

Administering a Multi-Versioned ArcSDE Geodatabase

Thomas Brown



ESRI

Today's Agenda

- Workflow Management
- ArcSDE Meta Data Schema
- Multi-Versioned Object's Schema
- Database Design
- Editing a Versioned Geodatabase
- Reconciliation and Post
 - How it affects the workflow process
- Compressing the database



ESRI

Intended Audience

- GIS database administrators
- Application developers
- Hard core junkies who just need to know everything



ESRI

What this session is not:

- An Introduction to versioning
- A conceptual overview
- Full of flashy demos!
- Entertaining



ESRI

Managing Workflow

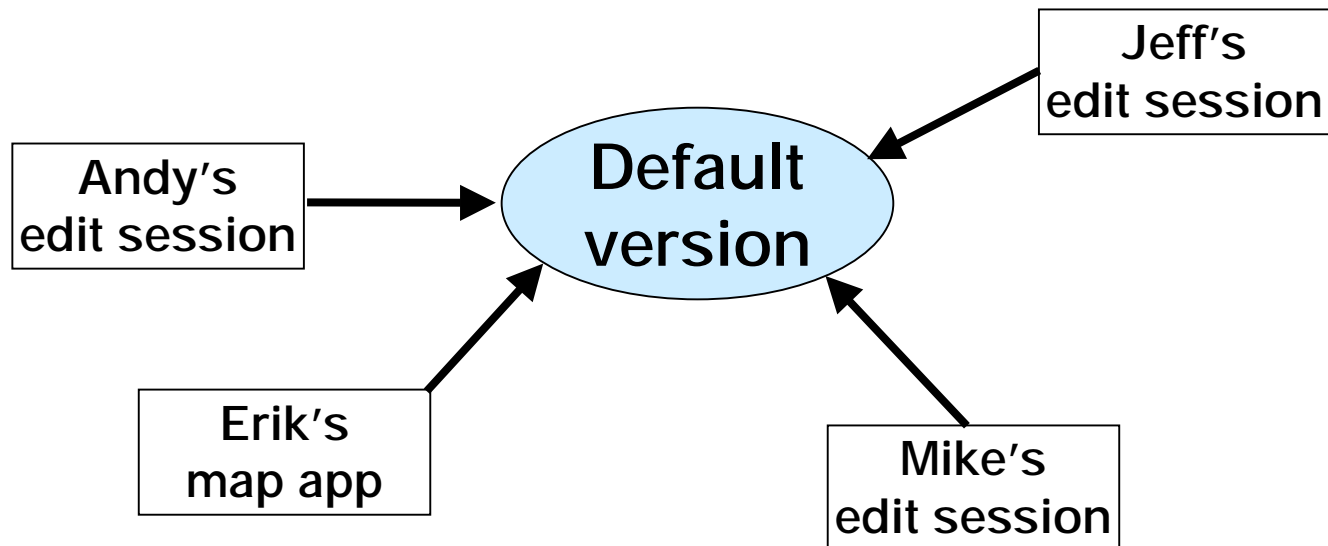
- Your organization's business process will impact the database administrator's responsibilities
 - version management
 - reconciliation/post
 - compress
 - and everyone's number one concern...
PERFORMANCE!



ESRI

Direct Editing of Default

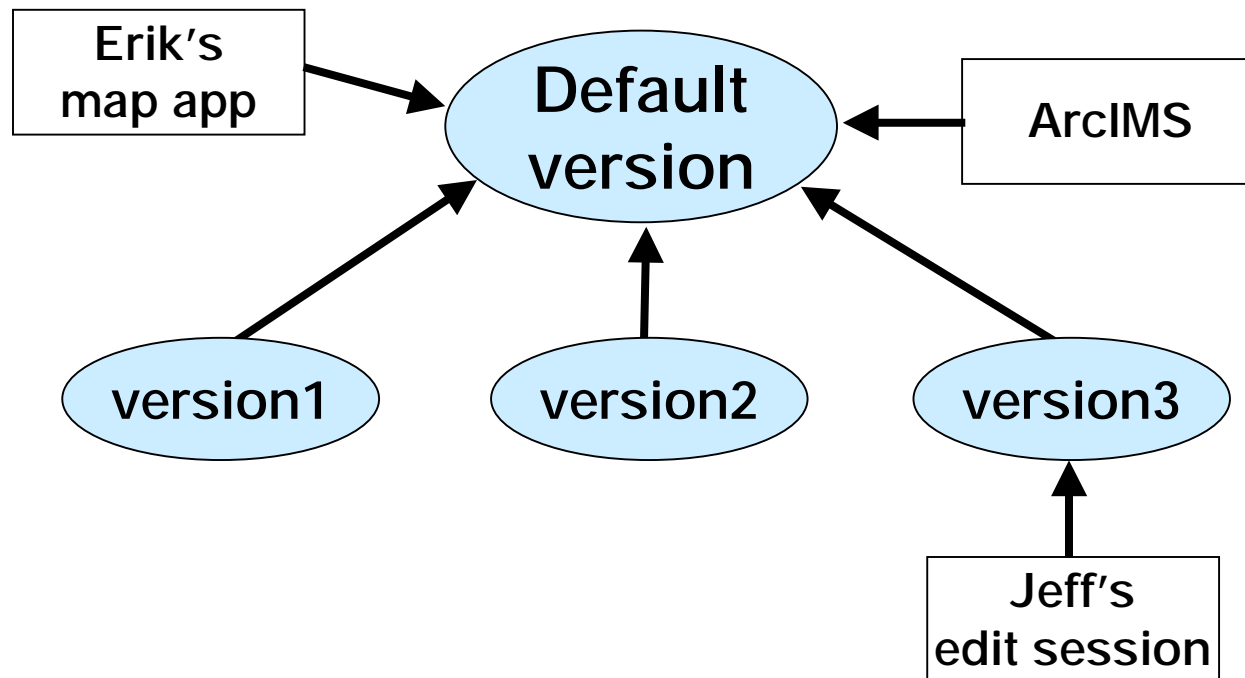
- All users can simultaneously edit the main database and save changes
 - Geodatabase insures read consistency and concurrency control



ESRI

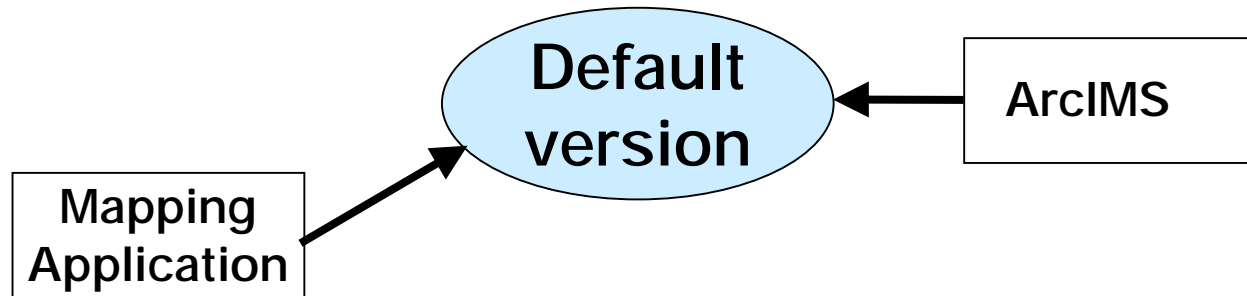
Work Order Processing

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



Cyclical Work Flows

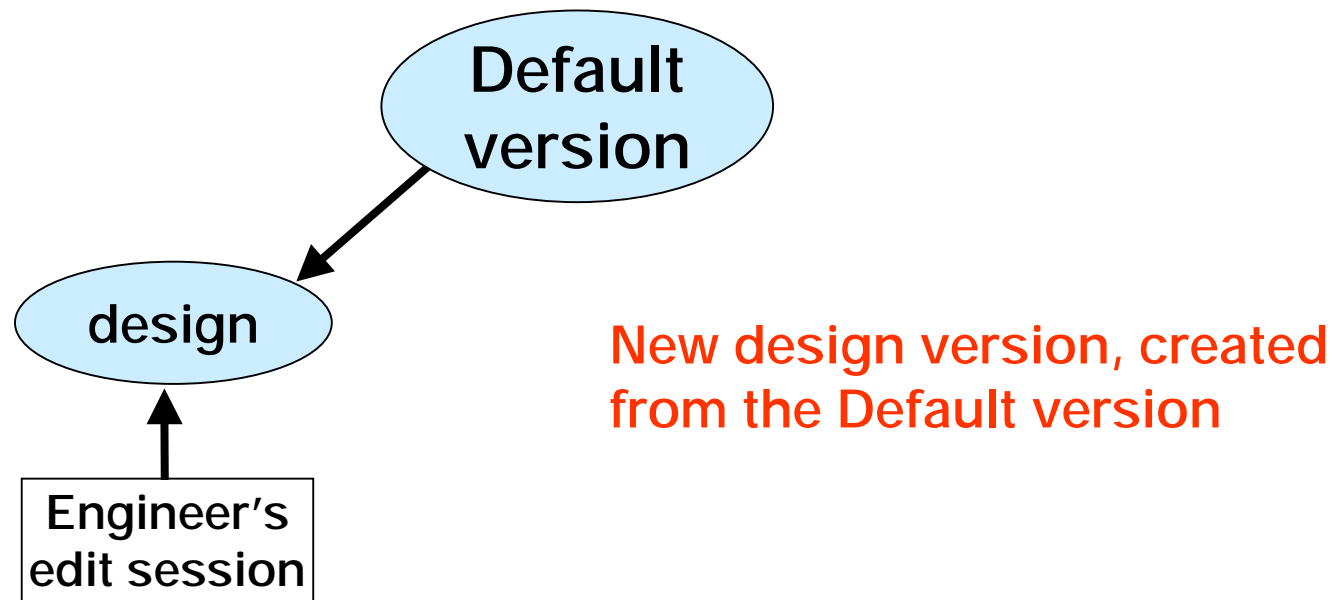
- Projects evolve through prescribed stages, or life cycle
 - design, approval, construction, as-built



ESRI

Cyclical Work Flows

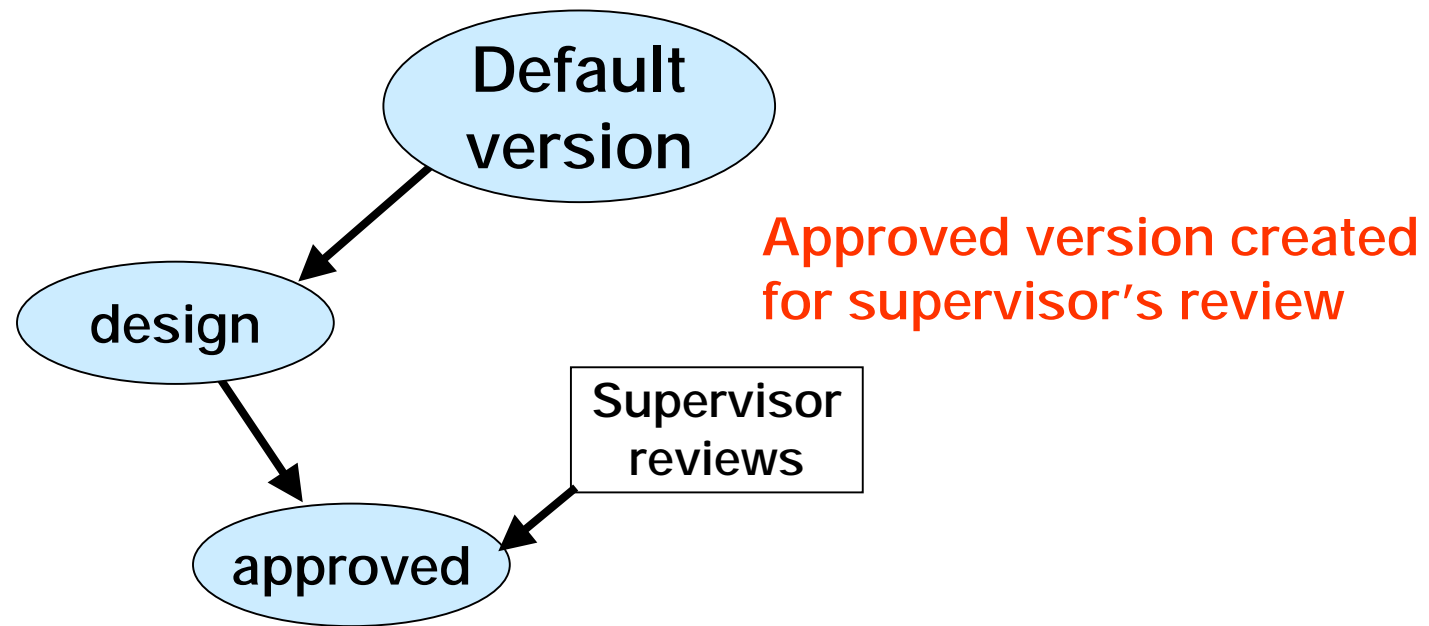
- Projects evolve through prescribed stages, or life cycles
 - design, approval, construction, as-built



