alternative Method if you do not get the "Unregistration Succeeded" message:** If the method above does not work, the reason is probably due to the "%CommonProgramFiles" environmental variable pointing to the wrong location, and/or Windows Vista or Windows 7 Security settings. The fix is to use a batch file that includes the full pathnames to "ESRIRegAsm.exe" and to the extension DLL and REG files. You may edit the BAT file yourself, or you may use the tool **Make_Batch_Files.exe** (located in your installation folder) to create new registration and unregistration batch files that are properly formatted to your system.

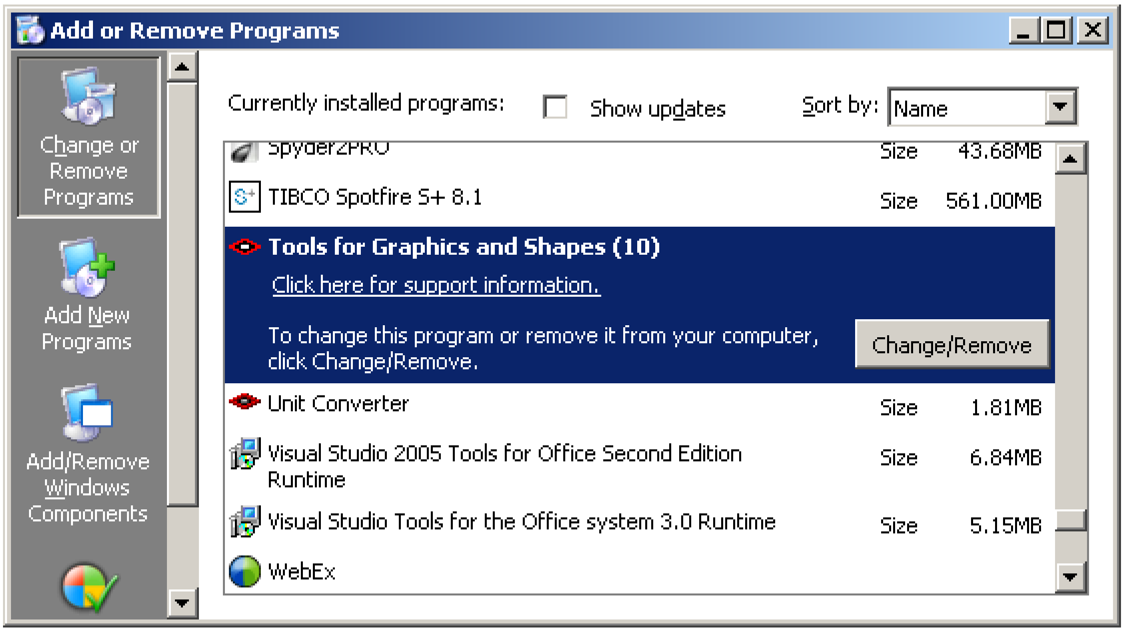
If using Windows XP: Simply double-click on the file **Make_Batch_Files.exe** to create the new batch files.

If using Windows Vista or Windows 7: Right-click on the file **Make_Batch_Files.exe** and click "Run as Administrator" to create the new batch files.



Repeat Step 3 above to unregister the tools in ArcGIS 10, but this time use the new BAT file *Unregister_Graphics_and_Shapes.bat*.

5) Click the Start button.
6) Open your Control Panel.
7) Double-click "Add or Remove Programs".
8) Scroll down to find and select "Tools for Graphics and Shapes 10".
9) Click the "Remove" button and follow the directions.



Note: For the concerned or curious, the batch file *Uninstall_Graphics_Shapes.bat* contains the following single line of text:

```
"%CommonProgramFiles%\ArcGIS\bin\ESRIRegAsm.exe" /p:Desktop /u
"Graphics_Shapes.dll"
```

It directs the ESRI installer *ESRIRegAsm* to unregister the DLL *GraphicsShapes.dll* within ArcGIS. *Uninstall_Graphics_Shapes.bat* may be opened and viewed using standard text editors such as Notepad or WordPad.

Troubleshooting

If Any of the Tools Crash

If a tool crashes, you should see a dialog that tells us what script crashed and where it crashed. I would appreciate it if you could copy the text in that dialog, or simply take screenshots of the dialog and email them to me at jeffj@jennessent.com. **Note:** Please make sure that the line numbers are visible in the screenshots! The line numbers are located on the far right side of the text. Use the scrollbar at the bottom of the dialog to make the line numbers visible.

"Object variable or With block variable not set" Error:

If you open ArcMap and immediately see the error dialog appear with one or more error messages stating that "Object variable or With block variable not set", then 90% of the time it is because ArcGIS was running when you installed the extension. The "Object" variable being referred to is the "Extension" object, and ArcGIS only sets that variable when it is initially turned on.

The solution is usually to simply close ArcGIS and restart it. If that does not work, then:

1) Close ArcGIS
2) Reinstall the extension
3) Turn ArcGIS back on.

RICHTX32.OCX Error:

If you see a line in the error dialog stating:

```
Component 'RICHTX32.OCX' or one of its dependencies not correctly registered: a file
is missing or invalid
```

This is almost always due to the fact that new installations of Windows 7 and Windows Vista do not include a file that the extension expects to find. This file, named "richtx32.ocx", is actually the "Rich Text Box" control that appears on some of the extension dialogs.

The solution is to manually install the missing file (richtx32.ocx) yourself. Here is how to do it:

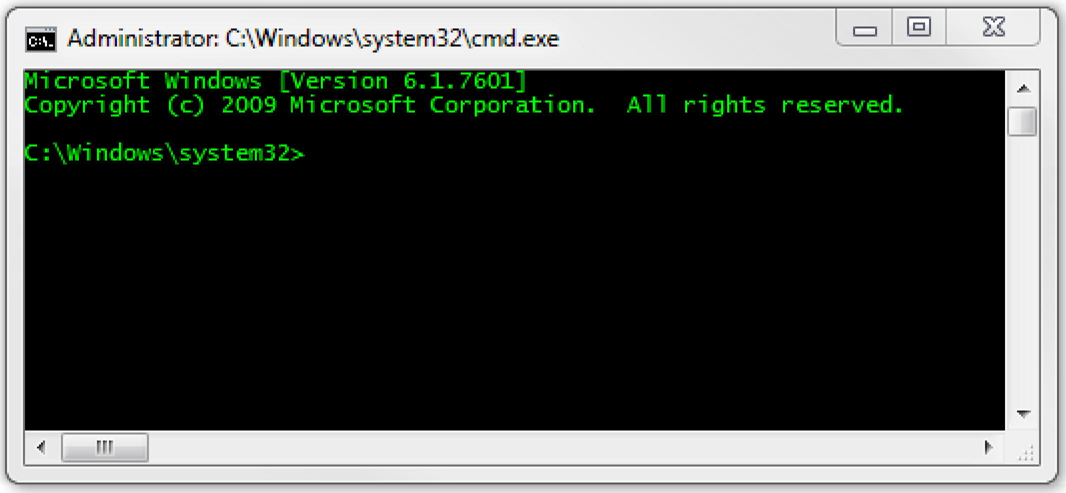
1) Open Windows Explorer and locate the file *richtx32.ocx* in your extension installation file.
2) If you are running a **32-bit version of Windows**, then copy *richtx32.ocx* to the directory

   `C:\Windows\System32\`

   If you are running a **64-bit version of Windows**, then copy *richtx32.ocx* to the directory

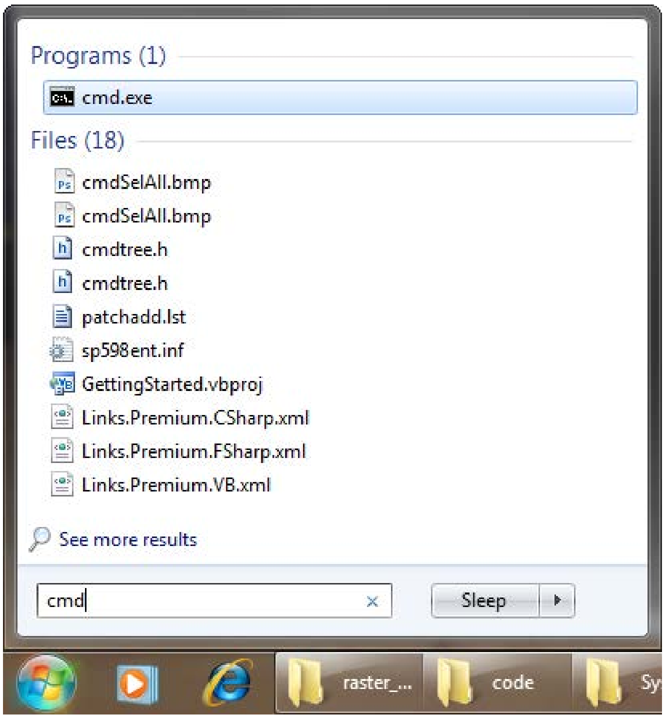
   `C:\Windows\SysWOW64\`

3) Open an "Elevated Command Prompt" window. This is the standard Windows Command Prompt window, but with administrative privileges enabled. You need these privileges enabled in order to register the OCX with Windows. **Note:** The Elevated Command Prompt opens up in the "..\windows\system32" directory, not the "..\Users\[User Name]" directory. The window title will also begin with the word "Administrator:"



a. **Method 1:** Click the “Start” button, then “All Programs”, then “Accessories” and then **right-click** on “Command Prompt” and select **Run as Administrator**.

b. **Method 2:** Click the “Start” button, and then click on the “Search Programs and Files” box. Type “cmd” and then click CONTROL+SHIFT+ENTER to open the Command window with Administrator privileges.



For more help on opening an Elevated Command Prompt, please refer to:

http://www.sevenforums.com/tutorials/783-elevated-command-prompt.html

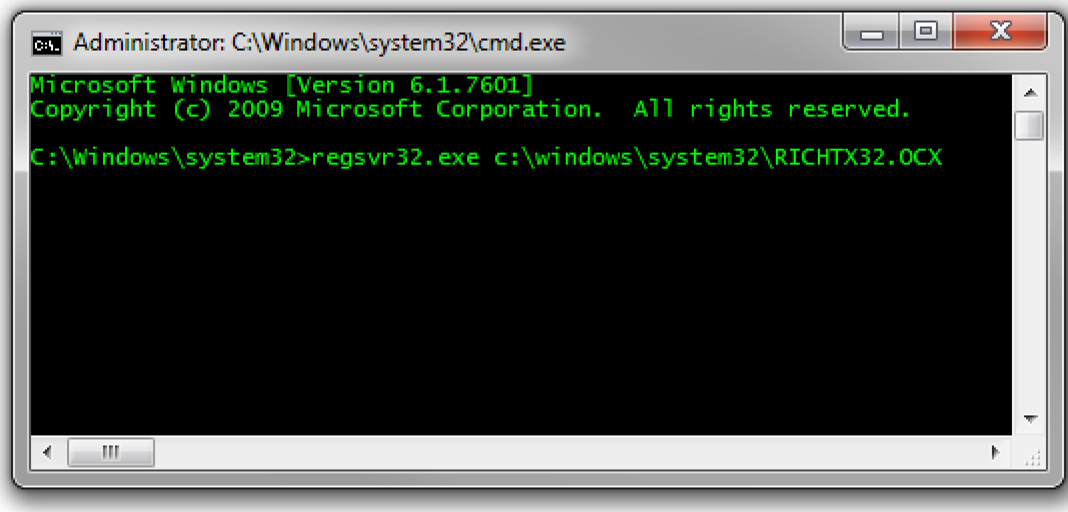
http://www.winhelponline.com/articles/158/1/How-to-open-an-elevated-Command-Prompt-in-Windows-Vista.html

Or simply do a search for “Elevated Command Prompt”.

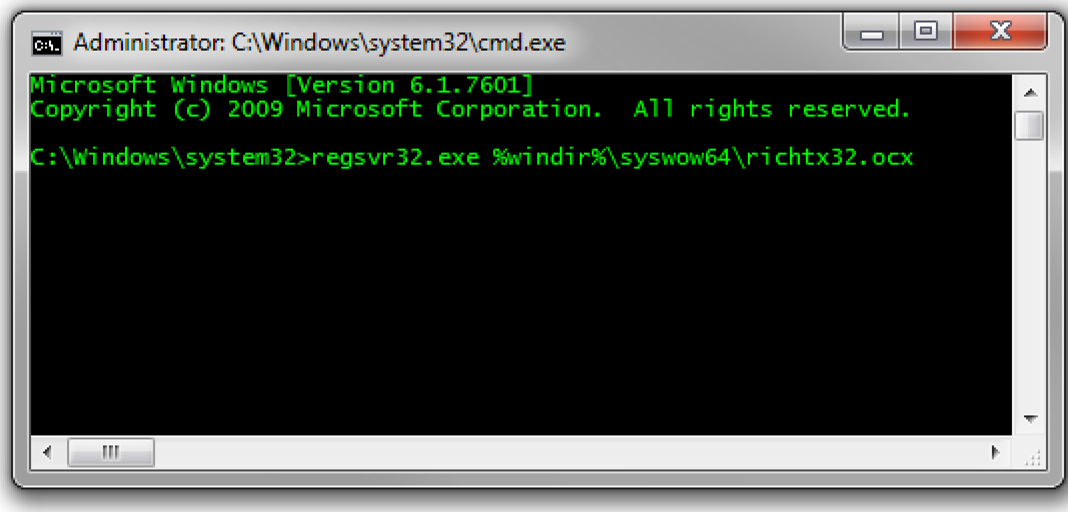
4) Register the file *richtx32.ocx* using the Windows RegSvr function:

a. If using a **32-bit version of Windows**, type the line

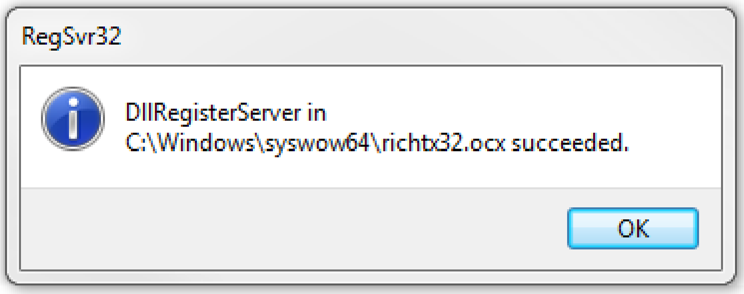
```
regsvr32.exe c:\windows\system32\richtx32.ocx
```



b. If using a **64-bit version of Windows**, type the line
`regsvr32.exe %windir%\syswow64\richtx32.ocx`

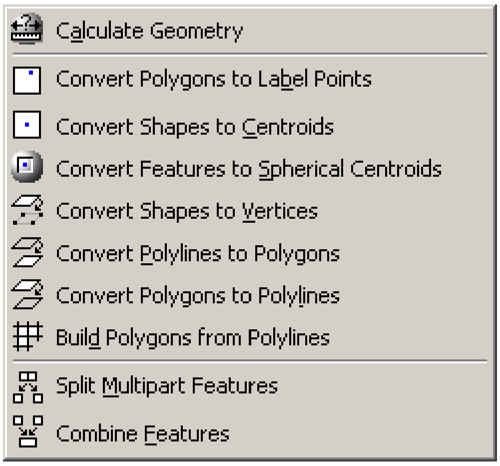


c. Click [ENTER] and you should see a message that the registration succeeded.



Tools for Graphics and Shapes

Tools for Shapes



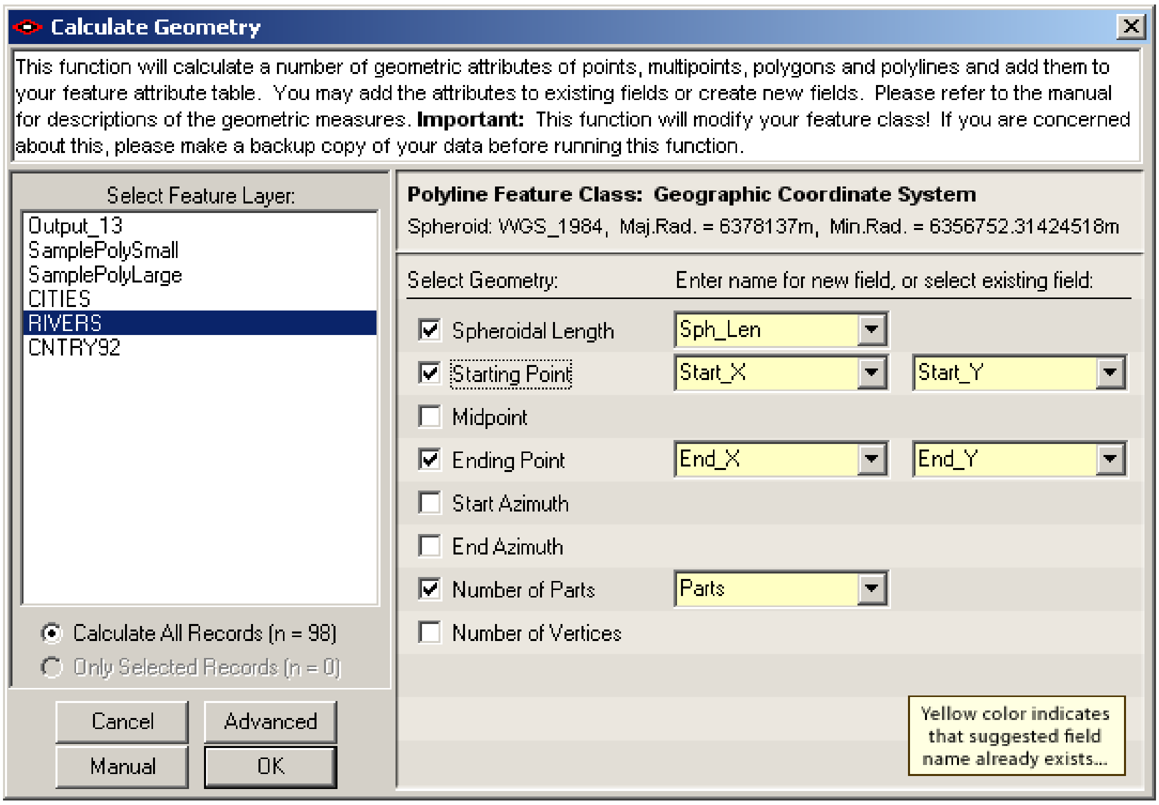
This extension includes many tools for calculating geometric attributes of vector features. Some of these tools are similar, but not identical, to existing tools available in various ArcGIS license levels. A few of these tools are not available at all in ArcGIS. All tools included with this extension are available at the ArcView license level and include features not currently offered in their standard ArcGIS counterparts.

NOTE: All of these tools will open a dialog and ask you to select the feature layer you want to work with from a list. If you click on a layer in the map Table of Contents before running this tool (i.e. so that it is selected in the TOC), then that layer will be automatically selected when the dialog opens.

Calculate Geometry

This function calculates a wide variety of geometric attributes of point, multipoint, polygon and polyline feature classes, including lengths, centroids and areas calculated on the sphere or spheroid. Some of these attributes may also be calculated using the standard ArcGIS "Calculate Geometry" function (i.e. open a feature class attribute table, right-click on the field name and select "Calculate Geometry"). However, this function provides many attributes that the standard ArcGIS function does not offer, and this function can add new fields to the attribute table automatically if necessary.

To run this function, click the menu item "Calculate Geometry":



The geometry options available will vary based on the projection of your feature class and what type of geometry the feature contains. See Table 1 for a full list of options for each type of feature class. Each geometry option is described in more detail after Table 1.

Note that if your feature class is projected, then you have the option to calculate both spherical and planimetric (projected) geometric attributes. The spherical attributes take longer to calculate but they are more accurate. Please refer to p. 42 for a comparison and discussion of spherical vs. projected geometric calculations.

You must specify a field name for each geometric attribute. You may either select one of the existing fields from the drop-down list or you may type a new field name in the box. If any of your specified field names already exist, they will be colored yellow in the dialog. You will be asked to confirm your choice if any of the field names already exist:

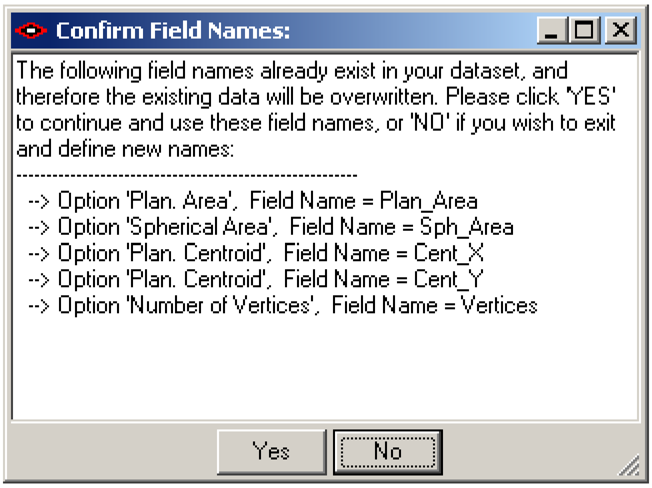


Table 1: Geometric Attributes available per feature type and projection.

Feature Type	Geometric Measure	Projection		
		Unknown	**Projected**	**Geographic**
Polygon	Planimetric Area	X	X	
	Spherical Area[1]		X[3]	X[3]
	Planimetric Centroid X	X	X	
	Planimetric Centroid Y	X	X	
	Spherical Centroid X[1]		X[4]	X
	Spherical Centroid Y[1]		X[4]	X
	Planimetric Perimeter Length	X	X	
	Spheroidal Perimeter Length[2]		X[3]	X[3]
	Number of Parts	X	X	X
	Number of Vertices	X	X	X
Polyline	Planimetric Length	X	X	
	Spheroidal Length[2]		X[3]	X[3]
	Start Point X	X	X	X
	Start Point Y	X	X	X
	Planimetric Midpoint X	X	X	
	Planimetric Midpoint Y	X	X	
	Spheroidal Midpoint X[2]		X[4]	X
	Spheroidal Midpoint Y[2]		X[4]	X
	End Point X	X	X	X
	End Point Y	X	X	X
	Projected Azimuth	X	X	

	Spheroidal Beginning Azimuth		X[4]	X[4]
	Spheroidal Ending Azimuth		X[4]	X[4]
	Number of Parts	X	X	X
	Number of Vertices	X	X	X
Point	X-Coordinate	X	X	
	Y-Coordinate	X	X	
	Latitude		X	X
	Longitude		X	X
Multipoint	Planimetric Centroid X	X	X	
	Planimetric Centroid Y	X	X	
	Spherical Centroid Latitude[1]		X[4]	X
	Spherical Centroid Longitude[1]		X[4]	X
	Number of Points	X	X	X

[1] Spherical measures based on a sphere with volume equal to selected spheroid volume (see p. 66).
[2] Spheroidal measures calculated from selected spheroid using Vincenty's equations (see p. 46).
[3] Value calculated from sphere or spheroid, but reported in meters or square meters
[4] Value calculated from sphere or spheroid, but reported in projected coordinates

NOTE: Polygon perimeters and polyline lengths and centroids are calculated on the spheroid of the original data (see p. 46). Polygon spherical areas and centroids are calculated on a sphere, regardless of whether the data projection uses a sphere or spheroid. If the data projection uses a spheroid, then the areas and centroids are calculated from a sphere containing the same volume as the original spheroid (see p. 66). Spheroids model the shape of the earth slightly better than spheres, and therefore I tried to go with the most accurate method I could find. In the case of polygon areas, however, I decided to settle for spherical rather than spheroidal methods because:

1) Even spheroids do not model the earth's surface perfectly, so there is bound to be some error regardless of the method, and
2) Topographic variation over the surface of the earth will more than wash out any marginal gains in accuracy produced by spheroidal methods, and
3) Most important of all, I simply could not find any existing formulas to help me calculate the area of a polygon over a spheroid, and I am not enough of a mathematician to easily figure them out for myself. Given (1) and (2) above, I felt that the spherical methods should provide sufficient accuracy.

