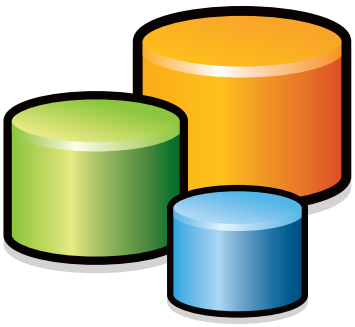


The Top Nine Reasons to Use a File Geodatabase

A scalable and speedy choice for single users or small groups

By Colin Childs, ESRI Education Services

Whether you are working with large or small datasets, file geodatabases optimized for use in ArcGIS are ideal for storing and managing geospatial data. Whether you are working on a single-user project or a project involving a small group with one or several editors, you really should consider using a file geodatabase rather than a personal geodatabase or collection of shapefiles. File geodatabases offer structural, performance, and data management advantages over personal geodatabases and shapefiles.

Structural	1. Improved versatility and usability 2. Optimized performance 3. Few size limitations
Performance	4. Easy data migration 5. Improved editing model 6. Storing rasters in the geodatabase
Data Management	7. Customizable storage configuration 8. Allows updates to spatial indexes 9. Allows the use of data compression

1 Improved versatility and usability

The file geodatabase is stored as a system folder that contains binary files that store and manage geospatial data. It is available at all ArcGIS license levels and functions in the same fashion on Windows and UNIX (Solaris and Linux) operating systems. This storage system is based on relational principles and provides a simple, formal data model for storing and working with information in tables.

Open a file geodatabase folder in Windows Explorer, and it looks like any other folder. You can view the files it contains: geographic data, attribute data, index files, lock files, signature files, and other files. Each feature class or table in the geodatabase is stored in two or more files.

2 Optimized performance

The data structure of a file geodatabase is optimized for performance and storage. Although individual feature classes can be as large as 1 terabyte (TB) in size and contain hundreds of millions of features, they still provide fast performance. File geodatabases significantly outperform shapefiles for operations involving attributes and allow scaling of dataset size limits way beyond those of shapefiles.

3 Few size limitations

Database size is limited only by available disk space. By default, individual tables and feature classes can be up to 1 TB. With the use of configuration keywords, this can be expanded to 256 TB.

4 Easy data migration

Because file geodatabases and personal geodatabases are both designed to be edited by a single user and do not support geodatabase versioning, data migration between them is easy.

5 Improved editing model

File geodatabases do not lock down the whole geodatabase if a user is editing a feature class. An edit model similar to that used for shapefiles is deployed. This model supports a single data editor and many data viewers concurrently. Stand-alone feature classes, tables, and feature dataset contents can be edited by different editors simultaneously without the entire geodatabase being locked. If a feature class in a feature dataset is being edited, all feature classes in that feature dataset are unavailable for editing, but features may still be viewed and selected in ArcMap.

6 Storing rasters in the geodatabase

Raster storage in a file geodatabase shares functionality from both the ArcSDE geodatabase and the personal geodatabase. Managed raster data is stored in the same way as in an ArcSDE geodatabase, and unmanaged raster data is stored in the same way as in a personal geodatabase.

Managed raster data is subdivided into small, manageable areas called tiles, stored as binary large objects (BLOBs) in the database. The tiling is automatic and invisible to end users.

These tiles are indexed and pyramided for fast display performance. Pyramiding allows the geodatabase to fetch only data at the specified resolution or level required for display.

Unmanaged rasters are maintained by users. Only the path to the location on the disk where the raster dataset is stored is maintained in the file geodatabase.

7 Customizable storage configuration

When creating a dataset, apply optional configuration keywords to customize data storage. Keywords optimize storage for a particular type of data to improve storage efficiency and performance.

Property	Settings
File geodatabase size	Technically no limit
Table or feature class size	1 TB (default) 256 TB (with keyword)
Number of feature classes and tables	2,147,483,647
Number of fields in a table or feature class	65,534
Number of rows in a table or feature class	4,294,967,295
Geodatabase name length	Operating system limits for a folder name
Table or feature class name length	160 characters
Field name length	64 characters
Character field width	2,147,483,647

File geodatabase parameters

Action in ArcCatalog	Method
Copying stand-alone feature classes or feature datasets	Use ArcCatalog's Copy and Paste commands.
Copying entire geodatabase	Use the Export > XML Workspace Document command to export the entire database to an XML file. Create a new, empty file geodatabase. Import > XML Workspace Document to import the data and schema from the XML file into a new file geodatabase.
Converting from low- to high-precision data	Automatically converted to high-precision with Copy/Paste and Export to XML Workspace Document.
Migrating domains	Copy referenced feature class using Copy/Paste.
Migrating unreferenced domains	Use the Export > XML Workspace Document command to export the entire database to an XML file. Import > XML Workspace Document to import from the XML file.
Incorporating shapefiles, coverages, and CAD data	Use Export > To Geodatabase or Import > To Geodatabase command.

Methods for moving data from a personal to a file geodatabase

In most cases, the DEFAULTS keyword is used. Use the keywords shown in the accompanying table if the dataset exceeds 1 TB or is less than 4 GB. If no configuration keyword is specified, the DEFAULTS keyword is used.

Allows updates to spatial index settings

Spatial indexes are used by ArcGIS to quickly locate features when you display, edit, or query data. An appropriate spatial index is important, especially when you are working with large datasets. While the spatial index of a personal geodatabase feature class uses a single grid size that cannot be modified, the spatial index of a file geodatabase feature class uses as many as three grid sizes, which can be modified. Additional grids allow feature classes with features of very different sizes to be queried more quickly.