ESRI

Cyclical Work Flows

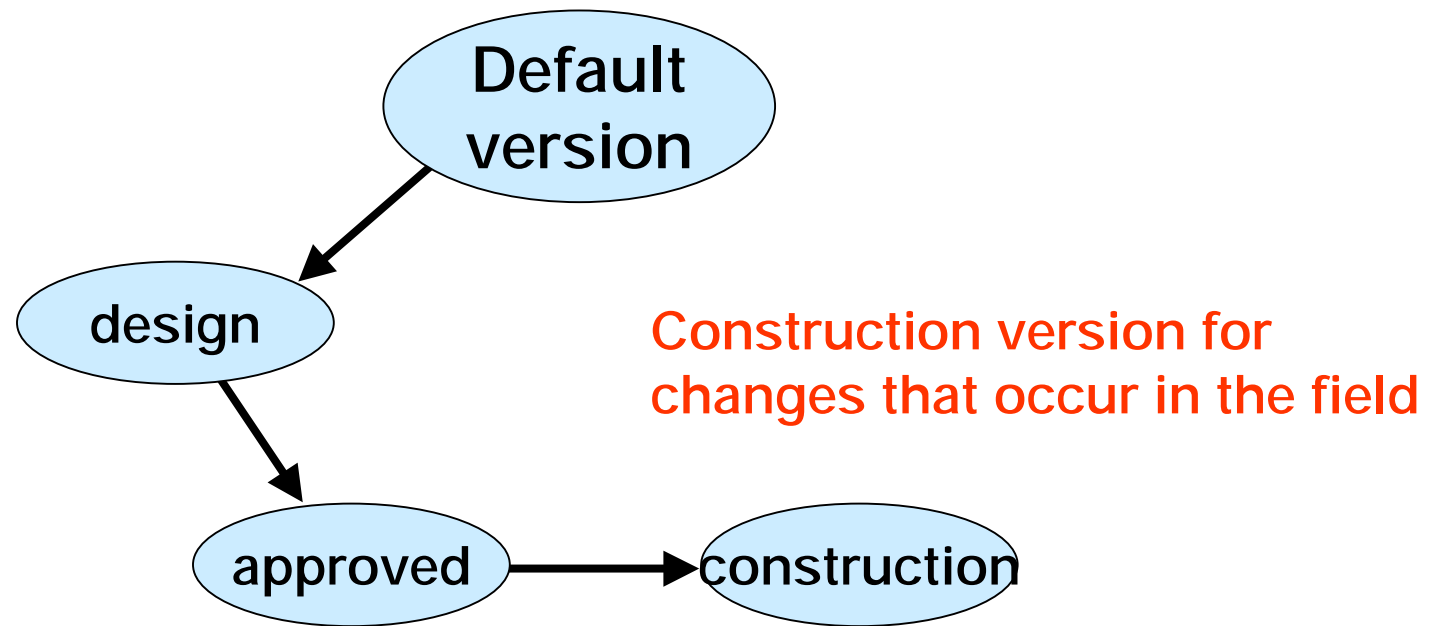
- Projects evolve through prescribed stages, or life cycles
 - design, approval, construction, as-built



ESRI

Cyclical Work Flows

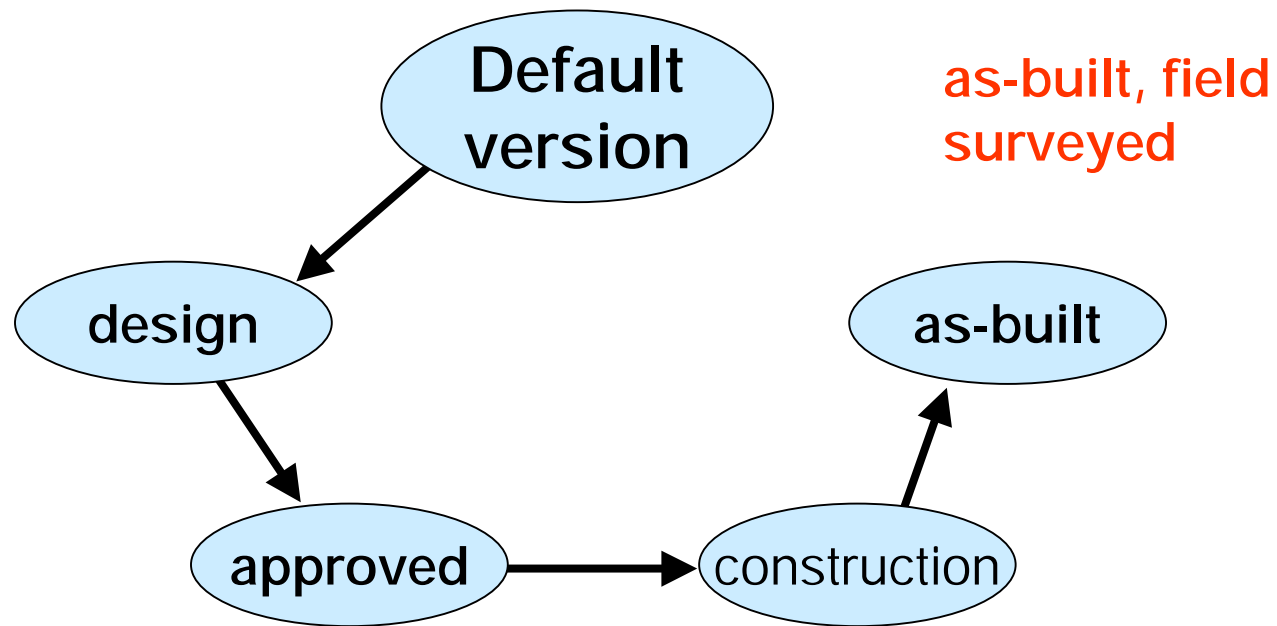
- Projects evolve through prescribed stages, or life cycles
 - design, approval, construction, as-built



ESRI

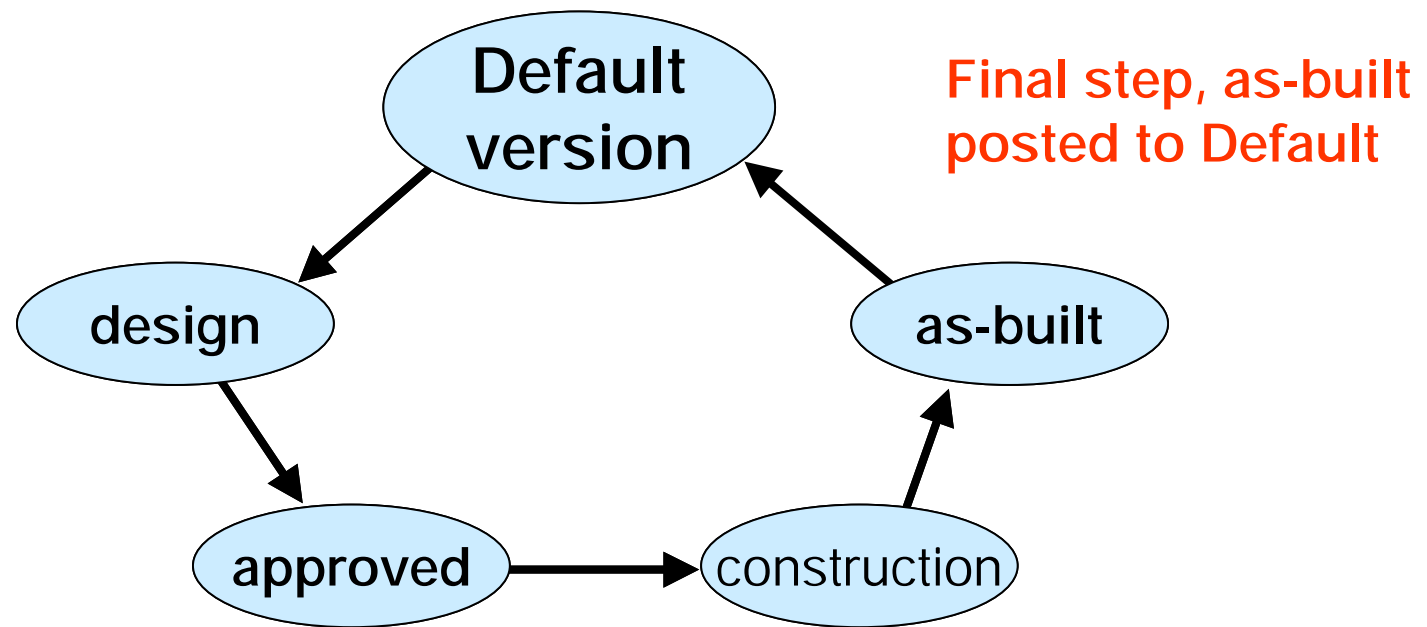
Cyclical Work Flows

- Projects evolve through prescribed stages, or life cycles
 - design, approval, construction, as-built



Cyclical Work Flows

- Projects evolve through prescribed stages, or life cycles
 - design, approval, construction, as-built



ESRI

ArcSDE Meta Data Schema

- Tables of interest
 - Versions
 - States
 - Layers
 - Table_registry
 - Mvtables_modified



ESRI

Some Definitions

- **Version**
 - A conceptual abstraction for a unit of work, such as work orders, design alternatives and the default database
 - the user defined “named” version references an internal database state
 - versions evolve over time through a succession of states



ESRI

Some Definitions

- State
 - Discrete snapshot of the database which has a constant schema and only differ by the set of rows for each table and the column values
 - states are organized in a tree structure



ESRI

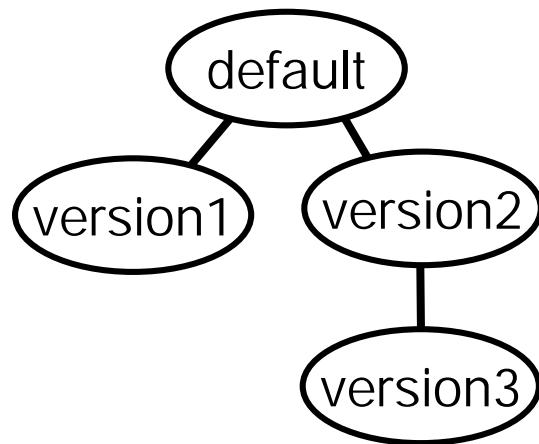
Versions Table

Name - case sensitive, 32 characters max length

Owner - user who created the version

Status - (Public, Protected, Private)

State id - current database state version references



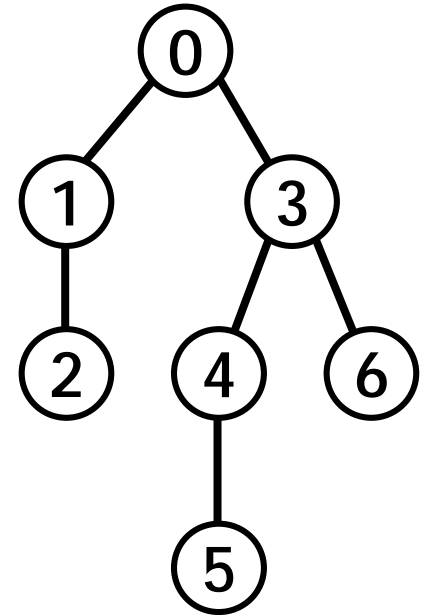
name	owner	status	state id
Default	SDE	protected	0
version1	gilligan	private	2
version2	skipper	private	6
version3	ginger	public	3



ESRI

States Table

id	owner	length	lineage
0	SDE	0	0
1	gilligan	1	(0)
2	gilligan	2	(0,1)
3	ginger	1	(0)
⋮			
6	skipper	2	(0,3)



Owner - user who created the state

Lineage length - number of preceding states

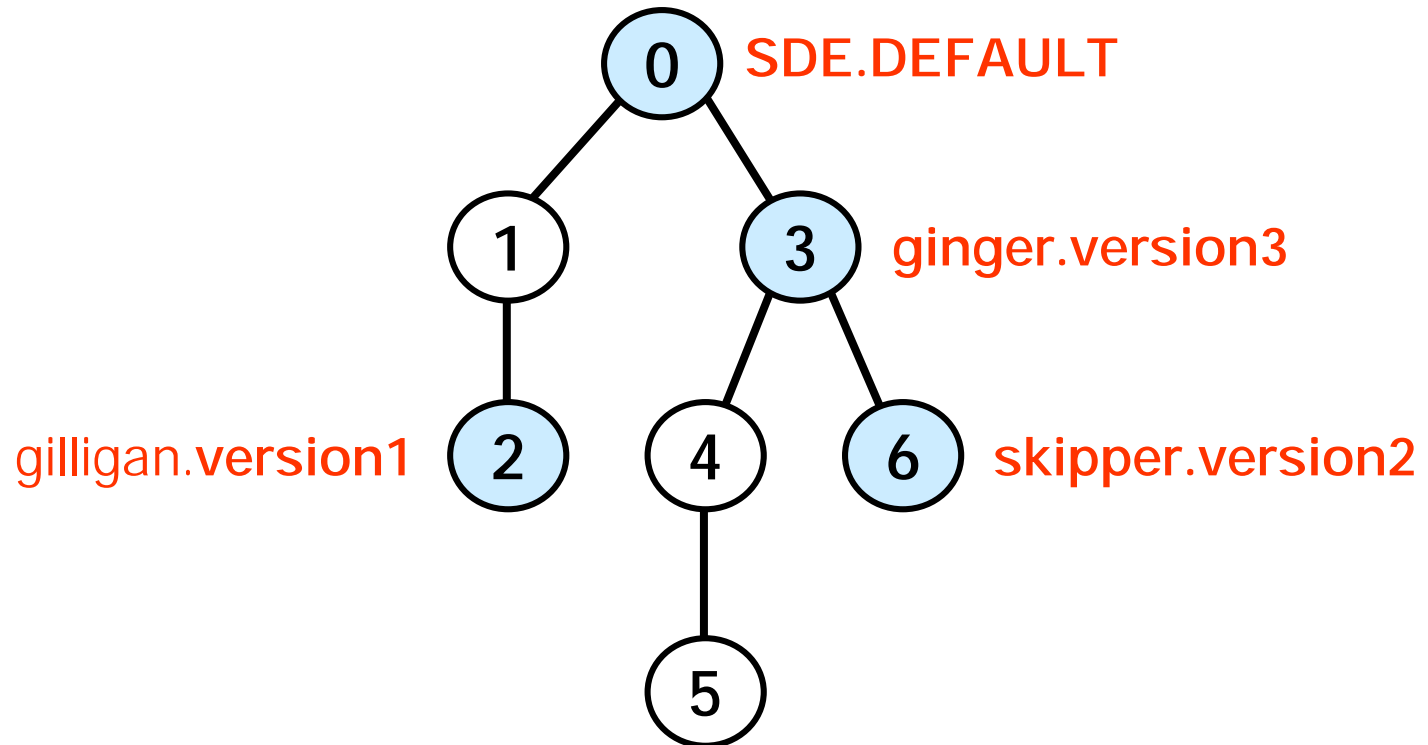
Lineage - binary storage of the lineage

(example: 0,3,4)



