An ESRI® White Paper • March 2010

Compressing ArcSDE® Geodatabases That Contain Replicas—Best Practices

ESRI 380 New York St., Redlands, CA 92373-8100 USA
TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

J-9842

Compressing ArcSDE Geodatabases That Contain Replicas—Best Practices

An ESRI White Paper

Contents **Page**

Introduction 1

Prerequisites 1

Compressing an ArcSDE Geodatabase 1

Example 1: Versioned Editing and Compression 2

Example 2: Versioned Editing and Compression with a One-Way Replica 5
Best Practice 1 7
Best Practice 2 7

Example 3: Versioned Editing and Compression with a Two-Way Replica 9
Best Practice 3 12

Versioned Editing and Compression with Checkout Replicas 14
Best Practice 4 14

Full Compression 15
Best Practice 5 15

Summary 15

Appendix A: Replica System Versions 17

Compressing ArcSDE Geodatabases That Contain Replicas—Best Practices

Introduction

Geodatabase replication is built on top of versioning. In replication, versioning is used during synchronization to determine which changes to send between replicas and where to receive changes. When creating a replica, the replica version is explicitly defined and can be either the DEFAULT version or a named versioned. When sending changes during synchronization, edits that have been made to the replica version are sent. When changes are received during synchronization, they are applied to the replica version.

When a replica is created, system versions are also created behind the scenes by ArcGIS®. These system versions are used internally by ArcGIS during synchronization. The synchronization process may remove existing system versions or create new ones. Directly interacting with these versions can result in errors or inconsistent results when performing geodatabase replication. System versions are hidden by design to protect against this. Hidden versions do not appear in ArcGIS and are not returned by ArcObjects™ (e.g., when using custom code).

Understanding the details of how these system versions are used by ArcGIS is not necessary to successfully implement geodatabase replication. However, these versions can have an effect when compressing an ArcSDE® geodatabase. This white paper describes several best practices for handling system versions when compressing an ArcSDE geodatabase that contains replicas.

Prerequisites

This is an advanced topic white paper, and the reader should have a good understanding of fundamental concepts related to ArcSDE geodatabases, versioned editing, and geodatabase replication. For background information related to these topics, review the following ESRI® white papers: *Versioning*, *Versioning Workflows*, and *An Overview of Distributing Data with Geodatabases*.

Compressing an ArcSDE Geodatabase

Data in a feature class or table that is not registered as versioned exists as a base table in the geodatabase. There may actually be several tables depending on the storage type, but for the purposes of this document, they are collectively referred to as a *base table*. When a feature class or table is registered as versioned, delta tables are created in the geodatabase and associated with the feature class or table. When edits are made to a feature class or table in ArcGIS, changes are stored in their associated delta tables. When you view the data, the contents of the base table are queried along with the contents of the delta tables for the version you are working with. If you have several versions, each with different edits, these queries return different results depending on the version.

Over time, some of the edits accumulated in these delta tables may no longer be needed. This increases the size of the geodatabase and can affect performance. Compression is a process an ArcSDE geodatabase administrator runs periodically that performs the following to remove unneeded edits from the delta tables:

- Looks for and removes changes that are no longer accessed by a version
- Looks for changes that are accessed by all versions, applies them to the base table, and removes them from the delta tables

The following examples show cases where unneeded changes have accumulated and how compression works on versions in these cases. In each diagram, the content of the base table is shown at the top of the diagram. Changes in the delta tables are also shown and described with a statement starting with the word *edit* (e.g., Edit 1: Parcel 3 is moved). Changes that will be removed from the delta tables when compression is applied are shown as grayed out in the diagrams.

Example 1: Versioned Editing and Compression

In this example, an ArcSDE geodatabase has a parcels feature class containing three features. The feature class is registered as versioned, causing new delta tables to be generated in the geodatabase. A new version named Project1 is also created from the DEFAULT version. This version allows a group within the organization to independently make edits to the parcels feature class. Since there are currently no changes in the delta tables, the same three parcels from the base table are displayed regardless of which version is being accessed (figure 1).

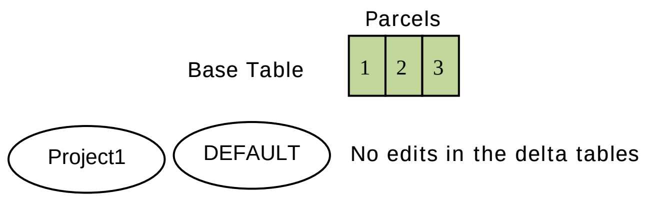

Figure 1: No edits have been applied to the parcels feature class.

An editor makes an edit in the DEFAULT version where parcel 3 is moved. At the same time, another editor working with the Project1 version adds a new feature: parcel 4. Both of these edits are stored in the delta tables of the parcels feature class. When connecting to the DEFAULT version and viewing parcels, ArcGIS queries the geodatabase and returns parcels 1 and 2 from the base table plus the edit to parcel 3 from the delta tables. When connecting to the Project1 version, ArcGIS queries the geodatabase and returns parcels 1, 2, and 3 from the base table as well as parcel 4 from the delta tables (figure 2).

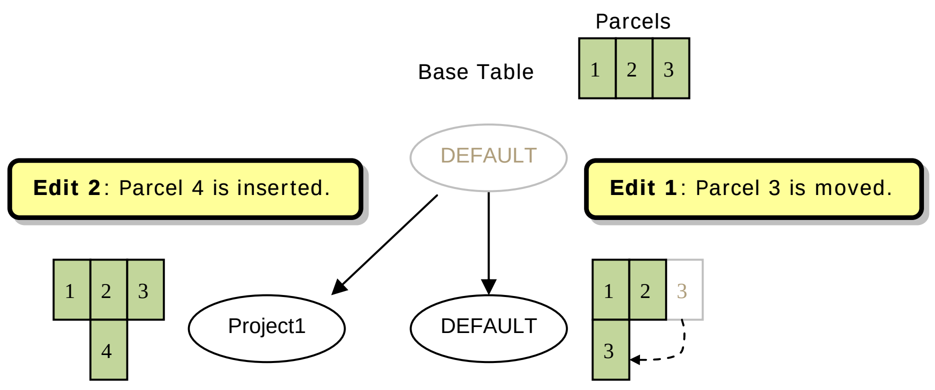

Figure 2: Parcel 3 is moved in the DEFAULT version (edit 1), and parcel 4 is added to the Project1 version (edit 2).

At this point, work in the Project1 version is complete and ready to be merged with the DEFAULT version. This is achieved by performing a reconcile and post procedure.

The reconcile process moves changes from the DEFAULT version into the Project1 version. The Project1 version is moved under DEFAULT, and the edit it references (i.e., parcel 3 is moved) is copied into Project1. If the same features have been edited in each version, a conflict occurs and a decision has to be made as to which change to accept. In this case, there are no conflicts (figure 3).

