

Building JOD Addons

December 9, 2019

1 Building JOD Addons

1.0.1 Introduction

JOD is distributed as three J addons: [jod](#), [jodsource](#) and [joddocument](#).

This notebook describes how to build the JOD addons.

1.0.2 JAL - the J package manager

J addons are managed with the [J package manager](#). As you might expect addons have "packaging" requirements. For each JOD addon you must supply:

1. `manifest.ijs` - a J script defining addon contents
2. `history.txt` - an optional addon revision history file
3. addon files - a directory of files referenced by `manifest.ijs`



To insure that all the required JOD addon files are generated and placed correctly you run a JOD macro script.
Yes, JOD builds itself!

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

```
j901/j64avx/windows/beta-s/commercial/www.jsoftware.com/2019-12-02T12:51:33
```

1.0.3 Running buildjodistribution

The following builds all JOD addons.

It assumes:

1. The jodsource addon has been installed.
2. The JOD dictionaries (joddev), (jod) and (utils) are available.
3. Required build directories - shown below - are present.

```
NB. relative J folder ~JOD is defined
jpath '~JOD'
```

```
NB. JOD addon
~JOD/joddev/alien/stage/jod
~JOD/joddev/alien/stage/jod/jodbak
~JOD/joddev/alien/stage/jod/joddoc
~JOD/joddev/alien/stage/jod/joddoc/pdfdoc
~JOD/joddev/alien/stage/jod/jodexts
~JOD/joddev/alien/stage/jod/jodlabs
```

```
NB. JODSOURCE addon
~JOD/joddev/alien/stage/jodsource
```

```
NB. JODDOCUMENT addon
~JOD/joddev/alien/stage/joddocument
~JOD/joddev/alien/stage/joddocument/pdfdoc
```

4. jod.pdf - is built from LaTeX source (see this [GitHub repository](#)) and copied to the
~JOD/joddev/alien/stage/joddocument/pdfdoc directory.

<https://github.com/bakerjd99/joddoc>

If these conditions are satisfied the following builds all JOD addons. When the macro runs it displays all the generated output files.

```
In [2]: clear ''
        load 'general/jod'
        dpset 'RESETME' [ portchars ''

        NB. open JOD dictionaries see:
        smoutput od ;:'joddev jod utils' [ 3 od ''

        NB. run build macro script and display output - note the location of the generated files
        rm 'buildjoddistribution'
```

```
+--+-----+-----+---+-----+
|1|opened (rw/ro/ro) ->|joddev|jod|utils|
+--+-----+-----+---+-----+
NB.*buildjoddistribution s-- full JOD distribution build.

NB. create temporary working locale
cocurrent 'base'
coclass tmploc_AAAjodbuild999_=: 'AAAjodbuild999' [ coerase <'AAAjodbuild999'
coinert 'ijod'

NB. record open dictionaries and open JOD dictionaries
ooo=: }. did 0
od ;:'joddev jod utils' [ 3 od''
+--+-----+-----+---+-----+
|1|opened (rw/ro/ro) ->|joddev|jod|utils|
+--+-----+-----+---+-----+

NB. get JOD build utilities and version tracking nouns
```

```

    tmploc get }. grp 'buildjod'
+-+-----+
|1|57 word(s) defined|
+-+-----+

NB. set distribution directories
jddir=: 'JODSTAGEDIR JODGITDIR JODSOURCESTAGEDIR JODSTAGEPDFDIR JODSTAGEDOCDIR'
jddir=: jddir , ' JODGITDOCDIR JODADDONDIR JODSCRIPTDIR JODEXTSDIR'
(jddir)=: setbuilddirs 0

NB. generate distribution scripts
updatejodmanifest 0
setting standard JOD interface
/VERSION=: '1.0.0' /VERSION=: '1.0.0'
/VERSION=: '1.0.0' /VERSION=: '1.0.0'
/VERSION=: '1.0.0' /VERSION=: '1.0.0'
JOD manifests updated
    JODVMD buildjodcompressed JODSTAGEDIR;JODGITDIR;JODADDONDIR;JODSCRIPTDIR
NB. System: JOD  Author: John D. Baker  Email: bakerjd99@gmail.com
NB. Version: 1.0.0  Build Number: 32  Date: 09 Dec 2019 09:24:19

compressing ...
output file locations:
c:/j64/j64-901beta/addons/general/jod/jod.ijs
c:/jod/joddev/script/jod.ijs
c:/jod/joddev/alien/stage/jod/jod.ijs
82147 script bytes
JOD compressed build complete
    JODTOOLSVMDBuildjodtoolscompressed JODSTAGEDIR;JODEXTSDIR;JODSCRIPTDIR
NB. System: jodtools  Author: John D. Baker  Email: bakerjd99@gmail.com
NB. Version: 1.0.0  Build Number: 32  Date: 09 Dec 2019 09:24:20

compressing jodtools ...
output file locations:

```

```

c:/j64/j64-901beta/addons/general/jod/jodexts/jodtools.ijs
c:/jod/joddev/script/jodtools.ijs
c:/jod/joddev/alien/stage/jod/jodexts/jodtools.ijs
8062 script bytes
JODTOOLS compressed build complete
    JODVMD updatejoddistribution JODSTAGEDIR;JODGITDIR
c:/jod/joddev/alien/stage/jod/jodparms.ijs
c:/jod/joddev/alien/stage/jod/jodprofile.ijs
c:/jod/joddev/alien/stage/jod/history.txt
c:/jod/joddev/alien/jodgit/jod/history.md
c:/jod/joddev/alien/stage/jod/jodbak/jodprofilebak.ijs
c:/jod/joddev/alien/stage/jod/jodbak/jodparmsbak.ijs
c:/jod/joddev/alien/stage/jod/jodbak/joduserconfigbak.ijs
c:/jod/joddev/alien/stage/jod/manifest.ijs
c:/jod/joddev/alien/stage/jod/jodlabs/labexample001.ijs
c:/jod/joddev/alien/stage/jod/jodlabs/labdump.ijs
c:/jod/joddev/alien/stage/jod/jodlabs/jodintro.ijt
c:/jod/joddev/alien/stage/jod/jodlabs/joddump.ijt
c:/jod/joddev/alien/stage/jod/jodlabs/jodbest.ijt
c:/jod/joddev/alien/stage/jod/joddoc/pdfdoc/jodpdf.txt
c:/jod/joddev/alien/stage/jod/joddoc/linuxnotes.txt
c:/jod/joddev/alien/stage/jod/joddoc/macnotes.txt
JOD staging directories updated
    JODVMD updatejodsourcedumps JODSOURCESTAGEDIR
c:/jod/joddev/alien/stage/jodsource/manifest.ijs
c:/jod/joddev/alien/stage/jodsource/joddev.ijs
c:/jod/joddev/alien/stage/jodsource/jod.ijs
c:/jod/joddev/alien/stage/jodsource/utils.ijs
c:/jod/joddev/alien/stage/jodsource/jodsourcesetup.ijs
JODSOURCE stage directory updated
    JODVMD releasejod JODSTAGEDIR;JODSTAGEPDFDIR;JODSTAGEDOCDIR;JODGITDOCDIR
c:/jod/joddev/alien/stage/joddocument/manifest.ijs
c:/jod/joddev/alien/stage/joddocument/history.txt
c:/jod/joddev/alien/jodgit/joddocument/history.md

```

```
c:/jod/joddev/alien/stage/joddocument/pdfdoc/jod.pdf
JODDOCUMENT updated
JOD distribution build complete
```

```
NB. destroy build locale
cocurrent 'base'
coerese <tmploc_AAAjodbuild999_
```

1

1.0.4 Build Details

The rest of this notebook is for those brave souls that want to dive into the complexities of the JOD addon build code.

It's a good news bad news situation.