ESRI

Organization of Versions and Database States



Version and State Ids

- Pre 8.1 - generated and managed by the iomngr process (in memory)
- 8.1 - generated and managed by database sequences
 - sde.version_id_generator
 - sde.state_id_generator



ESRI

Joining the Versions and States Tables

```
create view version_info as select a.name
  "VERSION NAME", a.owner, b.state_id,
  b.parent_state_id, DECODE (a.status, 0,
  'PRIVATE', 1, 'PUBLIC', 2, 'PROTECTED')
  "ACCESS" from versions a, states b
  where a.state_id(+) = b.state_id;
```

VERSION NAME	OWNER	STATE ID	PARENT_STATE_ID	ACCESS
DEFAULT	SDE	0	0	PROTECTED
version1	GILLIGAN	2	0	PRIVATE
version3	GINGER	3	2	PUBLIC
		4	1	
		5	1	
version2	SKIPPER	6	5	PRIVATE



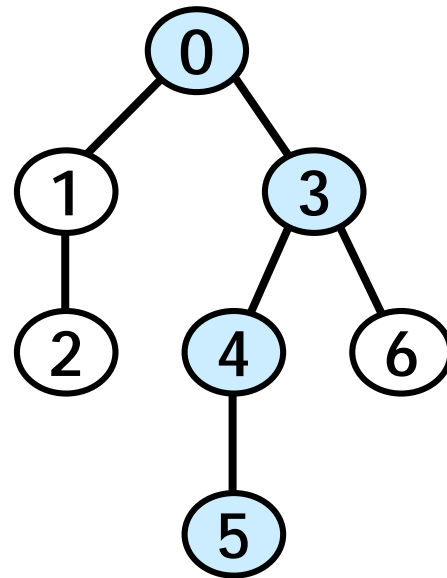
ESRI

Hierarchical query

```
select s.state_id from states s start  
with s.state_id = 5 connect by prior  
decode (s.state_id, 0, -1,  
s.parent_state_id) = s.state_id;
```

STATE_ID

5
4
3
0



ESRI

Table_registry and Layers Tables

- **Table_registry**
 - registration id uniquely identifies every table registered with ArcSDE
- **Layers**
 - layer id uniquely identifies every layer in the database

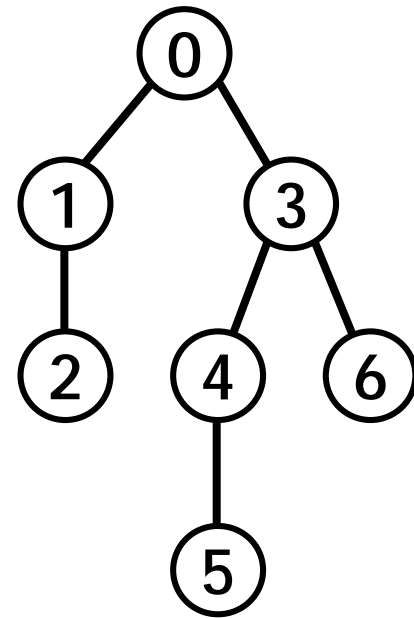


ESRI

Mvtables_modified

- Maintains the state id and the corresponding table registration id modified for that state

state id	table id
1	14
1	39
2	14
3	173
5	26
6	2



ESRI

Multi-versioned Object Schemas

- Business table (base table)
- Feature table
- Spatial table
 - Versioning (delta) tables
- Adds table
- Deletes table
 - and relevant table indexes



ESRI

Business Table (base table)

- User defined table for managing spatial attribute information

Columns	Index Name
OBJECTID (sde_row_id)	R<#>_SDE_ROWID_UK
SHAPE	A<#>_IX1
< user defined >	



ESRI

Feature table (F#)

- Manages the shape's geometry and related information such as area, length and type

Columns	Index Name
FID	F<#>_UK1
AREA	F<#>_AREA_IX2
LEN	F<#>_LEN_IX3
POINTS (BINARY)	



ESRI

Spatial Table (S#)

- An indexed table that stores references to the shapes based on a simple, regular grid

Columns	Index Name
SP_FID	S<#>_IX2
< other columns >	S<#>_IX1



ESRI

Id Generation

- ArcSDE 8.1 row ids and feature ids are generated by a <owner>.sequence
 - R<#> = row_id sequence
 - I<#> = shape sequence



ESRI

Adds Table, A<#>

- Maintains information for each inserted and updated row and corresponding state id

Columns	Index Name
OBJECTID	R<#>_ROWID_IX1
SHAPE	A<#>_IX1_A
< user defined >	
SDE_STATE_ID	A<#>_STATE_IX2



ESRI

Deletes Table, D<#>

- Maintains information for each deleted and updated row and corresponding state id

Columns	Index Name
SDE_STATE_ID	D<#>_IDX1 (col 1)
SDE_DELETES_ROW_ID	D<#>_IDX1 (col 2)
DELETED_AT	D<#>_IDX2



ESRI

Database Design

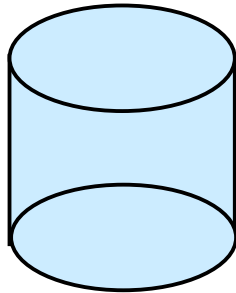
- Correctly loading your data is critical for database performance
 - avoid disk i/o contention
 - indexes on separate physical devices than the tables
 - accurate initial and next extent sizes
 - requires defining your keywords and parameters in the dbtune.sde file



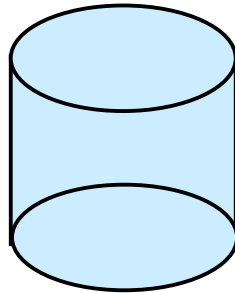
ESRI

Data Distribution

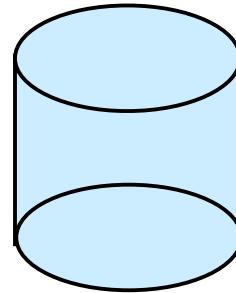
Business
Tables



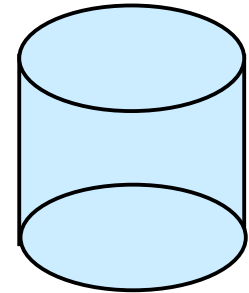
Feature
Tables



Spatial
Tables



Indexes



Delta Tables

Temporary Tables

Network Topology Tables



ESRI

Registering Objects as Versioned

- ArcSDE 8.0.2
 - set the dbtune.sde a_tblsp parameter
 - set the index_tablespace parameter



ESRI

Registering Objects as Versioned

- ArcSDE 8.1 (additional parameters)
 - a_tblsp parameter
 - a_index_1 and a_index_2 parameters
 - d_tblsp parameter
 - d_index_1 and d_index_2 parameters



ESRI

New Dbtune Table

- sdedbtune -o import/export utility to update the dbtune table

Columns
KEYWORD
PARAMETER_NAME
CONFIG_STRING

CONFIG_STRING is the objects storage clause



ESRI

Editing a Geodatabase

- Edit session is not visible to other users until the edit session is saved
- The version being edited continues to reference the initial state
 - provides read consistency



ESRI

Editing a Geodatabase

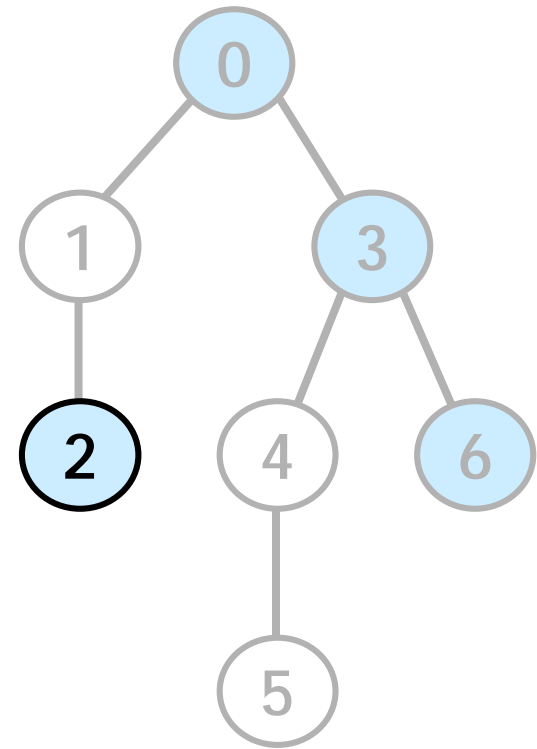
- States are created for each edit operation
 - provides undo/redo capability
- Saving commits the changes to the version
 - Clients refresh workspace to view the changes