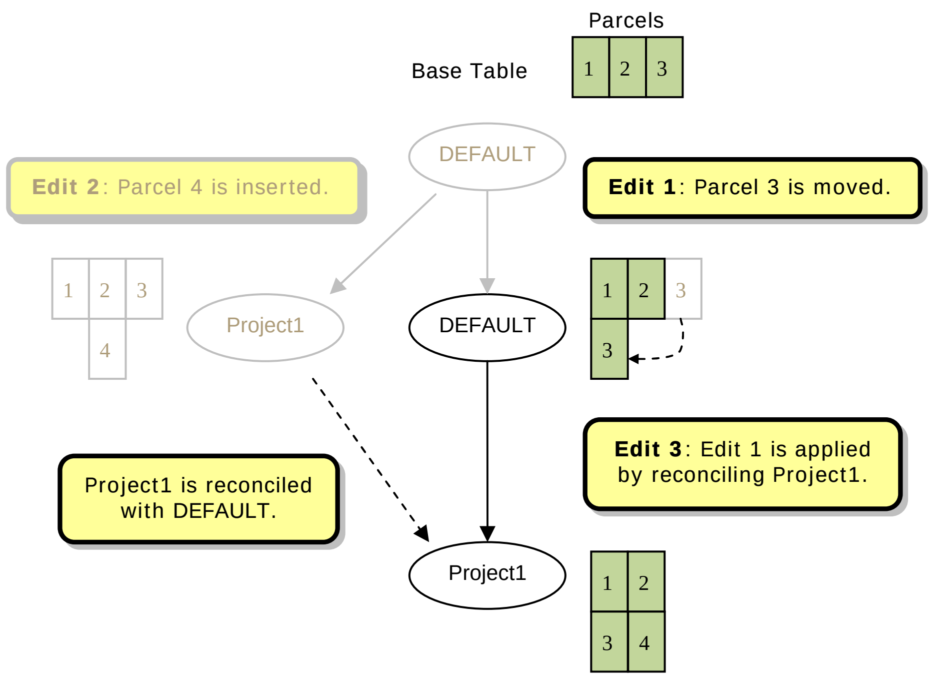
Parcels
Base Table
1 2 3
DEFAULT
Edit 2: Parcel 4 is inserted.
Edit 1: Parcel 3 is moved.
1 2 3 4
Project1
DEFAULT
1 2 3 3
Edit 3: Edit 1 is applied by reconciling Project1.
Project1 is reconciled with DEFAULT.
Project1
1 2 3 4

Figure 3: Project1 is reconciled with DEFAULT. This results in moving Project1 under DEFAULT and copying the edits it references (edit 3). Edit 2 will be removed during compression.

In figure 3, edit 2 is grayed out, indicating that if the compress operation is called at this point, the change will be removed from the delta tables because it is no longer referenced by a version. The Project1 version, which was the only version referencing edit 2, was moved under DEFAULT by reconciling and no longer references it.

Once reconciliation is completed, the post process can be executed. Posting always follows a reconcile operation. The post process moves changes from the Project1 version into the DEFAULT version (i.e., parcel 4 is added). Both versions now access the same edits in the delta tables. When connecting to either the DEFAULT version or the Project1 version and viewing parcels, ArcGIS queries the geodatabase and returns parcels 1 and 2 from the base table and parcels 3 and 4 from the delta tables (figure 4).

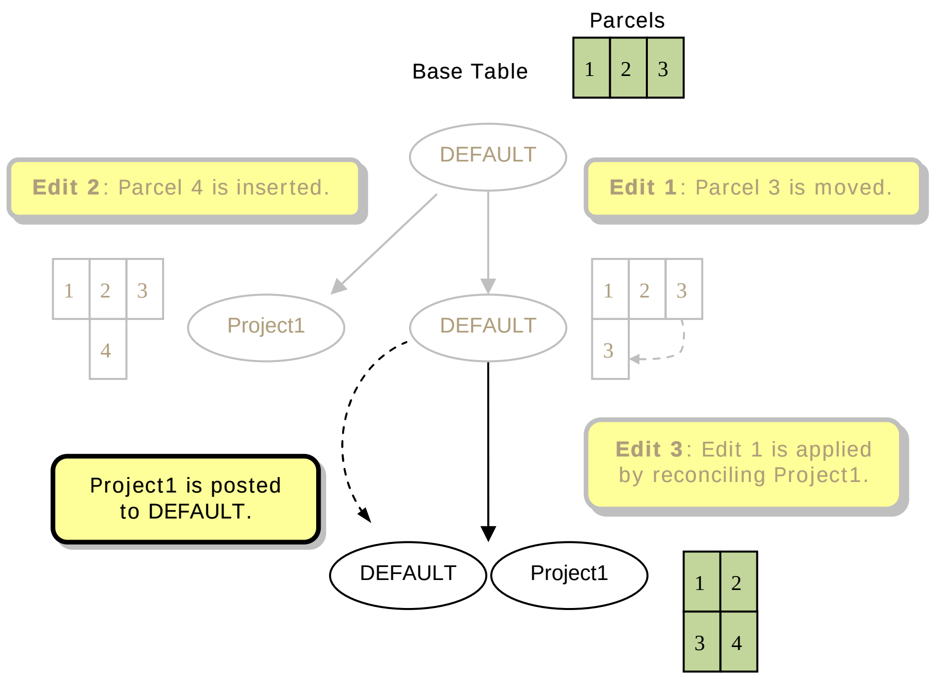
Parcels
Base Table
DEFAULT
Edit 2: Parcel 4 is inserted.
Edit 1: Parcel 3 is moved.
Project1
DEFAULT
Edit 3: Edit 1 is applied by reconciling Project1.
Project1 is posted to DEFAULT.
DEFAULT
Project1

Figure 4: The DEFAULT version is posted with the Project1 version. Compression will now move edit 1 and edit 3 to the base table.

In figure 4, edit 1 and edit 3 are grayed out, indicating that they will be removed from the delta tables by a compress operation. Since edit 1 and edit 3 are referenced by all versions in the geodatabase (i.e., Project1 and DEFAULT), compression will move them to the base table and remove them from the delta tables.

Next, compression is applied to the geodatabase; edit 1, edit 2, and edit 3 are removed from the delta tables for the reasons described above. The Project1 version is also manually deleted since the work is completed and the changes have been reconciled and posted to DEFAULT. Figure 5 shows the results of the compression.

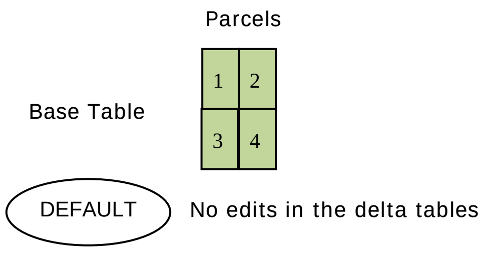

Figure 5: Edit 1, edit 2, and edit 3 are removed from the delta tables by compression.

