

Ruby master - Bug #16962

IO.close behaviour

06/16/2020 11:54 AM - ioquatix (Samuel Williams)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN

Description

I discussed this with [Eregon \(Benoit Daloze\)](#) and I think the goal here is to try and figure out a way these interfaces can be a bit less confusing.

1. I don't understand this behaviour:

```
STDOUT.close
STDOUT.puts "Hello World"
# => closed stream
```

vs

```
IO.for_fd(STDOUT.fileno, autoclose: true).close
STDOUT.puts "Hello World"
# => Hello World
```

2. IO.for_fd(..., autoclose: true/false)

The documentation for autoclose is:

If the value is false, the fd will be kept open after this IO instance gets finalized.

But it also seems to affect #close - i.e. calling close does not close underlying file descriptor.

Should we fix the documentation or is the implementation wrong? Maybe the name autoclose: is very confusing. My initial interpretation was it was just 'automatically close this I/O when it is garbage collected'.

3. IO.for_fd(..., autoclose: false) default

In most cases, it seems like autoclose: false would make more sense as the default, since the file descriptor must come from some other place.

History

#1 - 06/16/2020 11:55 AM - ioquatix (Samuel Williams)

- Description updated

#2 - 06/16/2020 11:57 AM - ioquatix (Samuel Williams)

- Description updated