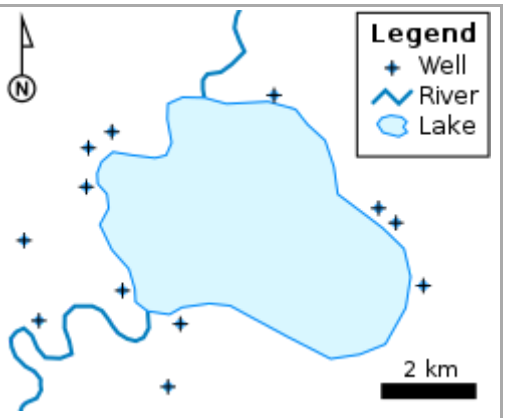
Shapefile

From Wikipedia, the free encyclopedia

The **shapefile** format is a geospatial vector data format for geographic information system (GIS) software. It is developed and regulated by Esri as a mostly open specification for data interoperability among Esri and other GIS software products.[1] The shapefile format can spatially describe vector features: points, lines, and polygons, representing, for example, water wells, rivers, and lakes. Each item usually has attributes that describe it, such as *name* or *temperature*.

Shapefile



A vector map, with points, polylines and polygons

Filename extension	`.shp`, `.shx`, `.dbf`
Internet media type	`x-gis/x-shapefile`
Developed by	Esri
Type of format	GIS
Standard	Shapefile Technical Description (http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf)

Overview

The shapefile format is a digital vector storage format for storing geographic location and associated attribute information. This format lacks the capacity to store topological information. The shapefile format was introduced with ArcView GIS version 2 in the early 1990s. It is now possible to read and write geographical datasets using the shapefile format with a wide variety of software.

The shapefile format stores the geometry as primitive geometric shapes like points, lines, and polygons. These shapes, together with data attributes that are linked to each shape, create the representation of the geographic data. The term "shapefile" is quite common, but the format consists of a collection of files with a common filename prefix, stored in the same directory. The three *mandatory* files have filename extensions `.shp`, `.shx`, and `.dbf`. The actual *shapefile* relates specifically to the `.shp` file, but alone is incomplete for distribution as the other supporting files are required. Legacy GIS software may expect that the filename prefix be limited to eight characters to conform to the DOS 8.3 filename convention, though modern software applications accept files with longer names.

Mandatory files

- `.shp` — shape format; the feature geometry itself {content-type: x-gis/x-shapefile}
- `.shx` — shape index format; a positional index of the feature geometry to allow seeking forwards and backwards quickly {content-type: x-gis/x-shapefile}
- `.dbf` — attribute format; columnar attributes for each shape, in dBase IV format {content-type: application/octet-stream OR text/plain}

Other files

- `.prj` — projection description, using a well-known text representation of coordinate reference systems {content-type: text/plain OR application/text}
- `.sbn` and `.sbx` — a spatial index of the features {content-type: x-gis/x-shapefile}
- `.fbn` and `.fbx` — a spatial index of the features that are read-only {content-type: x-gis/x-shapefile}
- `.ain` and `.aih` — an attribute index of the active fields in a table {content-type: x-gis/x-shapefile}
- `.ixs` — a geocoding index for read-write datasets {content-type: x-gis/x-shapefile}
- `.mxs` — a geocoding index for read-write datasets (ODB format) {content-type: x-gis/x-shapefile}
- `.atx` — an attribute index for the `.dbf` file in the form of *`shapefile`*`.`*`columnname`*`.atx` (ArcGIS 8 and later) {content-type: `x-gis/x-shapefile` }
- `.shp.xml` — geospatial metadata in XML format, such as ISO 19115 or other XML schema {content-type: application/fgdc+xml}
- `.cpg` — used to specify the code page (only for `.dbf`) for identifying the character encoding to be used {content-type: `text/plain` OR `x-gis/x-shapefile` }
- `.qix` — an alternative quadtree spatial index used by MapServer and GDAL/OGR software {content-type: x-gis/x-shapefile}

In each of the `.shp`, `.shx`, and `.dbf` files, the shapes in each file correspond to each other in sequence (i.e., the first record in the `.shp` file corresponds to the first record in the `.shx` and `.dbf` files, etc.). The `.shp` and `.shx` files have various fields with different endianness, so an implementer of the file formats must be very careful to respect the endianness of each field and treat it properly.

Shapefile shape format (`.shp`)

The main file (`.shp`) contains the geometry data. Geometry of a given feature is stored as a set of vector coordinates.[1]:5 The binary file consists of a single fixed-length header followed by one or more variable-length records. Each of the variable-length records includes a record-header component and a record-contents component. A detailed description of the file format is given in the *ESRI Shapefile Technical Description*.[1] This format should not be confused with the AutoCAD shape font source format, which shares the `.shp` extension.

