

The method returns a `WriteResult` object with the status of the operation. Upon successful update, the `WriteResult` object should have `"nModified"` equal to 1.

Step 4: Remove existing documents from the `admin.system.users` collection.

```
db.getSiblingDB("admin").system.users.remove( {} )
```

The method returns a `WriteResult` object with the number of documents removed in the `"nRemoved"` field.

Step 5: Copy documents from the `admin.system.backup_users` collection. Copy all documents from the `admin.system.backup_users`, created during the 2.6 upgrade, to `admin.system.users`.

```
db.getSiblingDB("admin").system.backup_users.find().forEach(
  function (userDoc) {
    status = db.getSiblingDB("admin").system.users.insert( userDoc );
    if (status.hasWriteError()) {
      print(status.writeError);
    }
  }
);
```

Step 6: Update the version document for the `authSchema`.

```
db.getSiblingDB("admin").system.version.update(
  { _id: "authSchema" },
  { $set: { currentVersion: 1 } }
)
```

For a sharded cluster, repeat the downgrade process by connecting to the *primary* replica set member for each shard.

Note: The cluster's mongos instances will fail to detect the authorization model downgrade until the user cache is refreshed. You can run `invalidateUserCache` on each mongos instance to refresh immediately, or you can wait until the cache is refreshed automatically at the end of the user cache invalidation interval. To run `invalidateUserCache`, you must have privilege with `invalidateUserCache` (page 421) action, which is granted by `userAdminAnyDatabase` (page 412) and `hostManager` (page 410) roles.

Result The downgrade process returns the user data to its state prior to upgrading to 2.6 authorization model. Any changes made to the user/role data using the 2.6 users model will be lost.

Downgrade Updated Indexes

Text Index Version Check If you have *version 2* text indexes (i.e. the default version for text indexes in MongoDB 2.6), drop the *version 2* text indexes before downgrading MongoDB. After the downgrade, enable text search and recreate the dropped text indexes.

To determine the version of your text indexes, run `db.collection.getIndexes()` to view index specifications. For text indexes, the method returns the version information in the field `textIndexVersion`. For example, the following shows that the text index on the `quotes` collection is version 2.

```
{
  "v" : 1,
  "key" : {
    "_fts" : "text",
    "_ftsx" : 1
  }
}
```