



# Migrating Data From ArcStorm™ or Map Librarian to the Geodatabase

*An ESRI® Technical Paper • June 2003*

Copyright © 2003 ESRI  
All rights reserved.  
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

#### **U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS**

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, the ESRI globe logo, ArcStorm, AML, ArcInfo, ArcGIS, ArcTools, ArcSDE, ArcIMS, ArcCatalog, SDE, ArcPlot, ArcMap, ArcDoc, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

# Migrating Data From ArcStorm or Map Librarian to the Geodatabase

## An ESRI Technical Paper

<b>Contents</b>	<b>Page</b>
Introduction .....	1
Overview .....	1
Suggested Resources .....	2
Comparison .....	2
Geodatabase Benefits Over ArcStorm and Map Librarian .....	3
Effects of Migrating .....	3
Database Design .....	3
Data Types .....	3
Database .....	4
Data Locking .....	4
Topology .....	4
Workflow .....	4
Migrating Features .....	7
Using ArcSDE for Coverages .....	7
Using ArcInfo Workstation .....	7
Using ArcSDE Commands .....	8
Using Exported Coverages .....	8
Migrating Annotation .....	9
Migrating ArcStorm History .....	10
Conclusion .....	12



# Migrating Data From ArcStorm or Map Librarian to the Geodatabase

**Introduction** Many user sites have a significant investment and history with using ESRI® ArcStorm™ or Map Librarian to store, manage, and maintain their spatial data. They have established workflow procedures and customized functionality using ARC Macro Language (AML™) and open development environment (ODE) to create an environment that is successful and stable. ArcInfo™ Workstation, part of the ArcGIS® family of products, contains a vast wealth of tools for sites to create, edit, convert, analyze, manage, and store spatial data. The ArcGIS Desktop software architecture offers substantial improvements for storage, manipulation, and presentation of that spatial data. One of the major improvements is the geodatabase, which manages spatial information (spatial data, business rules, relationships, topology, and behavior) in a relational database management system (DBMS). This paper presents options for migrating data from ArcStorm or Map Librarian databases to the geodatabase as well as provides an introduction to some of the topics that need to be addressed when migrating.

**Overview** Map Librarian was ESRI's first geospatial data management tool. It stored data in coverages by cutting the data into tiles. Working with the data involves locking a tile to ensure file integrity and making a temporary copy of the data to work on. ArcStorm extended this model by introducing individual feature locking and the historical tracking of individual features. It was still based on tiles using ArcInfo coverages, which had to be locked while storing or retrieving the data, even if only temporarily. ArcInfo users have a wide variety of uses for ArcStorm and map library databases, for example, forest stand management; state, county, and local government cadastral management; utility and asset inventory; and much more. The Land Records application found in ArcTools™ is a good example of what can be done with data stored in ArcStorm or map libraries. Parcel mapping and management are easily understandable examples of how ArcInfo can be used. Overlay processing, parcel splits, merges, queries (attribute and topological), selections, creation, modification, and deletion are just some of the types of operations performed on spatial data, of any type, in any database. The georelational data model supports all of these explicitly using the coverage as the backbone of the data store.

ArcGIS introduces a new object-oriented data model called the geodatabase data model. The geodatabase data model allows the logical representation to be closer to the physical world. Geodatabases can be created as personal (stored in Microsoft® Access) for single

user editing and display or using ArcSDE® (stored in IBM® DB2®, IBM/Informix®, Microsoft SQL Server™, or Oracle®) for multiuser editing and display.

Note: Data is stored in a personal geodatabase or stored in a DBMS using ArcSDE.

***Suggested Resources***

Technical papers available for download from the ESRI Support Center at support.esri.com include

- What is ArcGIS?*
- Migrating from ArcInfo Workstation*
- Working with the Geodatabase*
- Modeling and Using History in ArcGIS*

In addition, the ArcGIS product box ships with many books such as

- Modeling Our World*
- Building a Geodatabase*
- Geodatabase Workbook*
- Editing in ArcMap*

Some of these books can also be purchased from the ESRI Store at gisstore.esri.com or from major bookstores.

***Comparison***

The geodatabase has several important differences from other spatial data management tools. These differences are the result of improvements incorporated in the geodatabase model based on input from customers using the Map Librarian and ArcStorm products.

