

# Xindice 1.2 Commandline Tool Guide

\$Revision: 598114 \$

by Jay Kalafus, Kimbro Staken

---

NOTICE:

---

## Table of contents

1 General Information.....	3
1.1 Collection context.....	3
1.2 Common parameters.....	4
2 Collection Management Commands.....	4
2.1 Adding a Collection.....	4
2.2 Deleting a Collection.....	6
2.3 List Collections.....	7
3 Document Management Commands.....	8
3.1 Adding a Document.....	8
3.2 Adding a Binary Resource.....	9
3.3 Adding Multiple Documents.....	10
3.4 Deleting a Document.....	11
3.5 Retrieving a Document.....	12
3.6 List Resources.....	12
3.7 Importing a Directory Tree.....	13
3.8 Exporting a Directory Tree.....	14

4 Collection Indexer Actions.....	15
4.1 Adding a Collection Indexer.....	15
4.2 Deleting a Collection Indexer.....	17
4.3 Listing Collection Indexers.....	18
5 Quering a Collection.....	19
5.1 Xpath Query.....	19
5.2 XUpdate Query.....	20
5.3 Full Text Query.....	21
6 Miscellaneous Actions.....	21
6.1 Shutting down the Server.....	21
6.2 Accessing Help.....	22

**Note:**

If you notice incorrectness in this documentation, please [notify](#) Xindice community. Your feedback will help to create better documentation.

## 1. General Information

### 1.1. Collection context

Collection context refers to a collection location. It can be either collection canonical name or complete XML:DB URL. The collection context points to must already exist.

Collection canonical name starts with '/', then has a database name and the collection path. So, a collection 'foo' under a collection 'bar' in database 'db' will have canonical name /db/bar/foo. Canonical names are used for convenience and are converted to complete URLs by the command tool.

Complete URL has the following format:

`xmldb:<driver_protocol>://<host>:<port><path>`, where

**driver\_protocol**

Either `xindice` (XML-RPC driver) or `xindice-embed` (embedded driver).

Driver choice depends on whether or not `-l` switch was specified.

**host and port**

Host name or IP address and port number where Xindice server is running. This information only is used for XML-RPC driver, it can be present in URL when embedded driver is used, but will be ignored, so usually it is omitted. For XML-RPC, host name and port number can be omitted, too, then default address 127.0.0.1:8888 is used.

**path**

Collection canonical name

#### Examples

```
xmldb:xindice://10.1.1.1:8889/db1/col1/col2
```

Context points to a collection `col2`, which is a child of the collection `col1`, located in database `db1` that is available on Xindice server running on host `10.1.1.1` with port number `8889`. To communicate with the server, XML-RPC protocol will be used.

```
xmldb:xindice-embed:///db1/col1/col2
```

Context points to a collection `col2`, which is a child of the collection `col1`, located in database `db1` that is available on embedded Xindice instance. Command tool will use embedded

driver.

```
/db1/col1/col2
```

If `-l` switch was specified, this canonical name will be converted to `xmlldb:xindice-embed:///db1/col1/col2`. Otherwise, the result will be `xmlldb:xindice:///db1/col1/col2` (please note, that the default target address 127.0.0.1:8888 will be used).

## 1.2. Common parameters

**-c <context>**

The collection context that points to Xindice collection under which the action will be executed. The context can be either canonical name of the collection or complete XML:DB URL. The collection context points to must already exist.

**-l**

Use a local database rather than going over the network. This is done by using different driver to access database, embedded driver will be used instead of XML-RPC. This parameter should be combined with `-d` to specify system configuration to use.

**-d <file path>**

Path to the database configuration to use for the local database. Only applies if `-l` is specified. If this parameter is omitted, Xindice will look for system property "xindice.configuration", which can be set by setting environment variable `XINDICE_CONFIG` or by specifying it as the `-D` parameter for java command, which is what usually xindice command does itself.

**-v**

Show verbose output, additional information and exceptions.

**-h**

Print help screen for an action.

## 2. Collection Management Commands

### 2.1. Adding a Collection

Adds a new collection to the database. When adding a collection under an existing collection hierarchy all parent collections must already exist in the database.

