

JOD Best Practices Lab

December 15, 2021

1 JOD Best Practices Lab



1.0.1 Introduction

Software tools are like loaded guns: powerful weapons for slaying your coding demons but also dangerous when used improperly. Have you ever shot yourself in the foot? I know I'm missing a few toes and I bet you are as well.

This lab outlines a number of “best practices” or guidelines for using JOD. I’ve learned the hard way; if you take my advice to heart you *might* be spared!

```
[1]: NB. show J version
9!:14 ''
```

```
j903/j64avx2/windows/beta-w/commercial/www.jssoftware.com/2021-12-05T18:25:00/cloa
ng-13-0-0/SLEEP=1
```

```
[2]: load 'general/jod'
```

```

NB. use portable box drawing characters
portchars ''

NB. Verb to show large boxed displays
NB. in notebook without ugly wrapping
sbx=: ' ... ' , "1~ 73&{."1@":

NB. Verb to show first 12 lines of character
NB. lists in notebook without wrapping
tlf=: ] , ((10{a.)"_ = {:) }. (10{a.)"_
stx=: {{sbx 12 {. ];._2 tlf y -. CR}}

NB. show JOD version
smoutput JODVMD_ajod_

```

```

+-----+-----+
|1.0.23 - dev|1|14 Dec 2021 10:09:40|
+-----+-----+

```

1.0.2 Test for JOD user defined folders

WARNING: THIS LAB REQUIRES A NUMBER OF USER DEFINED JOD FOLDERS.

This lab requires a number of JOD folders to run. The next step tests for the existence of these folders.

If (TestJODDirectories) lists any undefined JOD directories configure these directories before running this lab. [Instructions on how to do this](#) can be found in (jod.pdf). Install the [joddocument](#) addon to get (jod.pdf).

```
[3]: TestJODDirectories_ajod_=3 : 0
```

```

NB.*TestJODDirectories v-- test user configured JOD directories.
NB.
NB. This verb checks that required JOD lab directories are
NB. defined. "Defined" does not mean the directories exist only
NB. that (jpath) expands to something other than its default.

```

```

NB.
NB. monad:  clMsg =. TestJODDirectories uuIgnore

NB. when a relative directory does not exist (jpath) echoes its argument
jodudirs =. '~'&,&.> ;:'JOD JODDUMPS JODSOURCE JODTEST'
jodudefs =. jpath&.> jodudirs  NB. !(*)=. jpath
mask =. jodudirs = jodudefs
if. 1 e. mask do.
  > 'missing JOD folders - define before running JOD labs' ; mask # jodudirs
else.
  > 'all JOD folders are defined' ; jodudefs
end.
)

TestJODDirectories 0

```

```

all JOD folders are defined
c:/users/john.baker/onedrive - jackson companies/jod
c:/users/john.baker/onedrive - jackson companies/jod/joddumps
c:/jodtest/labtesting
c:/jodtest/test

```

1.0.3 Lab cleanup

WARNING: DICTIONARIES CREATED BY PRIOR RUNS OF LAB THIS WILL BE DELETED IN THE NEXT STEP.

When the Best Practices lab runs it creates a number of dictionaries in the (~JODSOURCE) directory. The next step will remove any of these dictionaries. **IF YOU CARE ABOUT THESE DICTIONARIES STOP NOW.**

```

[4]: RemoveLabBestDictionaries_iod=: 3 : 0
if. IFWIN do.
  shell 'rd /s /q "',(jpath '~JODSOURCE'),' '
  smoutput 'Lab temporary best practices (win) dictionaries erased'
elseif. IFUNIX do.
  NB. avoid blanks in Mac and Linux paths

```

```

shell 'rm -rf ',jpath '~JODSOURCE'
smoutput 'Lab temporary best practices (mac/linux) dictionaries erased'
elseif.do.
smoutput 'Erase any previous temporary best practices dictionaries manually.'
end.
)

```

Remove Best Practices lab dictionaries.

```

[5]: NB. clear any previously opened dictionaries in master file
dpset 'RESETME'

NB. unregister dictionaries - IGNORE ERRORS
3 regd> ;:'bpcopy bptest' [ 3 od ''

NB. delete dictionary files
RemoveLabBestDictionaries 0

```

Lab temporary best practices (win) dictionaries erased

1.0.4 JOD does not belong in the J tree

My first and most important bit of advice is simply:

NEVER NEVER NEVER store your JOD dictionaries in J install directories!