Users accessing parcels in the DEFAULT version before compression (figure 4) and after compression (figure 5) will see the same results. However, after the compress operation, there are no edits in the delta tables. Therefore, the geodatabase has fewer records, and less data is accessed to return the same results.

This example describes a geodatabase with a single version and a few edits. In many cases, however, a multiuser geodatabase editing environment involves many edits and multiple versions. In these cases, an automated reconcile and post process is normally used.

Example 2: Versioned Editing and Compression with a One-Way Replica

This example is similar to example 1, except in this case, a one-way replica (Replica1) is created with the DEFAULT version as the replica version. The replica creation process creates a system version from DEFAULT for Replica1 in the parent geodatabase. A system version works like other versions except that it is associated with a replica and is maintained by ArcGIS. Figure 6 shows the parent replica geodatabase with the system version (Replica1). In this document, system versions are represented in the diagrams with the replica name and a dashed outline.

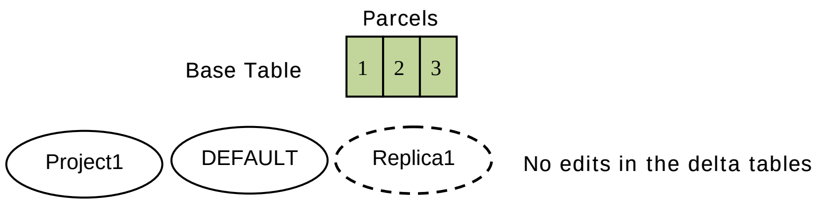

Figure 6: A system version is created when Replica1 is created. No edits have been applied to the parcels at this point.

The same editing activity from example 1 up to figure 4 is then performed:

- An editor moves parcel 3 in DEFAULT.
- Another editor adds parcel 4 in the Project1 version.
- A reconcile and post operation is performed between Project1 and DEFAULT.

Figure 7 shows the results.

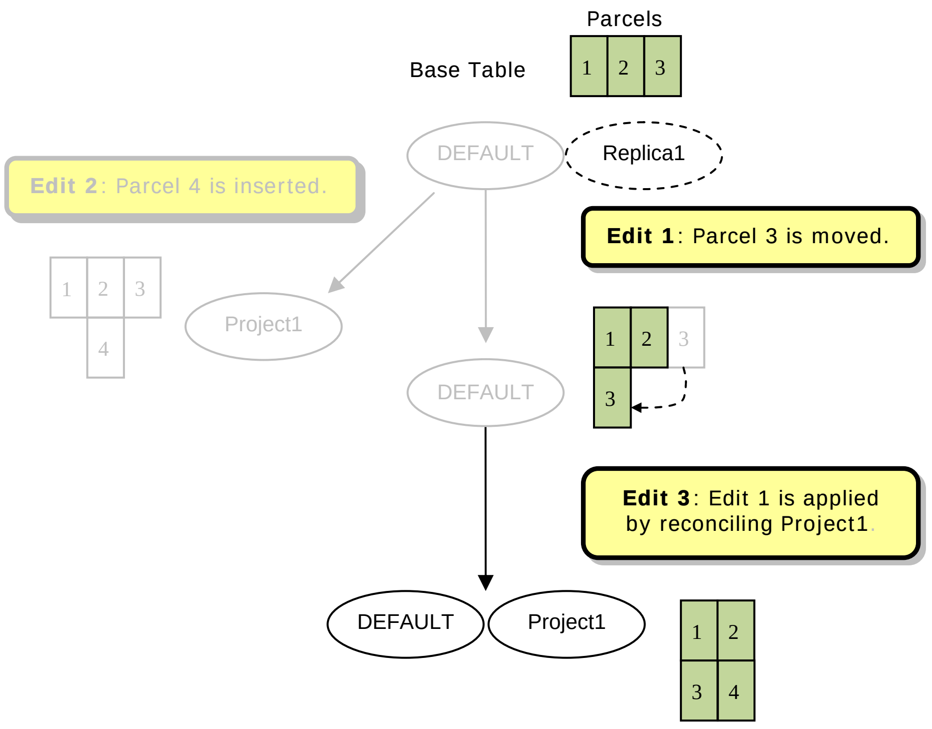
Parcels
Base Table
1
2
3
DEFAULT
Replica1
Edit 2: Parcel 4 is inserted.
Edit 1: Parcel 3 is moved.
1
2
3
4
Project1
DEFAULT
1
2
3
3
Edit 3: Edit 1 is applied by reconciling Project1.
DEFAULT
Project1
1
2
3
4

Figure 7: Changes up to figure 4 from example 1 are applied. Parcel 3 is moved in the DEFAULT version, parcel 4 is added in the Project1 version, and the Project1 version is reconciled and posted with DEFAULT.

Comparing figure 4 and figure 7, observe that if a compress operation is executed at this point, it will be much less effective than it was in example 1. In this example, only edit 2 will be removed from the delta tables instead of all edits at the same point in example 1. Edit 1 and edit 3 cannot be moved to the base table in figure 7, because they are not referenced by all versions (i.e., edit 1 and edit 3 are not referenced by the Replica1 system version).

Next, Replica1 is synchronized where changes are sent from this parent geodatabase to the relative replica geodatabase. The synchronization process sends edit 1 and edit 3 to the relative replica, since these changes were applied to the replica version (DEFAULT) after the replica was created. In this example, a connected synchronization is performed where an acknowledgment is automatically applied after the relative replica receives the changes. Upon receiving the acknowledgment message, the existing system version is removed and a new one is created from DEFAULT. Figure 8 shows the results after the synchronization.

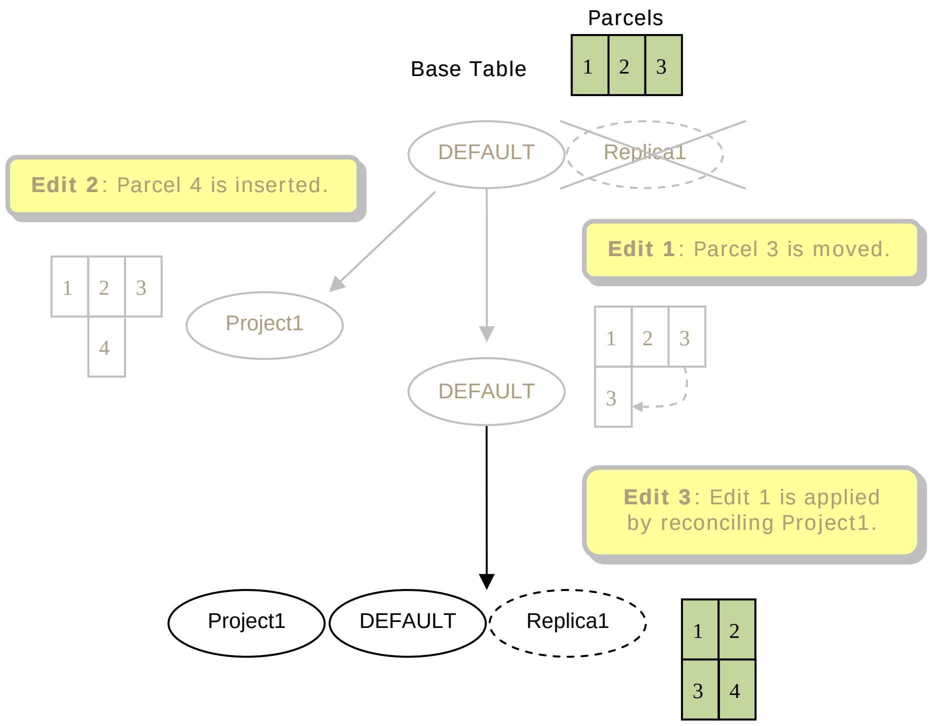
Parcels
Base Table
1 2 3
DEFAULT
Replica1
Edit 2: Parcel 4 is inserted.
Edit 1: Parcel 3 is moved.
1 2 3 4
Project1
1 2 3 3
DEFAULT
Edit 3: Edit 1 is applied by reconciling Project1.
Project1
DEFAULT
Replica1
1 2 3 4