Polygon Measures:

1. Planimetric Area: The area of the polygon assuming that the polygon lies on a flat surface, calculated from the coordinates of the polygon vertices. Values are reported in square map units. This is the standard area value reported by most analytical functions. The accuracy can range from very high to very low depending on the projection used. **NOTE:** Planimetric area should **not** be calculated on geographically-projected polygons! Latitude/Longitude values are polar coordinates (i.e. they reflect angular units) rather than Cartesian coordinates (i.e. X,Y units), and the methods used to calculate area from Cartesian coordinates are not appropriate for polar coordinates. A polygon area reported in "square degrees" is completely nonsensical.
2. Spherical Area: The area of the polygon assuming that it is draped over a sphere. Area values are reported in square meters. This area can only be calculated on a geographic polygon. If the original polygon is projected, it will be unprojected to latitude/longitude coordinates before the area is calculated. This area value should be very close to accurate regardless of the projection of the data, and features from different projections can therefore be compared directly without concern for projection. Please refer to p. 49 for

geometric methods used in this function, and p. 42 for a comparison between spherical area and projected area.

3. Planimetric Centroid X and Y Coordinates: The coordinates of the polygon center of mass, assuming that the polygon lies on a flat surface and has vertices in Cartesian coordinates. The methods used to calculate the planimetric centroid are only appropriate if the polygon vertices are in Cartesian coordinates (i.e. if the polygon is projected), and therefore this option is not available if the polygon is in a geographic projection (i.e. if the vertices are in polar coordinates).
4. Spherical Centroid X and Y Coordinates: The coordinates of the polygon center of mass, assuming that the polygon lies on a sphere and has vertices in polar coordinates (i.e. Latitude/Longitude values). These values can only be calculated using geographic coordinates so this option will not be offered if the projection is unknown. If the polygon is projected, then it will be unprojected to latitude/longitude values before the centroid is calculated. The centroid will then be projected back into the original polygon projection in order to get the projected X and Y Coordinates. Please refer to p. 49 for geometric methods used in this function, and p. 42 for a comparison between spherical centroids and projected centroids.
5. Planimetric Perimeter Length: The length of the polygon boundaries, calculated in the units of the map projection.
6. Spheroidal Perimeter Length: The length of the polygon boundaries as they lay over the data sphere or spheroid. Spheroidal (i.e. ellipsoidal) lengths are marginally more accurate than spherical lengths and much more accurate than projected lengths. Length values are reported in meters. Please refer to p. 46 for a description of methods used in this function.
7. Number of Parts: Some polygons are composed of multiple parts. This value simply reports the number of separate sub-polygons in each polygon feature.
8. Number of Vertices: The number of vertices in each polygon feature.

Polyline Measures:

1. Planimetric Length: The length of the polyline in the units of the map projection. This option is only appropriate for polygons in Cartesian coordinates and therefore is not available if your feature class is in a geographic projection.
2. Spheroidal Length: The length of the polyline as it truly lies on the surface of the data sphere or spheroid. Length values are reported in meters. See p. 46 for methods used in this function.
3. Start Point X and Y Coordinates: The coordinates of the polyline starting point, in the units of the polyline coordinate system.
4. Planimetric Midpoint X and Y Coordinates: The coordinates of the midpoint along the polyline, in the units of the polyline projection. The methods used to identify the planimetric midpoint are only appropriate for projected polylines, so this option is not available for geographic polylines.
5. Spheroidal Midpoint X and Y Coordinates: The coordinates of the midpoint of the polyline as it lies on the data sphere or spheroid. The methods used to identify the spheroidal midpoint are not appropriate for projected data, so projected polylines will be unprojected to latitude/longitude coordinates before this point will be calculated. This option is not available if the polyline projection is unknown. See p. 46 for methods used in this function.
6. End Point X and Y Coordinates: The coordinates of the ending point of the polyline, in the units of the polyline coordinate system.

7. Projected Azimuth: The average azimuth of the polyline, calculated as the azimuth (in degrees clockwise from North) of the straight line connecting the beginning of the polyline to the end of the polyline (see p. 57 for methods).

8. Spheroidal Starting Azimuth: The beginning azimuth of the geodesic curve connecting the beginning of the polyline to the end of the polyline. When calculating azimuth on a spheroid, the azimuth almost always changes constantly over the length of a geodesic connecting two points. This function calculates the azimuth of the geodesic as it leaves the starting point. See p. 46 for details.

9. Spheroidal Ending Azimuth: The ending azimuth of the geodesic curve connecting the beginning of the polyline to the end of the polyline. When calculating azimuth on a spheroid, the azimuth almost always changes constantly over the length of a geodesic connecting two points. This function calculates the azimuth of the geodesic as it reaches the ending point. See p. 46 for details.

10. Number of Parts: Some polylines are composed of multiple parts. This value simply reports the number of separate sub-polylines in each polyline feature.

11. Number of vertices: The number of vertices in each polygon feature.

Point Measures:

1. X and Y Coordinates: The X and Y coordinates in the Cartesian coordinate system of the feature class projection.

2. Latitude and Longitude: The polar coordinates (i.e. latitude and longitude) of the point. If the point is projected, then the point is unprojected to the projection geographic coordinate system before these values are identified.

Multipoint Measures:

1. Planimetric Centroid X and Y Coordinates: The coordinates of the multipoint center of mass, assuming the multipoint lies on a flat surface and has points in Cartesian coordinates. The methods used to calculate the planimetric centroid are only appropriate if the points are in Cartesian coordinates (i.e. if the multipoint is projected), and therefore this option is not available if the multipoint is in a geographic projection (i.e. if the points are in polar coordinates).

2. Spherical Centroid X and Y Coordinates: The coordinates of the multipoint center of mass, assuming the multipoint lies on a sphere and has points in polar coordinates (i.e. Latitude/Longitude values). These values can only be calculated using geographic coordinates, so this option will not be offered if the projection is unknown. If the multipoint is projected, then it will be unprojected to latitude/longitude coordinates before the centroid is calculated. The centroid will then be projected back into the original multipoint projection in order to get the projected X and Y Coordinates. For a review of the methods used in this function, see the discussions on calculating the polygon weighted mean centroid (p. 52) and triangle centroid (p. 58)

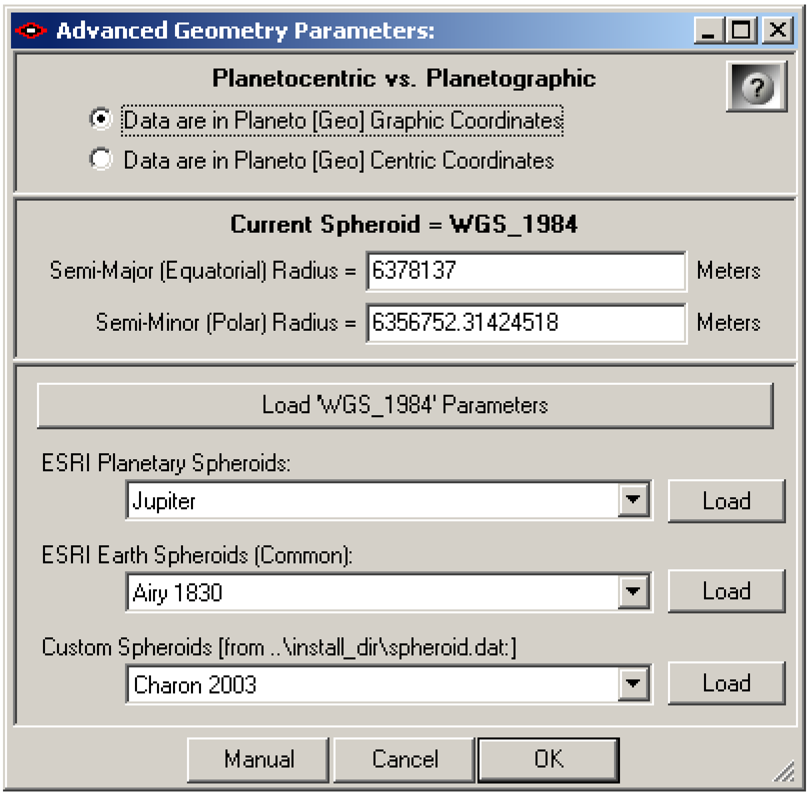
Advanced Features:

Note: The advanced features only apply to geographically-projected multipoint, polyline or polygon feature layers. The "Advanced" button will be disabled if the selected layer is not geographically projected or if it is a point layer.

Spherical and spheroidal measures are calculated on the sphere or spheroid of the data. As you select a feature layer in the "Calculate Geometry" dialog, notice that the spheroid parameters appear automatically above the list of geometry output options. In the illustration above, the "RIVERS" feature layer is seen to be in a Geographic Coordinate System based on the WGS 84 spheroid, with semi-major axis radius = 6378137m and semi-minor axis radius = 6356752.31424518m.

However, you are not forced to use this spheroid, and neither are you forced to assume that the data are in geographic (or planetographic) coordinates (see p. 67 for a comparison of planetocentric and planetographic coordinate systems).

If you wish to modify the spheroid parameters, click the “Advanced” button to open the “Advanced Geometry Parameters” dialog:



If your data are in planetocentric coordinates, then you may designate it here and the tool will adjust the analytical algorithms accordingly. You may also enter any semi-major or semi-minor radii values you wish, or enter them automatically by selecting from lists of predefined terrestrial or planetary spheroids.

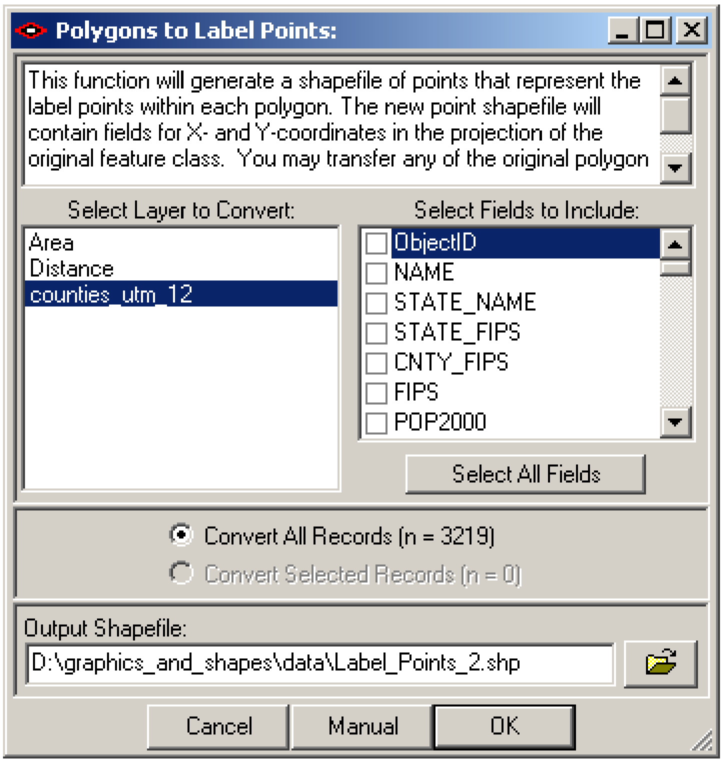
Note: If you wish to use a predefined spheroid, then select the spheroid from one of the three lists and then click the “Load” button to the right of the list. Simply selecting the named spheroid in the list will not change the current spheroid parameters.

Convert Polygons to Label Points

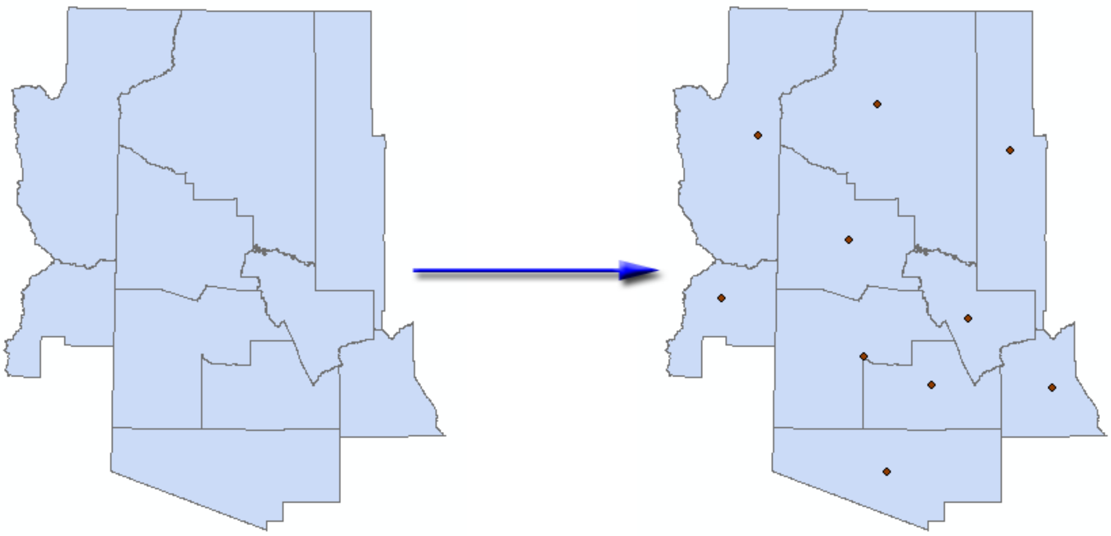
This function generates a point shapefile of label points for each polygon. Each polygon has a single label point that is guaranteed to be somewhere inside the polygon. Multipart polygons will have a single label point that is located inside one of the sub-polygons. You have the option to calculate label points for all polygons or only the selected set of polygons.

This function is similar to the ArcInfo-level Geoprocessing tool **FeatureToPoint**, using the option “Label Point”.

To run the function, click the menu item “Convert Polygons to Label Points”:



Select the polygon feature class you wish to convert, then select any or all fields to transfer to your new point shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new point shapefile will be added to your current active map frame.



NOTE: Regardless of the fields you choose to transfer over, your new point shapefile will contain additional fields containing unique ID values and X- and Y-coordinates.

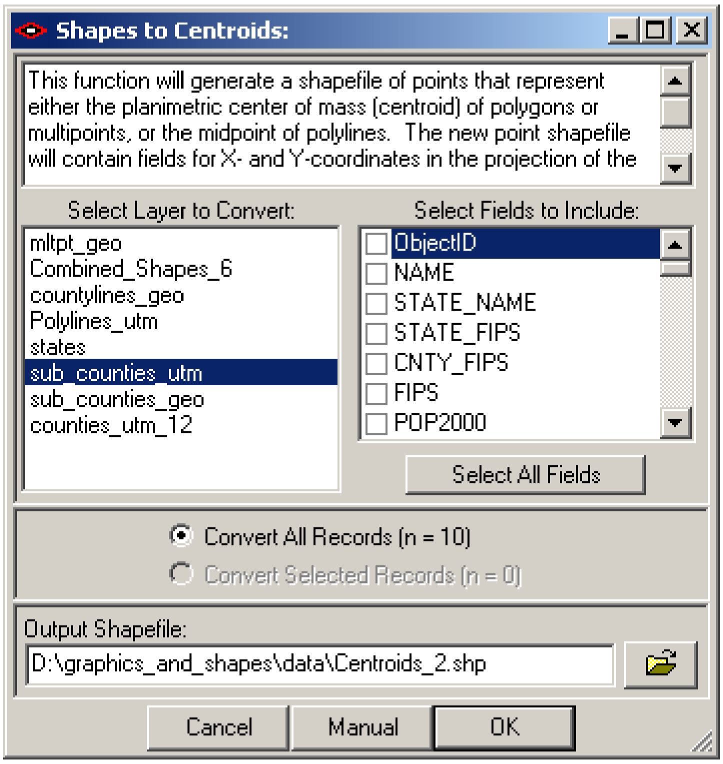
Convert Shapes to Centroids

This function generates a point shapefile of the centers of mass for each feature. This function works with Polygons, Polylines and Multipoints. For polygons, this function will generate the center of mass of that polygon, based on the projection of that polygon. For polylines, this function will return the midpoint of each line. For multipoints, this function will return the average location of all the points. In general, this function is similar to the ArcInfo-level Geoprocessing tool **FeatureToPoint**, using the option "Centroid".

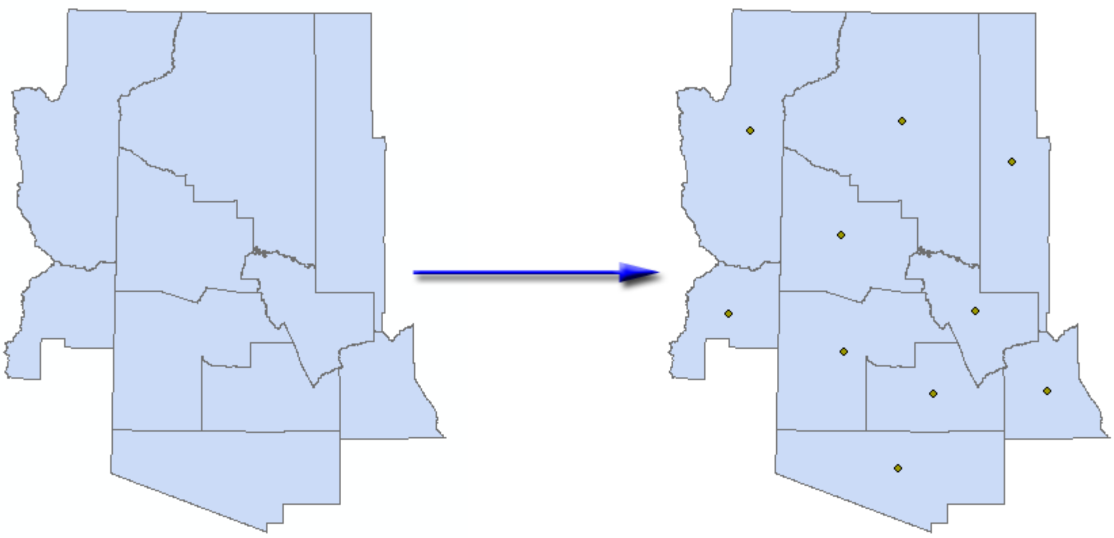
For polygons, the center of mass may not actually be inside the polygon border. If you need to generate points that are guaranteed to be within the polygon border, use the "Convert Polygons to Label Points" function above.

NOTE: The output produced by this function is only as accurate as the projection will allow it to be. Some projections are better than others at generating the true center of mass of a feature. If you want to guarantee an accurate center of mass, regardless of the projection, consider using the "Convert Features to Spherical Centroids", which calculates the center of mass directly on the sphere rather than on the projected flat surface.

To run the function, click the menu item "Convert Shapes to Centroids":



Select the feature class you wish to convert, then select any or all fields to transfer to your new centroid point shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new point shapefile will be added to your current active map frame.



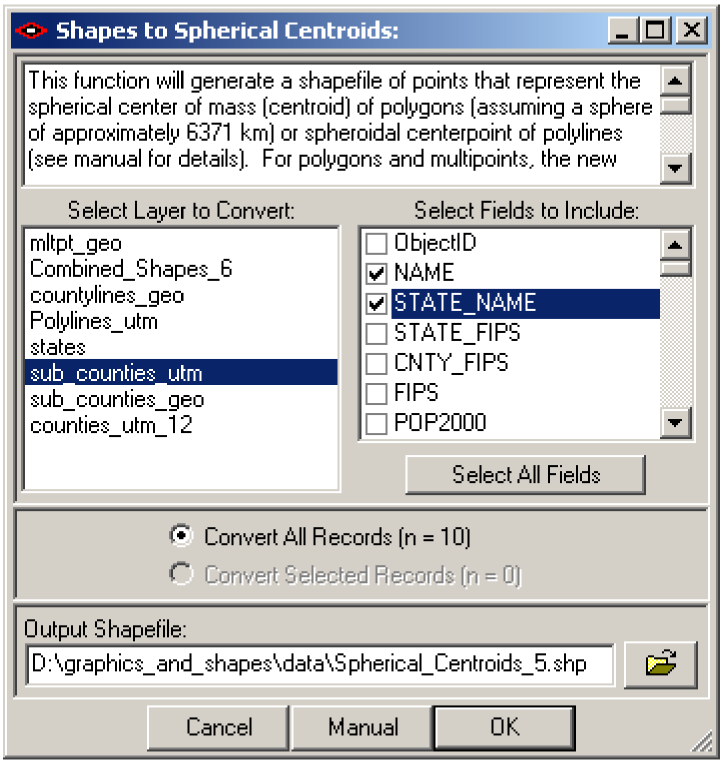
NOTE: Regardless of the fields you choose to transfer over, your new point shapefile will contain additional fields containing unique ID values and X- and Y-coordinates.