#### Summary

```
xindice add_collection -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

Command name abbreviation: ac

## Parameters

**-c <context>**

The collection context. Required.

**-n <name>**

Name for the collection to be created, must be present unless collection configuration is specified. It can also be a collection path rather than a name, in that case all parent collections must already exist.

**-f <file>**

Name of the file that holds collection configuration. If specified, the rest of command-specific parameters (like collection name) are ignored. A configuration may include child collections or indexers, all of them will be created at the same time as well.

**--filer <name>**

Collection filer, can be HashFiler or BTreeFiler (default: BTreeFiler). A filer defines which data structure is going to be used for storing information. HashFiler generally is more suitable for collections that are not expected to grow significantly.

**--pagesize <number>**

Page size for file pages (default: 4096 bytes).

**--pagecount <number>**

Initial number of pages in the primary storage (default: 1024).

- BTreeFiler: number of pages filer will be created with. This number can grow as more and more documents are added to the collection
- HashFiler: the size of the hash table main storage. Main storage size is constant. If main storage is at its capacity, documents are stored in overflow storage, affecting efficiency of the hash table

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

## Examples

Create a top level collection pebbles in database db. Collection will use BTreeFiler with

default values for page count and page size.

```
xindice add_collection -c /db -n pebbles
```

Create a child collection boulder in collection pebbles.

```
xindice add_collection -c /db/pebbles -n boulder
```

Create a child collection stone in collection boulder.

```
xindice add_collection -c /db/pebbles -n boulder/stone
```

Create a child collection sand in collection pebbles that uses HashFiler with page size equal to 1024 bytes and the size of the hash table main storage equals to 2048.

```
xindice add_collection -c /db/pebbles -n sand --filer HashFiler
--pagesize 1024 --pagecount 2048
```

Create a child collection in collection pebbles with the configuration specified in the file /tmp/collections/config.xml. If collection configuration contains any child collections or indexers, they will be created as well.

```
xindice add_collection -c /db/pebbles -f /tmp/collections/config.xml
```

## 2.2. Deleting a Collection

Deletes a collection or subcollection from the database. If deleting a collection that has subcollections, indexers and/or documents they will also be deleted.

### Summary

```
xindice delete_collection -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

Command name abbreviation: dc

### Parameters

**-c <context>**

The collection context. Required.

**-n <name>**

Name of the collection to delete. Required. It can also be a collection path rather than a name.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Delete a top level collection pebbles and all its subcollections, indexers and documents.

```
xindice delete_collection -c /db -n pebbles
```

Delete a child collection boulder from collection pebbles and all its subcollections, indexers and documents.

```
xindice delete_collection -c /db/pebbles -n boulder
```

Delete a child collection stone from collection boulder. Collection boulder is not affected.

```
xindice delete_collection -c /db/pebbles -n boulder/stone
```

## 2.3. List Collections

List all child collections under the given collection context.

### Summary

```
xindice list_collections -c <context> [-l [-d <path>]] [-v]
```

Command name abbreviation: lc

### Parameters

**-c <context>**

The collection context. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

**Examples**

List the top level collections in database db.

```
xindice list_collections -c /db
```

List child collections of collection pebbles.

```
xindice list_collections -c /db/pebbles
```

**3. Document Management Commands****3.1. Adding a Document**

Adds a document to a collection. Adding a document requires the file path to the document. If a document key is not provided an auto-generated system key will be used for the document. When entering the file path be sure to include the path and file extension.

**Summary**

```
xindice add_document -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

Command name abbreviation: ad

**Parameters****-c <context>**

The collection context. Required.

**-f <file>**

File path for the file with the document being added to the collection. Required. File content must be a well-formed XML document.

**-n <key>**

Key assigned to the document key. If this parameter is omitted, database will generate an unique key and assign it to the document.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l



is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Add a document to the collection pebbles with a key "stones".

```
xindice add_document -c /db/pebbles -f /tmp/stones.xml -n stones
```

Add a document to the collection boulder with an automatically generated key.