Figure 8: Replica1 is synchronized. This removes the old Replica1 system version and creates a new one from DEFAULT.

Figure 8 shows that a compress operation executed at this point will be much more effective. Observe that edit 1 and edit 3 are now referenced by all versions, so they will be applied to the base table and removed from the delta tables.

Best Practice 1: Plan to synchronize replicas where you send changes on a regular basis. As example 2 shows, this process removes system versions that may be limiting the effectiveness of compression. Also note that when using disconnected synchronization, the system versions are only removed after changes are sent and an acknowledgment message is received. One approach is to schedule synchronizations to occur on a regular basis, for example, once a week or once each night. Another approach is to define the synchronization schedule around times when you anticipate having an appropriate volume of changes to send.

Best Practice 2: It is also important to remove replicas that you no longer plan to synchronize from the geodatabase. If the replica in this example is never synchronized, the system version added at replica creation is never removed. Over time, as more edits are applied to DEFAULT, this will prevent many edits from being removed during compression. Removing unused replicas removes unused system versions, which can lead to a more effective compression.

The final step in this example is to remove the Project1 version, then execute a compress operation. Figure 9 shows the results where all changes are removed from the delta tables.

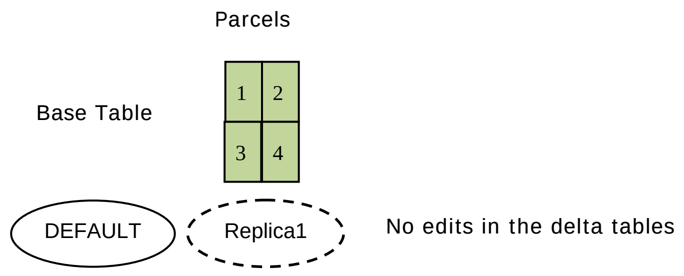

Figure 9: Edit 1, edit 2, and edit 3 are removed from the delta tables by compression.

This example describes a one-way replica and only the parent replica geodatabase is shown. This is because no system versions are created in a child replica geodatabase for a one-way replica. As a result, one-way replicas have no effect on compression in the child replica geodatabase.

Example 2 is a simple example involving a single named version and a single one-way replica. However, this can be easily expanded to include many named versions with many edits, as well as several one-way replicas. In this expanded scenario, a similar result can be achieved by doing the following:

- Reconcile and post each named version with DEFAULT, then manually delete each version.
- Synchronize each replica with changes being sent and an acknowledgment received.
- Compress the geodatabase.

The end result is similar to figure 9, only instead of a single system version, there would be one system version per replica (figure 10).

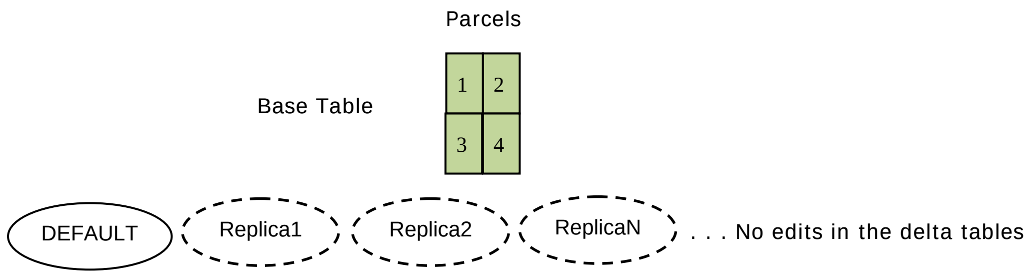

Figure 10: This shows the end result of example 2 with multiple named versions/replicas.

Another variation on this example would be where the Project1 version is used as the replica version instead of DEFAULT. In this instance, the Project1 version would still be reconciled and posted with DEFAULT, then the replica is synchronized. This end result would again be similar, except in this case the Project1 version cannot be deleted since it is associated with a replica (figure 11).

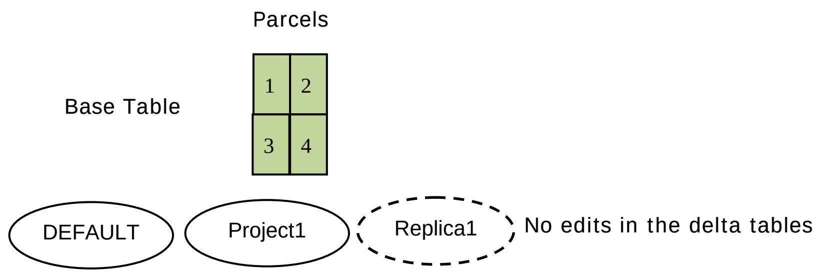

Figure 11: This is the end result of example 2 where Project1 is the replica version.

Example 3: Versioned Editing and Compression with a Two-Way Replica

This example starts at the same point as the previous examples, except in this case, a two-way replica (Replica2) is created with the DEFAULT version as the replica version. The replica creation process creates a system version from DEFAULT for Replica2 in the parent geodatabase and in the child geodatabase (figure 12).

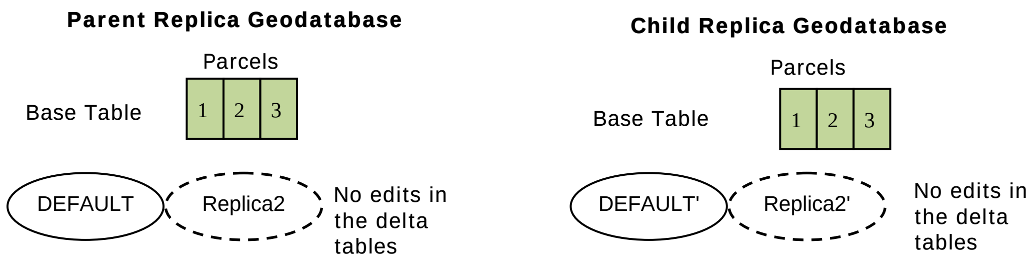
Parent Replica Geodatabase
Parcels
Base Table
1
2
3
DEFAULT
Replica2
No edits in the delta tables
Child Replica Geodatabase
Parcels
Base Table
1
2
3
DEFAULT'
Replica2'
No edits in the delta tables

Figure 12: A system version is created in the parent and in the child geodatabase when a two-way replica named Replica2 is created.