The 2D axis ordering of coordinate data assumes a Cartesian coordinate system, using the order (X Y) or (Easting Northing). This axis order is consistent for Geographic coordinate systems, where the order is similarly (longitude latitude). Geometries may also support 3- or 4-dimensional Z and M coordinates, for elevation and measure, respectively. A Z-dimension stores the elevation of each coordinate in 3D space, which can be used for analysis or for visualisation of geometries using 3D computer graphics. The user-defined M dimension can be used for one of many functions, such as storing linear referencing measures or relative time of a feature in 4D space.

The main file header is fixed at 100 bytes in length and contains 17 fields; nine 4-byte (32-bit signed integer or int32) integer fields followed by eight 8-byte (double) signed floating point fields:

Header of a .shp file format

Bytes	Type	Endianness	Usage
0–3	int32	big	File code (always hex value `0x0000270a`)
4–23	int32	big	Unused; five uint32
24–27	int32	big	File length (in 16-bit words, including the header)
28–31	int32	little	Version
32–35	int32	little	Shape type (see reference below)
36–67	double	little	Minimum bounding rectangle (MBR) of all shapes contained within the dataset; four doubles in the following order: min X, min Y, max X, max Y
68–83	double	little	Range of Z; two doubles in the following order: min Z, max Z
84–99	double	little	Range of M; two doubles in the following order: min M, max M

The file then contains any number of variable-length records. Each record is prefixed with a record header of 8 bytes:

Bytes	Type	Endianness	Usage
0–3	int32	big	Record number (1-based)
4–7	int32	big	Record length (in 16-bit words)

Following the record header is the actual record:

Bytes	Type	Endianness	Usage
0–3	int32	little	Shape type (see reference below)
4–	–	–	Shape content

The variable-length record contents depend on the shape type, which must be either the shape type given in the file header or Null. The following are the possible shape types:

Value	Shape type	Fields
0	Null shape	None
1	Point	X, Y
3	Polyline	MBR, Number of parts, Number of points, Parts, Points
5	Polygon	MBR, Number of parts, Number of points, Parts, Points
8	MultiPoint	MBR, Number of points, Points
11	PointZ	X, Y, Z *Optional*: M
13	PolylineZ	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points, Z range, Z array *Optional*: M range, M array
15	PolygonZ	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points, Z range, Z array *Optional*: M range, M array
18	MultiPointZ	*Mandatory*: MBR, Number of points, Points, Z range, Z array *Optional*: M range, M array
21	PointM	X, Y, M
23	PolylineM	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points *Optional*: M range, M array
25	PolygonM	*Mandatory*: MBR, Number of parts, Number of points, Parts, Points *Optional*: M range, M array
28	MultiPointM	*Mandatory*: MBR, Number of points, Points *Optional Fields*: M range, M array
31	MultiPatch	*Mandatory*: MBR, Number of parts, Number of points, Parts, Part types, Points, Z range, Z array *Optional*: M range, M array

Shapefile shape index format (`.shx`)

The index contains positional index of the feature geometry and the same 100-byte header as the `.shp` file, followed by any number of 8-byte fixed-length records which consist of the following two fields:

Bytes	Type	Endianness	Usage
0–3	int32	big	Record offset (in 16-bit words)
4–7	int32	big	Record length (in 16-bit words)

Using this index, it is possible to seek backwards in the shapefile by, first, seeking backwards in the shape index (which is possible because it uses fixed-length records), then reading the record offset, and using that offset to seek to the correct position in the `.shp` file. It is also possible to seek forwards an arbitrary number of records using the same method.

It is possible to generate the complete index file given a lone `.shp` file. However, since a shapefile is supposed to always contain an index, doing so counts as repairing a corrupt file.[2]

Shapefile attribute format (.dbf)

This file stores the attributes for each shape; it uses the dBase IV format. The format is public knowledge, and has been implemented in many dBase clones known as xBase. The open-source shapefile C library, for example, calls its format "xBase" even though it's plain dBase IV.[3]

The names and values of attributes are not standardized, and will be different depending on the source of the shapefile.

Shapefile spatial index format (.sbn)

This is a binary spatial index file, which is used only by Esri software. The format is not documented by Esri. However it has been reverse-engineered and documented by the open source community. The 100-byte header is similar to the one in `.shp`.[4] It is not currently implemented by other vendors. The `.sbn` file is not strictly necessary, since the `.shp` file contains all of the information necessary to successfully parse the spatial data.

