

# The navigation of BaseX databases

## Principles

The resource navigation of FOXPath cannot only be applied to the physical file system, but also to the virtual file system exposed by **BaseX databases**. URIs referring to BaseX databases or their resources (files and folders) must be prefixed with `basex://`. URI pattern:

```
basex://databaseName/path/to/file/or/folder
```

For example, the URI

```
basex://yogi02/frameworks/xforms/catalog.xml
```

references a document in database `yogi02` at path `frameworks/xforms/catalog.xml`. Similarly, the URI

```
basex://yogi02/frameworks/xforms
```

references a folder in database `yogi02` at path `frameworks/xforms`.

Special cases:

```
basex://
```

references the virtual root folder whose child folders represent databases.

```
basex://foo
```

references the database `foo`.

Navigation of BaseX databases and navigation of the physical file system are governed by the same rules of syntax and semantics. For example, the expression

```
basex://yogi02/frameworks/xforms/*.xsd[file-sdate >= '2016-01-01']
```

selects all XSD documents found in database `yogi02`, in its folder `frameworks/xforms`, with a last modification date greater or equal to 2016-01-01.

## Examples

A few examples should convey a feeling what FOXPath navigation of the BaseX databases is like. For more examples, see `foxpath-intro.pdf`. Although the examples in that article navigate the physical file system, they can be also used as models how to navigate BaseX databases: folders and files are treated exactly the same way, no matter if they are file system or database resources.

```
basex://*
```

Value: A list of all BaseX databases.

```
basex://y*
```

Value: A list of all BaseX databases whose name starts with “y” (case insensitive).

```
basex://yogi02/*
```

Value: Top-level files and folders in the database `yogi02`.

```
basex://yogi02/**[is-dir()]
```

Value: All folders in the database `yogi02`.

```
basex://yogi02/**[is-file()][file-sdate() lt '2016-11-07']
```

Value: All files in the database `yogi02`, which have a last modification date less than “2016-11-07”.

```
basex://yogi02/**/*.xsd
```

Value: All XSDs in the database `yogi02`.

```
basex://yo*/**/*.xsd
```

Value: All XSDs in databases whose name starts with `yo` (case insensitive).

```
basex://yogi02//frameworks/**/*.xsd
```

Value: Alls XSDs in database `yogi02` which are directly or indirectly contained by folder `frameworks`.

```
basex://yogi02//frameworks/(mathml, ooxml)/**/*.xsd
```

Value: Alls XSDs in database `yogi02` which are directly or indirectly contained by framework `mathml` or `ooxml`.

```
basex://yogi02//frameworks/mathml/(* except ~2.0)/**/*.xsd
```

Value: Alls XSDs in database `yogi02` which are directly or indirectly contained by a child folder of framework `mathml`, excluding the contents of child folder `2.0`. (Note the `~` character which escapes a digit appearing as the first character of a folder name.)

```
basex://yogi02//frameworks/mathml/**/*.xsd*\@targetNamespace  
=> distinct-values() => sort()
```

Value: A sorted list of all target namespaces used by XSDs of the `mathml` framework in the database `yogi02`. This example shows that the navigation of folders and files can be combined with navigation of the node trees which they contain.

For more examples, see article `foxpath-intro.pdf`. Although those examples navigate the physical file system, rather than BaseX databases, there is no difference except for the `basex://` prefix of URIs which refer to BaseX database resources.

### Tip – creation and use of database document catalogs

A selection of database documents can be translated into a **catalog document** (a “dcat”). XQuery code may use such a catalog in order to accomplish access to all selected documents. The creation of a catalog document is accomplished by FOXpath function `dcat`.

Example: the expression

```
basex://yogi02//frameworks//*.sch => dcat()
```

creates a catalog describing all schematron documents found in the frameworks of database `yogi02`. The catalog looks like this:

```
<dcat targetFormat="xml" count="61"
      t="2016-11-07T23:37:48.073+01:00"
      onlyDocAvailable="false">
  <doc href="yogi02/frameworks/dita/DITA-OT/plugins/org.dita-ng.doctypes/checkShell.sch"/>
  <doc href="yogi02/frameworks/dita/DITA-OT2.x/plugins/org.dita-ng.doctypes/checkShell.sch"/>
  <doc href="yogi02/frameworks/dita/resources/dita-1.2-for-xslt2-links-checker.sch"/>
  ...
</dcat>
```

Such a catalog enables XQuery code to access the complete set of selected documents very elegantly:

```
...
let $dcat := 'dcat.xml'
let $docs := doc($dcat)//@href/doc(.)
...
```