Next, parcel 3 is moved in the DEFAULT version of the parent replica geodatabase. Also, parcel 4 is added in the DEFAULT' version of the child replica geodatabase (figure 13).

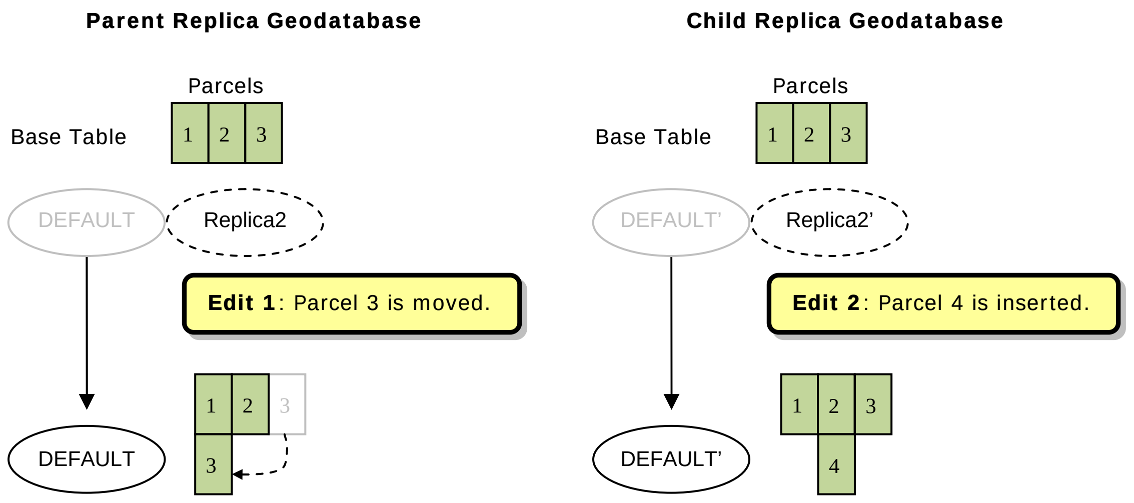
Parent Replica Geodatabase
Child Replica Geodatabase
Parcels
Base Table
1
2
3
DEFAULT
Replica2
Edit 1: Parcel 3 is moved.
DEFAULT
DEFAULT'
Replica2'
Edit 2: Parcel 4 is inserted.
4

Figure 13: Parcel 3 is moved in the DEFAULT version of the parent replica geodatabase (edit 1), and parcel 4 is added in the DEFAULT' version of the child replica geodatabase (edit 2).

Replica2 is then synchronized where changes are sent from the parent replica to the child replica (figure 14). In this case, connected synchronization is used and thus the changes are acknowledged implicitly. In this example, there are no conflicts; however, had the same feature been edited on both the parent and the child replica geodatabases, there would have been a conflict. See the synchronization and versioning help topic for more detailed information.

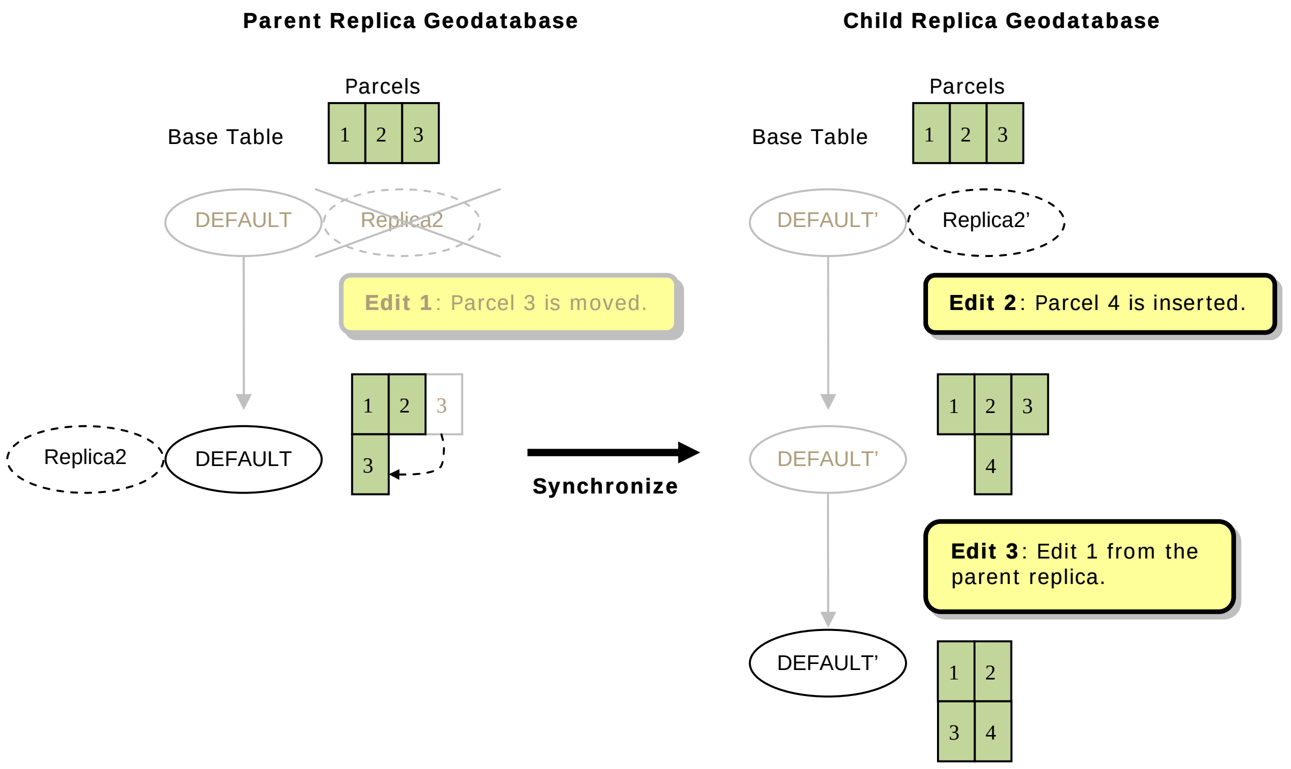
Parent Replica Geodatabase
Child Replica Geodatabase
Parcels
Base Table
1
2
3
DEFAULT
~~Replica2~~
Edit 1: Parcel 3 is moved.
Replica2
DEFAULT
Synchronize
DEFAULT'
Replica2'
Edit 2: Parcel 4 is inserted.
4
Edit 3: Edit 1 from the parent replica.

