# File geodatabase
# ArcGIS 9.2

ESRI®

# File geodatabase at 9.2

- Introduction
  - Comparisons and capabilities
- Storage
  - Storage limits and requirements, configuration keywords, rasters
- Compression
- Migrating to the file geodatabase

# Why the file geodatabase?

- An alternative to the personal geodatabase
  - Support full geodatabase model
  - Work the same way
- Eliminate database size limit of 2 GB
- Require high performance
  - Personal gdbs slow after database reaches ~ 500 MB
- Make available for more platforms
  - Eliminate dependency on JET engine

# Advantages over personal geodatabase

- No storage size limit
- Improved performance
- Reduced storage requirements
- Customize storage
  - Compression of vector data
  - Configuration keywords
- Additional raster data management functionality
- More platforms supported
  - Windows and UNIX (solaris and linux)

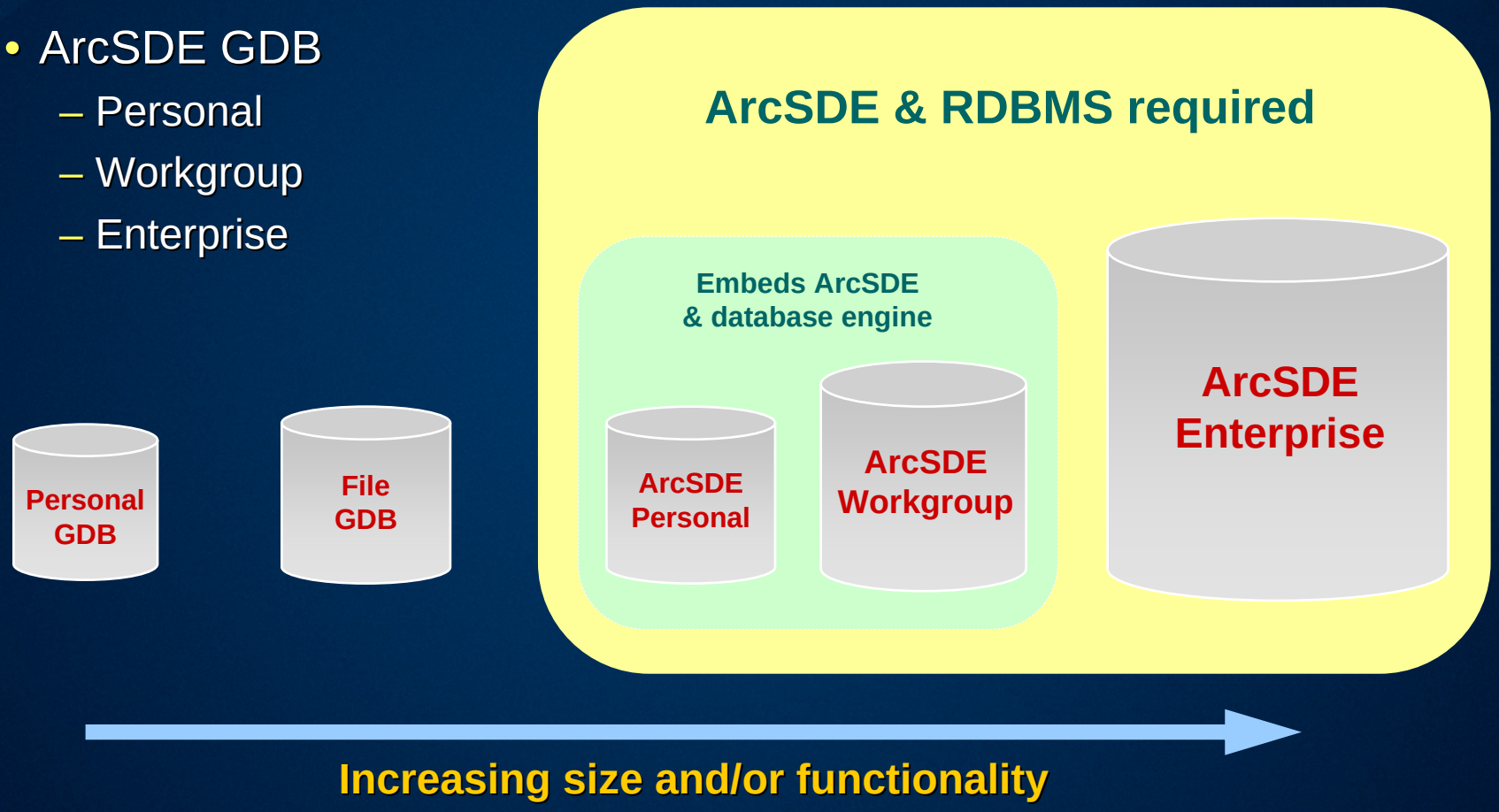
*ESRI will continue to support the personal geodatabase, but users are encouraged to migrate for these benefits*