ArcGIS automatically rebuilds the spatial index at the end of some update operations to ensure the index and its grid sizes are optimal. However, in some rare cases, you may need to manually recalculate the index.

Allows the use of data compression

Vector data can be stored in a file geodatabase in a compressed, read-only format that reduces storage requirements. Compression reduces the geodatabase's overall footprint on disk without reducing the performance. Once compressed, display and query performance are comparable to uncompressed data. Compressed data is in a direct-access format, so there is no need to uncompress the data because ArcGIS and ArcReader can read it directly.

Continued on page 14

The Top Nine Reasons to Use a File Geodatabase

Continued from page 13

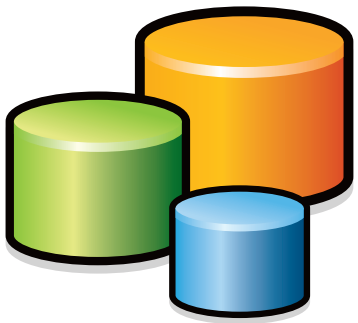
Sample data	Number/Type of features	Number of fields	Uncompressed size	Compressed size	Compression ratio (1:x)
Germany streets	7,118,614 lines	83	2.3 GB	511 MB	4.5
U.S. census blocks	8,205,055 points	11	705 MB	162 MB	4.4
Europe rails	383,531 lines	12	58 MB	17 MB	3.4
Calgary addresses	285,285 points	8	21 MB	7.4 MB	2.8
Calgary buildings	319,000 polygons	9	48 MB	20 MB	2.4
Europe water	232,375 polygons	10	176 MB	125 MB	1.4

Effects of compression by feature type and numbers

Compression is ideally suited to mature datasets that do not require further editing. However, if required, a compressed dataset can always be uncompressed and returned to its original, read/write format. The compression method applied is lossless, so no information is lost. One of the most important factors affecting spatial data compression is the average number of vertices per feature. As a general rule, the fewer vertices/features, the greater the compression. The tables above illustrate the effects of applying compression.

Ready to Try File Geodatabases?

The file geodatabase provides a widely available, simple, and scalable geodatabase solution for all users that can work across operating systems. It can scale up to handle very large datasets and still provide excellent performance. Its efficient data structure is optimized for performance and storage and uses about one-third the feature geometry storage required by shapefiles and personal geodatabases. File geodatabases also allow users to compress vector data to a read-only format that further reduces storage requirements.



Keyword name	Keyword setting	Usage
DEFAULTS	1 TB	Default settings
TEXT_UTF16	1 TB	Optimizes storage for non-Latin alphabet text
MAX_FILE_SIZE_4GB	4 GB	Restricts dataset to a maximum size of 4 GB; stores datasets less than 4 GB more efficiently than DEFAULTS keyword
MAX_FILE_ SIZE_256TB	256 TB	Used to create a dataset that is up to 256 TB in size

Keywords for customizing data storage

Action	Effect on spatial index
Create empty feature class.	Spatial index is created, but remains in unbuilt state. Grid sizes are set to 0,0,0.
Add/Create features in empty feature class.	Initial grid sizes 0,0,0 are automatically rebuilt as feature properties are updated.
Add different-sized features to feature class.	Manually update the index. The Indexes tab appears in the Feature Class Properties dialog box. Click the Recalculate button.
Load data using simple data loader or Append tool.	The spatial index is built as the final step of the loading process. Appropriate grid sizes are calculated for the new added features.
Import data.	The spatial index is automatically computed for the new feature class.
Copy/Paste feature class personal to a file geodatabase.	The spatial index is automatically rebuilt.
Copy/Paste feature class from file or ArcSDE geodatabase.	The grid sizes are copied from the source and the spatial index is automatically built.
Compress data.	An alternate indexing method is automatically applied and cannot be modified.
Uncompress data.	Precompression index is automatically reestablished.

File geodatabase and spatial indexes table

Compress	Result
Full geodatabase	All feature classes, feature datasets, and tables compressed
Stand-alone feature classes	A single feature class compressed
Feature dataset	All feature classes in the feature dataset compressed
Raster dataset	Compression already applied through pyramid creation

Effects of compression on data storage

Sample data	Uncompressed size	Compressed size	Compression percentage
U.S. census block centroids	705 MB	162 MB	77
California roads	329 MB	83 MB	74.7
Manhattan, NY parcels	4.42 MB	1.92 MB	56.6
Vegetation data	3.31 MB	2.32 MB	30
Riverside, CA buildings	644 KB	314.5 KB	51.2

Effects of compression by percentage

Compressing versus Compacting

These two operations are conceptually similar in that each can result in more compact storage, but as applied to file geodatabases, these are two unrelated operations.

Compaction

- Tidies up the storage of records in files by reordering them and eliminating free space
- Is recommended if data is frequently added and deleted
- Can reduce file sizes and improve performance
- Does not affect read/write capability
- Should be performed regularly

Compression

- Reduces storage requirements
- Improves database performance
- Makes database or feature classes read-only
- Should be performed as needed

Learn More

Several courses on building and using geodatabases are offered by ESRI. Visit www.esri.com/training to learn more about these offerings.

- *Building Geodatabases,* an instructor-led course
- *Creating and Editing Geodatabase Features with ArcGIS Desktop (for ArcEditor and ArcInfo),* a Web course
- *Introduction to the Multiuser Geodatabase,* an instructor-led course
- *Creating, Editing, and Managing Geodatabases for ArcGIS Desktop,* a Web course