**Table 1  
Functional Comparison Between Spatial Data Storage Methods**

	<b>Map Librarian</b>	<b>ArcStorm</b>	<b>Geodatabase</b>
<b>Stored in a Multiuser DBMS</b>	No	Yes <sup>2</sup>	Yes <sup>1</sup>
<b>Multiuser Performance</b>	Fair	Good	Better <sup>1</sup>
<b>Multiuser Access</b>	Limited	Limited	Yes <sup>1</sup>
<b>Long Transactions</b>	Yes	Yes	Yes <sup>1</sup>
<b>History Management</b>	No	Yes	Yes <sup>3</sup>
<b>Simple Points, Lines, Polygons</b>	Yes	Yes	Yes
<b>Polygons/Regions With Two-Dimensional Topology</b>	Yes	Yes	Yes

**Table 1 (continued)**

	<b>Map Librarian</b>	<b>ArcStorm</b>	<b>Geodatabase</b>
<b>Tiled Vector Data</b>	Yes	Yes	No
<b>Raster Storage</b>	No	No	Yes <sup>1</sup>
<sup>1</sup> ArcSDE geodatabases <sup>2</sup> Attribute data only <sup>3</sup> Snapshot in time model based on geodatabase versioning			

***Geodatabase Benefits Over ArcStorm and Map Librarian***

- Communication and data transfer via TCP/IP (ArcSDE).
- Security is handled by the DBMS (ArcSDE).
- Data access methods remain constant in application.
- A uniform storage repository for all (vector, raster, and tabular) data.
- Intelligent validation of features with stored business rules.
- Continuous storage of data sets for large areas.
- Long transactions or versioning (ArcSDE).
- Large selection of planar topology rules for spatial features.
- Stored network (linear) relationships.
- Consistent object display behavior.

**Effects of Migrating**

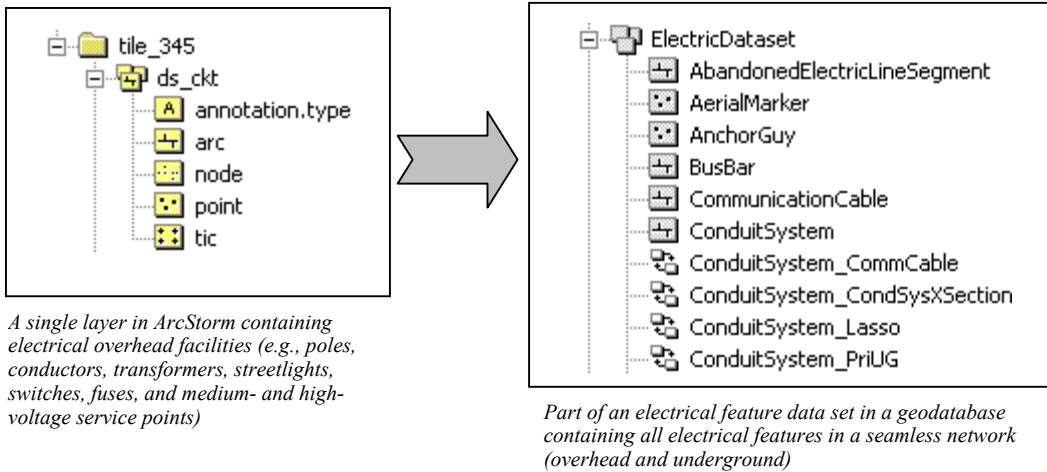
Migrating to a geodatabase for storage of spatial data can affect several aspects of an organization, from editing of features to determining how the new format fits into its business practices. The integration of the geodatabase into an organization may require the rewriting of applications or the updating of custom applications to access the data in the geodatabase format.

***Database Design***

Due to the different data models supported by ArcStorm/Map Librarian and the geodatabase, it is necessary to consider database design. Data models enable the logical representation of spatial features to more closely resemble the real world. Many sectors of the GIS user community have already met and worked out data models for their industry. The starting point for learning more is the ESRI Support Center ([support.esri.com/datamodels](http://support.esri.com/datamodels)).

**Data Types**

- ArcStorm and Map Librarian allow multiple entity types per coverage.
- Geodatabases allow one feature type per feature class. Feature data sets can contain multiple feature classes. Separate feature classes of the same type are allowed in the same feature data set.



The diagram above shows a coverage in an ArcStorm database representing multiple feature classes. The application logic must be able to sort out the different classes in order to represent them correctly and for any feature-specific editing rules required. The geodatabase model of the same data has the same features and classes, but they are separated into feature classes.