I would also avoid any OS managed directories like

c:\Program Files\..

Create a JOD master dictionary directory root that is completely independent of J. It's also a good idea to define a subdirectory structure that mirrors J's versions. JOD creates binary jfiles. These files are fairly stable but binaries can change when J changes.

```

[6]: NB. create a master JOD directory root outside of J's directories.
NB. This creation depends on a configured directory (~JODSOURCE).
smoutput newd 'bptest';(jpath '~JODSOURCE/bptest');'best practices test dictionary'

```

NB. This is an example - use another root for your dictionaries.

```
+--+-----+-----+-----+
|1|dictionary created ->|bptest|c:/jodtest/labtesting/bptest/|
+--+-----+-----+-----+
```

1.0.5 Load lab dictionary from dump script

To illustrate key features of JOD we need a nonempty dictionary. This next step loads the (bptest) dictionary from a dump script distributed with JOD.

```
[7]: NB. open new dictionary
od 'bptest' [ 3 od ''

NB. load from example dump script
O!:0 <jpath '~addons/general/jod/jodlabs/labdump.ijs'

NB. regenerate all references - show last three messages
_3 {. 0 globs&> }. revo''
```

```
+--+-----+-----+
|1|1 word(s) put in ->|bptest|
+--+-----+-----+
+--+-----+-----+
|1|35 word(s) put in ->|bptest|
+--+-----+-----+
+--+-----+-----+
|1|36 word explanation(s) put in ->|bptest|
+--+-----+-----+
+--+-----+-----+
|1|2 word document(s) put in ->|bptest|
+--+-----+-----+
+--+-----+-----+
|1|group <bstats> put in -> |bptest|
+--+-----+-----+
```

```
|1|group <sunmoon> put in ->|bptest|
+-+-----+-----+
NB. end-of-JOD-dump-file regenerate cross references with: 0 globs&> }. rev0 ''
+-+-----+-----+
|1|<var> references put in ->          |bptest|
+-+-----+-----+
|1|<yeardates> references put in ->    |bptest|
+-+-----+-----+
|1|<NORISESET> is a noun - no references|
+-+-----+-----+
```

1.0.6 Backup backup backup

A wise man once said, “You’re either backed up or f&%ked up!”

Care to go over the options again?

It is literally a snap to make a backup with JOD. So backup often!

```
[8]: NB. open the best practice dictionary
od 'bptest' [ 3 od ''

NB. back it up
smoutput packd 'bptest'

NB. restd recovers the most current backup
restd 'bptest'
```

```
+-+-----+-----+
|1|dictionary packed ->|bptest|0|
+-+-----+-----+
+-+-----+-----+
|1|dictionary restored ->|bptest|0|
+-+-----+-----+
```

1.0.7 Take a script dump

(packd) backs up binary jfiles but it's also a good idea to “dump” your dictionaries as plain text. JOD can dump all open dictionaries as a single J script. Script dumps are the most stable way to store J dictionaries. The jodsource addon distributes all JOD source code in this form.

```
[9]: NB. dump only (bptest)
      od 'bptest' [ 3 od ''

      NB. (make) creates a dictionary dump in the dump subdirectory
      bptestdump=: showpass_ajod_ make ''
```

```
+--+-----+
|1|object(s) on path dumped ->|c:/jodtest/labtesting/bptest/dump/bptest.ijs|
+--+-----+
```

1.0.8 Some uses of dump scripts

JOD dump scripts can be used to reload, copy and merge dictionaries.

```
[10]: NB. reload bptest
      od 'bptest' [ 3 od ''
      0!:0 {:bptestdump
```

```
+--+-----+
|1|1 word(s) put in ->|bptest|
+--+-----+
+--+-----+
|1|35 word(s) put in ->|bptest|
+--+-----+
+--+-----+
|1|36 word explanation(s) put in ->|bptest|
+--+-----+
+--+-----+
|1|2 word document(s) put in ->|bptest|
+--+-----+
```

```

+-+-----+-----+
|1|group <bstats> put in -> |bptest|
+-+-----+-----+
|1|group <sunmoon> put in ->|bptest|
+-+-----+-----+
+-+-----+-----+
|1|dictionary document updated ->|bptest|
+-+-----+-----+

