Xindice Release 1.1b1 Through 1.1b4 and 1.1 Upgrading How-To

$Revision: 598506 $

by Vadim Gritsenko

Table of contents

1. Intended audience

This How-to is aimed at developers or administrator performing upgrade of one of 1.1 beta releases or 1.1 final release of Xindice to the latest 1.2 release.

2. Prerequisites

Shutdown. Shutdown the database before starting upgrade process.

Backup. Always back up your database before starting an upgrade.

3. Upgrade Steps

3.1. Overview

All Xindice 1.1 beta releases are utilising same configuration files and backward compatible with the Xindice 1.1 final release. The areas where betas are different are:

- Changes to BTreeFiler and HashFiler. BTreeFiler has been changed to address incorrect handling of numeric indexes (see Bug #19203). HashFiler has been changed to address inefficient hash function implementation. Due to these changes, existing databases should be upgraded to the latest format using supplied database rebuild utility.
- Changes to release packaging. Since Xindice 1.1, there are only two releases are being shipped: source and binary. Binary release includes both xindice.war file ready for deployment, as well as xindice.jar for embedded applications.
- Changes to Xindice scripts. Xindice now comes with two main commands, `xindice` and `xindiced`. First one is a Xindice command line tool, and second provides a way to start Xindice database with embedded HTTP server (Jetty).

There are no format or layout differences between 1.1 final release and the latest 1.2 release.

3.2. Procedure

Upgrade procedure consists of the following:

- Install Xindice 1.2 release. If you are upgrading from 1.1 final release, this is the only step you need to perform. If you are upgrading from one of the beta releases, please follow the following steps.
- Point XINDICE_HOME environment variable to the Xindice installation directory.
- Copy existing database over to XINDICE_HOME/db. (Alternatively, set environment variable XINDICE_DB_HOME to point to directory which contains db directory.)
- In the shell, execute

(Unix):

```
      cd $XINDICE_HOME
      ./bin/xindice_rebuild rebuild db
```

(Windows):

```
      cd %XINDICE_HOME%
      bin\xindice_rebuild rebuild db
```

(If using XINDICE_DB_HOME variable, substitute 'db' with '$XINDICE_DB_HOME/db'.)

- Check tool output for any problems. If any of the collections or indices could not be rebuilt to new format, do not panic - just follow an alternate approach outlined below.
- Test resulting database using command line tool.

3.3. Alternate Procedure

If any of the collections or indices could not be converted to new format, follow these steps. To migrate collections over to the new database:

- Start old Xindice database (using backed up data files).
- Export all documents from affected collections (replace <name> with collection path):
  (Unix):

  ```
        mkdir export
        ./bin/xindiceadmin export -c /db/<name> -f export
  ```

  (Windows):

  ```
        mkdir export
        bin\xindiceadmin export -c /db/<name> -f export
  ```
- Stop old Xindice database.
- Start new database.
- Recreate collections in the new database.
- Delete corrupted collections from new database using command line tool. This should also remove all data files and index files of these collections.
- Import all documents into the new database:
  (Unix):

  ```
        ./bin/xindice import -c /db -f export/<name>
  ```

  (Windows):

  ```
        bin\xindice import -c /db -f export\<name>
  ```
- Create all indexes which were present in the old database.

For indices which can not be converted:

- Start new database.

- Drop corrupted indices.
- Recreate indices anew.

If you run into problems, don't worry. Be sure to send feedback so that we can improve these documents for the next person.