1. The good news: *all build code is in the three jodsource dictionaries.*
2. The bad news: *there is a lot of build code!*

Core JOD build code is in the JOD group (buildjod). Most JOD build macros (get) this group into a temporary locale. The following code fragment is found in a number of build macros.

```
cocurrent 'base'
coclass tmploc_AAAjodbuild999=: 'AAAjodbuild999' [ coerese <'AAAjodbuild999'
```

```
NB. record open dictionaries and open JOD dictionaries
ooo=: }. did 0
od ;:'joddev jod utils' [ 3 od''
```

```
NB. get JOD build utilities and version tracking nouns
tmploc get }. grp 'buildjod'
```

```
In [3]: NB. summary of group (buildjod) contents
        hg 'buildjod'
```

CR	carriage return character	
CRLF	carriage return line feed character pair	
JADDONJODPATH	fully qualified client jod JAL subversion directory	
JADDONJODROOTlinux	linux jaddon directory root path	
JADDONJODROOTmac	mac jaddon directory root path	
JADDONJODROOTwin	windows jaddon directory root path	
JADDONJODSOURCEPATH	fully qualified client jodsource JAL subversion directory	
JODTOOLSBUILDCNT	(jodtools) build count	
JODTOOLSVERSION	character string major-minor-point JODTOOLS version	
JODTOOLSVMID	JODTOOLS version, make and date	
JODVMD	version, make and date	
LF	line feed character	
TAB	tab character	
TEMPFILEO	temporary file must point to extant directory	
afterstr	retains string after first occurrence of (x)	
alltrim	trims all leading and trailing blanks	
allwhitetrims	trims all leading and trailing white space	
beforestr	retains string before first occurrence of (x)	
betweenstrs	select sublists between nonnested delimiters discarding delimiters	
blkaft	appends necessary blanks to J tokens	
buildjodcompressed	builds a fully compressed JOD load test script	
buildjodtoolscompressed	builds compressed version of the JOD tools class script	
changenqt	change not quoted	
changestr	replaces substrings - see long document	
charsub	single character pair replacements	
ctl	character table to newline delimited list	
dblquote	enclose all character lists in blcl in " quotes	
decomm	removes comments from j words	
dewhitejscript	removes all redundant blanks from J scripts	
fexist	1 if file exists 0 otherwise	
fwrite	write to file (UTF8 file names) returns #bytes if successful _1 otherwise	
jodsystempath	path of JOD system - \ terminated	
jpathsep	standarizes J path delimiter to unix/linux forward slash	

justfileext	file name and extension from fully qualified file
lastjodhistoryversion	extract last JAL compatible JOD version from build history
md5fromstring	computer md5 hash from character list
md5manifest	compute md5 checksums from manifest files
read	reads a file as a list of bytes
releasejod	final JOD release steps
setbuilddirs	sets paths used by JOD build words
setjaddoncopy	set distribution zip copy command for host OS
setjaddonpaths	sets jaddon paths for host OS
setjodinterface	sets interface list nouns for various systems
showpass	show and then pass noun
smoutput	session manager output
timestamp	formats timestamp as dd mmm yyyy hr:mn:sc
toCRLF	converts character strings to CRLF delimiter
toHOST	converts character strings to host delimiter
toJ	converts character strings to J delimiter LF
tstamp	standard j 8_07 library timestamp
updateJODTOOLSfiles	updates additional JOD addon locations with JODTOOLS script text
updateJODfiles	updates additional JOD addon locations with JOD script text
updatejoddistribution	updates files in JOD distribution
updatejodmanifest	set JOD addons manifests to current version
updatejodsourcedumps	update files in JODSOURCE distribution
wie	write ignoring errors
write	writes a list of bytes to file
+-----+	

1.0.5 A typical manifest.ijs

Each J add-on has a manifest file that lists the contents of the add-on. The following is the manifest for jod.

```
In [4]: NB. manifest for JOD
CAPTION='JOD J Object Dictionary'
VERSION='0.9.998 - dev'
FOLDER='general/jod'
```



```

PLATFORMS=: 'win linux darwin'
LABCATEGORY=: 'General'
FILES=: 0 : 0
jod.ijs
jodparms.ijs
jodprofile.ijs
manifest.ijs
history.txt
jodbak/jodparmsbak.ijs
jodbak/jodprofilebak.ijs
jodbak/joduserconfigbak.ijs
joddoc/linuxnotes.txt
joddoc/macnotes.txt
joddoc/pdftoc/jodpdf.txt
jodexts/jodtools.ijs
jodlabs/joddump.ijt
jodlabs/jodintro.ijt
jodlabs/jodbest.ijt
jodlabs/labexample001.ijs
jodlabs/labdump.ijs
)
RELEASE=: ''
DEPENDS=: 0 : 0
data/jfiles
)
DESCRIPTION=: 0 : 0
JOD (J Object Dictionary) is a J programming and refactoring
tool. JOD supports linked dictionary databases and provides
tools for building, documenting and testing J systems.