```

NB. end-of-JOD-dump-file regenerate cross references with: 0 globs&> }. revo ''

Copy a dictionary.

```

[11]: NB. copy/merge bptest dictionary
smoutput newd 'bpcopy';jpath '~JODSOURCE/bpcopy'
od 'bpcopy' [ 3 od ''
0!:0 {:bptestdump

NB. clear path
dpset 'CLEARPATH'

```

```

+-+-----+-----+-----+
|1|dictionary created ->|bpcopy|c:/jodtest/labtesting/bpcopy/|
+-+-----+-----+-----+
+-+-----+-----+
|1|1 word(s) put in ->|bpcopy|
+-+-----+-----+
+-+-----+-----+
|1|35 word(s) put in ->|bpcopy|
+-+-----+-----+
+-+-----+-----+
|1|36 word explanation(s) put in ->|bpcopy|
+-+-----+-----+
+-+-----+-----+
|1|2 word document(s) put in ->|bpcopy|

```



```

+-+-----+-----+
+-+-----+-----+
|1|group <bstats> put in -> |bpcopy|
+-+-----+-----+
|1|group <sunmoon> put in ->|bpcopy|
+-+-----+-----+
+-+-----+-----+
|1|dictionary document updated ->|bpcopy|
+-+-----+-----+
NB. end-of-JOD-dump-file regenerate cross references with: 0 globs&> }. revo ''
+-+-----+-----+
|1|path cleared ->|bpcopy|
+-+-----+-----+

```

1.0.9 Dump scripts are the best way to share and version control dictionaries

Dump scripts can be used to share and version control dictionaries. See this [Git repository](#) for examples.

[Example JOD Dump Script Repository](#)

1.0.10 Make a master re-register script

JOD only sees the dictionaries registered in the `jmaster.ijf` file so maintaining a list of registered dictionaries is a good idea. JOD can generate a re-register script that you can edit.

Generate a re-register script and put it in your main JOD dictionary directory root.

```

[12]: NB. generate re-register script
      rereg=: ;{: 5 od ''

      NB. save it in the master root
      rereg write_ajod_ jpath '~JODSOURCE/jodregister.ijs'

```

1.0.11 Set library dictionaries to READONLY

Open JOD dictionaries define a search path. The first dictionary on the path is the only dictionary that can be changed. It is called the “put” dictionary. Even though nonput dictionaries cannot be changed by JOD it’s a good idea to set them READONLY because:

READONLY dictionaries can be accessed by any number of JOD tasks. READWRITE dictionaries can only be accessed by one task.

Keeping libraries READONLY prevents accidental put’s as you open and close dictionaries.

```
[13]: NB. make bptest READONLY
      od 'bptest' [ 3 od ''
      dpset 'READONLY'

      NB. bpcopy is now the put dictionary
      od ;:'bpcopy bptest' [ 3 od ''

      NB. first group/suite sets path
      grp 'agroup'; ;:'datecheck yeardates today sunriseset0'

      NB. note dictionary path
      did ~ 0
```

```
+--+-----+
|1|+-----+--+-----+-----+-----+-----+-----+-----+
| ||      |--|Words|Tests|Groups*|Suites*|Macros|Path*      ||
| |+-----+--+-----+-----+-----+-----+-----+-----+
| ||bpcopy|rw|36  |0   |3     |0     |0     |/bpcopy/bptest||
| |+-----+--+-----+-----+-----+-----+-----+-----+
| ||bptest|ro|36  |0   |2     |0     |0     |/bptest      ||
| |+-----+--+-----+-----+-----+-----+-----+-----+
+--+-----+
```

1.0.12 Keep references updated

JOD stores word references. References enable many useful operations. References allow (getrx) to load words that call other words in new contexts.

```
[14]: NB. only put dictionary references need updating - show last five messages
      _5 {. 0 globs&> }. revo ''
```

```
+--+-----+-----+
|1|<tan> references put in ->      |bpcopy|
+--+-----+-----+
|1|<today> references put in ->    |bpcopy|
+--+-----+-----+
|1|<var> references put in ->      |bpcopy|
+--+-----+-----+
|1|<yeardates> references put in -> |bpcopy|
+--+-----+-----+
|1|<NORISESET> is a noun - no references|
+--+-----+-----+
```

1.0.13 Document dictionary objects

Documentation is a long standing sore point for programmers. Most of them hate it. Some claim it's unnecessary and even distracting. Others put in half baked efforts. In my opinion this "isn't even wrong!" Good documentation elevates code. In [Knuth's opinion](#) it separates "literate programming" from the alternative - "illiterate programming."