```
xindice add_document -c /db/pebbles/boulder -f /tmp/bookmarks.xml
```

## 3.2. Adding a Binary Resource

Adds a binary resource a collection, if support of binary resources has been turned on for the collection. If a resource key is not provided an auto-generated system key will be used for the resource. When entering the file path be sure to include the path and file extension.

The resource may or may not be XML document. Even if resource is XML document, it cannot be indexed, cannot be queried with XPath or modified with XUpdate.

### Summary

```
xindice add_resource -c <context> [-l [-d <path>]] [-v]  
[parameters...]
```

Command name abbreviation: ar

### Parameters

**-c <context>**

The collection context. Required.

**-f <file path>**

File path for the resource being added to the collection. Required.

**-n <key>**

Key assigned to the resource. If this parameter is omitted, database will generate an unique key and assign it to the resource.

**-l**

Use a local database rather then going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l

is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Add a resource to a collection pebbles with a key "stones".

```
xindice add_document -c /db/pebbles -f /tmp/stones.jpg -n stones
```

Add a resource to the collection boulder with an automatically generated key.

```
xindice add_resource -c /db/pebbles/boulder -f /tmp/bookmarks.html
```

### 3.3. Adding Multiple Documents

Adds multiple documents to a collection. This command requires the argument pointing to the directory path that contains documents, which are going to be stored in the collection specified by collection context. The collection must already exist. Documents added will be assigned their file name as the document key. The optional "extension" parameter can be used to only import documents with a certain file extension.

#### Summary

```
xindice add_multiple_documents -c <context> [-l [-d <path>]]
[-v] [parameters...]
```

Command name abbreviation: addmultiple

#### Parameters

**-c <context>**

The collection context. Required.

**-f <path>**

Directory name to import documents from. Required.

**-e <extention>**

The file extention that files must have in order to be added to the collection.

**-l**

Use a local database rather then going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Add all files from directory /tmp/mydocs to the collection pebbles.

```
xindice add_multiple_documents -c /db/pebbles -f /tmp/mydocs
```

Add all files from directory /tmp/mydocs that have an extension ".xml" to the collection pebbles.

```
xindice add_multiple_documents -c /db/pebbles -f /tmp/mydocs -e xml
```

## 3.4. Deleting a Document

Deletes an existing document from a collection.

### Summary

```
xindice delete_document -c <context> [-l [-d <path>]] [-v]  
[parameters...]
```

Command name abbreviation: dd

### Parameters

**-c <context>**

The collection context. Required.

**-n <name>**

The key of the document to be deleted. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Delete a document with the key "stones" from the collection pebbles.

```
xindice delete_document -c /db/pebbles -n stones
```

### 3.5. Retrieving a Document

Retrieves an existing document from a collection and stores it in the file. If the path passed in does not exist, it will be created. If the file specified by path argument already exists, it will be overwritten.

#### Summary

```
xindice retrieve_document -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

Command name abbreviation: rd

#### Parameters

**-c <context>**

The collection context. Required.

**-n <name>**

The key of the document to be retrieved. Required.

**-f <file>**

The file path to store the document under. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

#### Examples

Retrieve a document with the key "stones" from the collection pebbles and save it in the file /tmp/stones.xml.

```
xindice retrieve_document -c /db/pebbles -n stones -f /tmp/stones.xml
```

### 3.6. List Resources

List all resource keys under the given collection context.

#### Summary

```
xindice list_documents -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

Command name abbreviation: ld

### Parameters

**-c <context>**

The collection context. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

List all documents in the collection pebbles.

```
xindice list_documents -c /db/pebbles
```

## 3.7. Importing a Directory Tree

Adds multiple documents into a collection, copying the directory structure into the database as collections. This command takes a directory and creates a collection with the same name. Directories under the directory will also be created as child collections, duplicating the directory tree. The files within the specified directory are converted into documents and then stored into the newly created collection with their filenames as the document keys. The optional "extension" parameter can be used to only import documents with a certain file extension.

### Summary

```
xindice import -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

### Parameters

**-c <context>**

The collection context. Required.

**-f <path>**

The path to a directory to be imported as a collection. Required.

**-e <extention>**

The file extention that files must have in order to be added to the collection.