```

JOD requires the (data/jfiles) addon.

If you install JOD it is highly recommended that you
install the related addon JODDOCUMENT. JODDOCUMENT

installs detailed PDF documentation for JOD.

For more information see the JOD wiki pages (Info Button), run the lab "JOD lab JOD Introduction" - JOD labs are under the General lab category. If you do not see JOD labs listed on the Lab menus use the Lab browse button to access:

```
~addons/general/jod/jodlabs
```

More information is available on the JOD Page and GitHub at:

```
https://analyzethedatanotthedrive1.org/the-jod-page/
```

```
https://github.com/bakerjd99/jod
```

```
)
```

1.0.6 buildjodcompressed

All the code required to build JOD is available in the jodsource dictionaries. The following verb (buildjodcompressed) builds the "compressed" JOD load script. *It calls many JOD verbs.*

```
In [5]: buildjodcompressed=:4 : 0
```

```
NB.*buildjodcompressed v-- builds a fully compressed JOD load test script.
```

```
NB.
```

```
NB. dyad:  blVersion buildjodcompressed blclDirs
```

```
NB.
```

```
NB. verbatim:
```

```
NB.
```

```
NB. Builds a compressed JOD load script.
```

```
NB.
```

```
NB. Caveats:
```

```
NB.
```

```
NB. No error checking.
```

```
NB. Requires standard J profile.
```

```
NB. In addition to standard JOD directories this script also requires
```

NB. the staging directory: .../joddev/alien/stage/jod/
NB.
NB. author: John D. Baker
NB. changes: -----
NB. 08may28 rename/relocate compressed script to root of jod addon
NB. 11feb05 use (jpathsep) to show portable style file paths
NB. 11feb08 coinser 'ijod' added after moving JOD interface to ijod
NB. 11nov24 tested on linux
NB. 14nov14 adjusted for j 8.03 (runs on win32 J 8.03 - not 64 - no zfiles)
NB. 15mar15 remove zfiles - redirect build to .../joddev/alien/stage/... directories
NB. 17jan09 improve caveat notes for running on jconsole versions
NB. 17mar03 macro script converted to verb

'JODSTAGEDIR JODGITDIR JODADDONDIR JODSCRIPTDIR'=. y

NB. set vmd - version, make, date
'jodversion jodbuildcnt'=.2{x
jodbulddate=. alltrim tstamp'
jodbuildcnt=. >:jodbuildcnt

NB. increment build and save - must be done before
NB. script generation to insure header and class JODVMD match
JODVMD=.(jodbuildcnt;jodbulddate) (1 2){x

NB. save incremented version, make, date in dictionary
assert. >0{(WORD_ajod_,NVTABLE_ajod_) put 'JODVMD';WORD_ajod_;wrep_ajodstore_ 'JODVMD'

NB. set output files
JODFILE=. 'jod.ijs'