- Database**
- ArcStorm stores spatial data in coverage format and attribute information in INFO files; it also allows attribute information to be maintained in a DBMS separate from the spatial data.
  - Geodatabases can store many different data types (vector, raster, tables, and metadata) within the same DBMS.

**Data Locking**

**Topology** The georelational data model stores explicit topological relationships for features within the coverage. The object relational model of the geodatabase stores implicit relationships because the feature geometry is continually maintained as point, lines, and polygons. The topological relationships are stored as business rules (topology rules) within the geodatabase.

A geometric network is a special feature of the geodatabase that has explicit topology rules maintained by the object relational model.

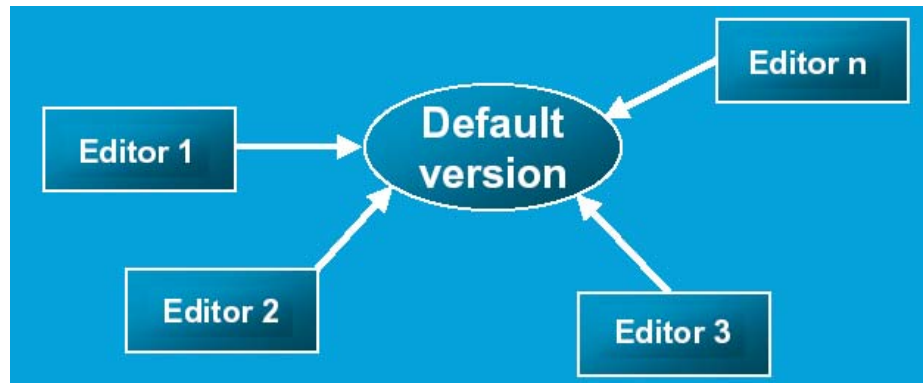
**Workflow** With ArcStorm and Map Librarian, data is checked out of the database or map library for editing. The checked out data is in ArcInfo Coverage format. Once data is checked out of the database, the underlying feature(s), or tile, is locked to prevent other users from editing the same data. This checked out data is the only copy of the changes being made. Once the edits have been made, they are posted back to the database either by the user making the edits or by a quality control manager.



When editing data in the geodatabase, the data remains in the database. The database provides a versioning model where separate versions of the same data may be created for editing purposes. In this scenario, it is possible for multiple editors to edit the same area at the same time. Any conflicts are resolved when the data is posted back to the default version. This is referred to as an optimistic locking model. Many organizations go through many iterations of planning before the real features are built. Versioning allows the planning process and the GIS system to be integrated and for all users involved in the construction and planning to see the changes prior to completion. Workflows supported by the geodatabase include the following:

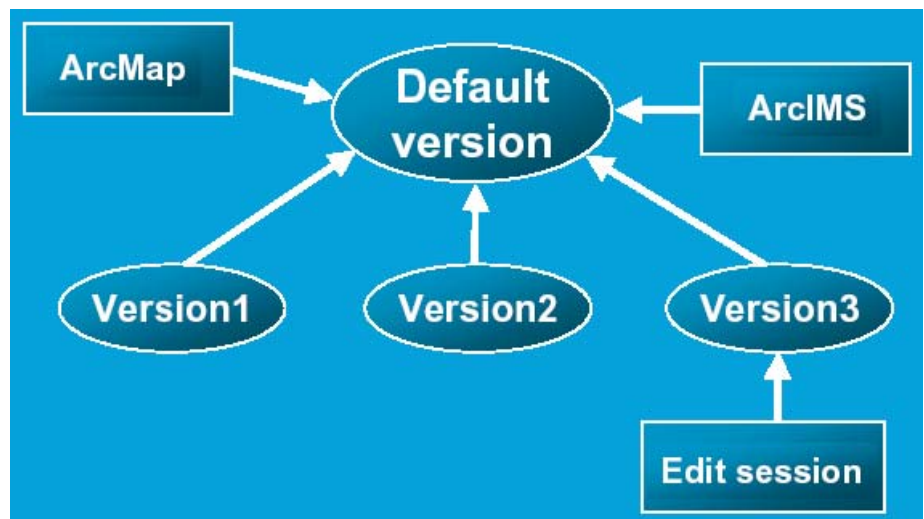
- Direct Editing of Default

All users can simultaneously edit the main database and save changes. The geodatabase ensures read consistency and concurrency. This workflow does not create child versions.