ESRI

Edit Session

`gilligan.version1`

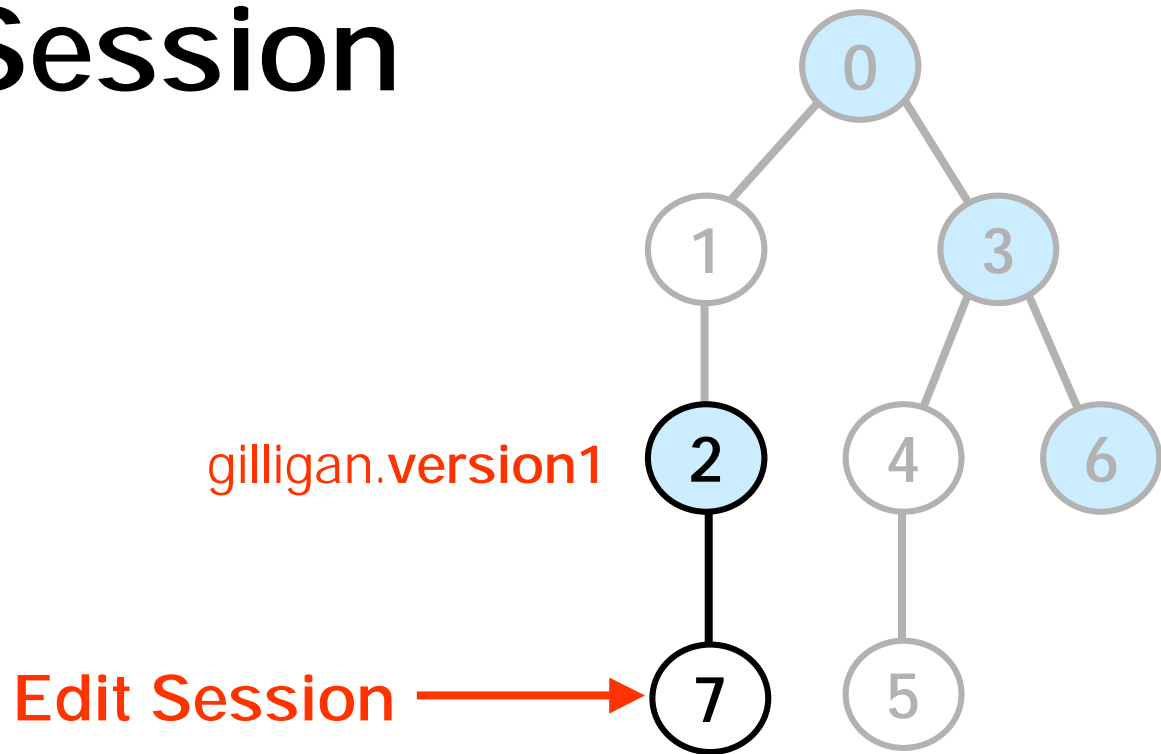


Gilligan starts editing version1



ESRI

Edit Session

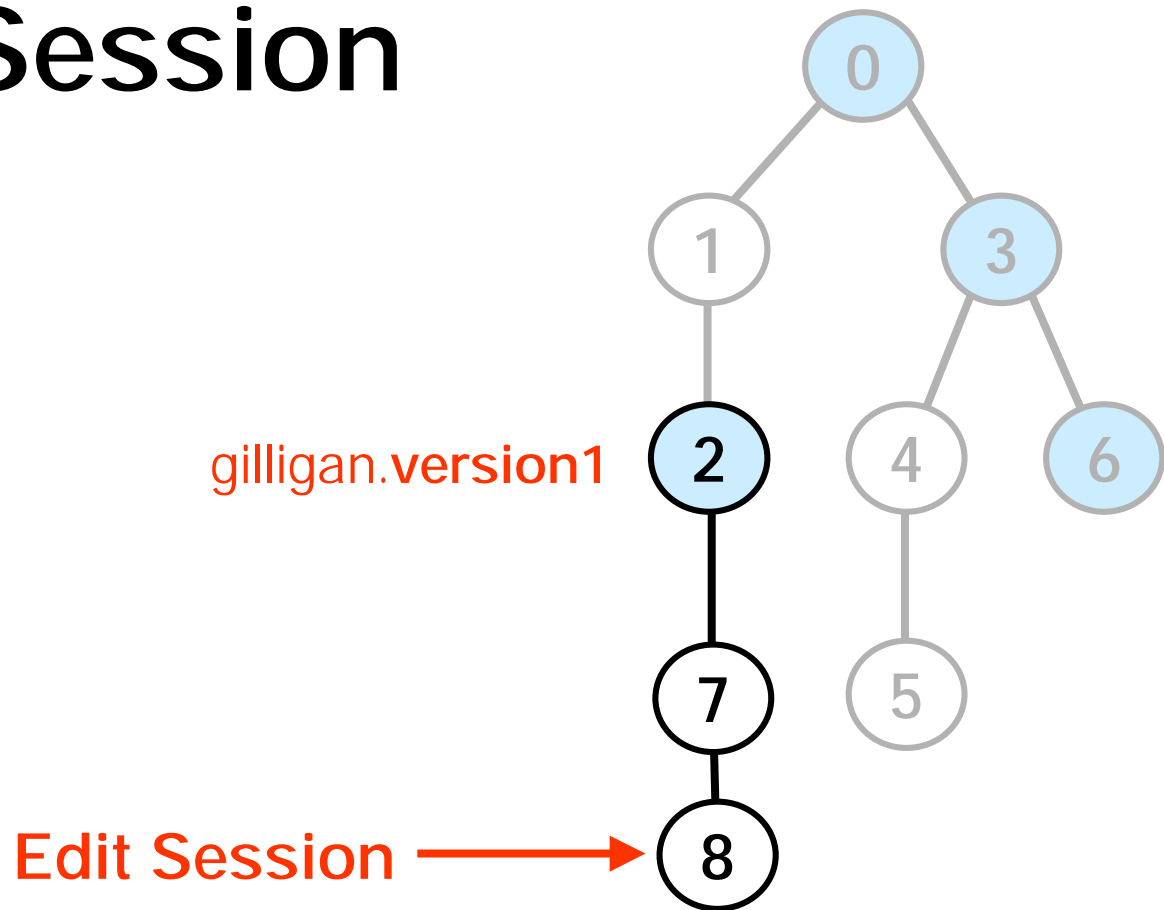


Gilligan inserts a new feature



ESRI

Edit Session

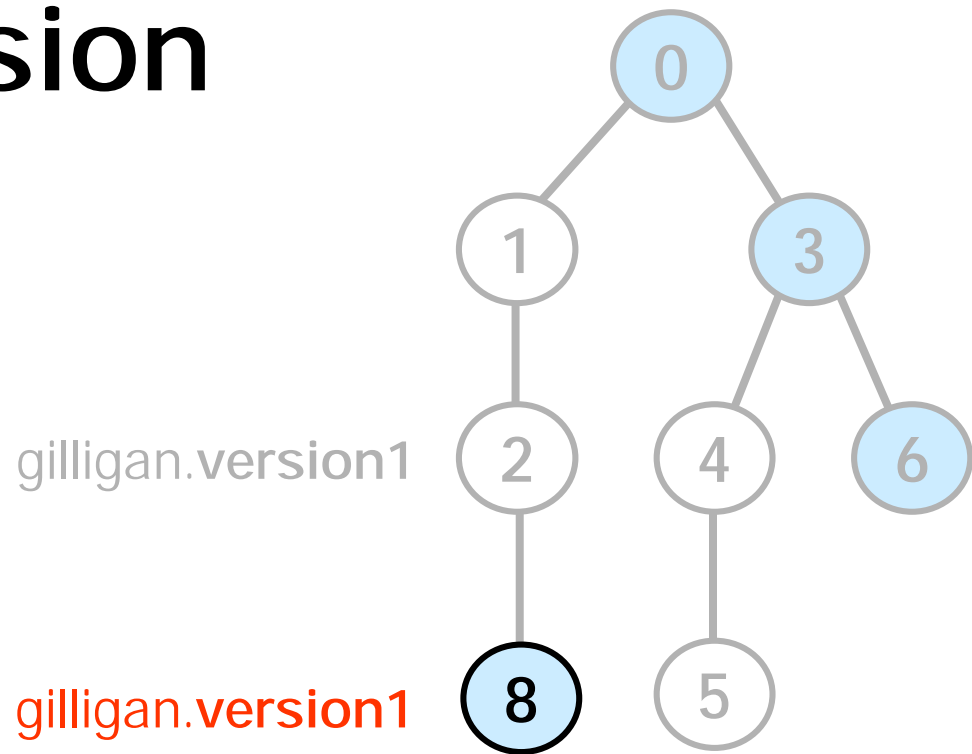


Gilligan updates a feature



ESRI

Edit Session



The skipper tells Gilligan to save



ESRI

ArcObjects

- Start and Stop Edit operations create database states

```
pWorkspace.StartEditOperation
```

```
Set pFeature = pFeatureClass.GetFeature(298)
```

```
fieldIndex = pFeatureClass.FindField("Owner")
```

```
pFeature.Value(fieldIndex) = "Donald Trump"
```

```
pFeature.Store
```

```
pWorkspace.StopEditOperation
```



ESRI

Inserting Features

- When new features or rows are inserted, a row is created in the A<#> table

object id	< user columns >	sde state id

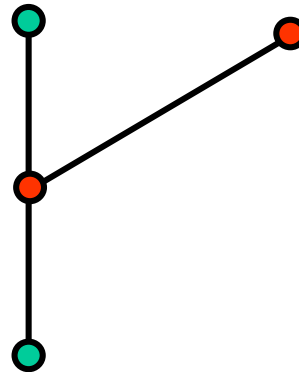


ESRI

Inserting Features

- When new features or rows are inserted, a row is created in the A<#> table

object id	< user columns >	sde state id
101	< null >	7

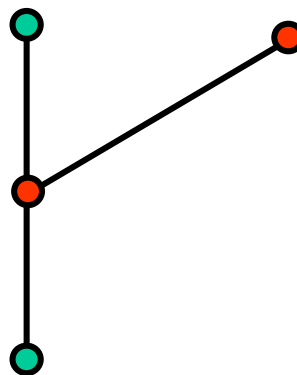


ESRI

Deleting Features

- When features or rows are deleted, a row is created in the D<#> table

sde state id	sde deletes row id	deleted at



ESRI

Deleting Features

- When features or rows are deleted, a row is created in the D<#> table

sde state id	sde deletes row id	deleted at
0	101	8



ESRI

Updating Features

- When features or rows are updated, a row is created in the A<#> and D<#> table

A<#>	object id	< user columns >	sde state id
D<#>	sde state id	sde deletes row id	deleted at



Updating Features

- When features or rows are updated, a row is created in the A<#> and D<#> table

A<#>	object id	< user columns >	sde state id
	278	< null >	9
D<#>	sde state id	sde deletes row id	deleted at
	0	278	9



Reconcile and Post

- How reconcile affects performance
 - Difference queries and conflict detection
- Post - why it's fast
- What is Autoreconcile?
- Object locks, what do those messages mean?



ESRI

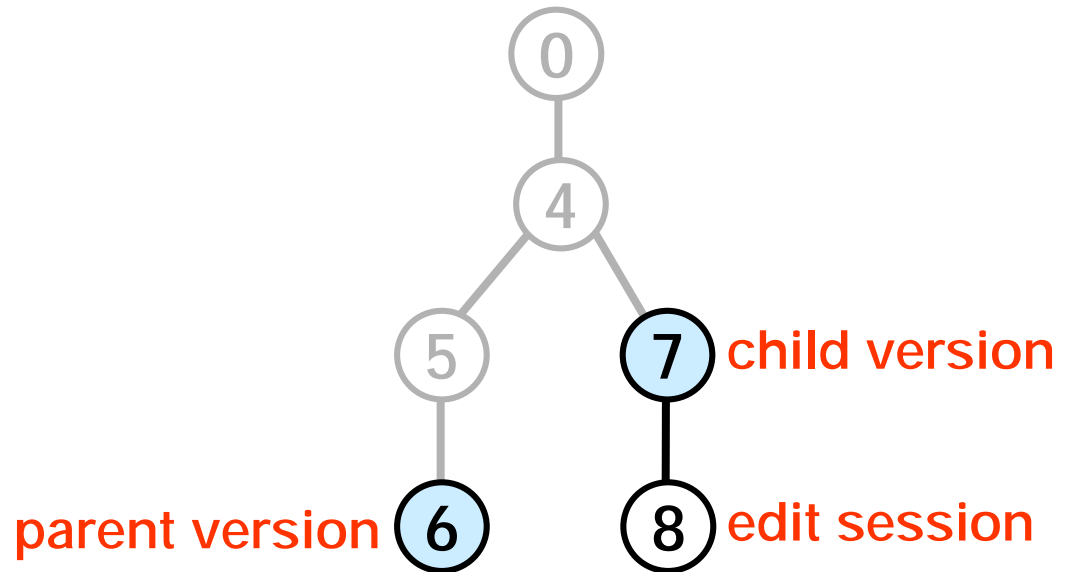
Reconcile

- Is the process of merging two versions and detecting conflicts
 - can occur when two users are editing the same version
 - or reconciling a child version and its parent version



ESRI

Example

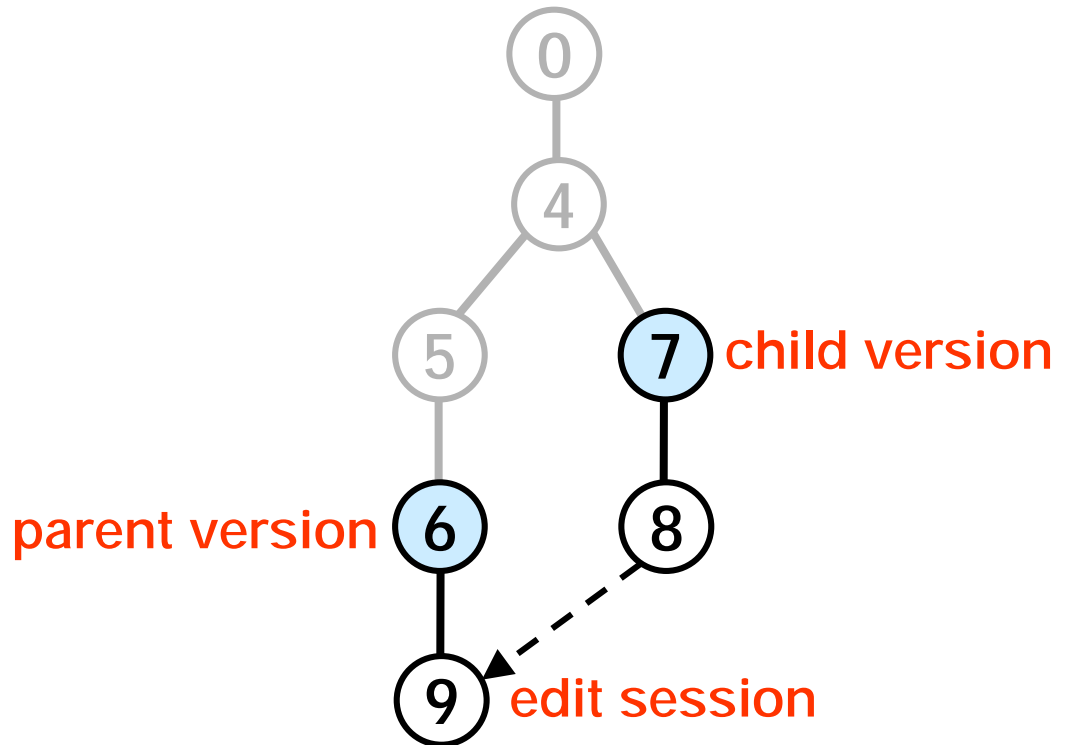


User starts editing the child version, makes one change then reconciles with the parent version



