XSpec

Release v2.2.4

cirulls

Dec 31, 2021

CONTENTS

XSpec is a unit test and behaviour-driven development (BDD) framework for XSLT, XQuery, and Schematron.

Note: This project is under active development.

CHAPTER

ONE

CONTENTS

1.1 What is XSpec

XSpec is a unit test and behaviour-driven development (BDD) framework for XSLT, XQuery, and Schematron. It is based on the Spec framework of RSpec, which is a BDD framework for Ruby.

XSpec consists of a syntax for describing the behaviour of XSLT, XQuery, or Schematron code, and some code that enables to test code against those descriptions.

1.1.1 What is BDD?

BDD is like Test Driven Development (TDD) in that it encourages to develop code by

- describing a behaviour (or writing a test)
- testing whether the code gives that behaviour
- if it doesn't, fixing the code until it does and the test passes

The difference between BDD and TDD is about how to write the tests. In BDD, the focus is on the **behaviour of the code**: the descriptions form the double role of both a **human-readable documentation** of what the code should do and runnable tests that can test whether the code does what it should do.

In BDD, the focus in on **scenarios** and the expected behaviour of the application with these scenarios. Scenarios fit particularly well with XSLT's source-driven (or template-driven) approach. For example, a scenario might be:

> when processing a `para` element, it should create a `p` element

or something more complex like:

> when processing a `fn` element with a `label` attribute in `footnote` mode,
> it should create a `p` element with a `sup` child holding the value of the `label` attribute

Scenarios written like this naturally map onto XSLT templates.

1.2 Installation

1.2.1 Requirements

Java

XSpec requires Java Runtime Environment.

XSpec is tested on Java 8 (recommended) and 11 (OpenJDK).

Saxon

XSpec requires Saxon XSLT processor running on Java.

Saxon can be downloaded from:

- Saxon HE (open source)
- Saxon PE/EE (commercial products)

XSpec is tested on Saxon 10, 9.9 (recommended) and 9.8 (deprecated).

Saxon 9.8 is deprecated

XSpec is partially tested on Saxon 9.8.

XSpec v2.3 will stop supporting Saxon 9.8 at all.

XQuery processor

Testing XQuery requires an XQuery 3.1 processor. XSpec is tested with the following XQuery processors:

- Saxon
- BaseX

1.2.2 Installation on Mac and Linux

Requirements

Make sure requirements are complete.

Installation

For this installation the Saxon JAR (e.g. `saxon9he.jar`) is assumed to be in `~/saxon`.

1. Extract the source code archive file of the latest `master` branch, or alternatively clone the `master` branch of XSpec from GitHub:

   ```
   git clone https://github.com/xspec/xspec.git
   ```

   For this installation XSpec is assumed to be extracted or cloned to `~/xspec`.

2. Set up the Saxon environment variable. You can either set `SAXON_CP` to the full classpath containing the Saxon jar file:

   ```
   export SAXON_CP=~/saxon/saxon9he.jar
   ```

   or alternatively set `SAXON_HOME` to the location of the Saxon jar file:

   ```
   export SAXON_HOME=~/saxon
   ```

   You can check that the environment variable is set with:

   ```
   echo $SAXON_CP
   ```

   `SAXON_CP` has precedence over `SAXON_HOME`.

   If you want to make the change permanent, set the environment variable in your shell profile (e.g. `~/.bashrc` or `~/.bash_profile`).

3. Open up a terminal, navigate to `~/xspec`, make sure that the file `bin/xspec.sh` is executable, and test the shell script with this command:

   ```
   bin/xspec.sh -h
   ```

   The output should be the following usage summary:

   ```
   Usage: xspec [-t|-q|-s|-c|-j|-catalog file|-h] file

     file           the XSpec document
     -t             test an XSLT stylesheet (the default)
     -q             test an XQuery module (mutually exclusive with -t and -s)
     -s             test a Schematron schema (mutually exclusive with -t and -q)
     -c             output test coverage report (XSLT only)
     -j             output JUnit report
     -catalog file  use XML Catalog file to locate resources
     -e             treat failed tests as error
     -h             display this help message
   ```

Congratulations! You have successfully installed XSpec!

To make the XSpec script more portable and invoke it from anywhere, add a script invoking `xspec.sh` in `/usr/bin`. To move `xspec.sh` out of the XSpec installation directory, you must set the environment variable `XSPEC_HOME` to the location where XSpec is stored.

1.2.3 Installation on Windows

Requirements

Make sure requirements are complete.

Installation

For this installation the Saxon JAR (e.g. `saxon9he.jar`) is assumed to be in `C:\saxon`.

1. Extract the source code archive file of the latest `master` branch, or alternatively clone the `master` branch of XSpec from GitHub:

   ```
   git clone https://github.com/xspec/xspec.git
   ```

   For this installation XSpec is assumed to be extracted or cloned to `C:\xspec`.

2. Open up a command prompt and set up the Saxon environment variable. You can either set `SAXON_CP` to the full classpath containing the Saxon jar file:

   ```
   set SAXON_CP=C:\saxon\saxon9he.jar
   ```

   or alternatively set `SAXON_HOME` to the location of the Saxon jar file:

   ```
   set SAXON_HOME=C:\saxon
   ```

   You can check that the environment variable is set with:

   ```
   set SAXON_
   ```

   ```
   C:\>set SAXON_
   SAXON_CP=C:\saxon\saxon9he.jar
   ```

   `SAXON_CP` has precedence over `SAXON_HOME`.

   If you want to make the change permanent, set the environment variable in System Control Panel (or `setx command`_).

3. Test the batch script with this command:

   ```
   C:\xspec\bin\xspec.bat -h
   ```

   The output should be the following usage summary:

   ```
   Usage: xspec [-t|-q|-s|-c|-j|-catalog file|-h] file

     file           the XSpec document
     -t             test an XSLT stylesheet (the default)
     -q             test an XQuery module (mutually exclusive with -t and -s)
     -s             test a Schematron schema (mutually exclusive with -t and -q)
     -c             output test coverage report (XSLT only)
     -j             output JUnit report
     -catalog file  use XML Catalog file to locate resources
     -e             treat failed tests as error
     -h             display this help message
   ```

Congratulations! You have successfully installed XSpec!

To make the XSpec script more portable and invoke it from anywhere, add `C:\xspec\bin` to your system path. To move `xspec.bat` out of the XSpec installation directory, you must set the environment variable `XSPEC_HOME` to the location where XSpec is stored.