Figure 14: Changes are synchronized from parent to child. This applies edit 1 from the parent to the child as edit 3.

In figure 14, since an acknowledgment was received by the parent, the initial system version is deleted and a new one is created. With the old system version removed, a compress operation executed at this point on the parent replica geodatabase would remove edit 1 from the delta tables and apply it to the base table. In the child replica geodatabase, edit 1 from the parent is applied to the replica version (DEFAULT') during synchronization and is labeled edit 3. Since the initial system version was left unchanged, a compress process executed on the child at this point will result in no changes being removed from the delta tables.

Next, Replica2 is synchronized again; this time changes are sent from the child replica to the parent replica (figure 15). Connected synchronization is used, and the changes are acknowledged implicitly.

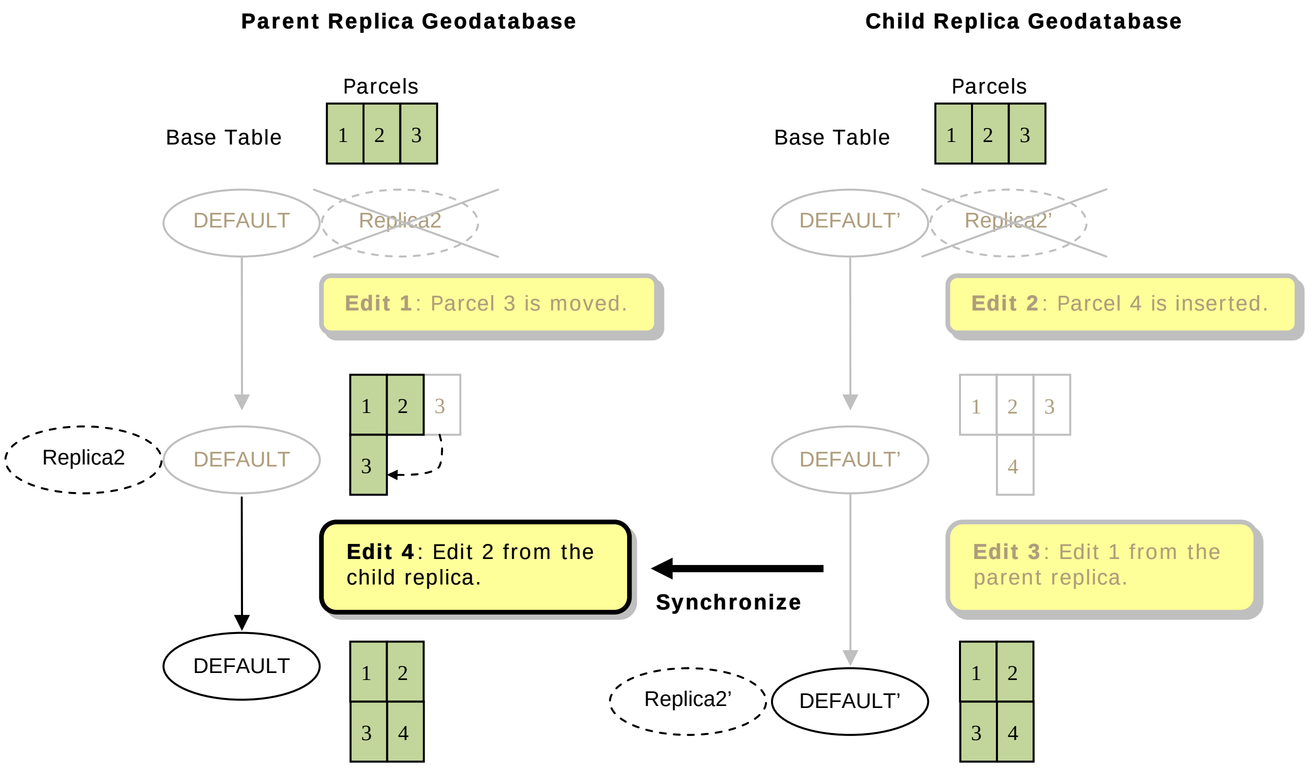
Parent Replica Geodatabase
Child Replica Geodatabase
Parcels
Base Table
1
2
3
DEFAULT
Replica2
Edit 1: Parcel 3 is moved.
Replica2
DEFAULT
Edit 4: Edit 2 from the child replica.
DEFAULT
4
Parcels
Base Table
DEFAULT'
Replica2'
Edit 2: Parcel 4 is inserted.
DEFAULT'
Edit 3: Edit 1 from the parent replica.
Synchronize
Replica2'
DEFAULT'

Figure 15: Changes are synchronized from child to parent. This applies edit 2 from the child to the parent as edit 4.

In figure 15, edit 2 from the child replica geodatabase is applied as edit 4 to the replica version (DEFAULT) in the parent replica geodatabase. Edit 3 is not sent because the system identified it as a change that already exists in the parent replica. In the child replica geodatabase, the existing system version is deleted and a new one (Replica2') is created from DEFAULT'. This allows compression of the child geodatabase to apply edit 2 and edit 3 to the base table and remove them from the delta tables. Compression of the parent geodatabase, however, would still leave edit 4 in the delta tables.

Best Practice 3: In some cases, changes predominantly need to be sent in one direction, but a two-way replica is still used. This may be done because changes may need to be sent back at a later time in the workflow, for example. In these situations, make sure to occasionally synchronize changes in the other direction even if there are few or no changes to send. This will remove system versions from the geodatabase that normally receives changes and allows compression to be more effective.

A third connected synchronization of Replica2 is then run to send changes from the parent to the child replica geodatabase (figure 16).