ESRI

Example

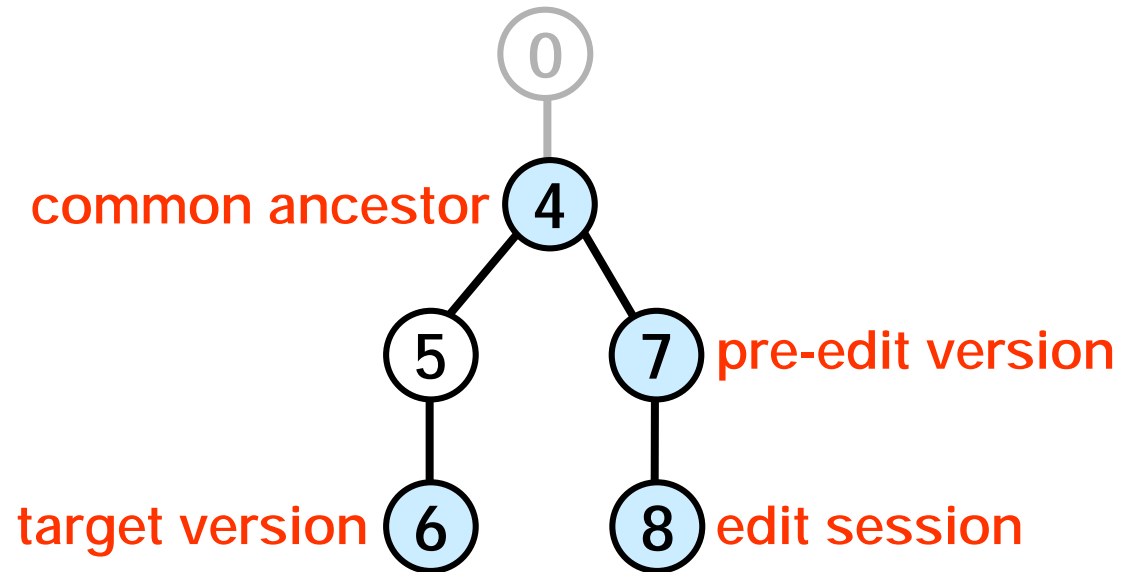


User starts editing the child version, makes one change then reconciles with the parent version



ESRI

Conflict Detection



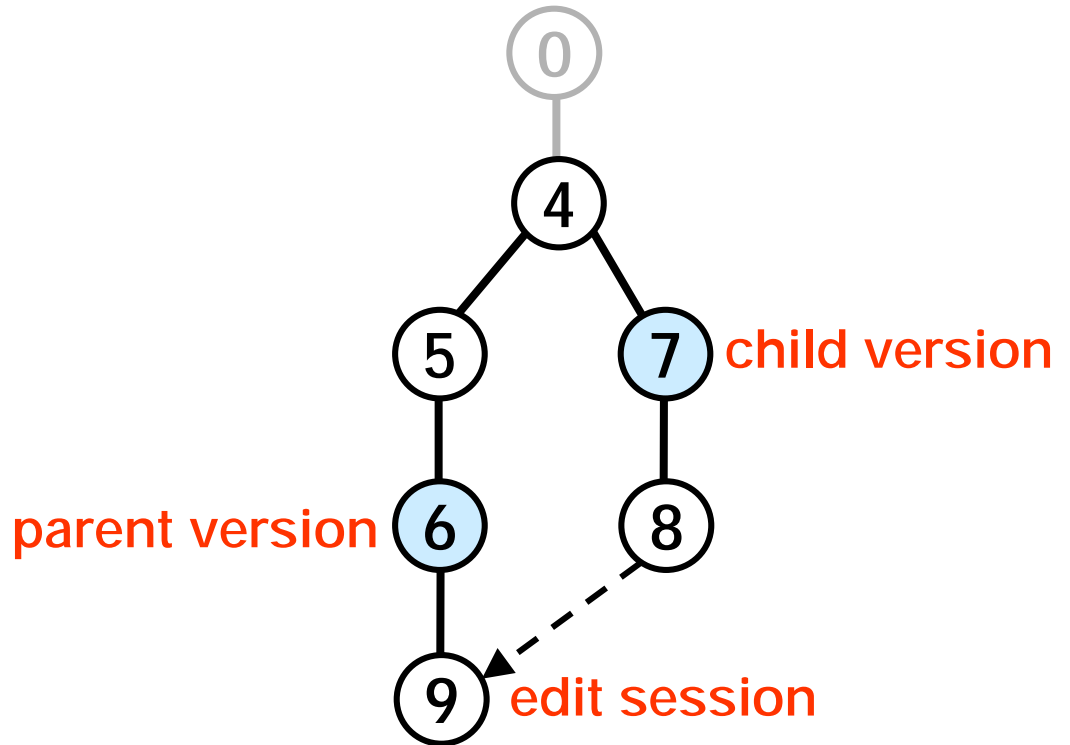
Each lineage is compared for all differences and feature conflicts:

- target version (4,5,6)
- edit session (4,7,8)



ESRI

Differences

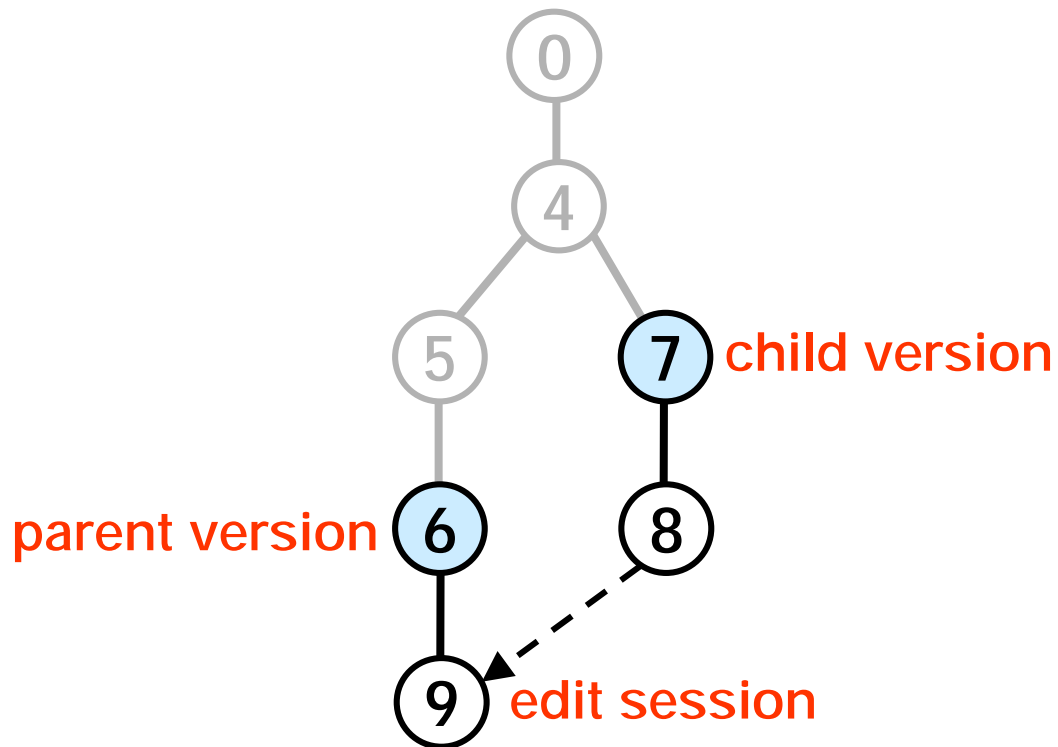


All features inserted, updated, deleted along the reconcile version lineage are applied to the reconcile state



ESRI

Post

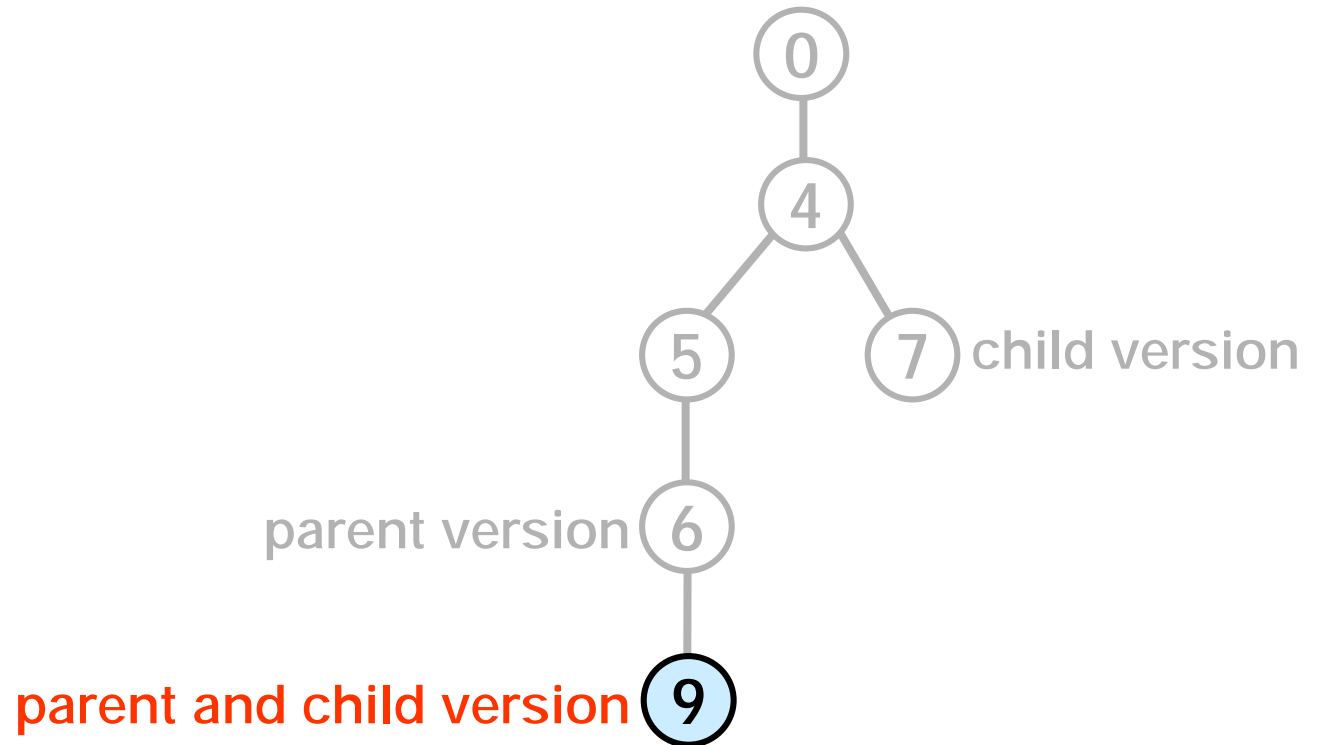


Automatically saves the edit session, and applies the reconciliation to the target version



ESRI

Post



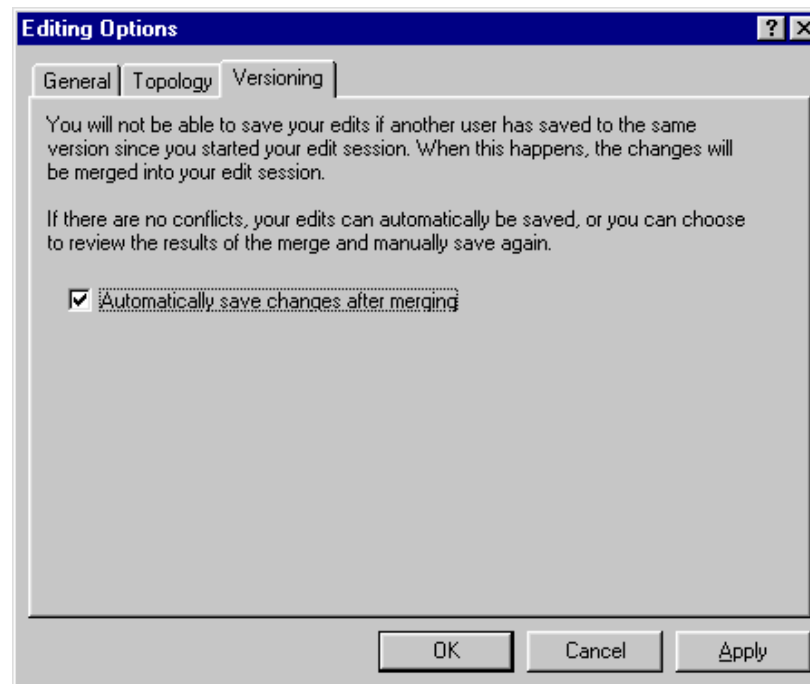
Automatically saves the edit session, and applies the reconciliation to the target version



ESRI

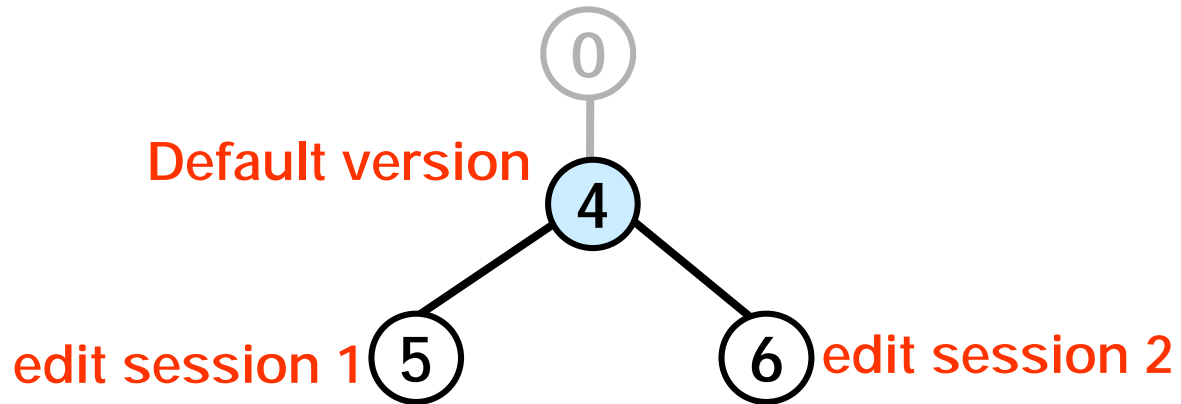
Auto Reconcile

- Editing option to automatically save changes (reconcile) if the version has been modified since you started editing