Convert Shapes to Spherical Centroids

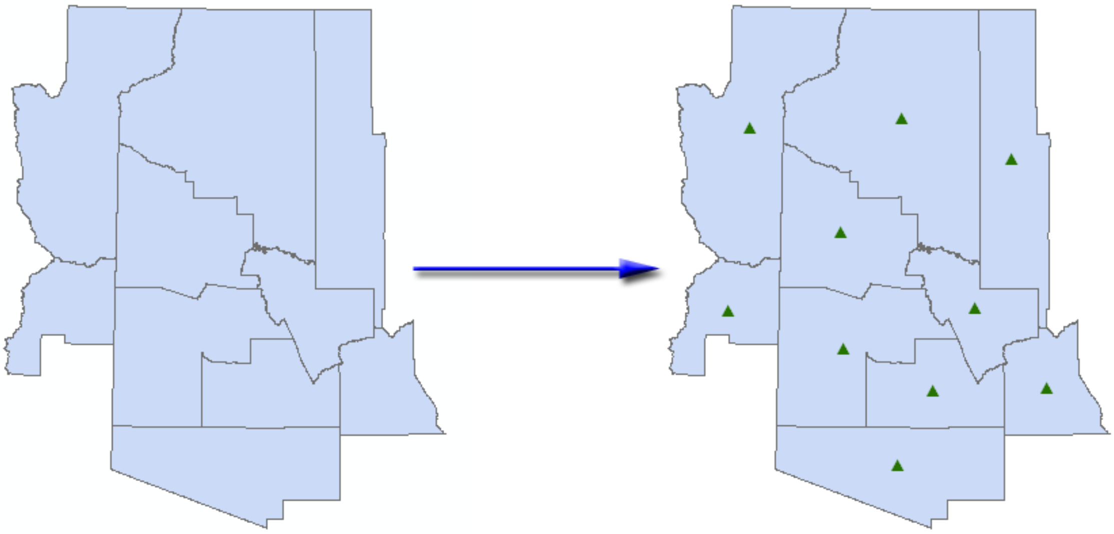
This function generates a point shapefile of the centers of mass for each feature using spherical and spheroidal functions to calculate the true centroid positions directly on the sphere or spheroid. This function works with polygons, polylines and multipoints. For polygons, this function will generate the surface center of mass of that polygon using spherical geometry (see p. 49). For polylines, this function will return the midpoint of each line using Vincenty's equations to calculate exact midpoints on the ellipsoid (see p. 47). For multipoints, this function will return the average location of all the points using spherical geometry (see discussion on polygon weighted mean centroid [p. 52] and triangle centroid [p. 58]). As far as I know, there is no standard ArcGIS tool that performs a similar function. See p. 42 for a comparison and discussion of spherical vs. projected geometric calculations.

For polygons, the center of mass may not actually be inside the polygon border. If you need to generate points that are guaranteed to be within the polygon border, use the "Convert Polygons to Label Points" function above.

To run the function, click the menu item "Convert Shapes to Spherical Centroids":



Select the feature class you wish to convert, then select any or all fields to transfer to your new centroid point shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new point shapefile will be added to your current active map frame.



NOTE: The new point shapefile will be in geographic coordinates, using the geographic basis of whatever projection the original feature class is in. For example, if the original feature class is in a UTM NAD27 projection, then the spherical centroid feature class will be in the NAD27 geographic coordinate system. If the original feature class is in a geographic coordinate system, then the new centroid feature class will be in the same coordinate system.

NOTE: This function will not work if the coordinate system of your data is unknown. In this case, the function will alert you to the problem and then bail out.

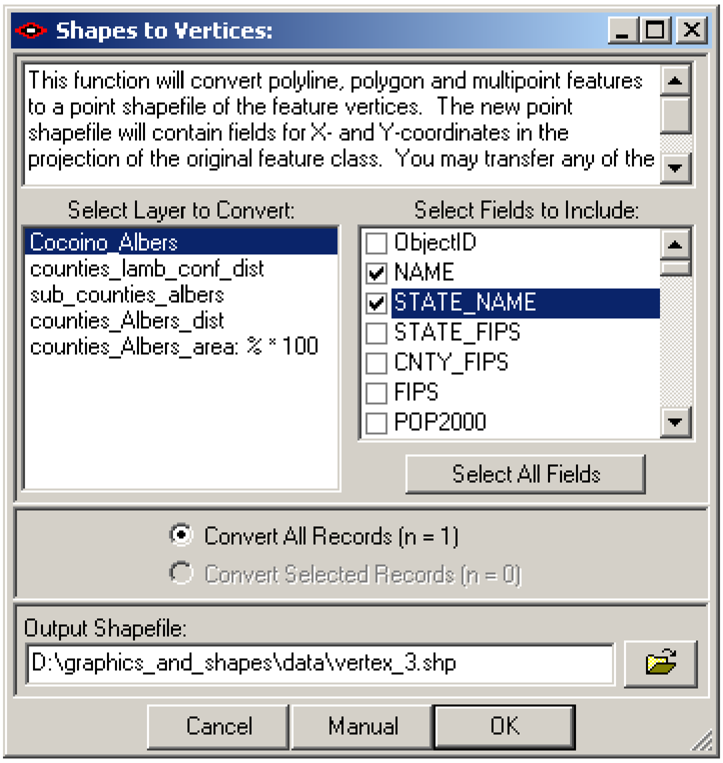
NOTE: Polygon and multipoint centroids are calculated on a sphere of approximately 6371km (see p. 66). Polyline midpoints are calculated on the WGS 84 Spheroid (see p. 47)

NOTE: Regardless of the fields you choose to transfer over, your new point shapefile will contain additional fields containing unique ID values and X- and Y-coordinates. Because the new shapefile is in a geographic projection, the X- and Y-coordinates will reflect the latitude and longitude of those points.

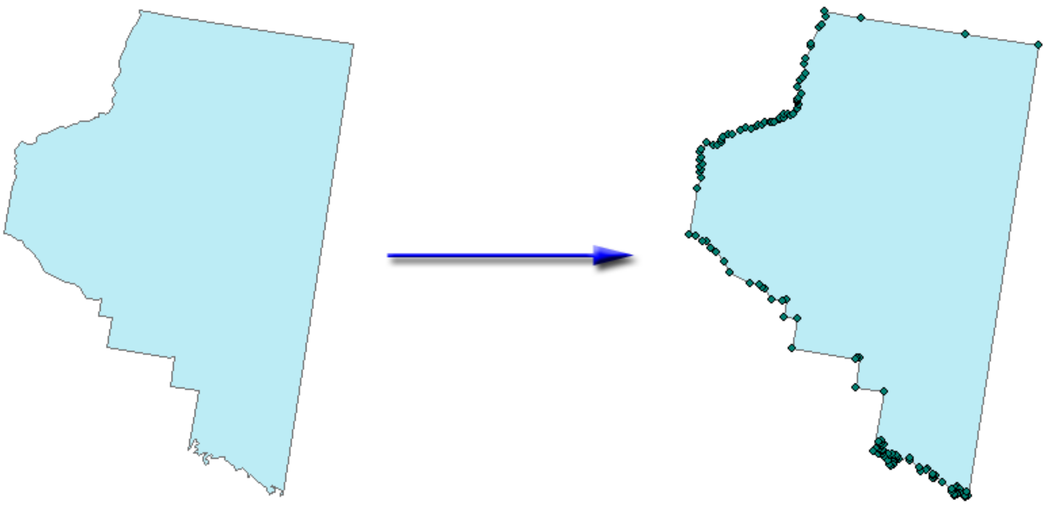
Convert Shapes to Vertices

This function converts polygon, polyline and multipoint features into a point shapefile of vertices. This is similar in function to the ArcInfo-level Geoprocessing tool **FeatureVerticesToPoints**, using the option “All”.

To run the function, click the menu item “Convert Shapes to Vertices”:



Select the feature class you wish to convert, then select any or all fields to transfer to your new vertex point shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new point shapefile will be added to your current active map frame.



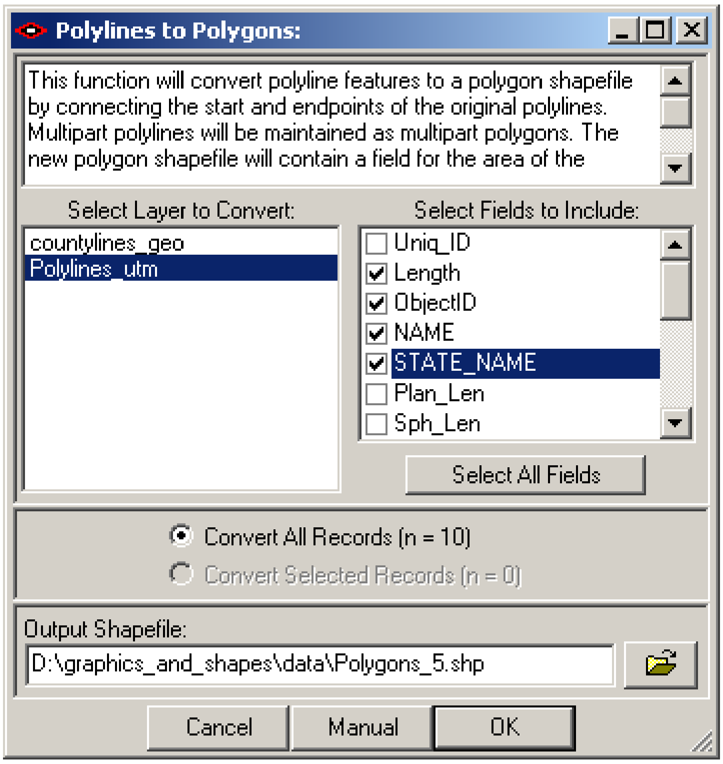
NOTE: Regardless of the fields you choose to transfer over, your new point shapefile will contain additional fields containing unique ID values and X- and Y-coordinates.

Convert Polylines to Polygons

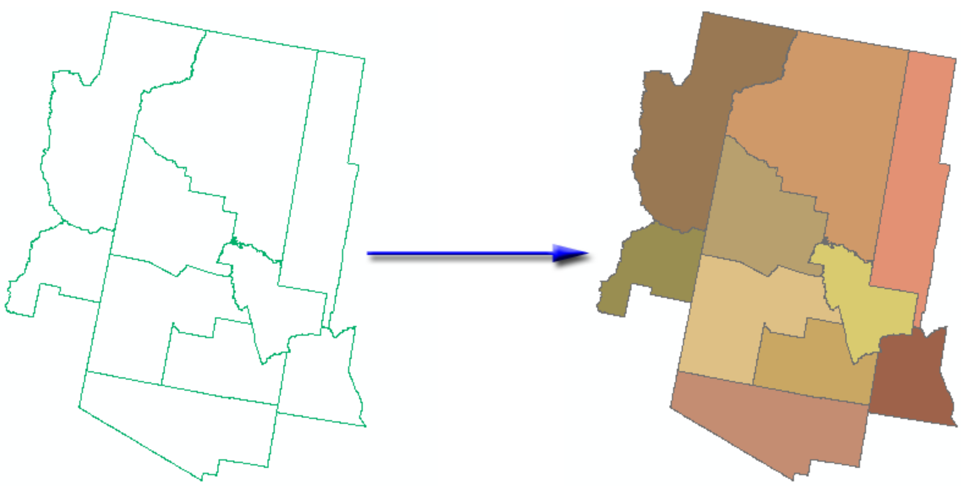
This function converts Polyline feature classes to Polygon shapefiles by connecting the starting and ending points of each polyline. In cases of multipart polylines, the new polygon may also be multipart. If a multipart polyline produces multiple overlapping sub-polygons, then the overlapping polygons will be dissolved into each other. This function is similar to the ArcInfo-level Geoprocessing tool **FeatureToPolygon**, except that it does not offer the option to insert attributes from a separate label feature class.

NOTE: This function is not the same as building polygons by identifying regions bounded by a set of intersecting polylines! The "Build Polygons" function is described on p. 30. This function converts polylines to polygons by simply connecting the end of the polyline back to its beginning.

To run this function, click the menu item "Convert Polylines to Polygons":



Select the polyline feature class you wish to convert, then select any or all fields to transfer to your new polygon shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new polygon shapefile will be added to your current active map frame.

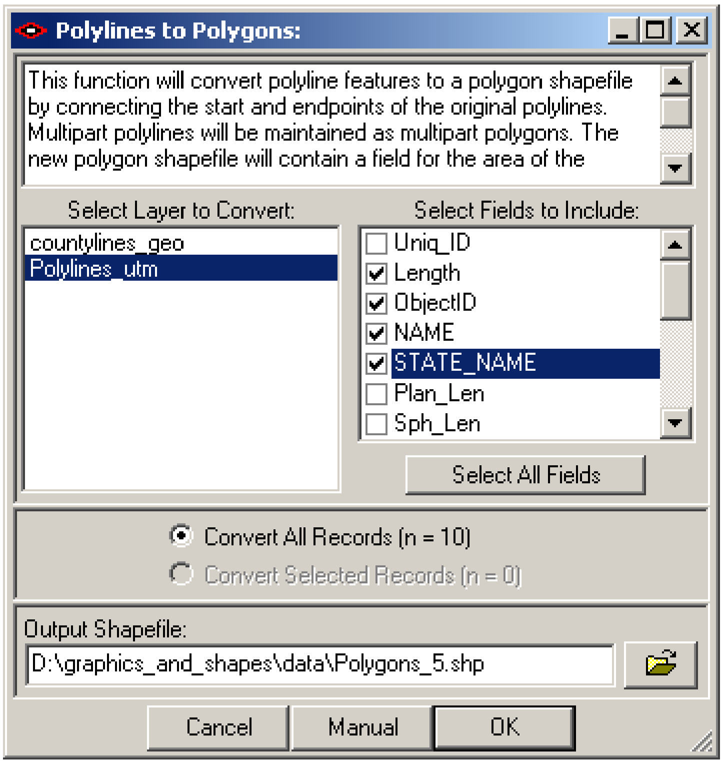


NOTE: Regardless of the fields you choose to transfer over, your new polygon shapefile will contain additional fields containing unique ID and area values for each polygon. Area values will be calculated in the coordinate system of the data. If your data are in geographic coordinates, then these area values will be very incorrect and you may wish to recalculate them using the "Calculate Geometry" function (see p. **17**).

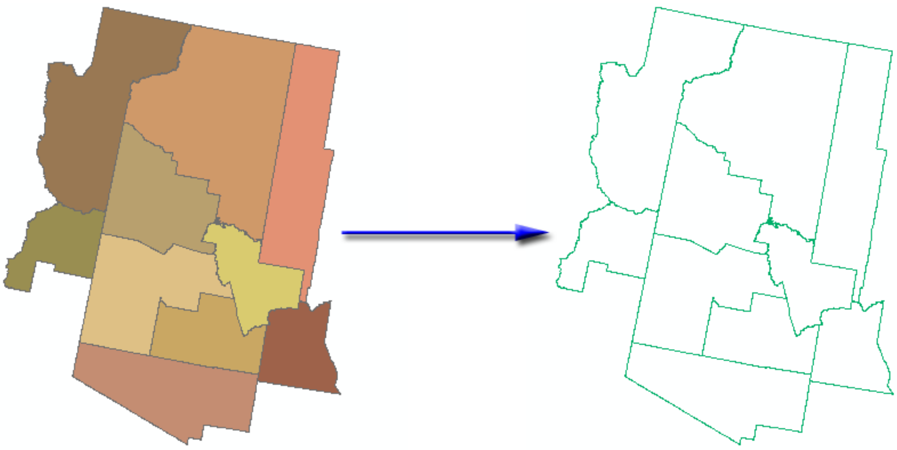
Convert Polygons to Polylines

This function converts Polygon feature classes to Polyline shapefiles by extracting the outer and inner rings of each polygon. In cases of multipart polygons, the new polyline will also be multipart. This function is similar to the ArcInfo-level Geoprocessing tool **FeatureToLine**.

To run this function, click the menu item "Convert Polygons to Polylines":



Select the polyline feature class you wish to convert, then select any or all fields to transfer to your new polygon shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new polyline shapefile will be added to your current active map frame.



NOTE: Regardless of the fields you choose to transfer over, your new polyline shapefile will contain additional fields containing unique ID and length values for each polyline. Length values will be calculated in the coordinate system of the data. If your data are in geographic coordinates, then these length values will be very incorrect and you may wish to recalculate them using the "Calculate Geometry" function (see p. **17**).

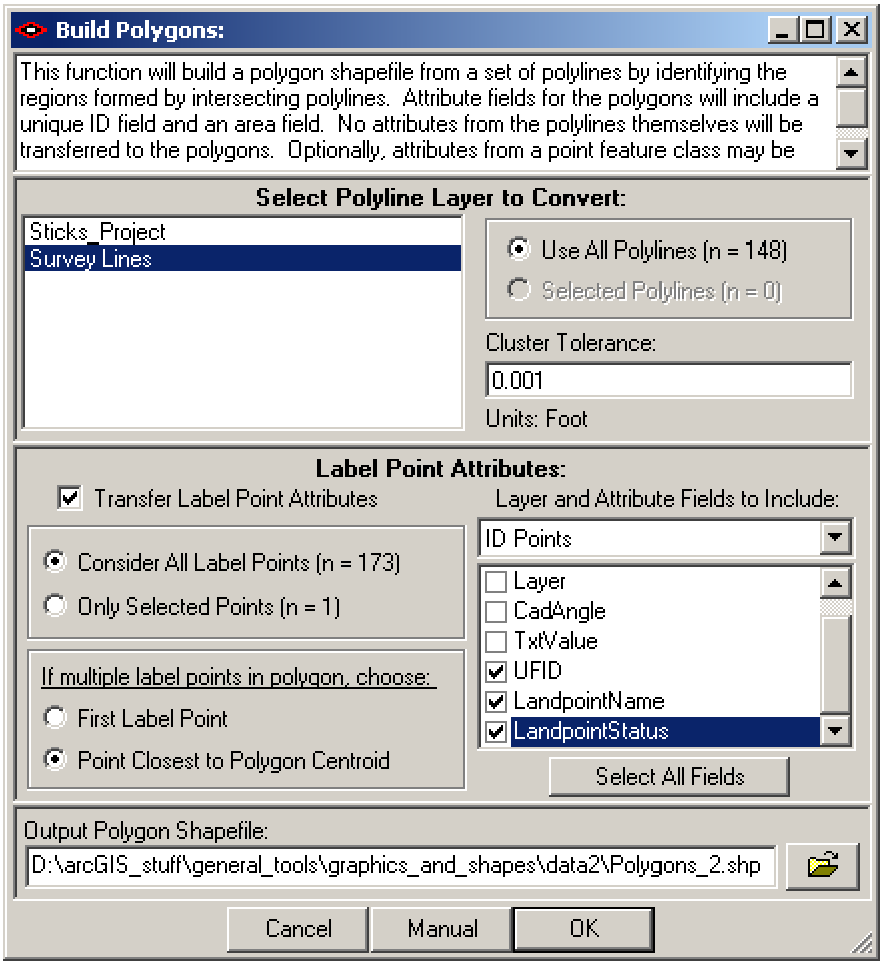
Build Polygons from Polylines

IMPORTANT: This function will only run on ArcGIS 9.2 or newer.

This function identifies regions bounded by a set of intersecting polylines and saves them to a polygon shapefile. **NOTE:** No attribute data from the polyline feature class will be transferred to the polygons because there is no way to know which polyline would represent a polygon when multiple polylines are used to create the polygon. **HOWEVER**, you can optionally select a point feature class to use to extract attribute data. This tool has a few options:

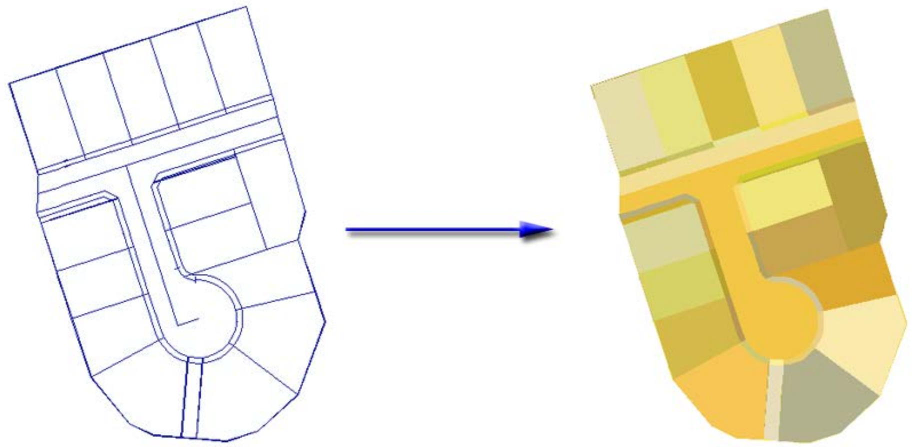
1) If any of your polylines are selected, you can choose to build your polygons using either all polylines, or only the selected polylines.
2) If you add attribute data from a point feature class, then you can choose to consider either all points or only selected points to extract data from.
3) If you add attribute data from a point feature class, and if multiple points intersect the polygon, you can choose to use either the first intersecting point or the intersecting point closest to the polygon centroid to extract data from.

To run this function, click the menu item "Build Polygons from Polylines":



Select the polyline feature class you wish to use. The **Cluster Tolerance** allows features that are very close to each other to be snapped together if they are within the specified distance from each other, and helps to prevent sliver polygons and cases where polygons fail to be generated because a polyline falls just short of an intersection point. All dangling nodes and unconnected polylines will be deleted from the final polygon shapefile.

If you wish to add attribute data from a point feature class, then select the point feature class and any attribute fields you want to transfer data from. If any features are selected, you may choose to consider only those selected features in the analysis. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new polygon shapefile will be added to your current active map frame.

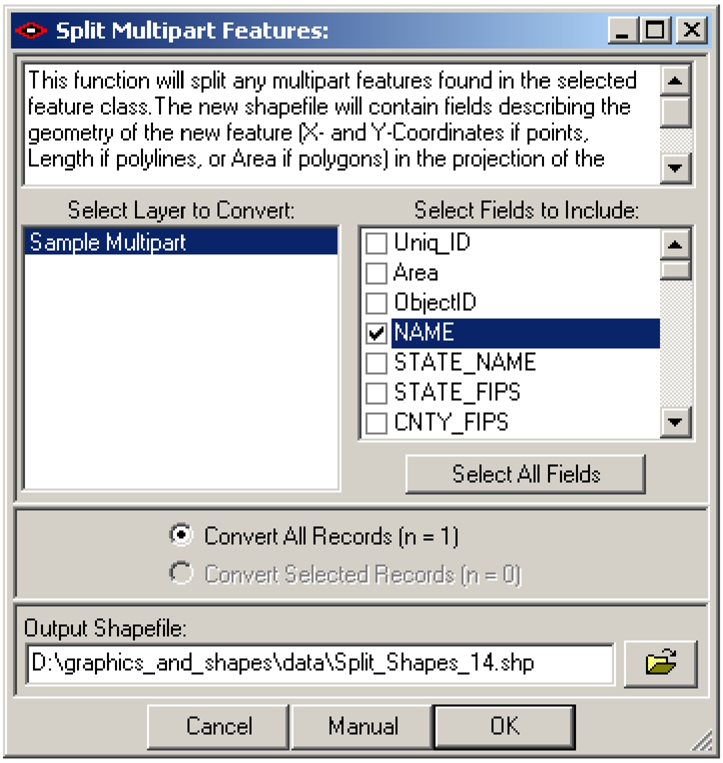


NOTE: Regardless of the point attribute fields you choose to transfer over, your new polygon shapefile will contain additional fields containing unique ID and area values for each polygon. Area values will be calculated in the coordinate system of the data. If your data are in geographic coordinates, then these area values will be very incorrect and you may wish to recalculate them using the "Calculate Geometry" function (see p. **17**).