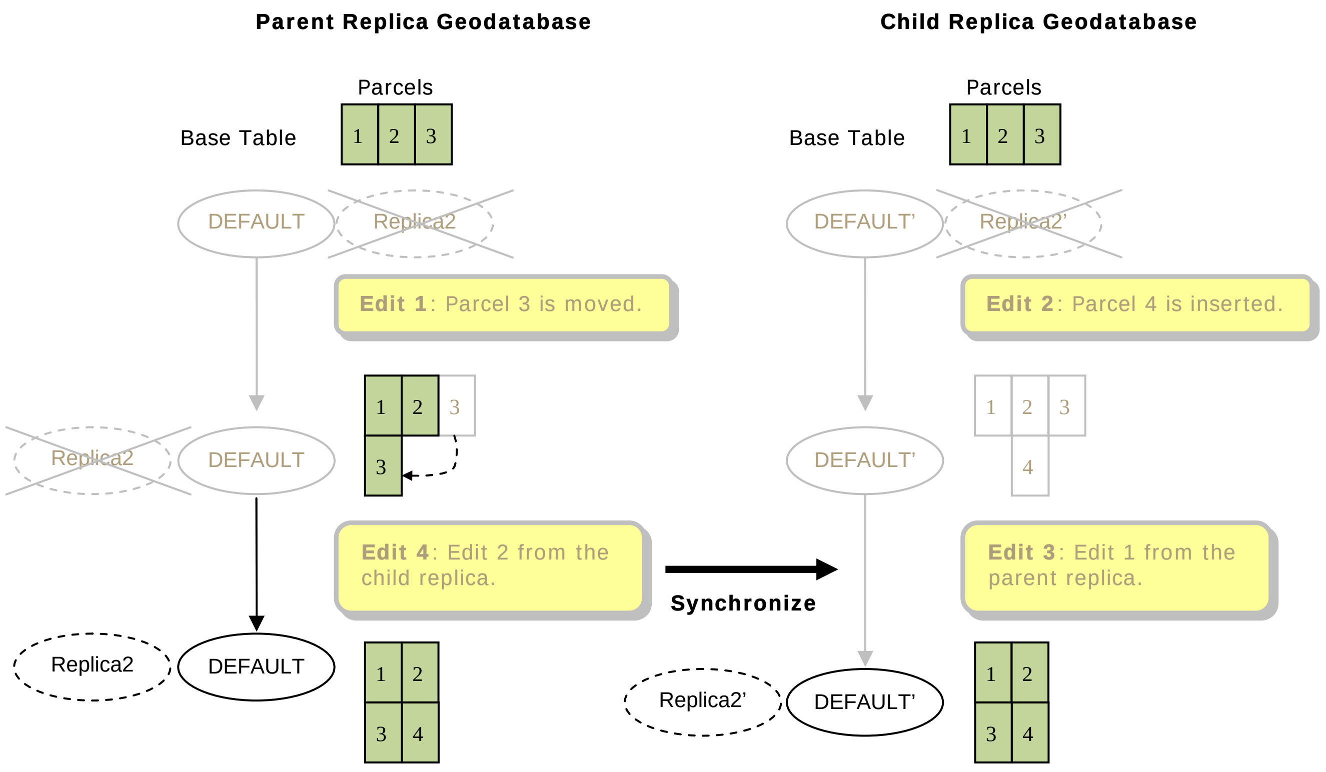
Parent Replica Geodatabase
Child Replica Geodatabase
Parcels
Base Table
1 2 3
DEFAULT
Replica2
Edit 1: Parcel 3 is moved.
Edit 4: Edit 2 from the child replica.
DEFAULT'
Replica2'
Edit 2: Parcel 4 is inserted.
Edit 3: Edit 1 from the parent replica.
Synchronize
1 2 3 4

Figure 16: Changes are synchronized from parent to child. No changes are sent during the synchronization.

In figure 16, edit 4 is identified as already existing in the child replica geodatabase and is therefore not sent. Since no other changes have been applied on the parent, no changes are sent to the child replica geodatabase. The acknowledgment received from the child replica geodatabase allows the old system version to be deleted on the parent replica geodatabase and a new one created. A compress operation run on the parent replica geodatabase at this point would move edit 1 and edit 4 to the base table and remove them from the delta tables.

Next, compression is executed on both replica geodatabases. This results in all edits being moved into the base tables and removed from the delta tables in both replica geodatabases (figure 17).

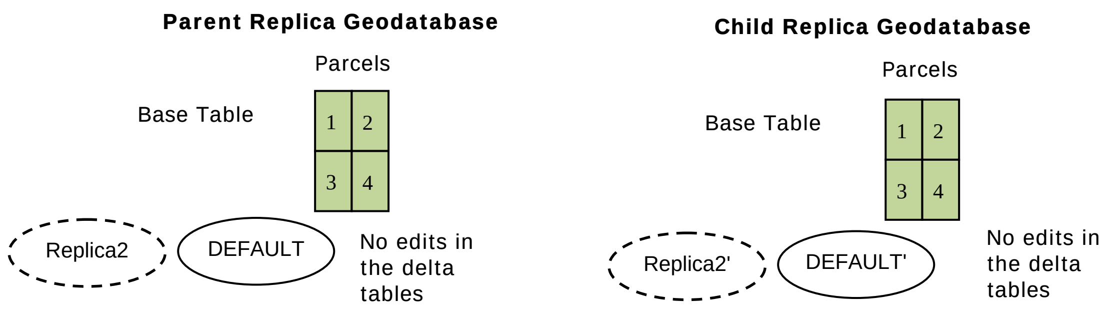
Parent Replica Geodatabase
Parcels
Base Table
1
2
3
4
Replica2
DEFAULT
No edits in the delta tables
Child Replica Geodatabase
Parcels
Base Table
1
2
3
4
Replica2'
DEFAULT'
No edits in the delta tables

Figure 17: Edits are removed from the delta tables during compression in both replica geodatabases.

For simplicity, this example shows edits applied only to the DEFAULT version. In the earlier examples, changes were reconciled and posted to DEFAULT from a named version, and the named version was deleted. Had the same pattern been followed for this example, where named versions are reconciled, posted, and deleted before the synchronizations, the end result would be the same.

Note that as the synchronizations are performed, more system versions than what are shown in the example diagrams may be created. These are not shown in the diagrams because they are not needed to explain the results in the examples. Think of the system versions in the diagrams as one or more system versions. See appendix A for more information on system versions.

Versioned Editing and Compression with Checkout Replicas

When a checkout replica is created, versions are created in the parent replica geodatabase as well as the child replica geodatabase (if the child is an ArcSDE geodatabase). These versions persist in the geodatabase until the checkout replica is synchronized (i.e., checked in). Synchronization removes the checkout replica as well as the system versions associated with it. A practice where you first check in these replicas, then perform the pattern of reconcile and post plus version deletion, will yield the same result as shown in figure 5 (i.e., end result of example 1).

Best Practice 4: The examples describe a pattern where changes from named versions are on occasion reconciled and posted with DEFAULT and optionally deleted. The geodatabase is then compressed. The following describes this type of workflow with ArcSDE geodatabases that have replicas of any type:

- First, synchronize (check in) your checkout replicas.
- Reconcile and post and optionally delete the nonreplica versions.
- If you plan on receiving changes for your two-way replicas, do so at this point. If you are using a manual reconcile policy, make sure to resolve any conflicts after the synchronization process.
- Reconcile and post the replica versions.
- Synchronize the replicas such that changes are sent and an acknowledgment received.
- Run the compress operation.

Note that when compression is complete, it is an ideal time to update statistics and rebuild indexes on the geodatabase datasets.

Full Compression

In all the examples discussed, the end result was that edits were removed from the delta tables. This is known as full compression. Although this is ideal, achieving full compression is often not necessary and may not even be practical. For instance, in example 2, if many replicas are involved, you will need to send changes for all replicas to achieve full compression. In your business workflow, you might not have a time period or opportunity to send changes for all replicas.

In example 3, a third synchronization was added that sent no changes to achieve full compression. If many two-way replicas exist in your ArcSDE geodatabase, you may need to perform several synchronizations with no changes to enable the geodatabase to fully compress. This again may not be practical.