ESRI

Example

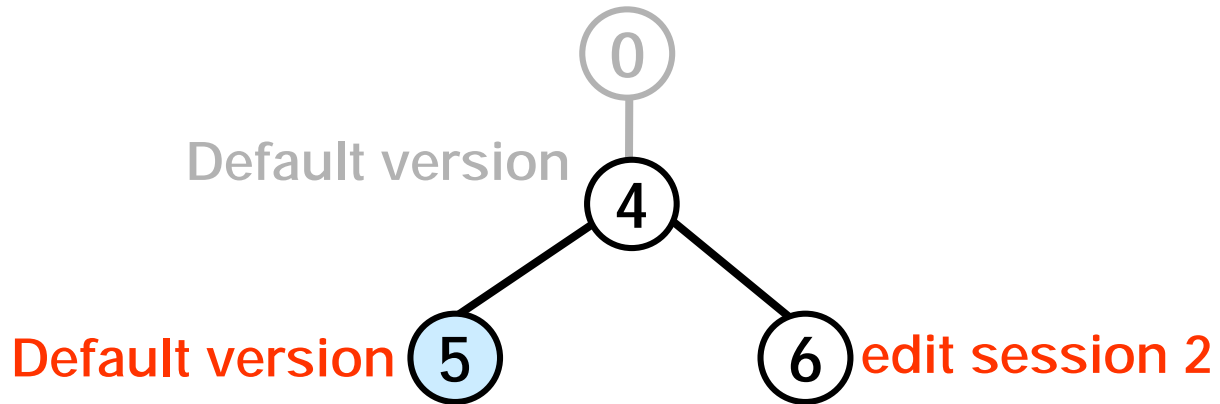


Edit session 1 stops editing and saves



ESRI

Example

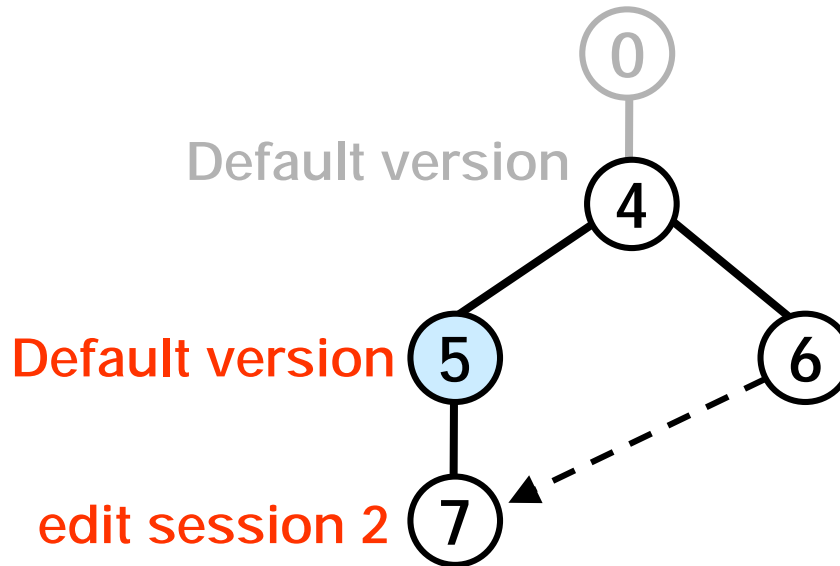


Edit session 1 stops editing and saves



ESRI

Example

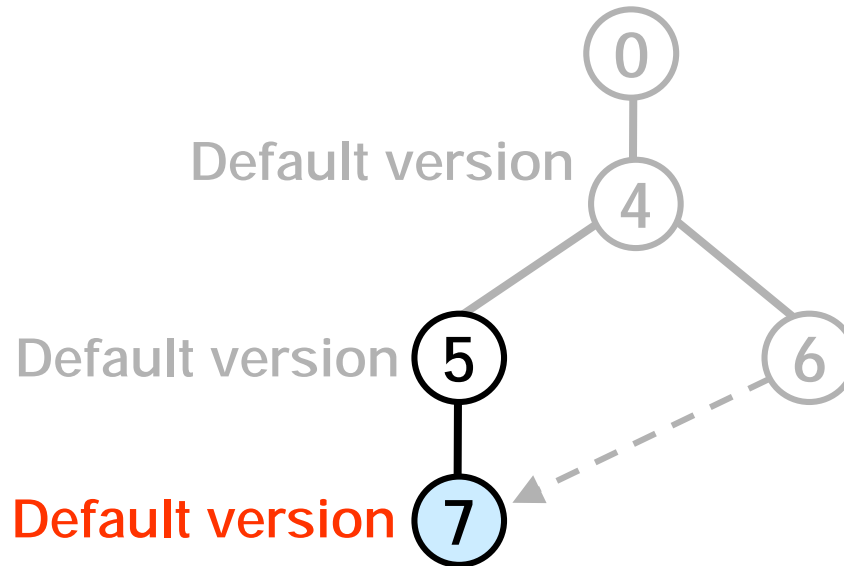


Edit session 2 saves, auto reconcile setting determines if the reconcile should be saved or not



ESRI

Example



When enabled version automatically saved



ESRI

Object locks

- Prevents versions from accidentally be reconciled by multiple users simultaneously
- Shared lock acquired on start editing, reconcile promotes shared lock to an exclusive lock
- Shared lock acquired on target version (prevents multiple reconciliation's)



ESRI

Warning Messages

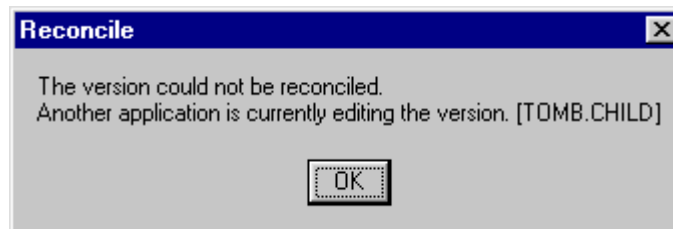
- Cannot start editing a version which is currently being reconciled