JOD provides a number of easy ways to document code. You can enter a single sentence or a large dissertation. I would recommend the former. See JOD documentation for more documentation options.

```
[15]: NB. for new words try a single line of documentation.
      afterlaststr=:] }.~ #@[ + 1&(i:~)@([ E. ])
      put 'afterlaststr'

      NB. insert sentence
      0 8 put 'afterlaststr';'retains string (y) after last occurrence of (x)'
```

```
+--+-----+-----+
|1|1 word explanation(s) put in ->|bpcopy|
+--+-----+-----+
```

JOD uses documentation in new contexts.

```
[16]: NB. for tacits the document sentence is fetched
smoutput disp 'afterlaststr'

NB. briefly describe sunmoon group - the payback for entering those sentences
sbx hlpnl }. grp 'sunmoon'
```

```
NB. retains string (y) after last occurrence of (x)
afterlaststr=:] }.~ #@[ + 1&(i:~)@([ E. ])
+-----+-----+-----+-----+-----+-----+-----+-----+
|NORISESET |indicates sun never rises or sets in (sunriseset0) and (sunr ...
|arctan    |arc tangent ...
|calmoons  |calendar dates of new and full moons ...
|cos       |cosine radians ...
|datecheck |checks dates in YYYY MM DD format ...
|fromjulian|converts Julian day numbers to dates, converse (tojulian) ...
|moons     |times of new and full moons for n calendar years ...
|round     |round y to nearest x (e.g. 1000 round 12345) ...
|sin       |sine radians ...
|sunriseset0|computes sun rise and set times - see long documentation ...
|sunriseset1|computes sun rise and set times - see long documentation ...
|tabit     |promotes only atoms and lists to tables ...
|tan       |tan radians ...
|today     |returns today's date ...
|yeardates |returns all valid dates for n calendar years ...
+-----+-----+-----+-----+-----+-----+-----+-----+

```

An example of long documentation.

```
[17]: NB. long document
stx 0 9 disp 'sunriseset0'
```

```
*sunriseset0 v-- sunrise and sunset times. ...
...
This verb has been adapted from a BASIC program submitted by ...
Robin G. Stuart Sky & Telescope's shortest sunrise/set ...
```

```

program contest. Winning entries were listed in the March ...
1995 Astronomical Computing column. ...
...
The J version of this algorithm has been vectorized. It can ...
compute any number of sunrise and sunset times in one call. ...
...
NB. verbatim: ...
...
```

1.0.14 Define your own JOD shortcuts

JOD words can be used within arbitrary J programs. If you don't find a JOD primitive that meets your needs do a little programming. There are many examples of JOD programming in JOD's source code. Install the jodsource addon to get JOD source code.

```

[18]: NB. examples of using JOD words to define new facilities

NB. describe a JOD group
hg_ijod=: [: hlpnl [: }. grp

NB. re-reference put dictionary show any errors
reref_ijod=: 3 : '(n,.s) #~ -.;0{"1 s=.0 globs&>n=.}.revo''' [ y'

NB. show words referenced by words in a group that are not in the group
jodg_ijod=: 'agroup'
nx_ijod=: 3 : '(allrefs }. gn) -. gn=. grp jodg'

NB. missing from (agroup)
nx ''
```

```

+-----+-----+---+---+-----+---+
|NORISESET|arctan|cos|sin|tabit|tan|
+-----+-----+---+---+-----+---+
```

1.0.15 Customize JOD edit facilities

The main JOD edit words (`nw`) and (`ed`) can be customized by defining a `DOCUMENTCOMMAND` script.

Note: Verbs that spawn J editors may not work in Jupyter labs. The following (`nw`) call opens a JQT or JHS editor in standard J front ends but does nothing here. This is because the J kernel is essentially a barebones `jconsole.exe` process that is not running JQT.

```
[19]: NB. define document command script - {~N~} is word name placeholder
DOCUMENTCOMMAND_ijod=: 0 : 0
smoutput pr '{~N~}'
)

NB. create a new word - opens an edit window in JQT and JHS
NB. nw 'bpword'
```

Find the (`bpword`) edit window and note how `DOCUMENTCOMMAND` text has been modified and insert. When the script window is saved with `CRTL-W` `DOCUMENTCOMMAND` runs.