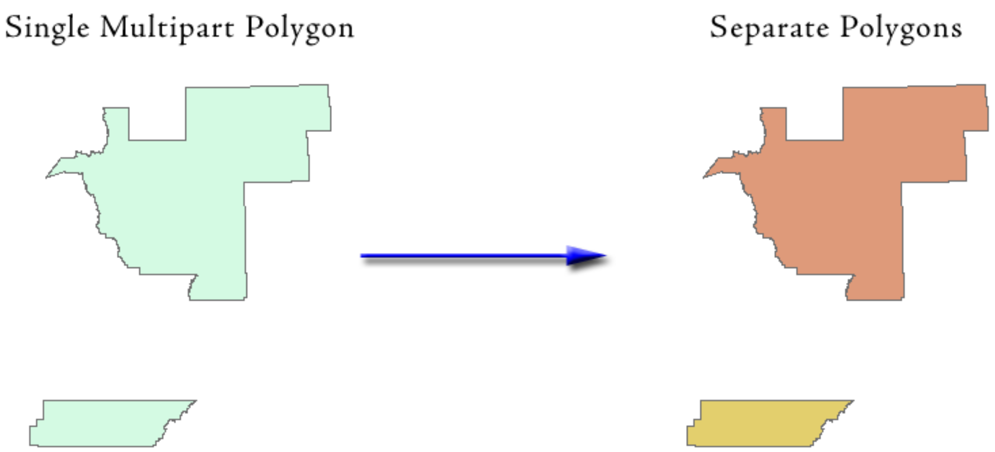
Split Multipart Features

This function splits all multipart polylines and polygons into single features, and breaks up multipoints into separate point features. This function is similar to the ArcView-level Geoprocessing tool "**MultipartToSinglepart**". Unlike the Geoprocessing tool, this function will also automatically recalculate the geometry of the split features.

To run this function, click the menu item "Split Multipart Features":



Select the feature class you wish to convert, then select any or all fields to transfer to your new feature shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new polygon shapefile will be added to your current active map frame.



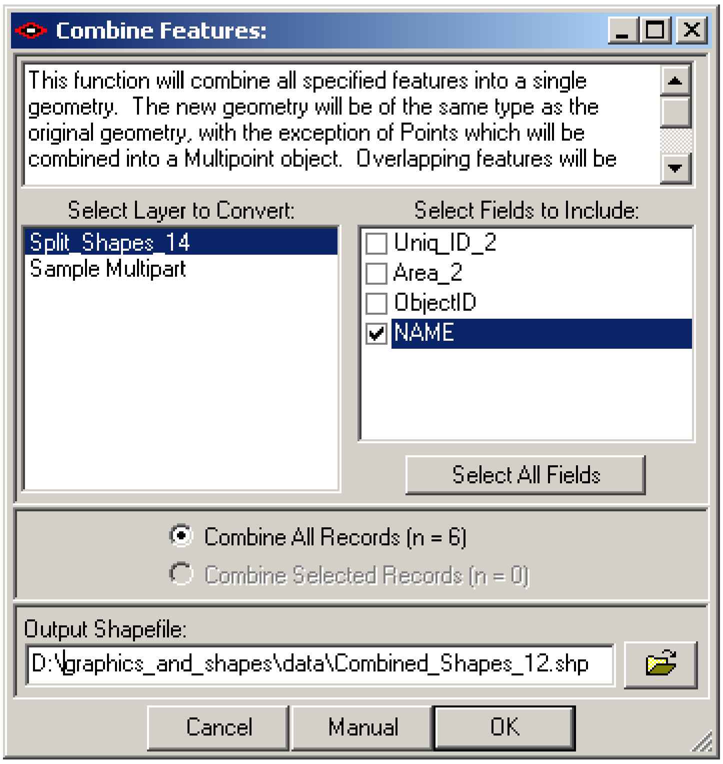
NOTE: Regardless of the fields you choose to transfer over, your new shapefile will contain additional fields containing unique ID and geometry values for each feature. Polyline shapefiles will get a field for "Length", polygon shapefiles will get a field for "Area" and point shapefiles will get fields for X- and Y-coordinates. Geometry values will be calculated in the coordinate system of the data. If your data are in geographic coordinates, then these values may be incorrect and you may wish to recalculate them using the "Calculate Geometry" function (see p. **17**).

Combine Features

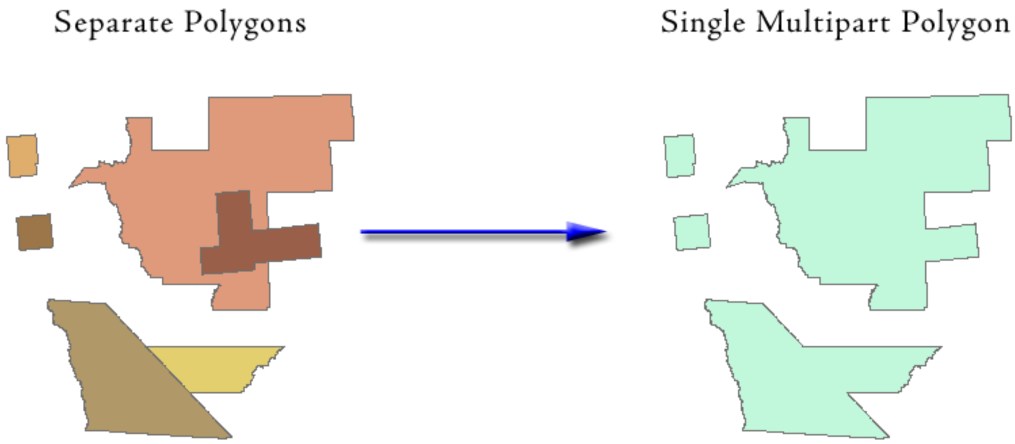
This function combines either all features or all selected features into a single feature and adds that feature to a new shapefile. I am not aware of any ArcGIS Geoprocessing tools that do this exact function, although the ArcView-level "Dissolve" and "Eliminate" tools come close. "Dissolve" combines features based on attribute values, and "Eliminate" combines all selected features into adjacent features. This "Combine Features" function is intended for cases where you want to combine a specific set of features into a single object.

If any of the features overlap, they will be dissolved into each other before being added to the new shapefile. If you are combining points into a Multipoint object, then all points with the same coordinates will be combined into a single point.

To run this function, click the menu item "Combine Features":

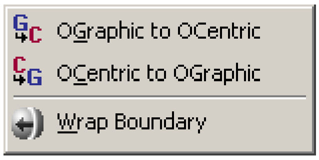


Select the feature class you wish to combine, then select any or all fields to transfer to your new feature shapefile. If any records are selected, you may choose to convert only those selected records. Finally, specify the path name for your output shapefile and click 'OK'. Upon completion, your new shapefile will be added to your current active map frame.



NOTE: Regardless of the fields you choose to transfer over, your new shapefile will contain additional fields containing unique ID and geometry values for each feature. Polyline shapefiles will get a field for "Length", polygon shapefiles will get a field for "Area" and multipoint shapefiles will get a field for the point count. Geometry values will be calculated in the coordinate system of the data. If your data are in geographic coordinates, then these values may be incorrect and you may wish to recalculate them using the "Calculate Geometry" function (see p. 17).

OCentric/OGraphic Geodesic Tools



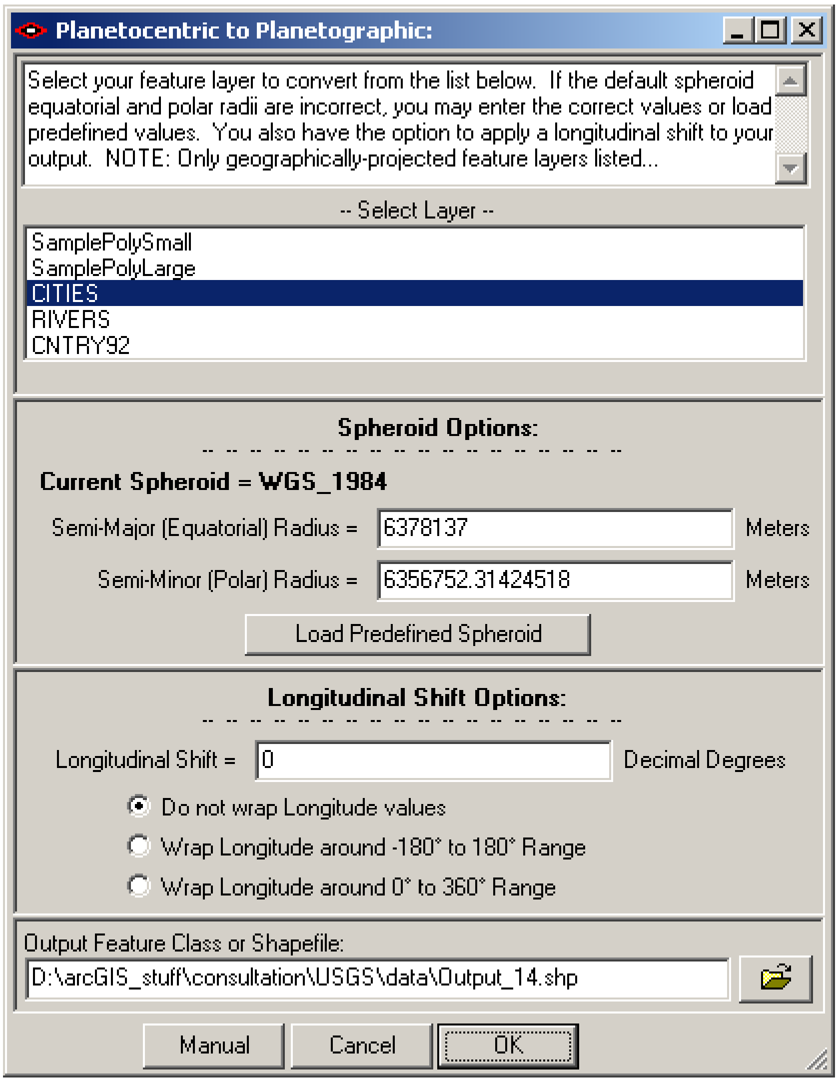
Note: These tools were developed with funding and guidance from the Astrogeology Team of the U.S. Geological Survey (see http://astrogeology.usgs.gov/). The author gratefully acknowledges their contribution.

OCentric / OGraphic Transformations

These functions transform a feature layer between planetographic and planetocentric coordinate systems, with options to specify the spheroid parameters and to apply a longitude shift. **Note:** This function only applies to geographically-projected data. You will not have the option to transform feature layers that are not in geographic coordinates.

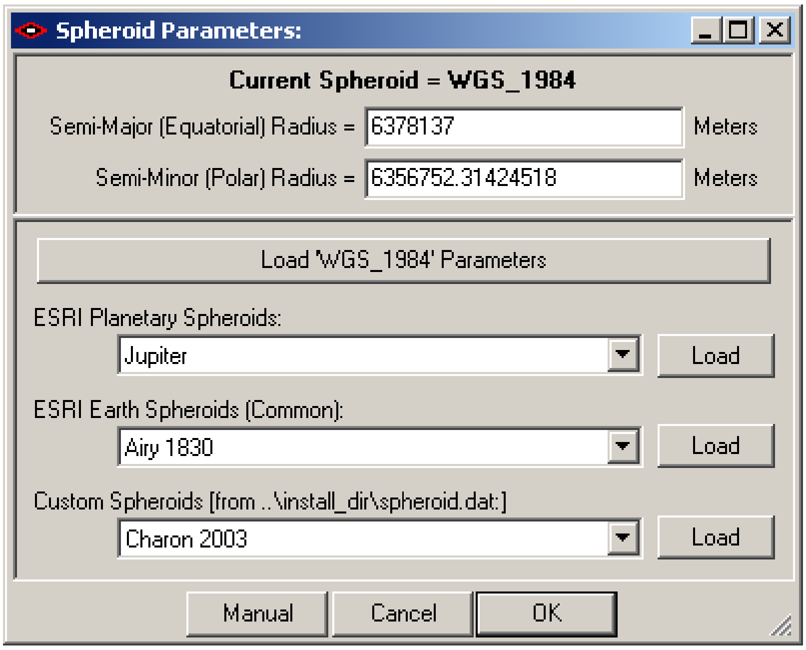
An explanation of planetocentric vs. planetographic latitudes is available on p. 67 of this manual.

Click either the "OCentric to OGraphic" or "OGraphic to OCentric" menu item to open the appropriate dialog:



Spheroid Options

The "Spheroid Options" section will automatically fill with the spheroid parameters of the currently selected feature layer. If you wish to specify a different spheroid, then click the "Load Predefined Spheroid" to open the "Spheroid Parameters" dialog:



Simply select the spheroid you prefer to use from one of the lists of pre-defined spheroids and then click the "Load" button next to that list. Alternatively, you may enter the spheroid semi-major and semi-minor radius values manually. Click 'Cancel' or 'OK' to return to the main dialog.

Longitudinal Shift Options

This section allows you to do two things:

a. Shift all features east or west by a specified number of degrees, and

b. Wrap all features around a predefined longitudinal range.

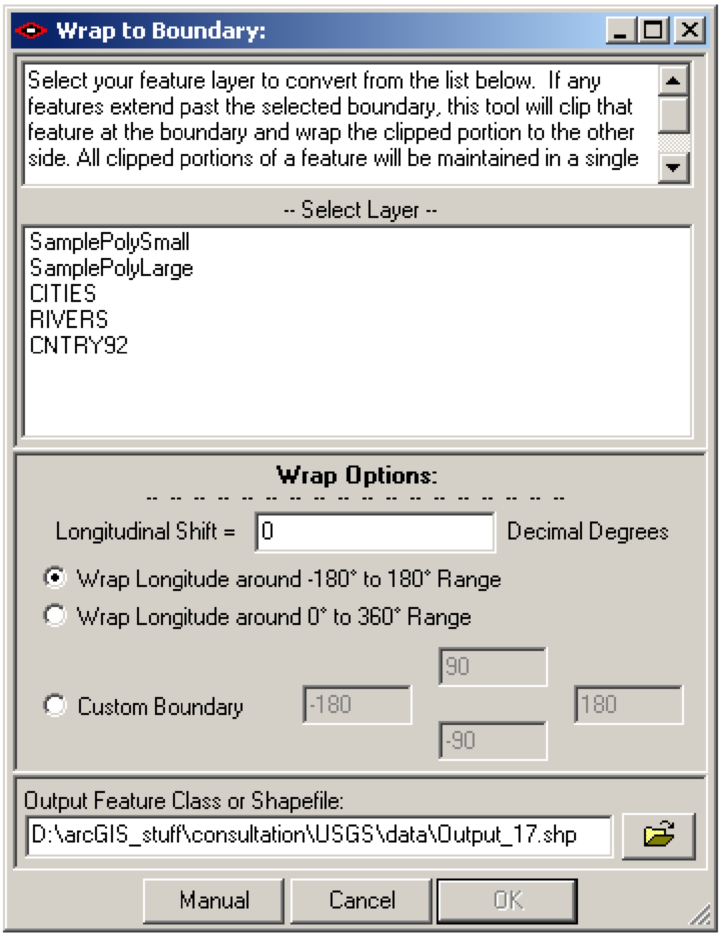
The "Longitudinal Shift" function will probably not be useful to most people working with terrestrial data. It is primarily intended for non-terrestrial datasets in which the prime meridian has been redefined. For example, if new observations lead us to redefine the coordinate system of some planet or satellite, then we might need to transform our existing datasets to account for a new prime meridian.

The "Wrap Longitude" function might be more generally useful. It forces the current dataset to fit within a pre-defined range of longitude values. If a feature extends past the eastern or western boundary, then that feature will be clipped and wrapped to the other side. Features that are split this way will still be represented with a single record in the attribute table.

Note: If you wish to shift or wrap the longitude values without also transforming between planetographic and planetocentric coordinate systems, then you may do so using the "Wrap Boundary" tool (see p. 36)

Wrap Boundary

This function allows you to shift features longitudinally and wrap them to a predefined longitudinal or latitudinal range. Click the "Wrap Boundary" menu item to open the dialog:

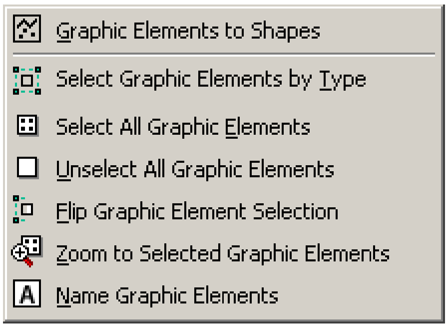


The "Longitudinal Shift" function will probably not be useful to most people working with terrestrial data. It is primarily intended for non-terrestrial datasets in which the prime meridian has been redefined. For example, if new observations lead us to redefine the coordinate system of some planet or satellite, then we might need to transform our existing datasets to account for a new prime meridian.

The "Wrap Longitude" function forces the current dataset to fit within a pre-defined range of longitude values. If a feature extends past the eastern or western boundary, then that feature will be clipped and wrapped to the other side. Features that are split this way will still be represented with a single record in the attribute table.

Tools for Graphics

Graphic Elements to Shapes



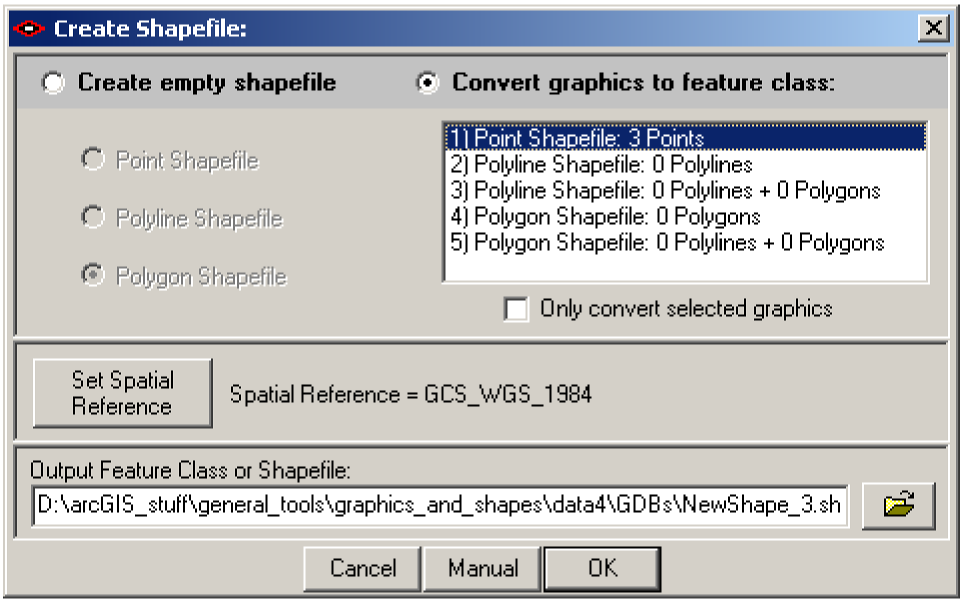
This function allows you to either create a new empty shapefile or convert graphic shapes to a shapefile. You may create either point, polyline or polygon shapefiles with this tool.

- Polygon shapefiles will include attribute fields for [Unique_ID] and [Area]. Area values will be in square meters and will be calculated directly on the sphere (see p. 49).
- Polyline shapefiles will include attribute fields for [Unique_ID] and [Length]. Length values will be in meters and will be calculated directly on the spheroid (see p. 46)
- Point shapefiles will include attribute fields for [Unique_ID], [X_Coord] and [Y_Coord]. X- and Y-coordinates will be in the projection you set for the shapefile.

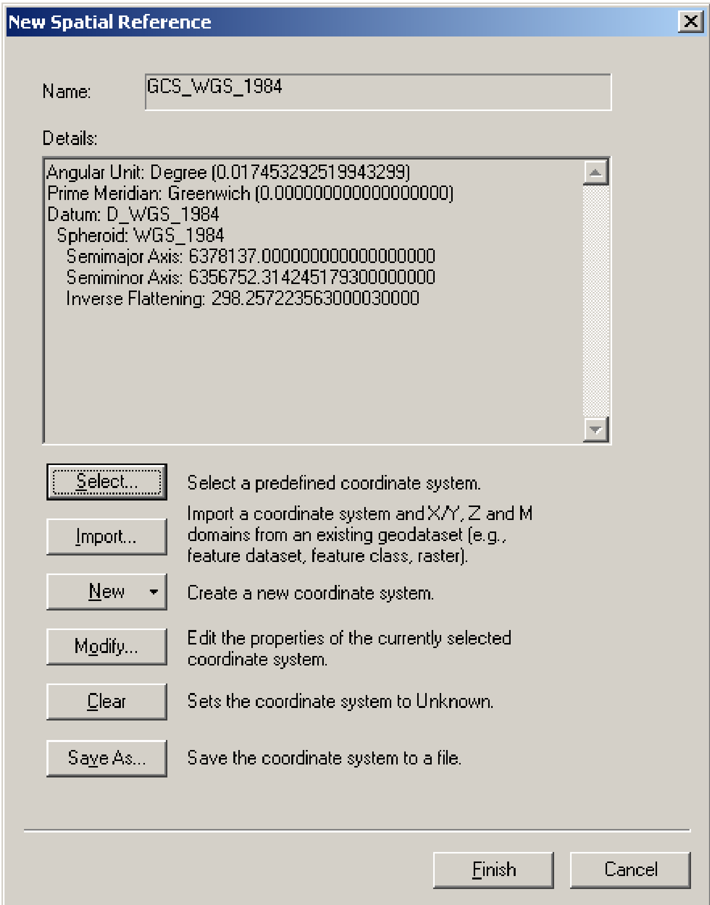
NOTE: If you are converting graphics to a shapefile, and if those graphics have names (right-click the graphic and check the properties to see if it has a name; see p. 41 for more info), then these names will also be added to the attribute table in a [Name] field.

This function also allows you to convert polyline graphics to polygons, or polygon graphics to polylines, if you wish. When opened, the tool will examine your map to see how many point, polyline or polygon graphics are available, and whether any of them are selected. The tool will show you how many of each type are available to convert. If you attempt to create a shapefile from existing graphics when there are no graphics to convert, you will be notified of this and asked if you would like to try a different shape type.