Limitations

Topology and the shapefile format

The shapefile format does not have the ability to store topological information. The ESRI ArcInfo coverages and personal/file/enterprise geodatabases do have the ability to store feature topology.

Spatial representation

The edges of a polyline or polygon are composed of points. The spacing of the points implicitly determines the scale at which the feature is useful visually. Exceeding that scale results in jagged representation. Additional points would be required to achieve smooth shapes at greater scales. For features better

represented by smooth curves, the polygon representation requires much more data storage than, for example, splines, which can capture smoothly varying shapes efficiently. None of the shapefile format types supports splines.

Data storage

The size of both `.shp` and `.dbf` component files cannot exceed 2 GB (or 2^{31} bytes) — around 70 million point features at best.[5] The maximum number of feature for other geometry types varies depending on the number of vertices used.

The attribute database format for the `.dbf` component file is based on an older dBase standard. This database format inherently has a number of limitations:[5]

- While the current dBase standard, and GDAL/OGR (the main open source software library for reading and writing shapefile format datasets) support null values, ESRI software represents these values as zeros — a very serious issue for analyzing quantitative data, as it may skew representation and statistics if null quantities are represented as zero
- Poor support for Unicode field names or field storage
- Maximum length of field names is 10 characters
- Maximum number of fields is 255
- Supported field types are: floating point (13 character storage), integer (4 or 9 character storage), date (no time storage; 8 character storage), and text (maximum 254 character storage)
- Floating point numbers may contain rounding errors since they are stored as text

Mixing shape types

Because the shape type precedes each geometry record, a shapefile is technically capable of storing a mixture of different shape types. However, the specification states, "All the non-Null shapes in a shapefile are required to be of the same shape type." Therefore, this ability to mix shape types must be limited to interspersing null shapes with the single shape type declared in the file's header. A shapefile must not contain both polyline and polygon data, for example, the descriptions for a well (point), a river (polyline), and a lake (polygon) would be stored in three separate datasets.

See also

- Geographic information system
- Open Geospatial Consortium
- Open Source Geospatial Foundation (OSGeo)
- List of geographic information systems software
- Comparison of geographic information systems software

External links

- Shapefile file extensions (http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Shapefile_file_extensions/005600000003000000/) – Esri Webhelp docs for ArcGIS 10.0 (2010)
- Esri – Understanding Topology and Shapefiles (http://www.esri.com/news/arcuser/0401/topo.html)

- shapelib.maptools.org (http://shapelib.maptools.org/) – Free c library for reading/writing shapefiles
- Python Shapefile Library (https://github.com/GeospatialPython/pyshp) – Open Source (MIT License) Python library for reading/writing shapefiles
- Shapefile Projection Finder - Detect unknown projection of a shapefile automatically (https://www.egger-gis.at/automatic-projection-detection/shapefile-projectionfinder/) [6][7]
- Java Shapefile (https://github.com/gallandarakhneorg/afc/tree/master/advanced/shapefile) and Dbase (https://github.com/gallandarakhneorg/afc/tree/master/advanced/dbasefile) Libraries – Open Source (Apache License) Java libraries for reading/writing shapefiles and the associated dBase files (libraries are part of the AFC Library (http://www.arakhne.org/afc/index.html) but could be used independently)

References

1. ESRI (July 1998). "ESRI Shapefile Technical Description" (http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf) (PDF). Retrieved 2007-07-04.
2. Rollason, Ed. "qgis - Creating missing .shx file?" (https://gis.stackexchange.com/a/306223). *Geographic Information Systems Stack Exchange*.
3. "Shapefile C Library V1.2" (http://shapelib.maptools.org/).
4. (PDF). 13 August 2016 https://web.archive.org/web/20160813212443/https://pyshp.googlecode.com/files/sbn_format.pdf (https://web.archive.org/web/20160813212443/https://pyshp.googlecode.com/files/sbn_format.pdf). Archived from the original on 13 August 2016. {{cite web}}: Missing or empty |title= (help)
5. "ArcGIS Desktop 9.3 Help – Geoprocessing considerations for shapefile output" (http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?TopicName=Geoprocessing%20considerations%20for%20shapefile%20output). Esri. April 24, 2009.
6. Egger, Manfred. "Shapefile Projectionfinder" (https://www.egger-gis.at/app/download/13175253496/POSTER_EGGER_FOSS4G.pdf?t=1485153945) (PDF). *www.egger-gis.at*.
7. "Shapefile Projectionfinder" (http://2016.foss4g.org/programme.html#poster-session).