Run and close the (`bpword`) edit window.

Edit a word in the dictionary. **JOD is always editing copies.** Originals can only be changed with explicit (`put`) operations.

```
[20]: NB. load dictionary word into edit window - requires JQT or JHS front ends
NB. also requires browser permissions and pop ups enabled

NB. ed 'sunriseset0'

NB. find the (sunriseset0) edit window and note DOCUMENTCOMMAND
NB. close edit window
```

1.0.16 Define JOD project macros

When programming with JOD you typically open a set of dictionaries. Load system scripts and define some handy nouns. This can be done in a project macro script.

```
[21]: NB. define a project macro - I use the prefix prj for such scripts
prjsunmoon=: 0 : 0
```

```

NB. standard j scripts
require 'debug task'

NB. local script nouns
jodg_iod=: 'sunmoon'
jods_iod=: 'sunmoontests'

NB. put/xref
DOCUMENTCOMMAND_z_ =: 'smoutput pr ''{~N~}'''
)

NB. store macro
4 put 'prjsunmoon';JSCRIPT_ajod_;prjsunmoon

```

```

+-+-----+-----+
|1|1 macro(s) put in ->|bpcopy|
+-+-----+-----+

```

Once project macros are defined it's easy to configure your J session. Open the required dictionaries and run the macor with (rm).

```

[22]: NB. setup project
rm 'prjsunmoon' [ od ;:'bpcopy bptest' [ 3 od ''

```

```

NB. standard j scripts
require 'debug task'
NB. local script nouns
jodg_iod=: 'sunmoon'
jods_iod=: 'sunmoontests'
NB. put/xref
DOCUMENTCOMMAND_z_ =: 'smoutput pr ''{~N~}'''

```

1.0.17 Edit your jodprofile.ijs

When JOD loads the script (~addons/general/jod/jodprofile.ijs) is executed. This script can be used to set up JOD the way you want. Note how you can execute project macros when JOD loads with sentences like:

```
rm 'prjsunmoon' [ od ;:'bpcopy bptest'
```

WARNING: store your profile in one of your dictionaries. This file is reset when JAL updates the JOD addon.

```
[23]: NB. display jodprofile.ijs
      stx read_ajod_ jpath '~addons/general/jod/jodprofile.ijs'
```

```
NB.*jodprofile s-- JOD dictionary profile.          ...
NB.                                                  ...
NB. An example JOD profile script. Save this script in ...
NB.                                                  ...
NB. ~addons/general/jod/                            ...
NB.                                                  ...
NB. with the name jodprofile.ijs                    ...
NB.                                                  ...
NB. This script is executed after all dictionary objects have ...
NB. been created. It can be used to set up your default JOD ...
NB. working environment.                            ...
NB.                                                  ...
```

1.0.18 Use JOD help and documentation

Install the (general/joddocument) addon to use JOD PDF help.

```
[24]: NB. location of jod.pdf - install addon (general/joddocument) with JAL
      smoutput jpath '~addons/general/joddocument/pdfdoc/jod.pdf'

      NB. opens jod.pdf if a pdf reader is available on your system
      jodhelp 0
```

```
c:/j64/j903/addons/general/joddocument/pdfdoc/jod.pdf
+-+-----+
```


|1|starting PDF reader|
+-+-----+

1.0.19 Summary and final remarks

1. JOD DOES NOT BELONG IN THE J TREE
2. BACKUP BACKUP BACKUP
3. TAKE A SCRIPT DUMP
4. MAKE A MASTER RE-REGISTER SCRIPT
5. SET LIBRARY DICTIONARIES TO READONLY
6. KEEP REFERENCES UPDATED
7. DOCUMENT DICTIONARY OBJECTS
8. DEFINE YOUR OWN JOD SHORTCUTS
9. CUSTOMIZE JOD EDIT FACILITIES
10. DEFINE JOD PROJECT MACROS
11. EDIT YOUR `jodprofile.ijs`
12. USE JOD HELP AND DOCUMENTATION

These are just some of the JOD practices I have found useful. As you use JOD you will probably find your own methods.

If you find an a useful method please let me know. I can be reached at:

John Baker
December 2021
bakerjd99@gmail.com