NOTE: Certain linear or areal graphic shapes are not technically polylines or polygons. Polygons that are defined by a circle or elliptic arc are not really “polygons” in the sense that they are not composed of a series of straight-line segments. This is also true for linear features that are constructed of Bezier curves. It is not possible to add true curves such as circles, ellipses or Bezier curves to a polyline or polygon shapefile, so this function will convert these curved shapes to standard polygons or polylines before adding them to the shapefile. It does this by generating 200 evenly-spaced points along the length or perimeter of the curve, and connecting these points with straight segments. Therefore, if the original graphic feature is composed of true curves, then the actual shape in the shapefile will be slightly different than the original feature.



You must set a spatial reference for the new feature class. If your map has a spatial reference set, then the map spatial reference will be the default value. You may easily change the spatial reference by clicking the "Set Spatial Reference" button and identifying the spatial reference you want:

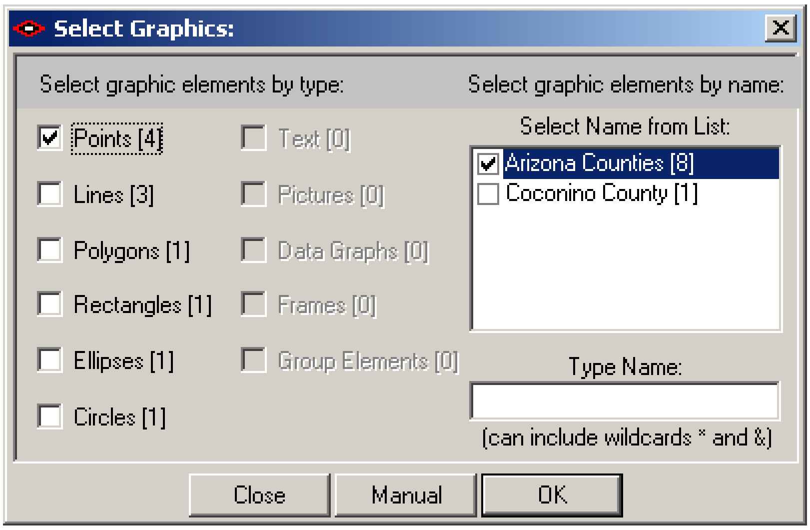


NOTE: This function adds the new feature class to the map, but does not delete existing graphics so you may not see the new shapefile when the shapes lie behind the graphics.

Select Graphic Elements by Type

If you have a lot of graphic elements in your active view, and if you want to select some specific subset of them, then it can get difficult and tedious to select them all individually by clicking on them. This function allows you to select graphics by shape type or by name (see p. 41 for information on naming graphics).

Click the “Select Graphic Elements by Type” menu item:



The check boxes on the left show you exactly how many graphics of each type are available to be selected. If no graphics of that type are available, the check box is disabled. You may also select graphics by name, either by selecting an existing name in the list or by typing in a name. The search text can include wildcards and it is not case-sensitive, so in the example above you could select all graphics named both “Arizona Counties” and “Coconino County” simply by typing in the text *count*.

NOTE: This function will not unselect any graphics that are already selected! For example, if you manually select a polygon graphic, and then use this tool to select all point graphics, then the polygon graphic will still be selected after you use the tool.

Select All Graphic Elements

This function simply selects all graphic elements in your active view. This function behaves the same way as the standard ArcMap function “Select All Elements” in the “Edit” menu. I just replicated the function here because I wanted all these function available in the same menu.

Unselect All Graphic Elements

This function simply unselects all graphic elements in your active view. This function behaves the same way as the standard ArcMap function “Unselect All Elements” in the “Edit” menu. I just replicated the function here because I wanted all these function available in the same menu.

Flip Graphic Element Selection

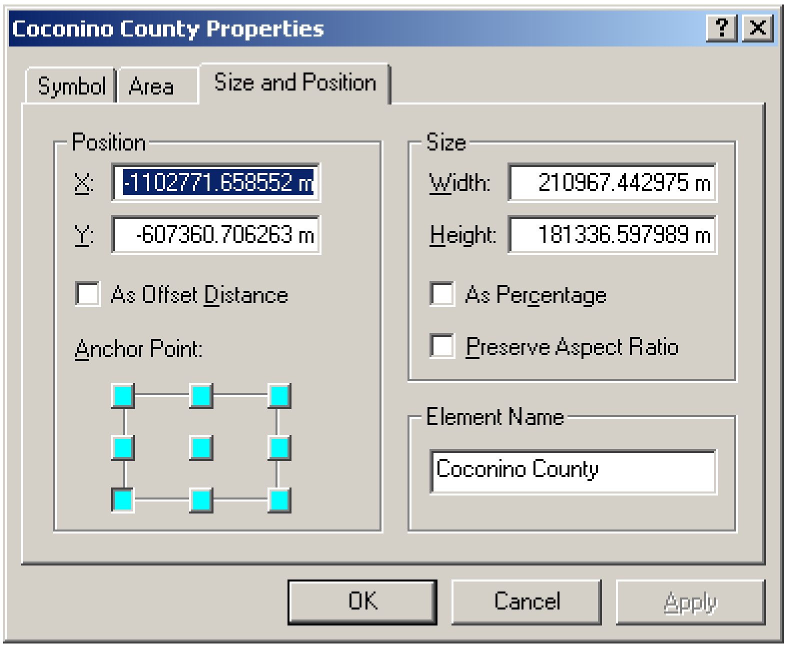
This function selects all graphics that are not currently selected, and unselects all graphics that are currently selected.

Zoom to Selected Graphic Elements

This function simply zooms to all selected graphic elements in your active view. This function behaves the same way as the standard ArcMap function “Zoom to Selected Elements” in the “Edit” menu. I just replicated the function here because I wanted all these function available in the same menu.

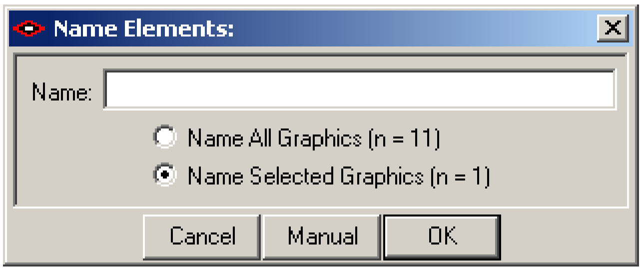
Name Graphic Elements

It is not commonly known, but graphic elements can have names attached to them. Beginning with ArcGIS 9.2, users could set or modify the name by right-clicking on the graphic, selecting "Properties" from the menu, and clicking the "Size and Position" tab:



Note that the "Properties" dialog caption also shows the element name.

This function provides a simple way to name multiple graphics simultaneously. If necessary, select the graphics you wish to name first (either by clicking on them manually, or by using the "Select Graphic Elements by Type" function on p. 40), and then click the menu item "Name Graphic Elements":



Type in your new name, specify whether you want to name all graphics or only the selected graphics, and click 'OK'. **NOTE:** If you want to remove existing names from a graphic, just clear all text in the "Name" text box and click 'OK'.

These names can actually be quite useful if they can be implemented in an efficient and simple way. For example, the Corridor Designer Evaluation tools (see http://www.corridordesign.org/downloads/) include several analytical functions that draw graphics on the screen, including polyline graphics with corridor widths in the names. If the user converts these graphics to a shapefile, then the names become useful attribute information.

NOTE: ArcGIS 9.1 does not offer a way to enter the name manually using the Graphic Properties dialog. This "Name Graphic Elements" tool is the only way to set a name for a graphic in ArcGIS 9.1.

NOTE: If you use the "Graphic Elements to Shapes" function (p. 38) to convert your graphics to a shapefile, then any graphic names will be added to the shapefile attribute table. If no graphics have names, then no "Name" field will be added to the attribute table.

An Examination of Errors Derived from Projected Data

I was curious to see how much difference it really made to calculate the area and centroid from projected data vs. geographic data. I used the polygon feature class of United States Counties (ESRI 2006) that comes with ArcGIS as a test dataset. This dataset includes 3,219 high-resolution polygons (some with more than 60,000 vertices) delineating county boundaries in the United States and Puerto Rico.

I projected these polygons into 3 common projections (UTM, Albers Equal Area Conic and Lambert Conformal Conic; See Snyder 1983; Iliffe 2000; and Kennedy & Kopp 2000 for thorough descriptions of these map projections) and calculated the area (projected and spherical) and centroid (projected and spherical) for each projection (see p. 49 for methods). All spherical and spheroidal calculations were based on the WGS 84 spheroid. I then compared the projected vs. spherical area values for each polygon, and also the distance between the projected and spherical centroids. I assume that the spherical values are the closest to truth and I describe the difference observed in the projected values as “error”.

I compared areas derived by spherical vs. projected methods by calculating the percent difference between projected and spherical area values:

$$\text{\% Difference} = \frac{\text{Projected Area} - \text{Spherical (True) Area}}{\text{Spherical (True) Area}}$$

A value of 50% would mean that the projected area was 1.5 × as large as the spherical area, while a value of -50% would mean the projected area was half as large as the spherical area.

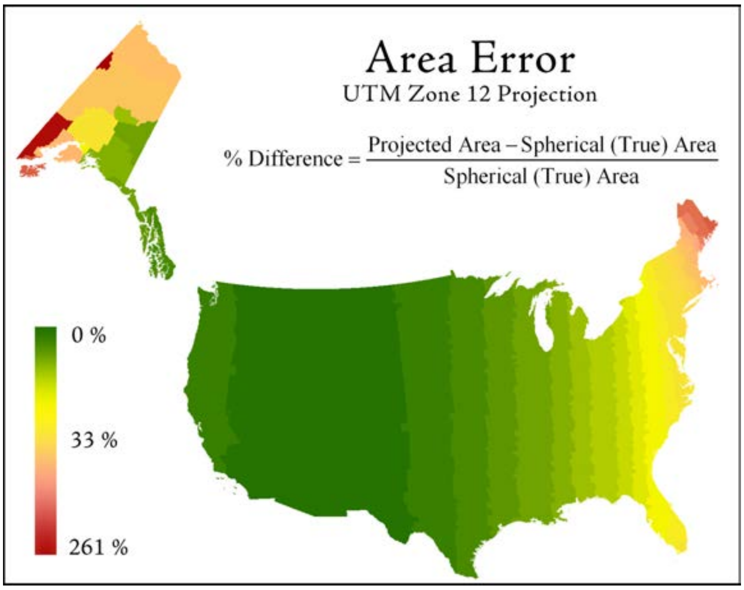
I measured the distance between spherical and projected centroids using Vincenty’s equations to calculate true distances between points over the spheroid (see p. 46).

Results for UTM Zone 12

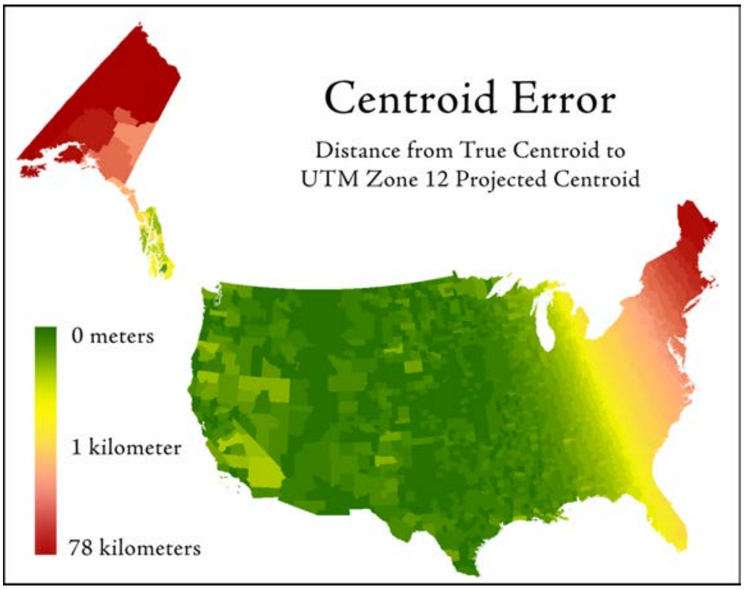
The Universal Transverse Mercator (UTM) projection is a cylindrical projection with lines of tangency along lines of longitude (hence “transverse” Mercator; the normal Mercator projection is a cylinder that is tangent on or around the equator). The UTM projection is divided into 60 zones, such that each zone has a different central meridian, and each central meridian is separated by 6° of longitude. The UTM cylinder actually intersects the planet surface along 2 longitudinal lines of tangency at ±3° east and west of the central meridian. The UTM projection is most accurate along these two lines of tangency. The central meridian for UTM Zone 12 is at Longitude -111°.

UTM is a conformal projection so it does well at maintaining the shapes of polygons. UTM Zone 12 is most accurate along the lines of Longitude -108° and -114°, and accuracy degrades as distance increases.

Not surprisingly, the errors were fairly small within the 6° band of the UTM zone. My home county (Coconino County in Arizona), located in the center of UTM Zone 12 at around 36° Latitude, has a true area of 48,272 km^2 and a projected area of 48,303 km^2, a difference of only 0.06%. Errors increase as we move east or west (see figure below), and error rates increase to approximately 0.3 – 0.6% within 300 km, and to around 1% at 600 km. Error rates increase quickly after that.



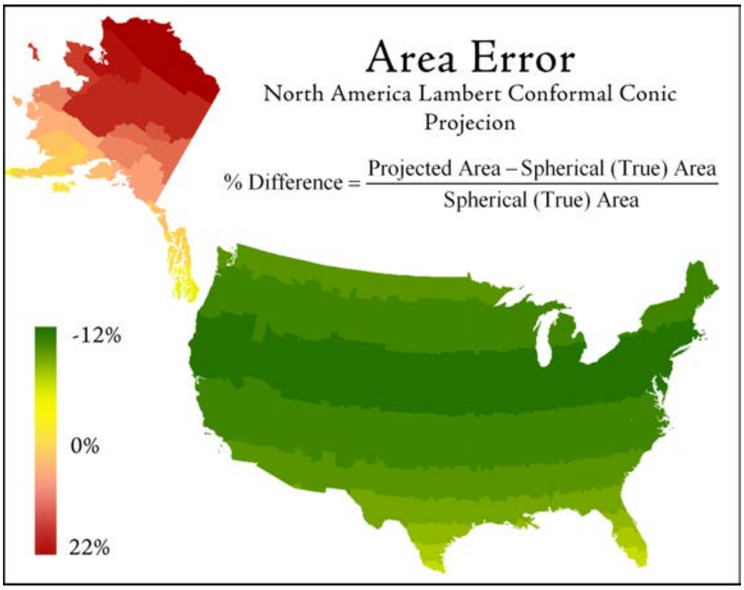
Centroid errors follow a similar pattern. These errors are affected by both the distance from the central meridian and the size of the county; smaller counties will naturally have smaller errors than larger counties that are located the same distance from the central meridian. Coconino county, which is one of the largest counties in the country, had a centroid location error of 24m, while counties ¼ the size 600km to the east had centroid locations errors in similar ranges. San Bernardino and Riverside counties, both large counties located about 150 km to the southwest of Coconino county, had centroid errors of around 180m.



Results for North America Lambert Conformal Conic Projection

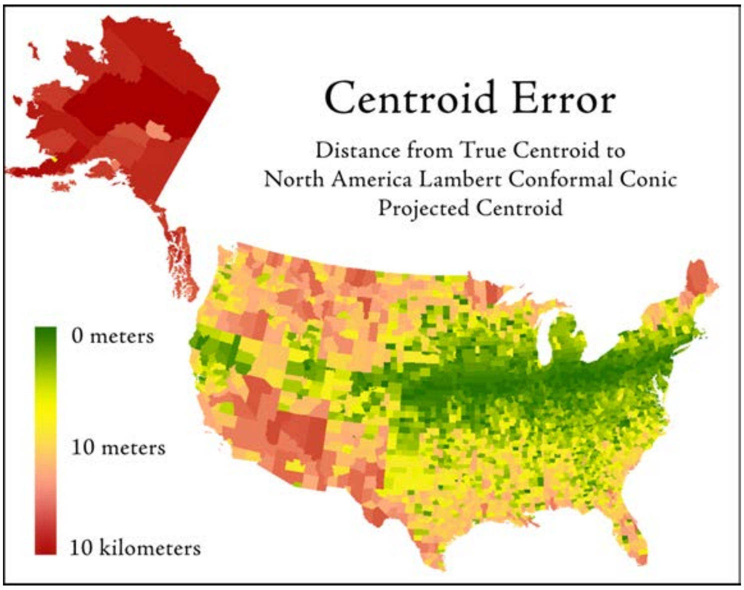
The Lambert Conformal Conic Projection is (not surprisingly) a conic projection with a latitude of origin and 2 standard parallels where the cone intersects the planetary surface (i.e. lines of tangency). The "North American" version has a latitude of origin at 40°, with lines of tangency at 20° and 60°. Therefore all distortions will be minimized along the 20° and 60° parallels of latitude. Area values tend to be compressed within the parallels of latitude and increased outside this range.

Coconino county is located fairly close to the latitude of origin and therefore the area was underestimated fairly dramatically (-11%, or over 5,300 km^2). Within the boundaries of the United States, Area values were most accurate in Hawaii (at around 20° latitude) and southern Alaska (at around 60° latitude).



As a "conformal" projection, it is intended to preserve shape as well as possible. Centroid location accuracy is associated with shape accuracy, so centroid accuracy decreases with increasing distance from the latitude of origin. Centroid accuracy also tends to decrease as the size of the polygon increases.

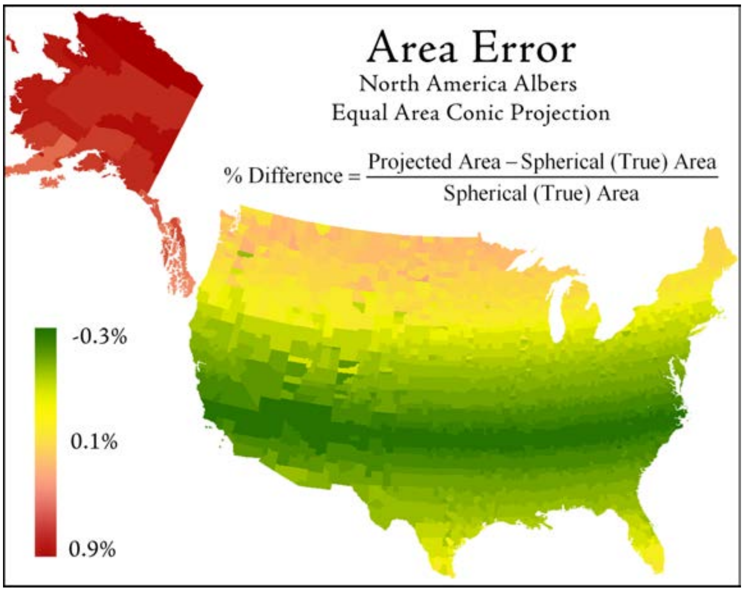
Coconino county had a centroid error of 147m, which was relatively large compared to many counties it the same latitude. For example, the 20 counties in the neighboring state of New Mexico that share the same general latitudinal range as Coconino county had an average centroid error of 14m. These 20 counties also have an average size only $\frac{1}{6}^{th}$ the size of Coconino county, which likely accounts for the difference.



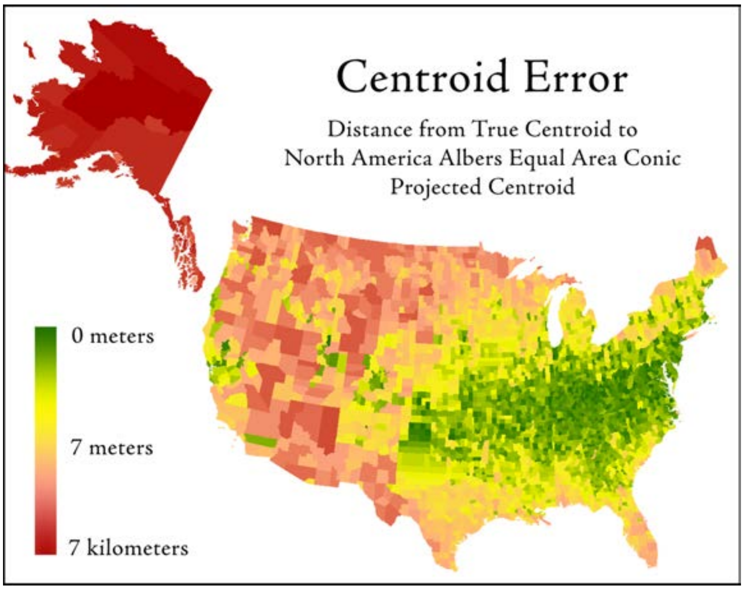
Results from North American Albers Equal Area Conic Projection

Like the Lambert Conformal Conic projection, the Albers Equal Area Conic Projection is a conic projection with a latitude of origin and two standard parallels where the cone intersects the planetary surface. It differs from Lambert in that it is designed to maintain area rather than shape (hence the name "Equal Area") and therefore the spacing between the parallels decreases toward the poles. The spacing between parallels in the Lambert projection increases toward the poles.

The North American version of Albers Equal Area has a latitude of origin at 40° and standard parallels at 20° and 60°. This projection did much better than both the UTM and Lambert projections at producing the true area of the county polygons. Accuracy does drop off with increasing distance from the latitude of origin, but the drop off is slow. The error for Coconino county, just south of 40° latitude, was only 0.01% (6 km^2).



The centroid errors were better than with the Lambert projection but worse than the UTM. Coconino county had a centroid error of 47 m while the 20 New Mexico counties at the same latitude range had an average error of 5 m.



In sum, projected datasets do have errors, and the degree of error varies dramatically depending on the projection and your location. In many cases we can get satisfactory results with projected data by choosing the right projection, but I suspect that many times people understand very little about the actual projection they are working with and they just choose whatever the default is. This can cause ridiculous errors when, for example, they try to calculate areas and distances on geographic data while treating the geographic coordinates as Cartesian coordinates (i.e. when someone calculates the area of the continental United States to be “817”, or the distance from Los Angeles to New York to be “45”).

Hopefully these tools for calculate areas and distances directly on the sphere or spheroid will help alleviate some of the confusion and provide more accurate data in general.