# File geodatabase editing

- As with personal geodatabases, Single-user editing
  - Does not support versioning
- However, different locking model
- Locks apply to feature dataset, standalone feature class, table
  - No database-wide lock
  - More than one editor at a time is possible, for different datasets

# Geodatabase options at 9.2

- Personal GDB
- File GDB
- ArcSDE GDB
  - Personal
  - Workgroup
  - Enterprise



# Demo

## *Introducing the file geodatabase*

# File geodatabase at 9.2

- Introduction
  - Comparisons and capabilities
- Storage
  - Storage limits and requirements, configuration keywords, rasters
- Compression
- Migrating to the file geodatabase

# Storage limits

- No database size limit
- Per table limit: 1 TB (default)
- Same data in a file geodatabase takes up less disk space than personal gdbs, shapefiles
  - Amount of reduction varies by dataset
  - Storage on disk generally reduces by 50 to 75%

# Storage requirements (uncompressed)

| | Shapefile | Personal gdb | File gdb |
|---|---|---|---|
| US rivers and streams | 2.19 GB | Exceeds 2 GB limit | 878 MB |
| California roads | 1.23 GB | 684 MB | 329 MB |
| US census block centroids | 838 MB | 1.8 GB | 705 MB |
| US traffic analysis zones | 249 MB | 295 MB | 68 MB |
| US counties | 3.2 MB | 3.2 MB | 1.6 MB |

# Configuration keywords

- Predetermined keywords stored within the geodatabase
  - Very few options, cannot be customized
- Vast majority of users should use the default
  - 1 TB per table
  - UTF8 text attribute storage, optimal for latin alphabets
- UTF16
  - Use when lots of text in non-latin alphabet
- 4GB, 256TB keywords

# Raster data

- Unmanaged rasters stored the same way as in personal gdbs
- Managed rasters store the same way as in ArcSDE gdbs
  - Images store as BLOBs within the gdb
  - Supports JP2000, JPEG, LZ77 compression
  - Appending to mosaics is easier
  - Control resampling method when building pyramids: better display performance
  - Support partial pyramid updating
  - Faster copying between file gdb and ArcSDE gdb

# File geodatabase at 9.2

- Introduction
  - Comparisons and capabilities
- Storage
  - Storage limits and requirements, configuration keywords, rasters
- Compression
- Migrating to the file geodatabase

# Compression

- Vector feature classes and tables can optionally be stored in a read-only compressed format
- Advantage: Further reduce storage requirements
- Compression is lossless
- Direct access format, no uncompressing required

# Compression

| | Uncompressed | Compressed | Ratio |
|---|---|---|---|
| US census block centroids | 705 MB | 162 MB | 4.4 |
| California roads | 329 MB | 83 MB | 3.9 |
| Calgary buildings | 48 MB | 20 MB | 2.4 |
| US rivers and streams | 878 MB | 442 MB | 2.0 |
| Mexico roads | 3.5 MB | 2.7 MB | 1.3 |

Less vertices / feature = more compression

# Compression

- Compress / Uncompress tools
  - Right-click context menu commands
  - Geoprocessing tools: Data Management Toolbox > File Geodatabase toolset
- What you can compress
  - Geodatabase: all feature classes and tables compress
  - Feature dataset: all feature classes compress
  - Standalone feature class
  - Table
- Editing not allowed
- Properties of a compressed dataset cannot be edited

**Demo**

***File geodatabase compression***

# File geodatabase at 9.2

- Introduction
  - Comparisons and capabilities
- Storage
  - Storage limits and requirements, configuration keywords, rasters
- Compression
- Migrating to the file geodatabase

# Migrating to the file geodatabase

- Reasons to migrate from personal gdbs
  - no database size limit
  - improved performance
  - reduce storage requirements
  - to work on UNIX platforms
- Reasons not to migrate
  - very small datasets only, no advantage to moving
  - require ability to leverage Access
- Most users will benefit from migrating

# Migrating to the file geodatabase

- From a personal geodatabase:
  - Copy Paste (for feature datasets, classes, and tables)
  - Export to XML Workspace Document
  - Existing GP conversion tools
- From shapefiles, coverages or other formats:
  - Right-click and Export
  - Existing GP conversion tools
- Models or scripts for moving many datasets
- Creating new datasets
  - Works the same as for personal geodatabases