NB. working JOD addon script
JODADDONFILE=. JODADDONDIR,JODFILE

NB. backup of JOD script placed in put dictionaries script directory

```
JODSCRIPTFILE=. JODSCRIPTDIR,JODFILE
```

```
NB. JOD staging directory under put dictionaries alien directory
```

```
JODSTAGEFILE=. JODSTAGEDIR,JODFILE
```

```
head='.NB. System: JOD Author: John D. Baker Email: bakerjd99@gmail.com',LF
```

```
head=.head, ('NB. Version: ',(jodversion),' Build Number: ',(":"jodbuildcnt),' Date: ',jodbuilddate),LF
```

```
smoutput head
```

```
NB. turn off white space preservation during JOD load
```

```
hc=.head,'(9!:41) 0',LF
```

```
NB. get preprocessor script - leave only header comments
```

```
s=. dewhitejscript (1;0 2){:: MACRO_ajod_ get 'jodprep'
```

```
NB. main JOD classes
```

```
g=. ;: 'jodon jod jodstore joddob jodmake jodutil'
```

```
NB. collect group dependents
```

```
deps=. gdeps&.> g
```

```
smoutput 'compressing ... '
```

```
NB. dewhite group headers
```

```
h=. dewhitejscript&.> {"1 >{: GROUP_ajod_ get g
```

```
NB. compress all nondependent words in each group (a cruncher)
```

```
code=. , }.@:compj&> ({.@:grp)&.> g) -.&.> deps
```

```
NB. combine header and word definitions
```

```
s=. s , ; , h ,. code
```

```
NB. append post processor script
```

```
s=. s, dewhitejscript (1;0 2){:: MACRO_ajod_ get 'jodpost'
```

```

NB. remove a few remaining redundant blanks after
NB. assignments - their presence annoys me!
s=. hc , '/' =./=./ =:/=: ' changenqt s

NB. save compressed load scripts
smoutput 'output file locations:'
(s=. toJ s) write showpass JODADDONFILE
smoutput s write showpass JODSCRIPTFILE
smoutput s write showpass JODSTAGEFILE
updateJODfiles s

NB. update build history - stored as plain text macro
JODBUILDHISTORY=. (1;0 2){:: MACRO_ajod_ get 'JODBUILDHISTORY'
JODBUILDHISTORY=. JODBUILDHISTORY,(2#LF),head
assert. >0{MACRO_ajod_ put 'JODBUILDHISTORY';TEXT_ajod_;JODBUILDHISTORY

NB.byte size of generated plain text script
smoutput (":#s), ' script bytes'
smoutput 'JOD compressed build complete'
)

NB. erase to prevent exeution out of context
erase 'buildjodcompressed'

```

1

1.0.7 buildjodcompressed uses union

The following displays all the words on the buildjodcompressed call tree. All of these words are in the source dictionaries.

```

In [6]: NB. find the uses union - unique list of words on the call tree
        'rc xr'=. 31 uses 'buildjodcompressed'

        NB. display all the words called by (buildjodcompressed)
        ('Direct calls';'Locatives - JOD internals') ,: 40&list &.> }. ,xr

```

+-----+		+-----+	
Direct calls		Locatives - JOD internals	
+-----+		+-----+	
ALPHA	CR	GROUP_ajod_	MACRO_ajod_
CRLF	CWSONLY	NVTABLE_ajod_	TEXT_ajod_
DEFAULT	DEPENDENTSEND	WORD_ajod_	wrep_ajodstore_
DEPENDENTSSTART	DICTIONARY	wttext__MK	namecats__MK
DOCUMENT	ERR001	obtext__UT	checkopen__ST
ERR002	ERR003	defwords__ST	getdicdoc__ST
ERR004	ERR008	getdocument__ST	getexplain__ST
ERR010	ERR014	getgstext__ST	getntstamp__ST
ERR015	ERR020	getobjects__ST	invfetch__ST
ERR023	ERR024	checkput__ST	putdicdoc__ST
ERR025	EXPLAIN	putexplain__ST	putntstamp__ST
GROUP	INCLASS	puttable__ST	puttexts__ST
INCREATE	INPUT	putwords__ST	JARGS__MK
INSIZE	JNAME	checkpath__ST	gslistnl__ST
LF	MACRO	putgs__ST	dnlsearch__ST
MACROTYPE	MAXNAME	rxall_jregex_	
NAMEALPHA	NVTABLE		
OBFUSCATE	OBFUSCCNT		
OBFUSCPFX	OBJECTNC		
PUTBLACK	SUITE		
TAB	TEST		
WORD	alltrim		
badblia	badbu		
badcl	badil		
badlocn	badrc		
badunique	beforestr		
blkft	boxopen		
changenqt	changetok		
checknames	checknttab		
checknttab2	compclut		
compj	compressj		

ctit	ctl		
decomm	dewhitejcr		
dewhitejscript	dnl		
exobrefs	firstone		
gdeps	get		
grp	gsmakeq		
isempty	islocref		
jderr	jnfrblcl		
ljust	nc		
nlargs	ok		
plt	put		
rv	rxall		
second	showpass		
smoutput	timestamp		
toJ	trimnl		
tstamp	updateJODfiles		
wie	write		
+-----+-----+			