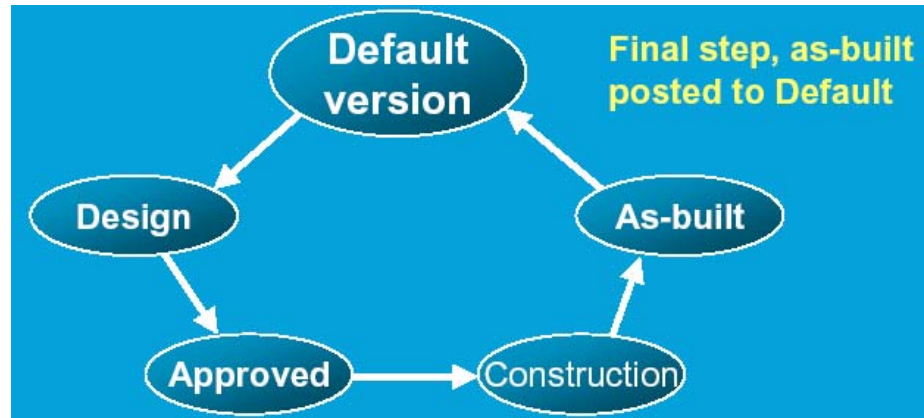
- Work Order Processing

Discrete work units are processed as work orders and posted to the database upon completion.



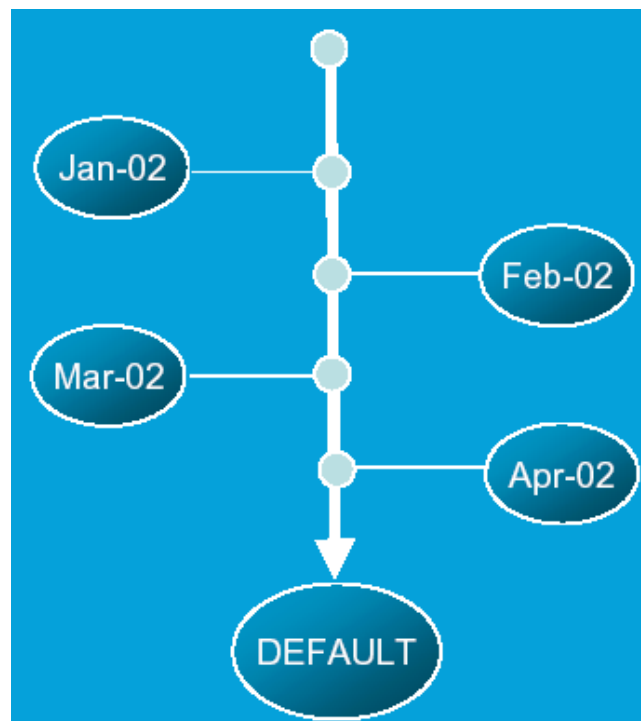
■ Cyclical Workflows

Projects evolve through prescribed stages or life cycles (e.g., design, approval, construction, as-built). New child versions are created for each stage of the project life cycle.



■ Workflow With History

Combines one of the previous workflow methods with periodic creation of versions that store a view of the database as it exists at the time the version is created. Historical versions remain part of the default lineage tree.



## Migrating Features

The following methods can be used to migrate data stored using ArcStorm or Map Librarian to an ArcSDE and/or personal geodatabase as noted. The method chosen will depend on the size of the ArcStorm/Map Librarian database and whether the migration is a one-time operation or will be executed frequently.

### *Using ArcSDE for Coverages Personal Geodatabase or ArcSDE Geodatabase*

Converting an ArcStorm or Map Librarian layer directly into a geodatabase can be done by using ArcSDE for Coverages, which is provided as part of ArcSDE and ArcIMS® (no additional cost or licensing is required). Once ArcSDE for Coverages has been installed and the layers registered to the service, they can be exported to a personal geodatabase or an ArcSDE geodatabase. ArcSDE for Coverages will show the latest iteration of the data, even as it is being edited.