General Geometric Functions

Vincenty's equations for Calculations on the Spheroid

Thaddeus Vincenty, among many other ground-breaking achievements (Chovitz 2002) wrote some of the primary methods to calculate distances over a spheroid and to determine the location of a new point on a spheroid given an origin and initial bearing. He used an iterative approach in both cases to narrow down the error to an acceptable level. The equations below are numbered according to Vincenty's 1975 paper, with some modifications by Veness (2007):

Terms used by Vincenty:

- $a, b =$ major and minor semiaxes of the ellipsoid
- $f =$ flattening $\dfrac{a-b}{a}$
- $\phi_1 =$ geodetic latitude, positive north of the equator, of first point P_1
- $\phi_2 =$ geodetic latitude, positive north of the equator, of second point P_2
- $L =$ difference in longitude, positive east
- $s =$ length of geodesic (i.e. distance between P_1 and P_2 in units of a, b)
- $\alpha_1 =$ initial azimuth of the geodesic connecting P_1 to P_2
- $\alpha_2 =$ ending azimuth of the geodesic connecting P_1 to P_2
- $\alpha =$ azimuth of the geodesic at the equator
- $u^2 = \dfrac{\cos^2 \alpha \left(a^2 - b^2\right)}{b^2}$
- $U_1 = \arctan\left(1-f\right)\tan\phi_1$ (i.e. reduced latitude of ϕ_1)
- $U_2 = \arctan\left(1-f\right)\tan\phi_2$ (i.e. reduced latitude of ϕ_2)
- $\sigma =$ angular distance between P_1 and P_2 on the sphere
- $\sigma_1 =$ angular distance on the sphere from the equator to P_1
- $\sigma_{\mathrm{m}} =$ angular distance on the sphere from the equator to midpoint on the line
- $\lambda =$ difference in longitude on an auxiliary sphere
- $\lambda' =$ iterated estimate of λ, beginning at 2π (Veness 2007)

Using Vincenty's Equations to Calculate Distance and Azimuths on a Spheroid

Vincenty gives both "Direct" and "Inverse" formulae. The "Inverse" formula is used to calculate the distance between two points on the spheroid, and the initial and final azimuths of the geodesic curve connecting those two points. Numbers on the right margin refer to Vincenty's equation numbers. **NOTE:** This extension automatically uses the WGS 84 spheroid in all spheroid-based calculations.

Do the following (**NOTE:** steps below include some minor algebraic modifications by Chris Veness which simplify the coding a bit):

Initially set $\lambda' = 2\pi$

while $\left|\lambda - \lambda'\right| > 10^{-12}$ (Threshold suggested by Veness [2007]; $\cong 0.006$ mm at equator)

$$\sin\sigma = \sqrt{\left(\cos U_2 \sin\lambda\right)^2 + \left(\cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos\lambda\right)^2} \quad [14]$$

$$\cos\sigma = \sin U_1 \sin U_2 + \cos U_1 \cos U_2 \cos\lambda \quad [15]$$

$$\sigma = \arctan[2]\left(\sin\sigma,\quad \cos\sigma\right)$$ [16][i]

$$\sin\alpha = \frac{\cos U_1 \cos U_2 \sin\lambda}{\sin\sigma}$$ [17]

$$\cos^2\alpha = 1 - \sin^2\alpha$$ (Trig identity; included by Veness [2007])

$$\cos\left(2\sigma_m\right) = \cos\sigma - \frac{2\sin U_1 \sin U_2}{\cos^2\alpha}$$ [18]

$$C = \frac{f}{16}\cos^2\alpha\left[4 + f\left(4 - 3\cos^2\alpha\right)\right]$$ [10]

$$\lambda' = \lambda$$ (Introduced by Veness [2007])

$$\lambda = L + \left(1 - C\right) f \sin\alpha\left\{\sigma + C\sin\sigma\left[\cos\left(2\sigma_m\right) + C\cos\sigma\left(-1 + 2\cos^2\left(2\sigma_m\right)\right)\right]\right\}$$ [11][ii]

Loop[iii] until $\left|\lambda - \lambda'\right| \le 10^{-12}$

$$A = 1 + \frac{u^2}{16384}\left\{4096 + u^2\left[-768 + u^2\left(320 - 175u^2\right)\right]\right\}$$ [3]

$$B = \frac{u^2}{1024}\left\{256 + u^2\left[-128 + u^2\left(74 - 47u^2\right)\right]\right\}$$ [4]

$$\Delta\sigma = B\sin\sigma\left\{\cos\left(2\sigma_m\right) + \tfrac{1}{4}B\left[\cos\sigma\left(-1 + 2\cos^2\left(2\sigma_m\right)\right) - \tfrac{1}{6}B\cos\left(2\sigma_m\right)\left(-3 + 4\sin^2\sigma\right)\left(-3 + 4\cos^2\left(2\sigma_m\right)\right)\right\}\right]$$ [6]

Final Distance $s = bA\left(\sigma - \Delta\sigma\right)$ [19]

Initial Azimuth[iv] $\alpha_1 = \arctan[2]\left(\cos U_2 \sin\lambda,\quad \cos U_1 \sin U_2 - \sin U_1 \cos U_2 \cos\lambda\right)$ [20][v]

Final Azimuth[iv] $\alpha_2 = \arctan[2]\left(\cos U_1 \sin\lambda,\quad -\sin U_1 \cos U_2 + \cos U_1 \sin U_2 \cos\lambda\right)$ [21][v]

Using Vincenty's Equations to Calculate the Position of a Point on the Spheroid

Vincenty gives both "Direct" and "Inverse" formulae. The "Direct" formula calculates the position of the new point on the spheroid given an initial point, bearing and distance. Numbers on the right margin refer to Vincenty's equation numbers.

$$\tan U_1 = \left(1 - f\right)\tan\phi_1$$

$$\cos U_1 = \frac{1}{\sqrt{1 + \tan^2 U_1}}$$ (Trig identity; included by Veness [2007])

$$\sin U_1 = \tan U_1 \cos U_1$$ (Trig identity; included by Veness [2007])

$$\sigma_1 = \arctan[2]\left(\tan U_1,\quad \cos\alpha_1\right)$$ [1][v]

$$\sin\alpha = \cos U_1 \sin\alpha_1$$ [2]

$$\cos^2\alpha = 1 - \sin_2\alpha$$ (Trig identity; included by Veness [2007])

$$A = 1 + \frac{u^2}{16384}\left\{4096 + u^2\left[-768 + u^2\left(320 - 175u^2\right)\right]\right\}$$ [3]

$$B = \frac{u^2}{1024}\left\{256 + u^2\left[-128 + u^2\left(74 - 47u^2\right)\right]\right\}$$ [4]

---

i Arctan[2] function described on p. 53.

ii Algebraic modification by Veness (2007) from original Vincenty (1975) equation.

iii Veness suggests setting a maximum number of iterations possible because (as Vincenty points out) the formulas could go into an infinite loop if the two points are nearly antipodal.

iv Azimuths are in radians. See p. 53 for information on converting back to degrees.

v Arctan[2] function described on p. 53.

$\sigma = \frac{s}{bA}$ (First approximation)

Initially set $\sigma' = 2\pi$

while $|\sigma - \sigma'| > 10^{-12}$ (Threshold suggested by Veness [2007]; $\cong$ 0.006 mm at equator)

$$\cos(2\sigma_m) = \cos(2\sigma_1 + \sigma) \qquad [5]^{vi}$$

$$\Delta\sigma = B\sin\sigma\left\{\cos(2\sigma_m) + \tfrac{1}{4}B\left[\cos\sigma\left(-1 + 2\cos^2(2\sigma_m)\right) - \tfrac{1}{6}B\cos(2\sigma_m)\left(-3 + 4\sin^2\sigma\right)\left(-3 + 4\cos^2(2\sigma_m)\right)\right\}\right] \qquad [6]$$

$\sigma' = \sigma$ (Introduced by Veness [2007])

$$\sigma = \frac{s}{bA} + \Delta\sigma \qquad [7]$$

Loop[vii] until $|\sigma - \sigma'| \le 10^{-12}$

$$\phi_2 = \arctan[2]\left(\sin U_1\cos\sigma + \cos U_1\sin\sigma\cos\alpha_1,\ (1-f)\sqrt{\left[\sin^2\alpha + \left(\sin U_1\sin\sigma - \cos U_1\cos\sigma\cos\alpha_1\right)^2\right]}\right) \qquad [8]^{viii,vi}$$

$$\lambda = \arctan[2]\left(\sin\sigma\sin\alpha_1,\quad \cos U_1\cos\sigma - \sin U_1\sin\sigma\cos\alpha_1\right) \qquad [9]^{viii,\ vi}$$

$$C = \frac{f}{16}\cos^2\alpha\left[4 + f\left(4 - 3\cos^2\alpha\right)\right] \qquad [10]$$

$$L = \lambda - (1-C)\, f\sin\alpha\left\{\sigma + C\sin\sigma\left[\cos(2\sigma_m) + C\cos\sigma\left(-1 + 2\cos^2(2\sigma_m)\right)\right]\right\} \qquad [11]$$

Reverse Azimuth[ix] $\alpha_2 = \arctan[2]\left(\sin\alpha,\quad -\sin U_1\sin\sigma + \cos U_1\cos\sigma\cos\alpha_1\right)$ [12][vi]

$P_2 = (\phi_2,\ \lambda_1 + L)$ [ix]

[vi] Algebraic modification by Veness (2007) from original Vincenty (1975) equation.

[vii] Veness suggests setting a maximum number of iterations possible because (as Vincenty points out) the formulas could go into an infinite loop if the two points are nearly antipodal.

[viii] Arctan[2] function described on p. 53.

[ix] Values are in radians. See p. 53 for information on converting back to degrees.

Calculating the Surface Area and Centroid of a Spherical Polygon

The CRC Handbook of Standard Mathematical Tables and Formulae (Zwillinger 2003:367) gives the following formula for the internal area of a polygon on a sphere:

$$A = \left(\sum_{i=1}^{n} \theta_i - (n-2)\pi \right) R^2$$

where R = radius of sphere

n = number of vertices

θ_i = internal angles of polygon in radians

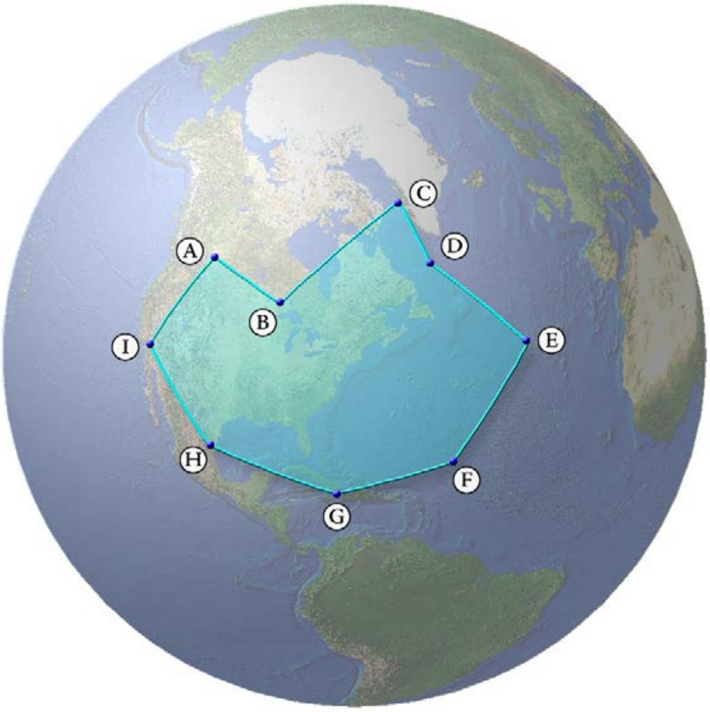
This extension does not use this method because it does not lend itself easily to multipart polygons or polygons with holes. Rather, this extension uses a slightly more complex method involving breaking the polygon down into a set of spherical triangles, and which requires approximately 33% more computation time but has the advantage of allowing us to calculate the surface centroid of that polygon at the same time.

The algorithm works as follows:

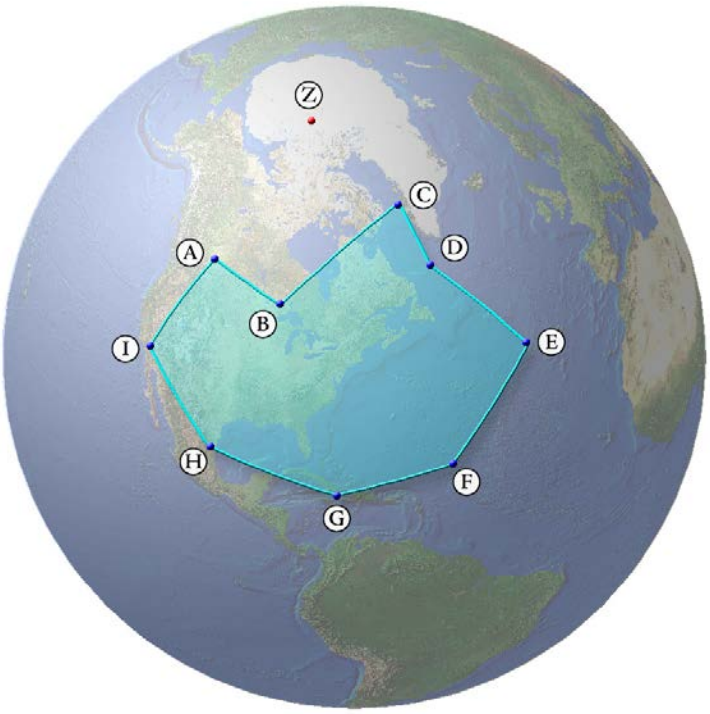
1) Break the polygon up into a set of triangles.
2) Calculate the area and center of mass of each triangle.
3) The polygon area is the sum of all the triangle areas.
4) The polygon centroid is the weighted average of all the triangle centroids, where the triangle centroid is weighted by the triangle area.

There is even an easy way to break up the polygon into triangles. We do not have to worry about generating triangles that occur only within the polygon because we can either add or subtract the triangle area depending on whether the vertices were entered clockwise or counter-clockwise. This means that triangles may be generated outside the polygon boundary, but the area of those triangles are subtracted from the cumulative area, and the final center of mass reflects only the area within the polygon boundary.

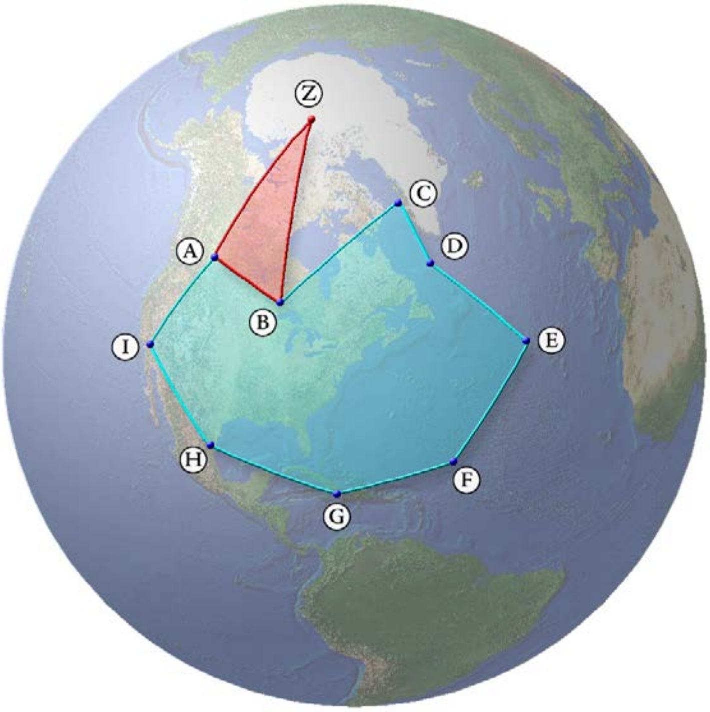
For example, if we had a polygon with 9 vertices labeled A..I:



We start by creating a new point, and then we generate triangles using that new point and all consecutive pairs of vertices. This new point does not have to be inside the polygon:

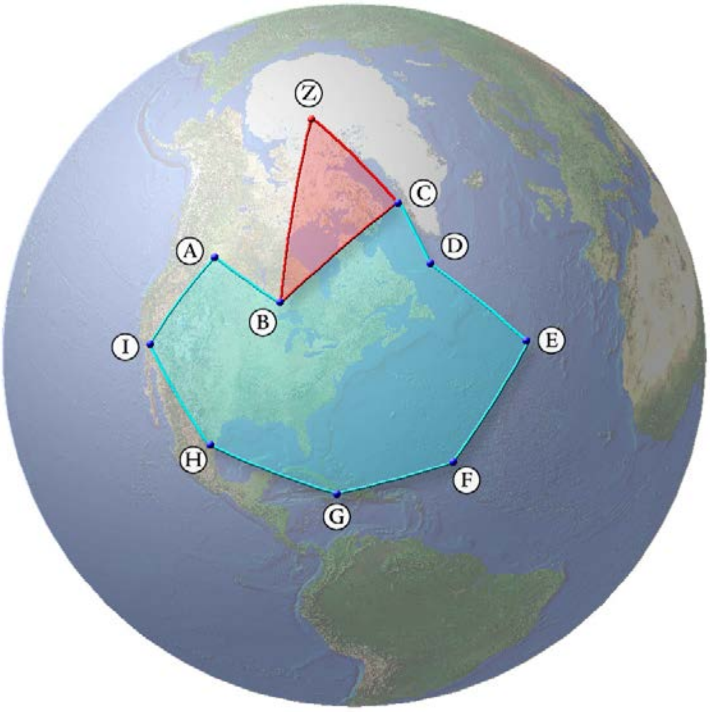


We generate triangles for each consecutive pair of vertices, such that each triangle is composed of vertices (V_i, V_{i+1}, New Point). The first triangle includes vertex A, vertex B and the New Point. Note that the vertices are entered in a counter-clockwise direction:



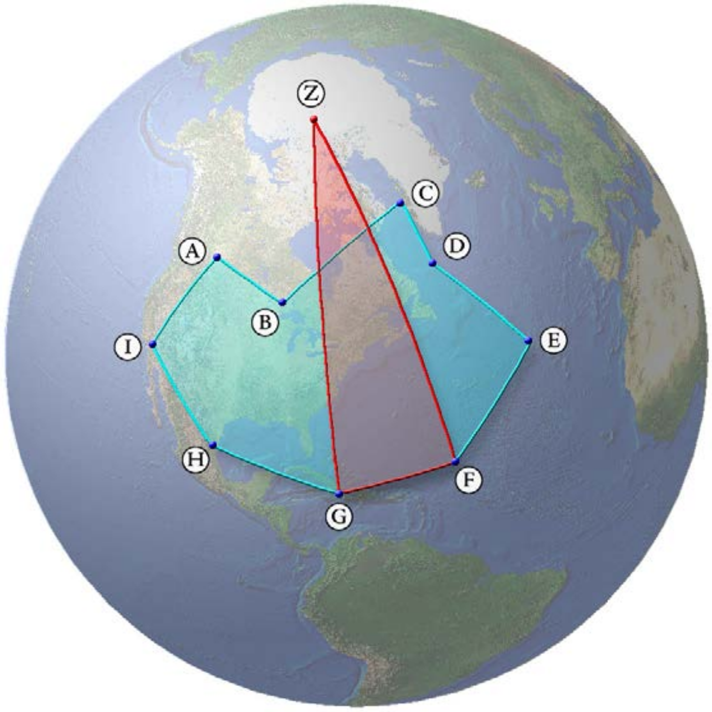
We calculate the area (p. 53) and center of mass (p. 58) of this triangle and multiply the centroid by the area to get a weighted centroid. Because these vertices are entered counter-clockwise, we set this area value to be negative.

The second triangle includes vertex B, vertex C and the New Point:



Again, these vertices are entered counter-clockwise and therefore this triangle is also assigned a negative area value.

Later on, we would generate a triangle from vertex F, vertex G and the New Point:



Note that in this case the vertices are entered in a clockwise direction. This is important because this means the sign of the area of ΔFGZ will be positive while the signs of the areas of ΔABZ and ΔBCZ were negative. The negative values of ΔABZ and ΔBCZ essentially clip out the excess area calculated from ΔFGZ . The final total will only reflect the area within the polygon. This method correctly handles holes and multipart polygons.

The final polygon centroid would be the weighted average of all the triangle centroids, calculated as the sum of the weighted centroids divided by the total polygon area. Centroids should be weighted with either positive or negative values depending on whether the vertices are clockwise or counter-clockwise.

TECHNICAL NOTE: In order to minimize the potential spherical triangle edge lengths, this extension uses the projected centroid of the polygon for the New Point (Point Z in the examples above). Theoretically any point would work, but it stands to reason that a point that minimizes the distances would be likely to introduce less rounding error into the final equations, especially when using the Haversine functions.

Spherical Polygon Centroid = Mean of all spherical triangle centroids, where each centroid is weighted by triangle area (Note that centroids are in Cartesian coordinates, not polar latitude / longitude values):

Spherical Polygon Centroid:

$$\mathrm{X_{SPC}} = \frac{\sum X_i A_i}{\sum A_i} \qquad \mathrm{Y_{SPC}} = \frac{\sum Y_i A_i}{\sum A_i} \qquad \mathrm{Z_{SPC}} = \frac{\sum Z_i A_i}{\sum A_i}$$

where A_i = Area of Spherical Triangle i (might be negative value if triangle is being subtracted)

These X,Y and Z values are all in Cartesian coordinates, so this final polygon centroid vector needs to be converted back to latitude and longitude values (see p. 63)

Calculating the Surface Area of a Spherical Triangle

The surface area of a spherical triangle is surprisingly easy to calculate, provided you can determine the three internal angles. Remember that the internal angles of a spherical triangle will always be greater than 180°. If the internal angles add to exactly 180°, then that triangle exists on a plane rather than a sphere. The amount by which the internal angles exceed 180° is called the *Spherical Excess*. The surface area of the spherical triangle is directly related to the spherical excess.

$$\text{Area of Spherical Triangle} = R^2 E$$

$$\text{where } R = \text{radius of sphere}$$

$$E = \text{triangle spherical excess in radians}$$

$$= \text{Sum of angles - 180 (in degrees)}$$

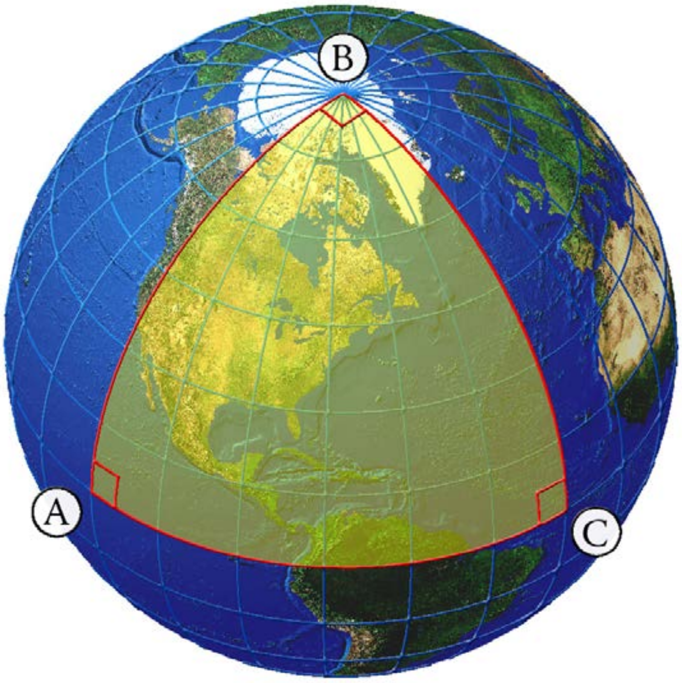
$$= \text{Sum of angles - } \pi \text{ (in radians)}$$

For a simple example, consider the triangle formed by the 3 coordinates:

Point A: Longitude = -120, Latitude = 0

Point B: Longitude = -120, Latitude = 90

Point C: Longitude = 0, Latitude = 0



This triangle, sometimes referred to as the "quadrantal triangle", is formed by 3 right angles. The sum of the internal angles is 270° and the spherical excess is therefore $270^{\circ} - 180^{\circ} = 90^{\circ}$, or $\frac{\pi}{2}$ radians. Note that if this were the unit sphere (i.e. a sphere with radius = 1), then $\frac{\pi}{2}$ would actually be the area of this triangle. The earth has a radius of approximately 6,371km (see p. 66), so the surface area of this large triangle = $6371^2 \frac{\pi}{2} \cong 63.8 \text{ million km}^2$.