ESRI

Warning Messages

- Cannot reconcile a version when multiple users are editing the version

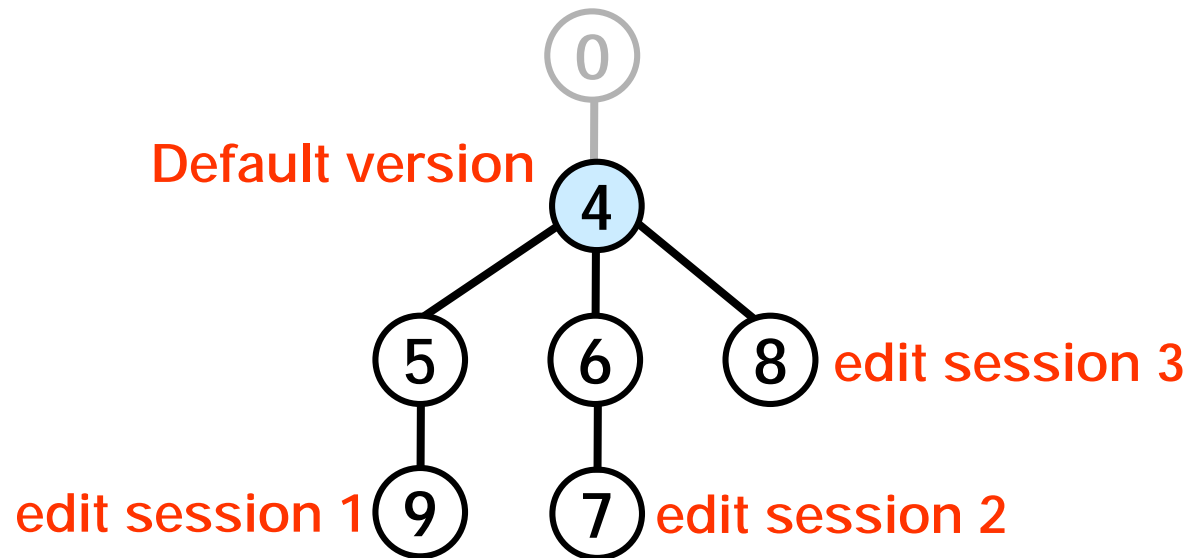


A Workflow Example

- Multiple users editing the Default version



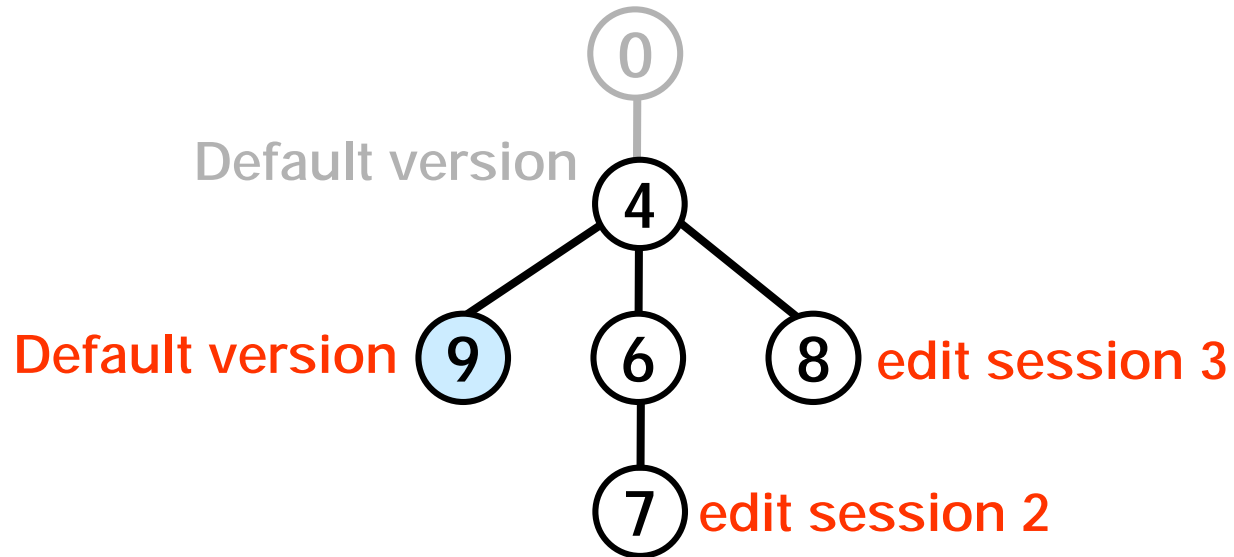
Multiple Users Editing



Three edit sessions start editing and make changes



Multiple Users Editing

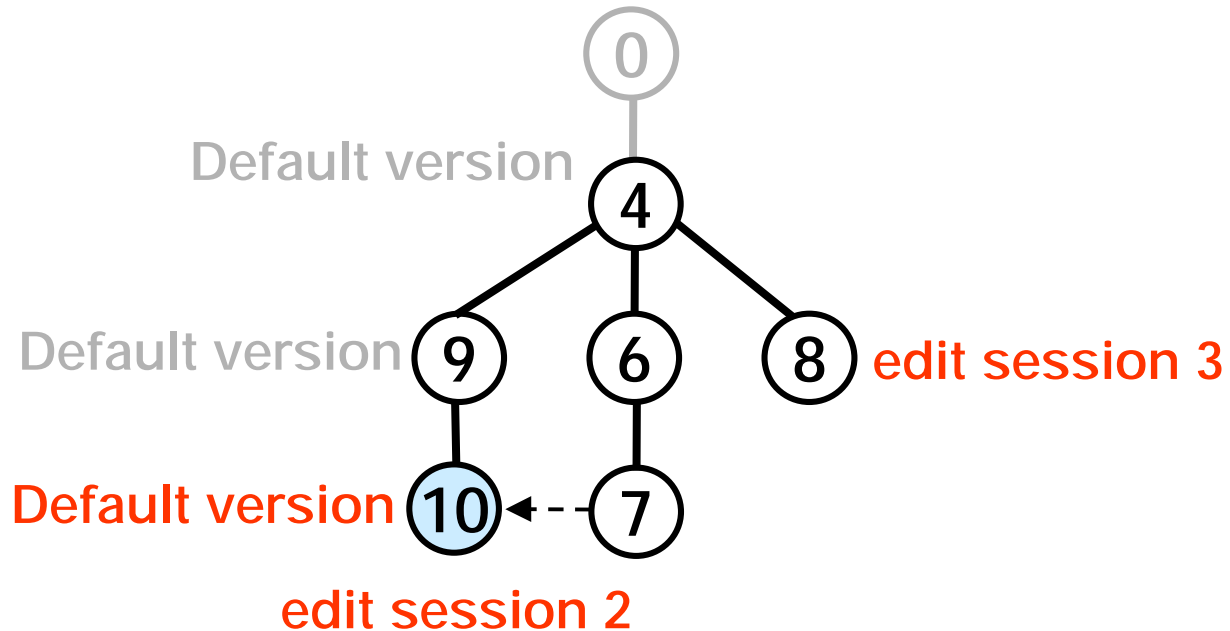


Edit session 1 stops editing and saves



ESRI

Multiple Users Editing

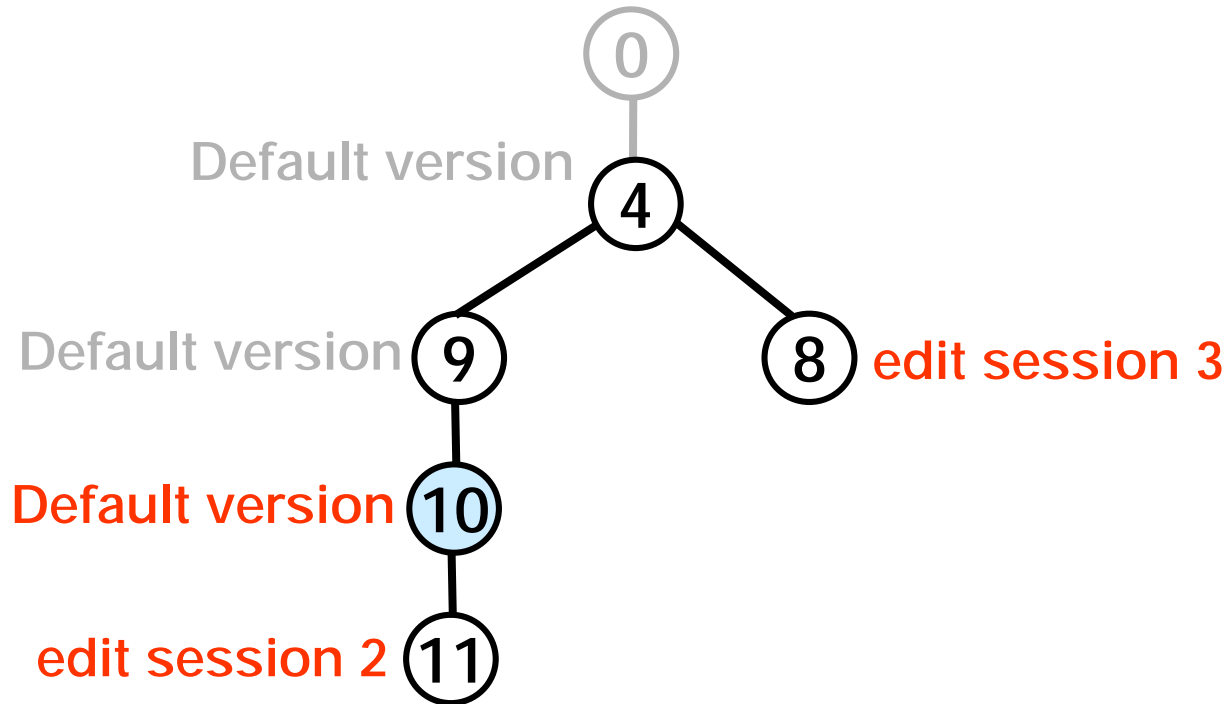


Edit session 2 saves



ESRI

Multiple Users Editing

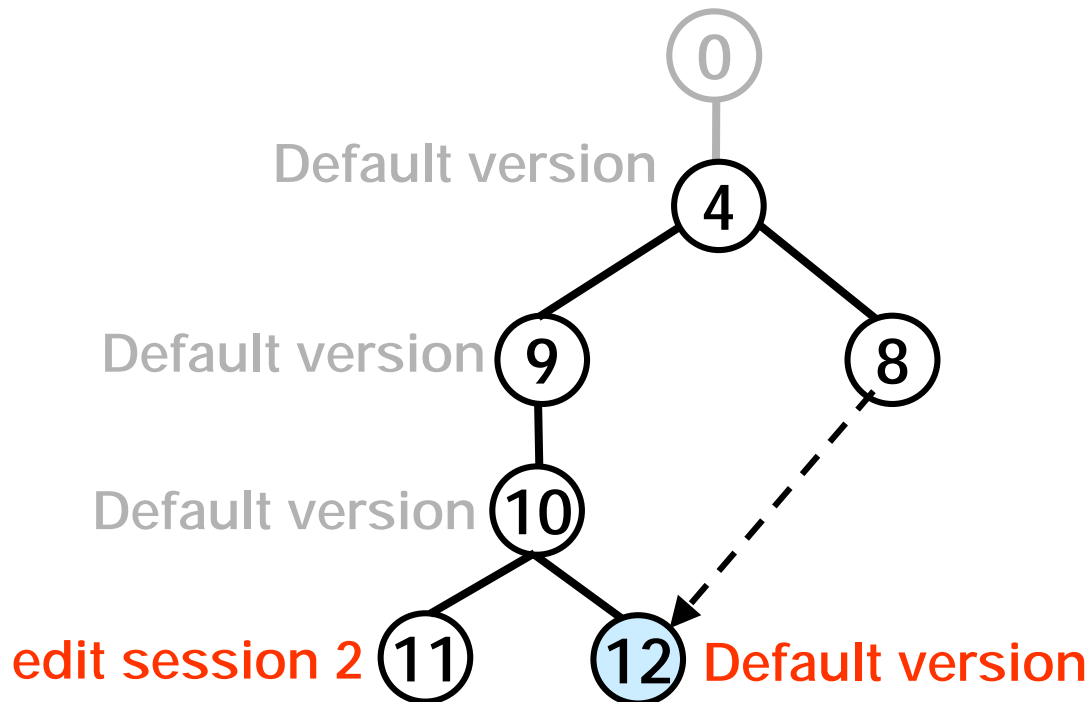


Edit session 2 continues editing



ESRI

Multiple Users Editing

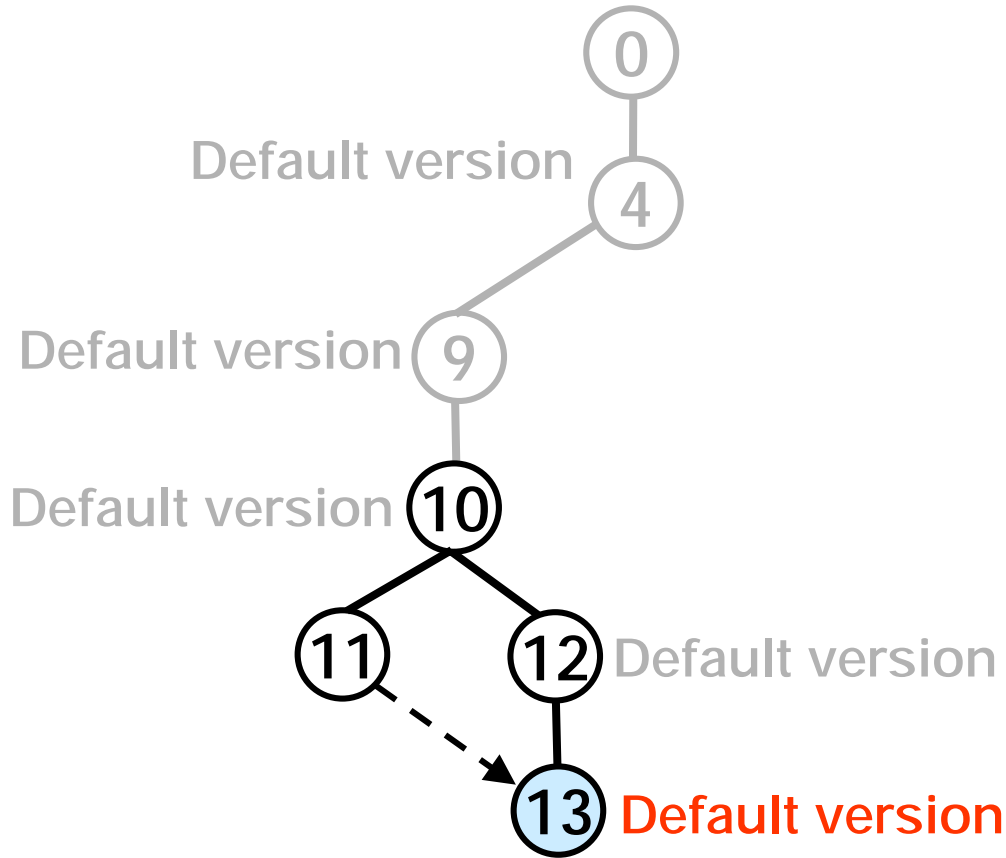


Edit session 3 stops editing and saves



ESRI

Multiple Users



Edit session 2 stops editing and saves



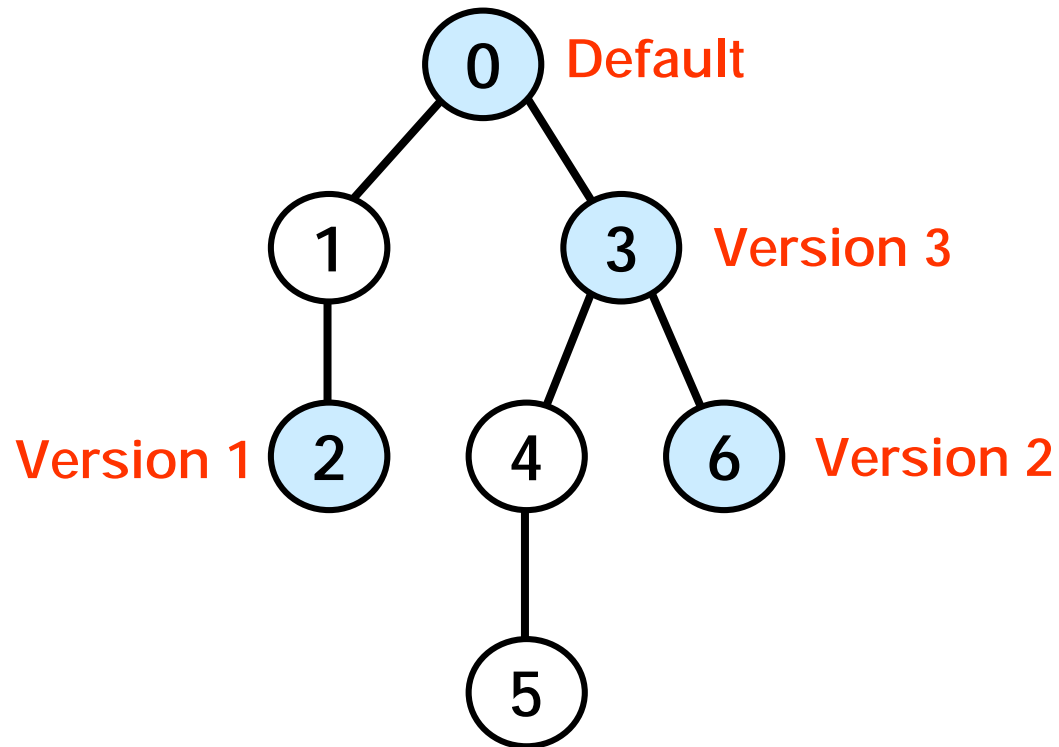
Compress

- What is the objective?
 - To improve **PERFORMANCE!**
- How?
 - By eliminating redundant rows and moving rows from the delta tables to the base table
 - minimizing the depth of the state tree



