

SPOOL - the forgotten device

Article

[Robert Cemper](#) · Sep 2, 2017



2m read

[Open Exchange](#)

SPOOL - the forgotten device

Hi all,

In recent discussions, answers, comments IO redirection ([with the famous code](#)) was quite often seen as the ultimate solution. It is definitely strong stuff and you can not only log your output but also do key-grabbing in input.

But I think for simple logging it's quite a significant overhead.

If you just need to get some text or dump your variables or objects then SPOOL should be sufficient for your needs.

The principle goes back to the roots and some utilities as `^%IS,^%SPOOL,%SPL*` written in quite ancient style are more interesting for programming archeology than really useful.

Online documentation on [SPOOL](#) is complete and tells you almost all details.

For daily use you should know

- Device = 2
- it has an integer documentId (default=1)
- it has an integer lineNumber (default =1)

You start it by `OPEN 2:documentId USE 2` and your redirection is done.

All further `WRITES` now end up in Global `^SPOOL(documentId,lineNumber)`

and when you are finished or your job dies `CLOSE 2` completes the redirection.

Or you return to your primary device by `USE $P` and keep the spool open for other output.

If you want to [append](#) to an existing document you have to get next line to write as start for the OPEN

```
set append=$s($o(^SPOOL(documentId,""),-1,val):$p(val,"",-1),1:0)
OPEN 2(documentId:append) USE 2
```

The SPOOL DEVICE ignores any `READ` and returns ""

So the easiest de-spooling - just lists the global:

```
FOR i=1:1 QUIT:'$DATA(^SPOOL(documentId,i),val) WRITE val
```

Find attached a small summary of all you need to use SPOOL for debugging:

SpoolDemo ;Essentials for using SPOOL device in debugging

```
;  
;; if we have no idea on free docId  
New()  
  SET docId=$INCREMENT(^SPOOL)  
  QUIT $$Open(docId)  
;  
;; we insist in a docId  
Open(docId)  
  DO Clean(docId)  
  OPEN 2:docId  
  USE 2  
  QUIT docId  
;  
;;continue to write to docId  
App(docId)  
  
SET append=$SELECT($ORDER(^SPOOL(docId,""),-1,val):  
$PIECE(val,"{",-1),1:0)  
  IF 'append QUIT $$Open(docId)  
  OPEN 2:(docId:append)  
  USE 2  
  QUIT docId  
;  
;; dump spooled output  
DeSpool(docId)  
  FOR i=1:1 QUIT:$DATA(^SPOOL(docId,i),val) WRITE val  
  WRITE !,"*** done ***",!  
  QUIT docId  
;  
;; clean Spool  
Clean(docId)  
  KILL ^SPOOL(docId)  
  QUIT
```

SPOOL - the forgotten device

Published on InterSystems Developer Community (<https://community.intersystems.com>)

This is a coding example working on Caché 2018.1.3 and IRIS 2020.2
It will not be kept in synch with new versions
It is also **NOT** serviced by InterSystems **Support** !

[#Caché](#) [#InterSystems](#) [IRIS](#)

[Check the related application on InterSystems Open Exchange](#)

170 2 0 0 669

Related posts

- SPOOL - the forgotten device
- [SPOOL as SQL Table](#)

Log in or sign up to continue
Add reply

Source URL: <https://community.intersystems.com/post/spool-forgotten-device>