Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'jjs.1' command

$ man jjs.1

jjs(1) Basic Tools jjs(1)

NAME

jjs - Invokes the Nashorn engine.

SYNOPSIS

jjs [options] [script-files] [-- arguments]

options

One or more options of the jjs command, separated by spaces. For more information, see Options.

script-files

One or more script files which you want to interpret using Nashorn, separated by spaces. If no files are specified, an interactive shell is started.

arguments

All values after the double hyphen marker (--) are passed through to the script or the interactive shell as arguments. These values can be accessed by using the arguments property (see Example 3).

DESCRIPTION

The jjs command-line tool is used to invoke the Nashorn engine. You can use it to interpret one or several script files, or to run an interactive shell.

OPTIONS

The options of the jjs command control the conditions under which scripts are interpreted by Nashorn.

-cp path

-classpath path

Specifies the path to the supporting class files To set multiple paths, the option can be repeated, or you can separate each path with a colon (:).

-Dname=value

Sets a system property to be passed to the script by assigning a value to a property name. The following example shows how to invoke Nashorn in interactive mode and assign myValue to the property named myKey:

```
>> jjs -DmyKey=myValue
jjs> java.lang.System.getProperty("myKey")
myValue
jjs>
```

This option can be repeated to set multiple properties.

-doe

--dump-on-error

Provides a full stack trace when an error occurs. By default, only a brief error message is printed.

-fv

--fullversion

Prints the full Nashorn version string.

-fx

Launches the script as a JavaFX application.

-h

-help

Prints the list of options and their descriptions.

--language=[es5]

Specifies the ECMAScript language version. The default version is ES5.

-ot

--optimistic-types=[true|false]

Enables or disables optimistic type assumptions with deoptimizing

recompilation. Running with optimistic types will yield higher final speed, but may increase warmup time.

-scripting

Enables shell scripting features.

-strict

Enables strict mode, which enforces stronger adherence to the standard (ECMAScript Edition 5.1), making it easier to detect common coding errors.

-t=zone

-timezone=zone

Sets the specified time zone for script execution. It overrides the time zone set in the OS and used by the Date object.

-v

-version

Prints the Nashorn version string.

EXAMPLES

Example 1 Running a Script with Nashorn

```
jjs script.js
```

Example 2 Running Nashorn in Interactive Mode

```
>> jjs
jjs> println("Hello, World!")
Hello, World!
jjs> quit()
>>
```

Example 3 Passing Arguments to Nashorn

```
>> jjs -- a b c
jjs> arguments.join(", ")
a, b, c
jjs>
```

SEE ALSO

jrunscript