ESRI

Simple Example

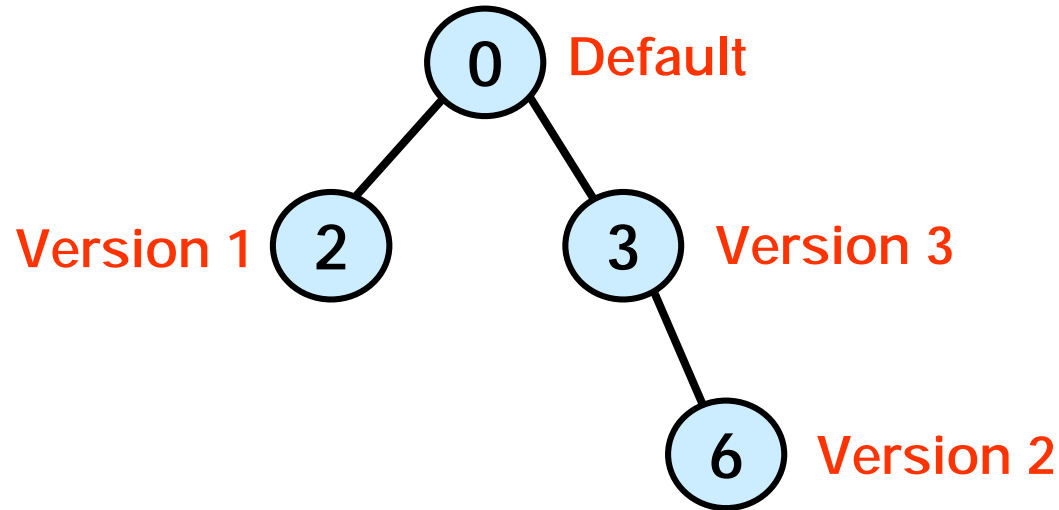


Prior to compress



ESRI

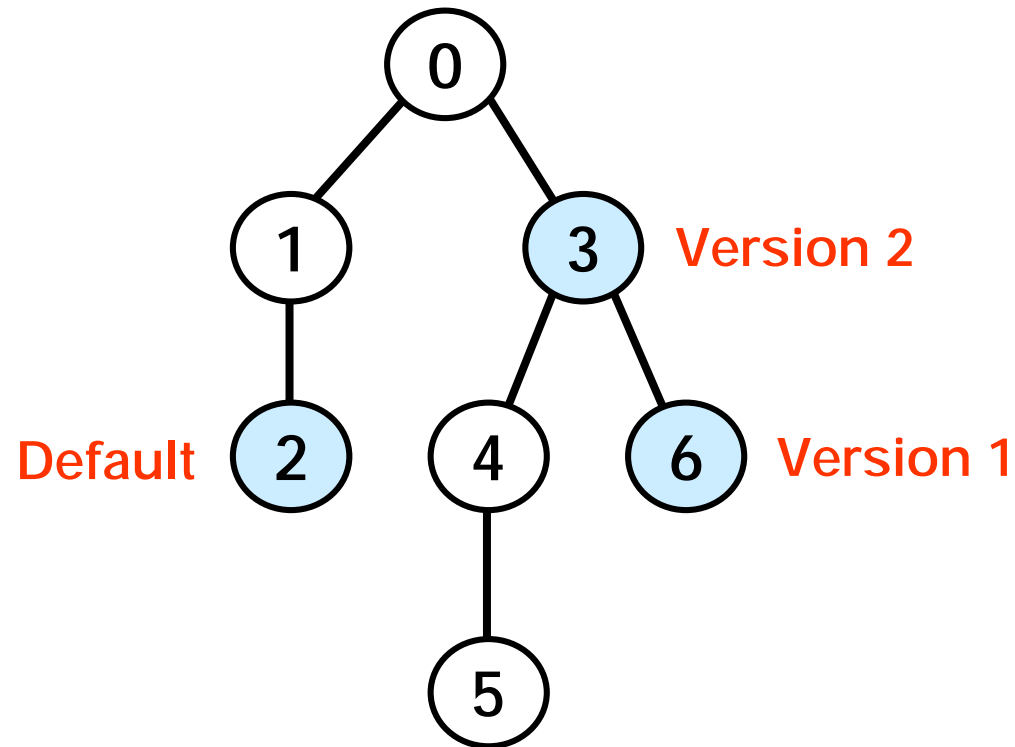
Simple Example



After compress - minimized the depth of the state tree



Example 2

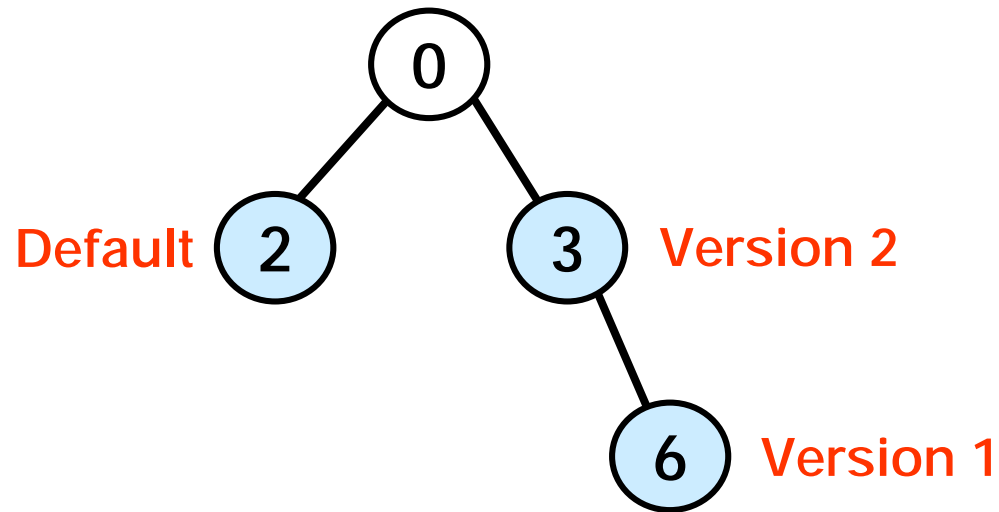


Prior to compress



ESRI

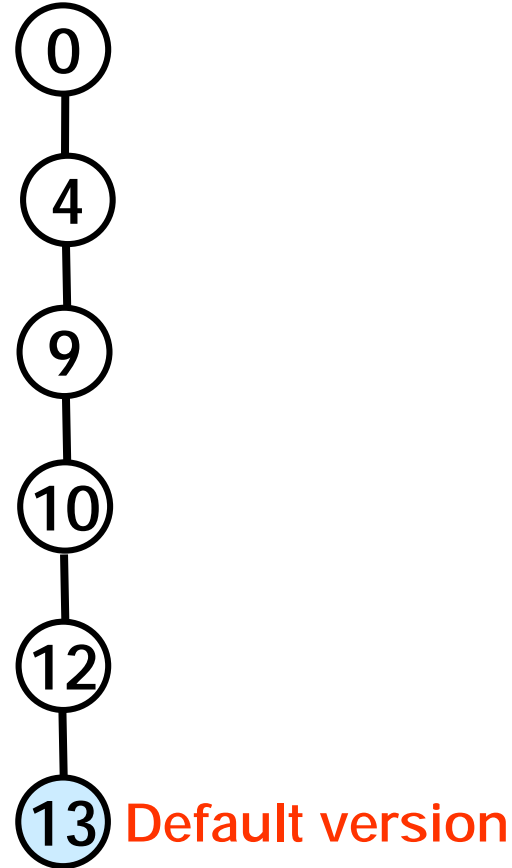
Example 2



After compress - minimized the depth of the state tree



Successful Example



Prior to compress



ESRI

Successful Example

① Default version

After compress - all rows in the delta tables moved back to the base table



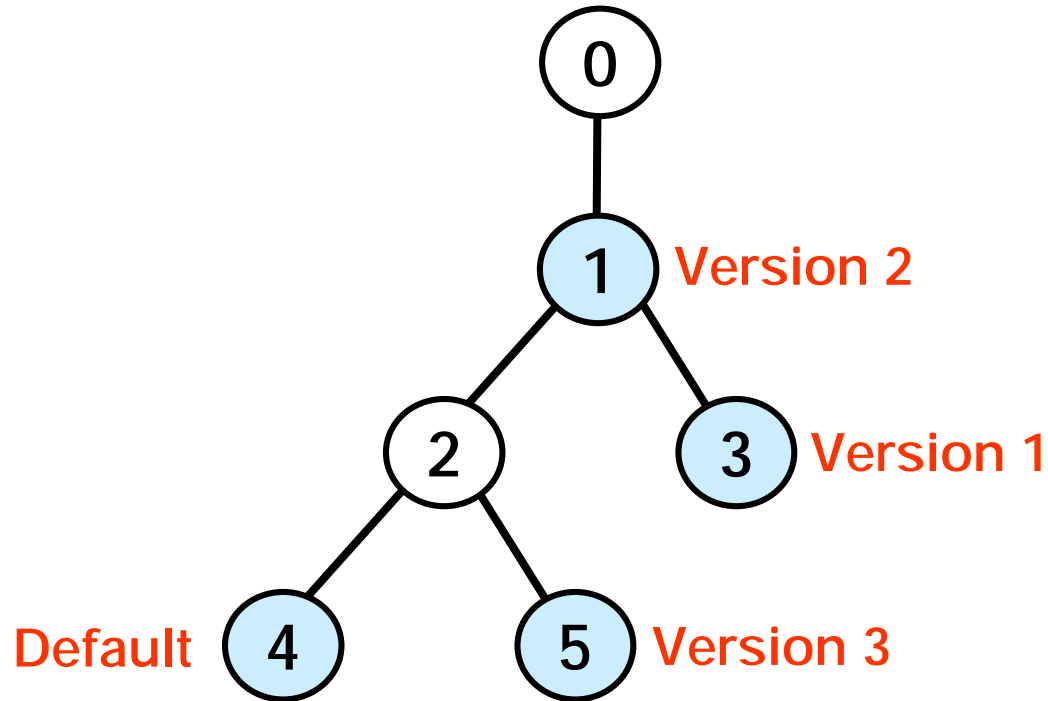
Compress

- How to determine if the compress was successful?
 - `select name, state_id from sde.versions;`
- If Default's state id = 0, it was successful
- But what if it is not 0?
 - Reconcile all versions with Default and compress again



ESRI

Complex Example



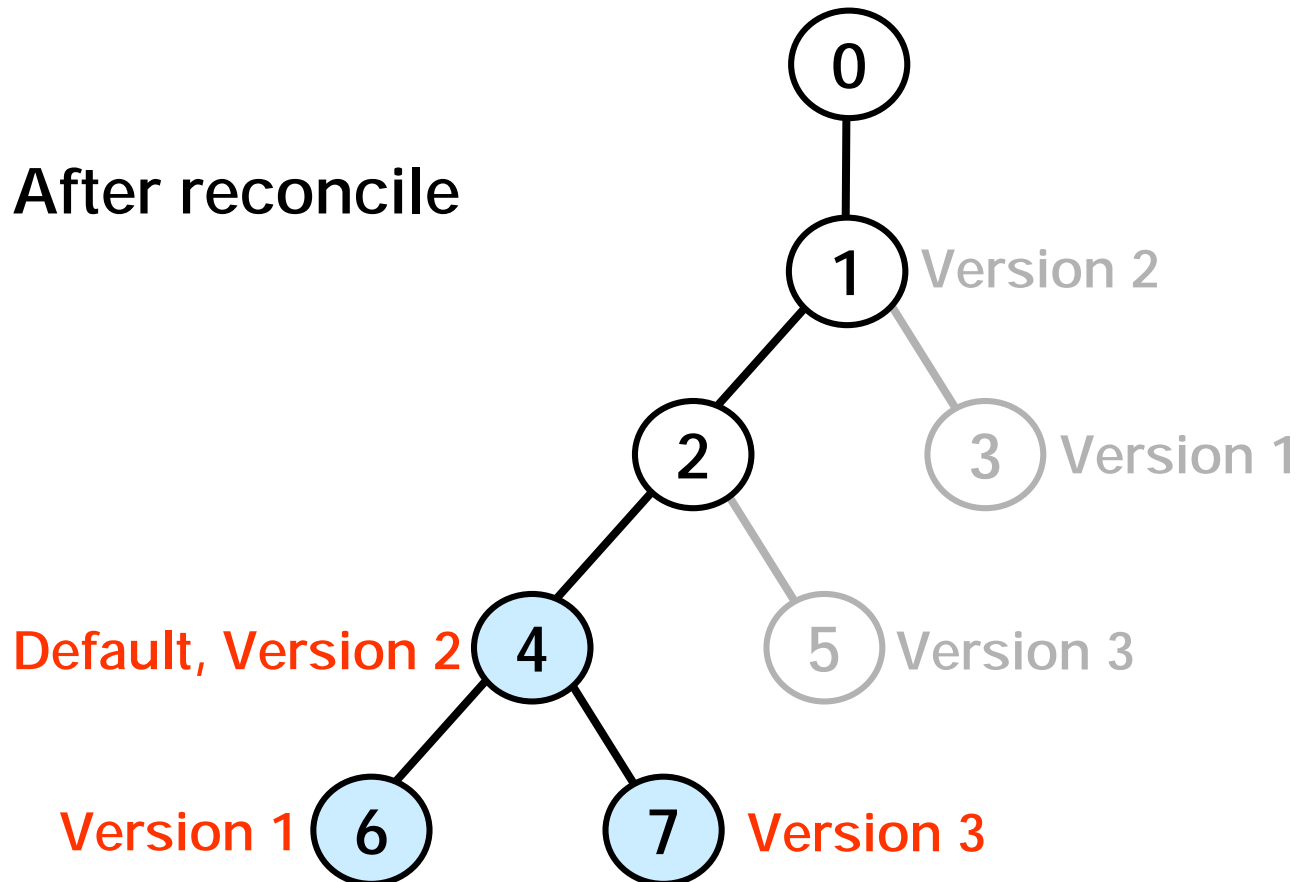
After compress, still requires reconciling and compressing again



ESRI

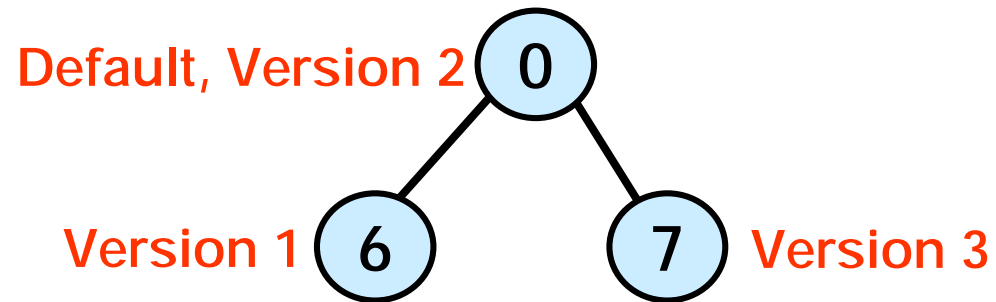
Complex Example

After reconcile



ESRI

Complex Example



Successful compression to base



ESRI

Compress

- Requires an exclusive lock on all database states
 - prevents inconsistent reads of the database
- 8.1 will support the ability to define a rollback segment for the compress transaction
 - (requires large rollbacks based on the number of rows in the delta tables)



ESRI

Final Slide !!!

- Update DBMS statistics (while the database is in production) and after running compress
 - analyze table <table> compute statistics
 - A/I 8.1 will contain a command in ArcCatalog to update statistics for the feature dataset and feature class



ESRI

Related sessions

- **Overview of the Geodatabase**
 - Wednesday/Thursday at 8:30, Room 6D (SDCC)
- **Designing and Using A Geodatabase**
 - Wednesday at 10:30, Room 3 (SDCC)
- **ArcSDE DBMS Administration: Oracle, SQL Server, Informix**
 - Thursday starting at 10:30, Room 3 (SDCC)



ESRI