1. Create a new geodatabase, or use an existing geodatabase.
2. In ArcCatalog™, connect to the ArcSDE for Coverages service (instance). Database Connections > Add SDE Connection.
3. Right click on the layer that you want to export and select Export > Geodatabase to Geodatabase.
4. In the Feature Class to Geodatabase menu
  - a. In Output Geodatabase enter the path, or browse to the desired geodatabase.
  - b. Use the default name for the new feature class, or enter a new one.
  - c. Use the Change Settings button to make any desired changes.
5. Click OK.

Note: ArcSDE for Coverages requires an integer item (column) that uniquely identifies the feature in the feature attribute table. This item is used as the spatial column for the layer and to stitch together features that are split by tile boundaries. With ArcStorm, this item is the OBJECT\_ID. Map Librarian layers do not, by default, have such an item. It is easy to load coverages into feature data sets using this method. Just create the feature data set and load the coverages into feature classes within the feature data set.

### *Using ArcInfo Workstation ArcSDE Geodatabase*

Use the ArcPlot™ LAYERIMPORT command to import the data from ArcStorm or the map library into ArcSDE. Once in ArcSDE the layer can be registered to a geodatabase from ArcCatalog. The advantage to this is that the unique item does not have to be an integer, and the layer does not have to be registered with ArcSDE for Coverages.

```
Arcplot: library states.Kentucky
Arcplot: Layer define st cover .counties polygon
Arcplot: layerquery st NAME = 'PIKE'
Arcplot: dataset connect Kentucky jorel esri_sde clark SDE
Arcplot: dataset createlayer Kentucky subexmp feature layer st polygon
        120000
Arcplot: layer define sub sde Kentucky subexmp feature polygon
Arcplot: layerimport sub layer st
Imported 1 features into sub from st
```

Note: Do not use to append to feature classes/tables that have behavior (relationship, behavior, network data, participate in a topology, etc.).

See the LAYERIMPORT command in ArcInfo Help (ArcDoc™) for complete syntax.

To create a feature data set, load the coverages with the same offset, scale, and spatial references. The resulting feature classes can be moved into a feature data set in ArcCatalog (drag/drop).

### ***Using ArcSDE Commands ArcSDE Geodatabase***

Use the ArcSDE command COV2SDE to import into ArcSDE for DBMS. Once in ArcSDE, the layer can be registered to a geodatabase in ArcCatalog. This method uses the tools provided with ArcSDE to do the conversion. For ArcStorm layers it will use the OBJECT\_ID item as a unique item for each feature. The unique item can be specified for Map Librarian data by using the -d (dissolve) option.

```
% cov2sde -o create -l counties,feature -f orldas.worldlib.country,polygon
-a all -s tiffany -i esri_av -u andy -p password -g 20,0,0
```

Note: Do not use to append to feature classes/tables that have behavior (relationship, behavior, network data, participate in a topology, etc.).

See the COV2SDE command in ArcSDE Developer Help for complete syntax.

To create a feature data set, load the coverages with the same offset, scale, and spatial references. The resulting feature classes can be moved into a feature data set in ArcCatalog (drag/drop).

### ***Using Exported Coverages Personal Geodatabase or ArcSDE Geodatabase***

Extract the data into a single coverage, then import using ArcGIS or ArcSDE tools. By doing this it is possible to do some quality control or preprocessing of the data prior to loading.

#### ■ Extracting the data layer from ArcStorm

```
Arcplot: library states.Kentucky
Arcplot: copyout layer ds_ckt arc ds_ckt
```

#### ■ Extracting the data layer from Map Librarian

```
Librarian: SETCOVER EXTAREA
Librarian: SETLAYERS NAMES POLYLAYER
Librarian: EXTRACT DISSOLVE my_workspace CLIP
Extracting coverage POLYLAYER from layer POLYLAYER
```

Note: With Map Librarian there is a 500-tile limit using the EXTRACT process that would require multiple extracts if the library tile structure has more than 500 tiles. This method is also resource intensive. An EXTRACT process takes 10 to 13 times the disk space of the resulting coverage to process. ArcStorm will take two to three times the disk space of the resulting coverage to process. If the data is sufficiently large, the build process may fail due to software limitations.

## Migrating Annotation

The following method can be used to migrate annotation features stored using ArcStorm or Map Librarian to an ArcSDE or personal geodatabase. ArcSDE for Coverages is used to serve the features to ArcGIS. The same technique can be used for any coverage that contains annotation.