**-l**

Use a local database rather then going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Import all directories and documents to the collection pebbles, starting with directory /tmp/flintstones.

```
xindice import -c /db/pebbles -f /tmp/flintstones
```

Import all directories and documents with an extension of ".xml" to the collection pebbles, starting with directory /tmp/flintstones.

```
xindice import -c /db/pebbles -f /tmp/flintstones -e xml
```

## 3.8. Exporting a Directory Tree

Creates an directory tree from a collection and its subcollections. The directory tree will be created starting at the directory passed in as an argument. That directory must exist already. The collection documents will be converted to files with the keys used as file names.

### Summary

```
xindice export -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

### Parameters

**-c <context>**

The collection context. Required.

**-f <path>**

The path to a directory where collection structure will be created. The directory must already exist. Required.

**-l**

Use a local database rather than going over the network. Should be combined with `-d` to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if `-l` is specified.

**-v**

Show verbose output, additional information and exceptions.

## Examples

Creating a directory tree identical to a collection pebbles tree in directory `/tmp/export`

```
xindice export -c /db/pebbles -f /tmp/export
```

## 4. Collection Indexer Actions

### 4.1. Adding a Collection Indexer

Adds a new collection index. In order to add the collection index, the collection it is being added to must already exist. This command has optional parameters for setting the index page size, index page count, and the index data type.

#### Summary

```
xindice add_indexer -c <context> [-l [-d <path>]] [-v]  
[parameters...]
```

Command name abbreviation: `ai`

#### Parameters

**-c <context>**

The collection context. Required.

**-n <name>**

The name of the indexer. Required.

**-p <pattern>**

The pattern used to create the index. Required, unless indexer configuration is specified. Name and Value indexers can only have one pattern, which is going to follow the `-p` switch. Full text indexer can have multiple patterns and its syntax is different, `-p` switch is followed by semicolon delimited list of patterns for indexer in the form `pattern=alias`.

**-t <type>**

The data type/index type.

**-f <file>**

Name of the file that holds indexer configuration. If specified, the rest of command-specific parameters (like indexer name) are ignored.

**--pagesize <pagesize>**

Page size for file pages (default: 4096). Does not apply to LuceneIndexer.

**--pagecount <pagecount>**

Number of pages in the primary storage (default: 1024). Does not apply to LuceneIndexer.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

Possible values of index type are:

Type	What is indexed	Indexer type
string	Non-normalized string value	ValueIndexer
trimmed	Normalized (whitespace stripped) string value	ValueIndexer
short	16-bit signed short integer	ValueIndexer
int	32-bit signed integer	ValueIndexer
long	64-bit signed integer	ValueIndexer
float	32-bit floating point value	ValueIndexer
double	64-bit floating point value (XPath number)	ValueIndexer
byte	8-bit signed byte	ValueIndexer
char	16-bit signed character	ValueIndexer
boolean	8-bit boolean value	ValueIndexer
name	Presence of the pattern in a document	NameIndexer
text	Pattern text content	LuceneIndexer



By default type will be "string" for element pattern or "trimmed" for attribute pattern.

### Examples

Create a Value indexer named "rockindex" in the pebbles collection and index value of a "rock" element as a string.

```
xindice add_indexer -c /db/pebbles -n rockindex -p rock
```

Create a Value indexer with name "hardindex" in the boulder collection for documents containing a rock elements with an attribute of "hard" and index value of an attribute "hard" of the "rock" element as normalized string.

```
xindice add_indexer -c /db/pebbles/boulder -n hardindex -p rock@hard
```

Create a Value indexer named "weight" in the pebbles collection and index value of an attribute "weight" of the "rock" element as a floating point value.

```
xindice add_indexer -c /db/pebbles -n weight -p rock@weight -t float
```

Create a Name indexer named "type" in the pebbles collection and index document keys of the documents that contain "rock" elements with an attribute of "type".

```
xindice add_indexer -c /db/pebbles -n type -p rock@type -t name
```

Create a Lucene indexer named "text" in the pebbles collection and index text content of element "origin" (with alias "origin") and text of attribute "donatedby" of the element "location" (with alias "donated").