Note that this result conforms with the known value of the surface area of a sphere:

$$\text{Surface Area of Sphere} = 4\pi r^2$$

$$\text{where } r = \text{radius of sphere}$$

The triangle with 3 right angles in the example above covers exactly $\frac{1}{8}^{\text{th}}$ of the globe:

$\left(\frac{1}{8}\times 4\pi r^2\right)=\frac{\pi}{2}r^2$, exactly as we calculated above.

The hard part is calculating the internal angles of the spherical triangle. This could be done using trigonometry to calculate the bearings to and from each pair of points (see p. 56 for example). I actually find bearings on a sphere to be somewhat confusing, though, because the bearing constantly changes as you follow a great circle route (unless you are following the equator or a line of longitude), so I found it to be a little more intuitive to calculate the triangle area using the triangle edge lengths instead. The following formula (also based on the triangle spherical excess) calculates the spherical area of the triangle from edge lengths (adapted from Zwillinger 2003:370).

$$\text{Area} = R^2 E$$

where:

$$R = \text{Sphere Radius}$$

$$E = \text{Spherical Excess}$$

$$\tan\left(\frac{E}{4}\right)=\sqrt{\tan\left(\frac{S}{2}\right)\tan\left(\frac{S-\overrightarrow{AB}}{2}\right)\tan\left(\frac{S-\overrightarrow{BC}}{2}\right)\tan\left(\frac{S-\overrightarrow{CA}}{2}\right)}$$

$$\overrightarrow{AB},\overrightarrow{BC},\overrightarrow{CA} = \text{Triangle Edge Lengths}$$

$$S=\frac{\overrightarrow{AB}+\overrightarrow{BC}+\overrightarrow{CA}}{2}=\text{Triangle Half-Perimeter}$$

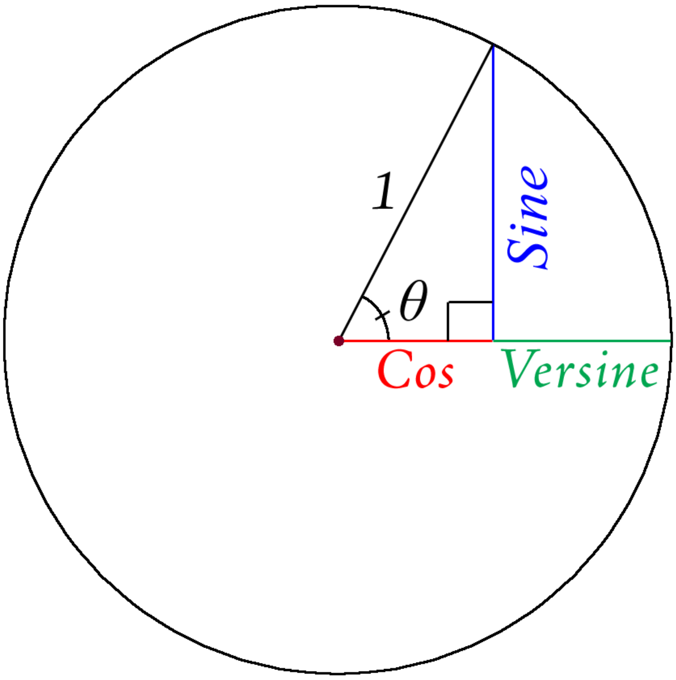
NOTE: It is worth remembering that whenever you draw a triangle on the sphere, you are actually drawing *two* triangles on the sphere. The region "outside" your triangular area of interest is also a perfectly valid triangle formed by 3 straight lines and 3 internal angles. Usually we automatically assume that the smaller of the two triangles is the one we are interested in (as the formula using edge lengths above does). However, if we are doing any function that requires calculating the internal angles of a triangle or polygon, we must take care that the angle reflects the inside of the object we are analyzing.

Calculating the Length of a Line on a Sphere

Spherical triangle edge lengths are calculated using an implementation of the Haversine formula (see Veness 2007a; Wikipedia 2007; Zwillinger 2003:373). The haversine is simply a trigonometric identity like the Sine or Cosine, although it is not used nearly as often as these more familiar identities. Haversines are useful for very small angles where cosine gets very close to 0. It has been used historically for geographic calculations over the surface of the earth because angles tend to be very small in these cases. The haversine is equal to half of the Versine, and is calculated as follows:

$$\text{Versin}(\theta) = 1 - \cos(\theta) = 2\sin^2\left(\frac{\theta}{2}\right)$$

$$\text{Haversin}(\theta) = \frac{\text{Versin}(\theta)}{2} = \sin^2\left(\frac{\theta}{2}\right)$$



The haversine is used to calculate the distance between two geographic points as follows:

$$\text{Distance} = RC$$

$$\text{where } R = \text{Radius of sphere in whatever units are appropriate}$$

$$C = 2\,\text{arctan[2]}\left(\sqrt{A},\ \sqrt{1-A}\right)$$

$$\text{arctan[2]} = \text{Arctangent, adjusted for quadrant (see General Geometric Functions)}$$

$$\text{Using Haversine: } A = \sin^2\left(\frac{\Delta\text{Lat}}{2}\right) + \cos(\text{Lat}_1)\cos(\text{Lat}_2)\sin^2\left(\frac{\Delta\text{Long}}{2}\right)$$

$$\Delta\text{Long} = \text{Long}_2 - \text{Long}_1\text{, in radians}$$

$$\Delta\text{Lat} = \text{Lat}_2 - \text{Lat}_1\text{, in radians}$$

$$\text{Lat}_1\text{, Long}_1 = \text{Latitude and Longitude at Point 1, in radians}$$

$$\text{Lat}_2\text{, Long}_2 = \text{Latitude and Longitude at Point 2, in radians}$$

Calculating the Azimuth Between Points on the Sphere

Calculating the azimuth or bearing between two points on the sphere is conceptually much more complicated than calculating the bearing on a plane. Except in the extremely rare instance where both points lie on the equator or along the same line of longitude, the bearing changes constantly as you travel over the great circle geodesic that connects those two points. For example, if you followed the great circle route from New York City, USA to Hanoi, Vietnam, you would leave New York on a bearing of 358°, travel in a straight line the entire way, and then arrive in Hanoi on a bearing of 182°.

There are various formulas available for calculating the initial bearing and the final bearing, and the bearing for points in between. The formula below calculates the initial bearing for the great circle line going from Point 1 to Point 2 (Williams 2006):

$$\text{Initial Bearing} = \arctan[2]\left(y,\ x\right)$$

where $\arctan[2] =$ Arctangent, adjusted for quadrant (see General Geometric Functions)

$$y = \sin\left(\Delta\text{Long}\right)\cos\left(\text{Lat}_2\right)$$

$$x = \cos\left(\text{Lat}_1\right)\sin\left(\text{Lat}_2\right) - \sin\left(\text{Lat}_1\right)\cos\left(\text{Lat}_2\right)\cos\left(\Delta\text{Long}\right)$$

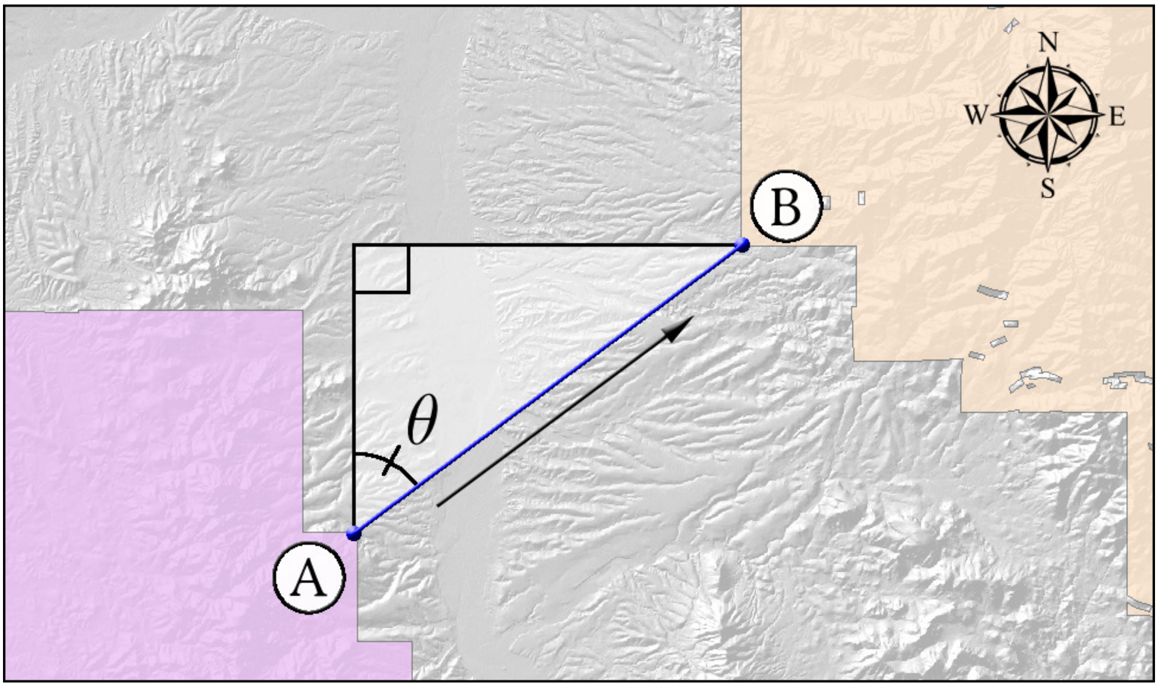
$\Delta\text{Long} =$ change in longitude from Point1 to Point2

$\text{Lat}_1 =$ Latitude at Point 1, in radians

$\text{Lat}_2 =$ Latitude at Point 2, in radians

Note: Result is in radians, where North = 0 and π = South

Calculating the Azimuth Between Points on a Plane



The azimuth (or bearing) between points A and B is calculated as follows:

Given that:

$\Delta x =$ Point B x-coordinate – Point A x-coordinate

$\Delta y =$ Point B y-coordinate – Point A y-coordinate

If If $\Delta x = 0$ and $\Delta y = 0$ then $\theta = -9999$ (i.e. No Direction)

Else if $\Delta y = 0$ then:

If $\Delta x < 0$ then $\theta = -90^{\circ}$

If Else If $\Delta x = 0$ then $\theta = 0^{\circ}$

Else If $\Delta x > 0$ then $\theta = 90^{\circ}$

Else:

$$\theta_a = \frac{\arctan\left(\frac{\Delta x}{\Delta y}\right) * 180}{\pi}$$

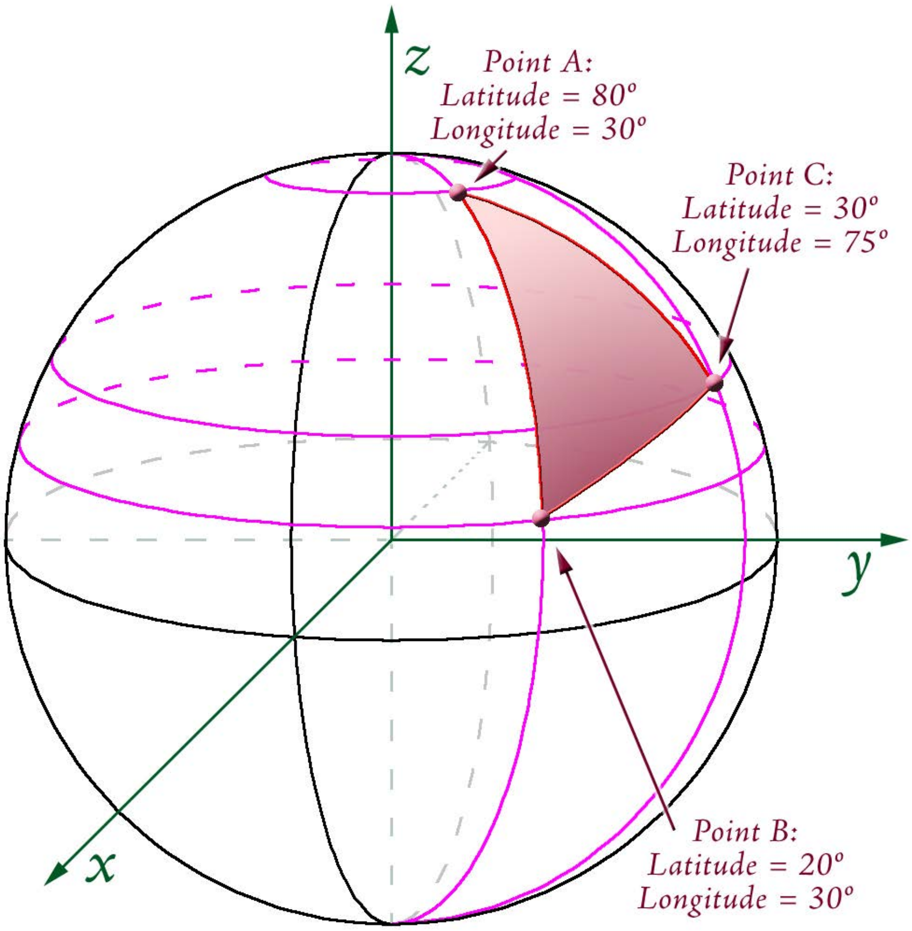
If $\Delta y \geq 0$ then $\theta = \theta_a + 180$

Else:

If $\Delta x \leq 0$ then $\theta = \theta_a$

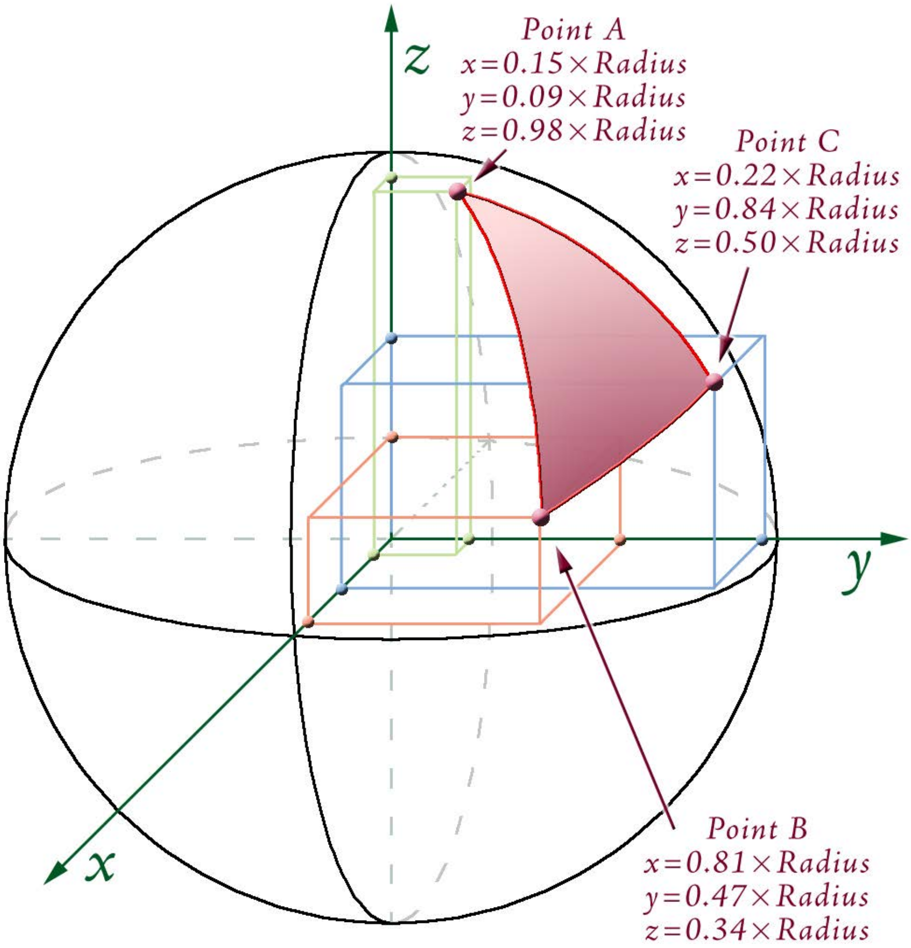
Else if $\Delta x > 0$ then $\theta = \theta_a + 360$

Calculating the Centroid of a Spherical Triangle



Calculating the centroid of a spherical triangle is similar in concept to calculating the centroid of a planar triangle. However, we must remember that longitude and latitude values are not so much coordinates as they are directions from the origin of the sphere. Therefore they cannot be added and divided the way that Cartesian coordinates can. However, they can easily be converted to Cartesian coordinates, and then the computations are simple:

Step 1: Convert Latitude and Longitude values to Cartesian coordinates (see p. 61):



Step 2: Calculate 3D Centroid of 3 points:

3D Triangle Centroid:

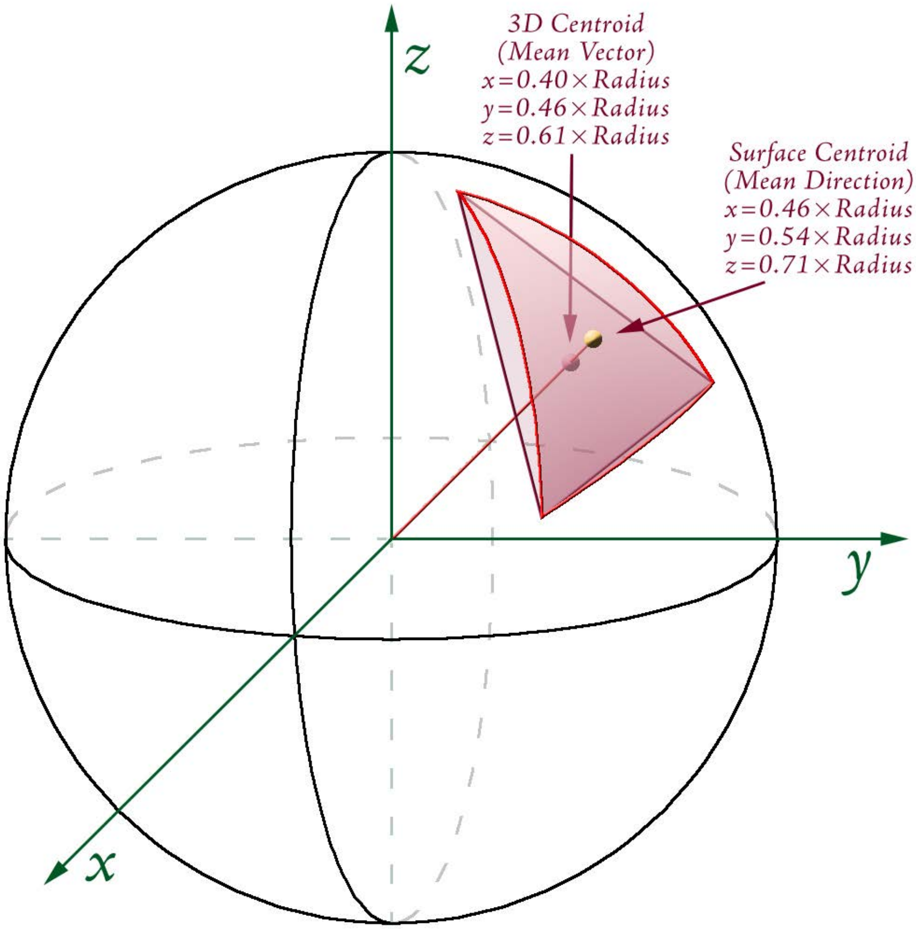
$$\mathrm{X_{3DC}} = \frac{\sum\left(X_1 + X_2 + X_3\right)}{3} \qquad \mathrm{Y_{3DC}} = \frac{\sum\left(Y_1 + Y_2 + Y_3\right)}{3} \qquad \mathrm{Z_{3DC}} = \frac{\sum\left(Z_1 + Z_2 + Z_3\right)}{3}$$

Step 3: This 3D Centroid is actually located inside planet. Shift point outward along centroid vector to the surface of the planet.

$$\text{Centroid Vector Length } = L = \sqrt{\mathrm{X_{3DC}}^2 + \mathrm{Y_{3DC}}^2 + \mathrm{Z_{3DC}}^2}$$

$$\text{Surface X} = \frac{\mathrm{X_{3DC}}}{\mathrm{L}} \qquad \text{Surface Y} = \frac{\mathrm{Y_{3DC}}}{\mathrm{L}} \qquad \text{Surface Z} = \frac{\mathrm{Z_{3DC}}}{\mathrm{L}}$$

NOTE: Mardia (2000; p. 163 – 167) describes Step 2 as calculating the mean vector, and Step 3 as calculating the mean direction of the 3 vectors. The mean direction normalizes the mean vector by the mean vector length. For calculating centroids on the surface of the sphere, the mean direction is more useful than the mean vector. In our case, all points have exactly the same length, so we normalize using the sphere radius. In other words, if we consider a Lat/Long coordinate to be composed of three components (R, ϕ, θ; see p. 61 for definitions), then Step 2 identifies the mean coordinate of ϕ and θ (i.e. mean vector), while Step 3 extends this mean vector outward to the surface of the sphere by setting the vector length = R.


z
3D Centroid
(Mean Vector)
x=0.40×Radius
y=0.46×Radius
z=0.61×Radius
Surface Centroid
(Mean Direction)
x=0.46×Radius
y=0.54×Radius
z=0.71×Radius
y
x

Converting between Latitude / Longitude and Cartesian Coordinates

Latitude / Longitude to Cartesian Coordinate on the Sphere:

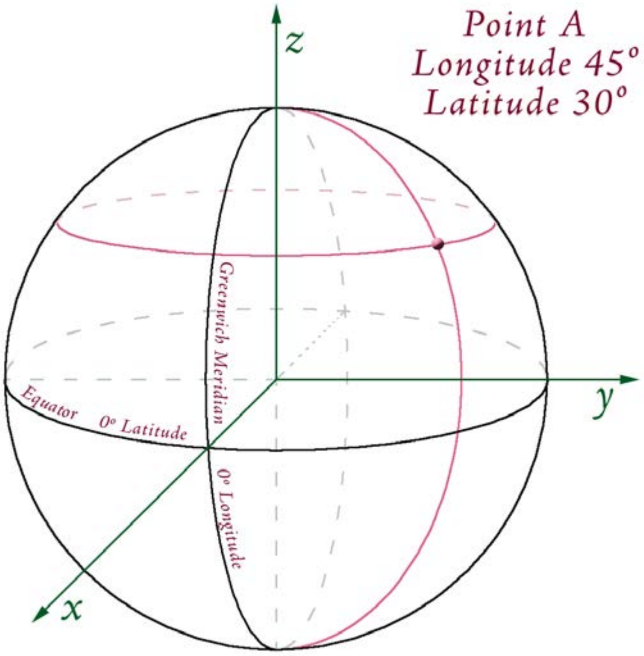
Given a Latitude and Longitude in Degrees, calculate X, Y and Z as follows:

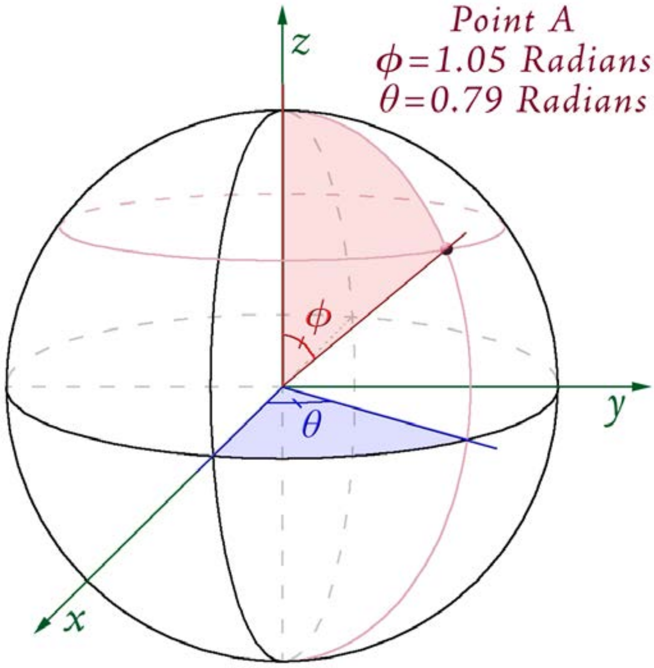
$$\mathrm{X} = R \sin\phi \cos\theta$$
$$\mathrm{Y} = R \sin\phi \sin\theta$$
$$\mathrm{Z} = R \cos\phi$$

where:

$$\phi = \frac{(90 - \mathrm{Latitude})\pi}{180} = \text{Angle from North Pole down to Latitude, in Radians}$$
$$\theta = \frac{(\mathrm{Longitude})\pi}{180} = \text{Angle from Greenwich to Longitude, in Radians}$$
$$R = \text{Radius of Sphere}$$

For example, a point at 30° Latitude and 45° Longitude would be converted to ϕ = 1.05 radians and θ = 0.79 radians:

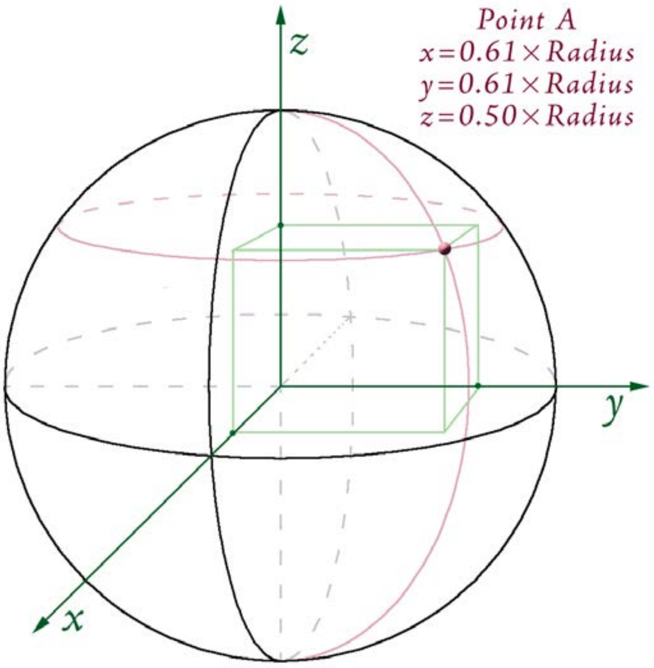




NOTE: Be careful with the terms ϕ and θ! This document treats ϕ as the angle south from the North Pole, and θ as the angle east from Greenwich. However, the literature is not consistent with these two terms! Much of the literature uses these exact two symbols, but for the other angle (i.e. ϕ for east/west and θ for north/south).

Using the formula above, calculate X, Y and Z from ϕ and θ.

$$\mathrm{X} = R\sin\phi\cos\theta = R\sin(1.05)\cos(0.79) = 0.61R$$
$$\mathrm{Y} = R\sin\phi\sin\theta = R\sin(1.05)\sin(0.79) = 0.61R$$
$$\mathrm{Z} = R\cos\phi = R\cos(1.05) = 0.5R$$



NOTE: As with ϕ and θ, the literature is inconsistent regarding how to write the X, Y and Z coordinates of a 3-dimensional point. If a point is located at X, Y, Z, some sources will write the coordinates as (X,Y,Z) while other sources will write it as (Z,X,Y). If you write coordinates in either of these formats, please identify which format you are using!

Cartesian to Latitude / Longitude Coordinates on the Sphere:

$$\phi = \text{arctan[2]}\left(\sqrt{X^2 + Y^2},\ Z\right) = \text{Angle from North Pole down to Latitude, in radians}$$

$$\theta = \text{arctan[2]}\left(Y,\ X\right) = \text{Angle from Greenwich to Longitude, in radians}$$

where $\text{arctan[2]} = $ Arctangent, adjusted for quadrant (see General Geometric Functions)

$$\text{Latitude} = 90 - \phi\left(\frac{180}{\pi}\right)$$

$$\text{Longitude} = \theta\left(\frac{180}{\pi}\right)$$

Converting Between Latitude / Longitude and Cartesian Coordinates on the Spheroid:

As illustrated in Chapter 2 of Iliffe (2000), the formulas above can be adapted to oblate spheroids (such as a planet). If you refer to Iliffe's text, please be aware that his formulae define ϕ as Latitude, rather than as $(90 - \text{Latitude})$. The formulae below are modified from Iliffe's formulae to make them consistent with the conversion formulae for the sphere above.

Latitude / Longitude to Cartesian:

Given a Latitude and Longitude in Degrees, Spheroid Equatorial and Polar Axes, and an optional height above the spheroid, calculate X, Y and Z as follows:

$$X = (\nu + h)\sin\phi\cos\theta$$
$$Y = (\nu + h)\sin\phi\sin\theta$$
$$Z = \left((1-e^2)\nu + h\right)\cos\phi$$

where:

$\phi = \dfrac{(90 - \text{Latitude})\pi}{180}$ = Angle from North Pole down to Latitude, in Radians

$\theta = \dfrac{(\text{Longitude})\pi}{180}$ = Angle from Greenwich to Longitude, in Radians

a = Spheroid equatorial radius (i.e. semi-major axis)

b = Spheroid polar radius (i.e. semi-minor axis)

e^2 = Spheroid eccentricity (squared) = $\dfrac{a^2 - b^2}{a^2}$

$\nu = \dfrac{a}{\sqrt{1 - e^2\cos^2\phi}}$

h = Height above the spheroid

<u>Cartesian to Latitude / Longitude Coordinates on the Spheroid:</u>

$$\text{Latitude} = 90 - \phi\left(\frac{180}{\pi}\right)$$
$$\text{Longitude} = \theta\left(\frac{180}{\pi}\right)$$

where:

$\phi = \arctan[2]\left(P - e^2 a\cos^3 u,\ Z + \varepsilon b\sin^3 u\right)$ = Angle from North Pole down to Latitude, in radians

$\theta = \arctan[2](Y,\ X)$ = Angle from Greenwich to Longitude, in radians

a = Spheroid equatorial radius (i.e. semi-major axis)

b = Spheroid polar radius (i.e. semi-minor axis)

$P = \sqrt{X^2 + Y^2}$

$u = \arctan[2](Za, Pb)$

e^2 = Spheroid eccentricity (squared) = $\dfrac{a^2 - b^2}{a^2}$

$\varepsilon = \dfrac{e^2}{1 - e^2}$

arctan[2] = Arctangent, adjusted for quadrant (see General Geometric Functions)

Arctan[2] Function

Various functions in this extension calculate arctangents. However, there is a problem with the basic function arctan because it does not account for quadrant. For example, given that $\tan A = \frac{\Delta Y}{\Delta X}$, then $\arctan \frac{\Delta . Y}{\Delta X} = A$, where A is in radians. However, this simple arctan function does not properly account for the signs of ΔX and ΔY and will only return values ranging between $\pm \frac{\pi}{2}$. The arctan[2] function checks the signs of ΔX and ΔY and returns a value of A radians that correctly ranges from $-\pi$ to π.

Unfortunately, Visual Basic 6 does not have a function to calculate arctangent in this manner. Many programming languages such as C++, PHP, C# and VB.NET include the "atan2" function which does the trick. You simply specify X and Y separately and the function determines the quadrant. **NOTE:** Microsoft Excel also has an "atan2" function, but for some reason the Excel version takes the ΔX and ΔY values in the order of (x, y) while all other implementations in the civilized world appear to take these values in the order of (y, x). Therefore be careful if you use this function in both your code and in Excel!

Given the lack of an Atan2 function in VB6 and VBA, I wrote my own function as follows:

```
Const dblPi As Double = 3.14159265358979
Public Function atan2(Y As Double, X As Double) As Double
  If X > 0 Then
    atan2 = Atn(Y / X)
  ElseIf X < 0 Then
    If Y = 0 Then
      atan2 = (dblPi - Atn(Abs(Y / X)))
    Else
      atan2 = Sgn(Y) * (dblPi - Atn(Abs(Y / X)))
    End If
  Else    ' IF X = 0
    If Y = 0 Then
      atan2 = 0
    Else
      atan2 = Sgn(Y) * dblPi / 2
    End If
  End If
End Function
```

Converting between Radians and Decimal Degrees

$$\text{Radians clockwise from North} = Degrees\left(\frac{\pi}{180}\right)$$

$$\text{Degrees clockwise from North} = Radians\left(\frac{180}{\pi}\right)$$

Spherical Radius Derived from Spheroid

When applying spherical functions, this extension assumes a sphere with the same volume as the actual data spheroid. For example, if the data used the WGS 84 spheroid (with semi-major axis = 6378137m, and semi-minor axis = 6356752.31424518m), then the spherical functions would be applied to a sphere with a radius of 6371000.79000915 meters derived as follows:

WGS 84 Spheroid:

$$\text{Semi-major axis } (a) = 6378137.0 \text{ m}$$

$$\text{Semi-minor axis } (b) = 6356752.31424518 \text{ m}$$

Volume of WGS 84 Spheroid:

$$V = \frac{4}{3}\pi a^2 b \qquad (\text{WGS 84 is an oblate spheroid})$$

Volume of Sphere with Radius R:

$$V = \frac{4}{3}\pi R^3$$

Therefore solve for R when $\frac{4}{3}\pi R^3 = \frac{4}{3}\pi a^2 b$

$$R^3 = a^2 b \qquad (\text{Like terms cancel})$$

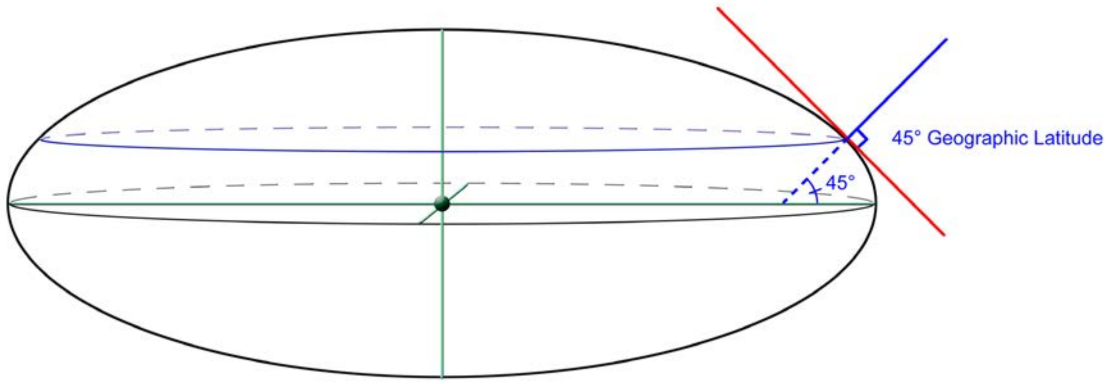
$$R = \sqrt[3]{a^2 b} = \sqrt[3]{(6378137.0^2)(6356752.31424518)}$$

$$= 6371000.79000915 \text{ m}$$

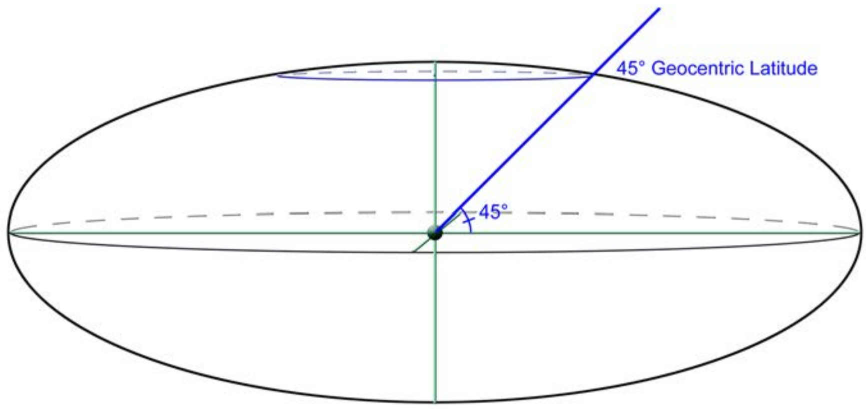
Planetocentric vs. Planetographic Coordinate Systems

For this tool, the terms Planetocentric (or Geocentric) and Planetographic (or Geographic, sometimes Geodetic) refer to the method of defining Latitude values on the surface of the spheroid.

Historically, latitude at a point on the earth has been determined by measuring the angle from the horizon at that point to some fixed reference point in space (the North Star, for example). Technically, this method produces the angle between the Normal of the spheroid (i.e. the vector perpendicular to the surface of the spheroid) at that point. Latitude coordinates produced by this method are referred to as Geographic (or Geodetic) coordinates. Planetary spheroids are typically slightly flattened spheres, so the Normal of the spheroid does not actually intersect the centroid of the planet. The illustration below greatly exaggerates the flattening to demonstrate this concept.



An alternative method would be to use the center of the spheroid as a reference point, and define "Latitude" as the angle between a line connecting the spheroid center to the point on the surface, and another line connecting the spheroid center to a point on the spheroid equator. Latitude coordinates produced by this method are referred to as Geocentric coordinates.



Because the prefix "Geo" implies Earth-based measures, I have substituted the prefix "Planeto" to create more general terms. **Note:** Most earth-based projections and datums are Planetographic, while most extra-terrestrial datums are Planetocentric.

Note: Longitude values are unaffected by whether the data are Planetocentric or Planetographic. The difference between Planetocentric and Planetographic latitude values depends on the flattening of the spheroid, where greater flattening leads to greater differences. If the spheroid were a true sphere (with no flattening), then Planetocentric and Planetographic latitudes would be identical.

EQUATIONS

(Adapted from p. 17 of Snyder [1987])

$$\text{Planetocentric Latitude } (\psi) = \frac{180\left(\arctan\left(\left(\frac{b}{a}\right)^2 \tan\left(\frac{\phi\pi}{180}\right)\right)\right)}{\pi}$$

$$\text{where } \phi = \text{Planetographic Latitude}$$
$$a = \text{Semi-major axis radius}$$
$$b = \text{Semi-minor axis radius}$$

$$\text{Planetographic Latitude } (\phi) = \frac{180\left(\arctan\left(\left(\frac{a}{b}\right)^2 \tan\left(\frac{\psi\pi}{180}\right)\right)\right)}{\pi}$$

$$\text{where } \psi = \text{Planetocentric Latitude}$$
$$a = \text{Semi-major axis radius}$$
$$b = \text{Semi-minor axis radius}$$

Revisions

Version 1.0

- Build 1.0.291 (January 1, 2008): Initial Release
- Build 1.0.293 (January 2, 2008): Minor aesthetic changes to dialogs; fixed some typos in manual.
- January 3, 2008: Corrected typo in manual.

Version 1.1

- Build 1.1.73 (January 9, 2008):
    - Added "Build Polygon" function
    - Added Polyline Azimuth options to "Calculate Geometry" tool.
    - Added code to disable "Build Polygon" tool if user was running a pre-9.2 version of ArcGIS.
    - Added code to prevent error messages when some functions were cancelled.
    - Added code to force shapefile field names to conform to dBASE restrictions, when field data were copied from non-shapefile sources.
    - Added scroll bars to the message section of the "Message Box" dialog.
    - Added persistence code so the document would remember the last save folder, and the Progress Dialog parameters, from one session to the next.
    - Fixed the link in the "About" dialog so that it pointed to the correct web page.
- Build 1.1.76 (January 10, 2008)
    - Fixed an error in the "Spheroidal Length" Geometry function in which it tried to write to the wrong field.
    - Modified "Azimuth" functions to return -9999 if the polyline was empty, or if the start and end points were coincident.
- January 14, 2008
    - Minor changes to manual.
- March 6, 2008
    - Minor changes to manual.
- Build 1.1.85 (March 28, 2008)
    - Repaired a bug in the "Calculate Geometry" dialog in which it might crash with the error message "Object variable or With block variable not set", occuring at line 48 of the form frmGeometry.frm, if no layers are selected in the Map Table of Contents.

Version 2.0

- Build 2.0.57 (February 12, 2009):
    - Modified sphere and spheroid functions to use spheroid from actual data rather than defaulting to WGS 1984.
    - Added functions to transform between planetocentric and planetographic coordinate systems.
    - Added functions to wrap data to pre-defined longitude ranges.
    - Added functions to shift longitude values.
    - Added output options to save to personal and file geodatabases as well as shapefiles.
    - Fixed a bug with the Spheroidal Multipoint Centroid in which it was calculating on the sphere rather than the spheroid.
    - Modified "Graphics to Feature Class" tool so that the new feature class would be symbolized with the symbol of the first graphic, if available.
    - Fixed bug in "Graphics to Feature Class" tool which caused a crash if there were grouped graphics available.
    - Disabled "Convert Graphics to Features" option if either map document currently opened to Layout view, or if no graphics were avaailable in map view.
    - Fixed bug in "Calculate Geometry" function in which sperical areas and centroids would occasionally crash if three vertices were on the same line of longitude.

- Fixed a bug in "Select Graphics" function in which it always selected graphics from focus map, even if document was set to Layout view.

- Build 2.1.72 (October 21, 2010)
  - Rearranged tools and menus
  - Added installation routines for ArcGIS 10.0.
  - Various cosmetic and efficiency modifications.
- November 11, 2010
  - Fixed typo on p. 62 of manual: Changed

$$\mathrm{Z} = R\cos\phi = R\cos(0.79) = 0.5R$$

to

$$\mathrm{Z} = R\cos\phi = R\cos(1.05) = 0.5R$$

- Build 2.1.80 (April 11, 2011)
  - Wrote work-arounds for an ArcGIS 10 bug which produced an error referring to an unknown error in the geometry system.
- Build 2.1.81 (May 12, 2011)
  - Fixed a bug in which spheroidal distances were being calculated incorrectly by a small amount because the spheroid flattening was incorrect. It was being calculated as $(a-b)/b$ instead of $(a-b)/a$.
- Build 2.1.82 (May 12, 2011)
  - Fixed a bug in which the suggested field names for starting and ending azimuths for spheroidal polylines were reversed, so the new "End Azimuth" values would (unless the user changed the suggested field name manual) show up in a field named "Start_Az" and vice versa.
- Build 2.1.85 (June 1, 2011)
  - Fixed a bug in the "Split Multipart" tool in which it would raise an error upon completion of splitting multipart polylines, with the message "Subscript out of range" at line 1301.
  - Fixed a bug in the "Split Multipart" tool in which it would crash when attempting to split a multipoint feature.

References

Chovitz, Bernard. 2002. In Memoriam: Thaddeus Vincenty. Available at http://www.gfy.ku.dk/~iag/newslett/news7606.htm. Last viewed December 28, 2007.

ESRI. 2006. ArcGIS 9 Media Kit: ESRI Data and Maps. County boundaries derived from US Census data. ESRI, Redlands, California, USA.

Iliffe, Jonathan. 2000. Datums and Map Projections for Remote Sensing, GIS and Surveying. Whittles Publishing, Scotland, UK. 150 pp.

Kennedy, Melita & Steve Kopp. 2000. Understanding Map Projections. ESRI, Redlands, California, USA. 110 pp.

Mardia, Kanti V., & Peter E. Jupp. 2000. Directional Statistics. John Wiley & Sons Ltd. West Sussux, England. 427 pp.

Snyder, John P. 1983. Map Projections Used by the U.S. Geological Survey, 2nd Ed. Geological Survey Bulletin 1532. U.S. Government Printing Office, Washington, DC, USA. 313 pp.

Snyder, John P. 1987. Map Projections – A Working Manual. Geological Survey Professional Paper 1395. U.S. Government Printing Office, Washington, DC, USA. 383 pp.

Veness, Chris. 2007a. Calculate distance, bearing and more between two latitude/longitude points. Available at http://www.movable-type.co.uk/scripts/latlong.html. Last viewed December 15, 2007.

Veness, Chris. 2007b. Vincenty formula for distance between two latitude/longitude points. Available at http://www.movable-type.co.uk/scripts/latlong-vincenty-direct.html. Last viewed December 15, 2007.

Vincenty, Thaddeus. 1975. Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations. Surv. Rev., XXII(176):88–93.

Wikipedia. 2007. Haversine formula. Available at: http://en.wikipedia.org/wiki/Haversine_formula. Last viewed December 25, 2007.

Williams, Ed. 2006. Aviation Formulary V1.43. Available at http://williams.best.vwh.net/avform.htm#Crs. Last viewed December 26, 2007.

Zwillinger, Daniel. 2003. CRC standard mathematical tables and formulae. CRC Press. www.crcpress.com. 912 pp.