Using the information in this white paper, you can develop a business practice to schedule the compress process around reconciling, posting, and synchronization workflows. Another approach is to simply schedule a compress process to occur on a regular basis (e.g., nightly). As long as the practices described in this document are applied, you should not see a case where changes that you expect to be removed during compression are not removed because of geodatabase replication.

It is sometimes assumed that full geodatabase compression is needed because external, non-ESRI applications (that can only access data in the base tables) are used with the geodatabase. In this case, consider creating multiversioned views and using them with the non-ESRI applications instead of trying to compress to base.

Best Practice 5: In general, try to achieve some level of compression rather than full geodatabase compression. Attempting to achieve full geodatabase compression may lead to many additional synchronizations, which may not be practical in your particular business workflow.

Summary

This white paper described how versioning is used by geodatabase replication. Aside from the compress process, it has little impact on existing versioning workflows. The information in this document can be used to achieve an effective compression with replicas in the geodatabase. Information on what the replica system versions are and how to find them is also included (see appendix A).

Several best practices were presented and are summarized below:

Best Practice 1: It is important to remove replicas that you no longer plan to synchronize from the geodatabase.

Best Practice 2: Plan to synchronize replicas where you send changes on a regular basis.

Best Practice 3: If you predominantly send changes in one direction with a two-way replica, occasionally synchronize changes in the other direction even if there are few or no changes to send.

Best Practice 4: Do the following to incorporate replication into an existing pattern where changes from named versions are reconciled and posted with DEFAULT and optionally deleted:

- First, synchronize (check in) your checkout replicas.
- Reconcile and post and optionally delete the nonreplica versions.
- If you plan on receiving changes for your two-way replicas, do so at this point. If you are using a manual reconcile policy, make sure to resolve any conflicts after the synchronization process.
- Reconcile and post the replica versions.
- Synchronize the replicas such that changes are sent and an acknowledgment received.
- Run the compress operation.

Note that when the compress process is complete, it is an ideal time to update statistics and rebuild indexes on the geodatabase datasets.

Best Practice 5: In general, try to achieve an effective compression rather than full geodatabase compression. Effective compression can be achieved by following the best practices in this document and scheduling compression to occur on a regular basis (e.g., nightly).

Appendix A: Replica System Versions

The examples in this document refer to replica system versions. These are versions that are used behind the scenes by ArcGIS to determine the changes to synchronize for a replica. A set of system versions exists for each replica in the geodatabase. Various system versions are created and deleted by ArcGIS as the replicas are synchronized.

These versions are hidden by design, which means they are not displayed in ArcGIS and not returned by ArcObjects. However, if you query the `SDE.VERSIONS` table in the underlying DBMS, you will be able to see these versions. You can identify them by the way they are named. The naming conventions for the replica system versions are as follows:

Two-way and one-way replicas

- `SYNC_RECEIVE_<replica id>_<generation number>`
- `SYNC_RECEIVE_REC_<replica id>_<generation number>`
- `SYNC_SEND_<replica id>_<generation number>`

Checkout replicas

- `REP_CO_SYNC_<replica id>`

For example, an ArcSDE geodatabase implemented on Oracle® has been built with a two-way replica named *rep_2way*, a one-way replica called *rep_1way*, and a parent checkout replica named *rep_co*. After connecting to Oracle using SQL*Plus, the following query is executed to list these replicas along with their replica IDs:

```
SQL> select id, name from sde.gdb_replicas;

 ID NAME
---------- --------------------------------
 57 rep_1way
 56 rep_2way
 58 rep_co
```

Next the `SDE.VERSIONS` table is queried as follows:

```
SQL> select name from sde.versions order by name;

NAME
------------------------------------------------------------
DEFAULT
SYNC_RECEIVE_56_0
SYNC_RECEIVE_56_1
SYNC_RECEIVE_56_2
```

```
SYNC_RECEIVE_REC_56_2
SYNC_SEND_56_0
SYNC_SEND_57_0
rep_co
```

In this geodatabase, observe that the only non-system versions are DEFAULT and *rep_co*. The *rep_co* version is the replica version and was created when the checkout replica was created. Applying the naming pattern information above, you can see that there are several system versions for *rep_2way* (replica ID 56) and one system version for *rep_1way* (replica ID 57). The following query returns only the system versions for *rep_2way:*

```
SQL> select name from sde.versions where name like
'SYNC_SEND_56%' or name like 'SYNC_RECEIVE_%56%';

NAME
-----------------------------------------------------------
SYNC_RECEIVE_56_0
SYNC_RECEIVE_56_1
SYNC_RECEIVE_56_2
SYNC_RECEIVE_REC_56_2
SYNC_SEND_56_0
```

In this example, the child checkout replica is hosted in another ArcSDE geodatabase implemented on Oracle. When the `SDE.GDB_REPLICAS` table in that geodatabase is queried, the following is returned:

```
SQL> select id, name from sde.gdb_replicas;

 ID NAME
---------- --------------------------------
 42 rep_co
```

Next, the `SDE.VERSIONS` table is queried in the same geodatabase as follows:

```
SQL> select name from sde.versions order by name;

NAME
-----------------------------------------------------------
DEFAULT
REP_CO_SYNC_42
rep_co
```

The results of the query show the DEFAULT version and the *rep_co* version as the only nonsystem versions. The *rep_co* version is the replica version and was created as part of the checkout process. Also listed is a version named `REP_CO_SYNC_42`, which, as shown in the naming pattern above, is a system version for the *rep_co* checkout replica.

Using this information, you can determine which system versions exist in your ArcSDE geodatabase and which replicas they are associated with. Note that the system versions listed will change over time as the replicas are synchronized.

It is very important that you never interact with these system versions manually. If you need to remove these versions, you should follow the examples in this white paper or delete the replica in ArcGIS.

About ESRI

Since 1969, ESRI has been helping organizations map and model our world. ESRI's GIS software tools and methodologies enable these organizations to effectively analyze and manage their geographic information and make better decisions. They are supported by our experienced and knowledgeable staff and extensive network of business partners and international distributors.

A full-service GIS company, ESRI supports the implementation of GIS technology on desktops, servers, online services, and mobile devices. These GIS solutions are flexible, customizable, and easy to use.

Our Focus

ESRI software is used by hundreds of thousands of organizations that apply GIS to solve problems and make our world a better place to live. We pay close attention to our users to ensure they have the best tools possible to accomplish their missions. A comprehensive suite of training options offered worldwide helps our users fully leverage their GIS applications.

ESRI is a socially conscious business, actively supporting organizations involved in education, conservation, sustainable development, and humanitarian affairs.

Contact ESRI

1-800-GIS-XPRT (1-800-447-9778)
Phone: 909-793-2853
Fax: 909-793-5953
info@esri.com
www.esri.com

Offices worldwide
www.esri.com/locations

ESRI
380 New York Street
Redlands, California
92373-8100 USA