# Migrating to the file geodatabase

- Geoprocessing:
  - File gdb supported in all tools
  - Create File GDB tool

# Migrating to the file geodatabase

- SQL same as for shapefiles, coverages
- SQL differs from personal geodatabases
  - supports a subset of features and functions
  - syntax differs slightly
- Dialogs you create SQL expressions with help you with the correct syntax
  - appropriate delimiters for fields and values
  - relevant keywords and operators
- SQL defined for a personal gdb layer may not work once you've migrated its source data…

# Migrating to the file geodatabase

- Syntax differs from personal geodatabases:
  - delimit fields with “field”, not [field]
  - string searches case sensitive
  - wildcards _ and %, not ? and *
  - UPPER and LOWER, not UCASE and LCASE
  - precede dates with *date*, not #

# Migrating applications

- Update WorkspaceFactory to get the app working on file gdb
  - change AccessWorkspaceFactory to FileGDBWorkspaceFactory
  - change extenstion from “.mdb” to “.gdb”
- Update any SQL syntax
- Use load only mode to maximize data transfer performance

```
Dim pFeatureClassLoad As IFeatureClassLoad
Set pFeatureClassLoad = pFeatureClass
pFeatureClassLoad.LoadOnlyMode = true
```

- No other differences in ArcObjects

# Demo

***File geodatabase migration***

# For more information…

- The following on-line help topics are useful resources
- To find them, go to the Search tab and search for the topic title:
  - Types of geodatabases
  - Migrating to the file geodatabase
  - How raster data is stored in a geodatabase
  - Configuration keywords for file geodatabases
  - Setting spatial indexes
  - About compressing file geodatabase data
  - Compacting file and personal geodatabases
  - SQL Reference

# Performance Tips and Tricks

- Maintain File System
  - Defragment hard drives
  - Provide adequate free space
- Compact the geodatabase
- Use appropriate resolution
  - Default 0.0001 meter is OK, but can save space and improve performance with larger values. E.g. Fedlands feature class (original shapefile was ~ 500 MB)
    - 1.0 m = 99.25 MB
    - 0.1 m = 113.59 MB
    - 0.01 m = 126.21 MB
    - 0.001 m = 170.55 MB
    - 0.0001 m = 215.08 MB
- Try recalculating spatial index grid size for possible improvement in spatial filtering performance.

# Performance and Scalability testing

- The test objective was to measure ArcMap viewer performance and scalability using a very large raster-based File Geodatabase. The end goal is to determine if there is any performance degradation as concurrent user load is added against the FGDB and if so, determine the source of the bottleneck

- <u>File Server Configuration</u>
  - 2 x 3.2 GHz Xeon
  - 4 GB RAM
  - IBM DS4300 Storage with 10x146 GB Fibre Channel drives
  - Windows 2003 Server, SP1
  - ArcGIS Desktop 9.2, SP1

# Performance and Scalability testing

- The test data consisted of an unmanaged FGDB referencing 939 GB of data containing 3.4 million records and over 5.3 million files.
- The ESRI Performance Tool Set was used to generate and replay the map displays against the provided MXD. The loader scripts consisted of 240 pre-canned map extents where there were 40 extents used for each of the following extent scales:
- 1:4,000,000
- 1:2,000,000
- 1:1,000,000
- 1:500,000
- 1:250,000
- 1:125,000
- The baseline script used by the single ArcMap workstation client was similar containing 60 total extents, 10 for each extent scale.

# Performance and Scalability testing

- The testing showed that ArcMap and the File Geodatabase scaled well even under considerable simulated user load. Baseline client response time was maintained as additional load was added to the data server and no system bottlenecks were reached. At some point the data server would begin to become a bottleneck but it is thought that this would be well past 150 users for this specific configuration. Assuming an extrapolation of CPU processing and along with the use of Gigabit networking shows that a dual CPU 3.2 GHz data server could support up to 320 users if no bottlenecks were encountered.

# File Geodatabase – post 9.2

- More SQL to meet developer requirements.
  - SELECT DISTINCT
  - GROUP BY, ORDER BY
  - Correlated sub-queries.
  - Select list expressions
  - More powerful INSERT, UPDATE, DELETE statements with ExecuteSQL

# File Geodatabase – post 9.2

- Open API.
  - Home grown open source spatial data API (e.g. GDAL, FDO), or
  - Standard data access API (e.g., ODBC, JDBC, ADO.Net)
- Authentication/Authorization
- Additional Compression tools to support customized compression