```
xindice add_indexer -c /db/pebbles -n text -p  
origin=origin;location@donatedby=donated -t text
```

## 4.2. Deleting a Collection Indexer

Deletes a collection index referred by the name. Deleting an index does not affect collection and its documents.

### Summary

```
xindice delete_indexer -c <context> [-l [-d <path>]] [-v]  
[parameters...]
```

Command name abbreviation: di

**Parameters****-c <context>**

The collection context. Required.

**-n <name>**

The name of the indexer being deleted. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

**Examples**

Delete index with name "hardindex" in the pebbles collection.

```
xindice delete_indexer -c /db/pebbles -n hardindex
```

**4.3. Listing Collection Indexers**

Lists all indexers of a collection.

**Summary**

```
xindice list_indexers -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

Command name abbreviation: li

**Parameters****-c <context>**

The collection context. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

## Examples

To list all collection indexes in the pebbles collection

```
xindice list_indexers -c /db/pebbles
```

## 5. Querying a Collection

### 5.1. Xpath Query

Execute XPath query against a collection. The query is executed for all documents in the collection.

#### Summary

```
xindice xpath -c <context> [-l [-d <path>]] [-v]  
[parameters...]
```

#### Parameters

**-c <context>**

The collection context. Required.

**-q <query>**

XPath query to execute. Required.

**-s <namespaces>**

Semicolon delimited list of namespaces for query in the form prefix=namespace-uri.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

#### Examples

Run XPath query against all documents in the pebbles collection and find all "rock" elements that have the "type" attribute equals to "hard".

```
xindice xpath -c /db/pebbles -q "/rock[@type='hard']"
```

Run Xpath query against all documents in the pebbles collection and find all "rock" elements that have the "type" attribute equals to "hard" when the rock element is in the namespace `http://www.bedrock.com`.

```
xindice xpath -c /db/pebbles -s "br=http://www.bedrock.com" -q
"/br:rock[@type='hard']"
```

## 5.2. XUpdate Query

Execute XUpdate query against a document or a collection.

### Summary

```
xindice xupdate -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

### Parameters

**-c <context>**

The collection context. Required.

**-f <file path>**

Name of the file which holds XUpdate query. Required.

**-n <name>**

Document key, if query to be run against single document.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Run XUpdate query from file `/path/to/xupdate.xml` against all documents in the test collection.

```
xindice xupdate -c /db/test -f /path/to/xupdate.xml
```

Run XUpdate query from file `/path/to/xupdate.xml` against the document with the key "document-to-update.xml" in the test collection.

```
xindice xupdate -c /db/test -n document-to-update.xml -f
/path/to/xupdate.xml
```

### 5.3. Full Text Query

Execute Lucene query against a collection. This type of query requires existing LuceneIndexer for the collection.

#### Summary

```
xindice xpath -c <context> [-l [-d <path>]] [-v]
[parameters...]
```

#### Parameters

**-c <context>**

The collection context. Required.

**-q <query>**

Lucene query to execute. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

#### Examples

Find documents that have word 'Xindice' and do not have word 'XPath' in the text content of "title" element in the book collection, assuming that element "title" is included in LuceneIndexer with alias "chaptertitle".

```
xindice xpath -c /db/pebbles -q "+chaptertitle:Xindice
-chaptertitle:XPath"
```

## 6. Miscellaneous Actions

### 6.1. Shutting down the Server

Shuts down the Xindice server. Shutdown command only works with embedded driver.

Usually, this command is unnecessary, because all other commands will execute shutdown command after their main work has been done, if embedded driver is used. For XML-RPC driver, command can be executed, but it does not do anything, as the lifecycle is managed by application container (such as Tomcat).

### Summary

```
xindice shutdown -c <context> [-l [-d <path>]] [-v]
```

### Parameters

**-c <context>**

The collection context. Required.

**-l**

Use a local database rather than going over the network. Should be combined with -d to specify the configuration to use.

**-d <path>**

Path to the database configuration to use for the local database. Only applies if -l is specified.

**-v**

Show verbose output, additional information and exceptions.

### Examples

Shutdown the server

```
xindice shutdown -c /db
```

## 6.2. Accessing Help

Show available commands

### Summary

```
xindice --help
```

Command name abbreviations: -h, help