1. Begin by determining the scale (reference scale) that will be used to capture the annotation. This is the scale that the text size specified is relative to. Usually this would be the scale most used to view or render the annotation.
  - Open ArcMap™.
  - Add the desired ArcSDE for Coverages layer.
  - Zoom to a scale that would be typically used when viewing the annotation.
  - Set desired symbology and size for the text.

Note: Pseudo items will override symbol properties set up in ArcMap (\$SIZE, \$OFFSETX, \$OFFSETY).

\$SIZE does not equal font point size; calculate to 0 before loading into the geodatabase to match ArcMap settings.

2. Create the empty feature class in ArcCatalog.
  - Open ArcCatalog.
  - Create the new annotation feature class in a geodatabase.
    - Set Reference Scale to the same as copied from ArcMap. (For example, do not use commas in scales such as 234,567,890.)
  - If the new feature class is not in a feature data set
    - In New Feature Class select Shape.
    - In Field Properties select the Spatial Reference '...' button.
    - Set or import the desired coordinate system.

If you do not do this, no annotation features will convert.

  - If the new feature class is in a feature data set, the coordinate system must be set.
  - Close ArcCatalog.

3. Add the Convert Coverage Annotation command to ArcMap.
  - Select Tools > Customize.
  - In the Commands tab select Label in Categories.
  - Drag Convert Coverage Annotation to desired toolbar location.
  
4. Convert the annotation.
  - Add the annotation layer(s) that you want to convert.
  - Change the properties of the annotation symbology as desired.
  - Click the Convert Coverage Annotation button.
  - Select the annotation layer(s) that you want to convert.
  - Browse to and select the annotation feature class that you created earlier.
  - Click Convert.

## Migrating ArcStorm History

ArcStorm maintains historical data if configured to do so. This data can be used for on-the-fly viewing of the data at a point in time. Migrating this data to individual feature classes or a combined feature class is possible, although the geodatabase does not automatically maintain history for layers. Versions can be created to show the state of the database at certain points in time. See the *Modeling and Using History in ArcGIS* white paper online for more details.

Note: Caution should be exercised when performing this process because it requires the use of ArcStorm system tables. Under no circumstances should it be attempted while the database is in use.

1. The ArcStorm "transactions.closed" table has a transaction number and completed date in an epoch date format.

ID	LIBRARY	USER	COMPLETED
1	red_lib	tom	865018712
2	red_lib	tom	865021695
12	red_lib	tom	867273047
13	red_lib	tom	869240664

Note: The epoch date is seconds elapsed from the epoch—4 p.m. December 31, 1969.

2. The ArcStorm "data\_sources.closed" table has the layers involved in the transaction.

ID	TYPE	NAME	NAME2
1	4	property	
1	6	property	red_property
1	2	property	
1	3	property	

Note: Where the TYPE field can be interpreted as follows:

Code	Feature Class
-1	SQL
0	INFO
1	Points
2	Arcs
3	Polygons
4	Nodes
5	Tics
6	Annotation
7	Sections
8	Routes
9	Links
10	COGO points
11	Regions

3. We can use [formatdate] to convert from epoch dates to usable dates.

```
Arc: &type [formatdate '865018712' '%x %H:%M:%S']
05/30/1997 11:58:32
```

4. Create a historical view using the date calculated for each transaction.

```
Arcplot: asconnect reddb
Arcplot: library reddb.red_lib
Arcplot: historicalview create view1 layer .property '05/30/1997
11:58:32'
Arcplot: historicalview list
```

Name	Time	Type	Valid?	Object
VIEW1	05/30/1997 11:58:32	Layer	-	REDDB.RED_LIB.PROPERTY

5. Select features with create\_\_id = transaction\_\_id (from step 1.)

```
Arcplot: reselect reddb.red_lib.view1 create__id = 1
```

6. Copy out features into a coverage.

```
Arcplot: copyout layer view1 polygon nowait
```

7. If one feature class is desired, append data into a single coverage, keeping sufficient information about the completed date from step 1.
8. Load the resulting data into ArcSDE.

## **Conclusion**

The ultimate decision should be made on a case-by-case basis. There are many benefits to migrating to the geodatabase that should be explored even if a complete conversion from ArcStorm or Map Librarian is not immediately possible. Some of these benefits include setting up relationships between feature classes, versioning, utilizing subtypes inside a feature class to represent individual feature types inside a single feature class, and storing a single seamless data store of your data accessible to clients all across your available network.