Writing Portable J addons

J 2012 Conference

July 23-24, Toronto Ontario

John D. Baker

http://bakerjd99.wordpress.com/

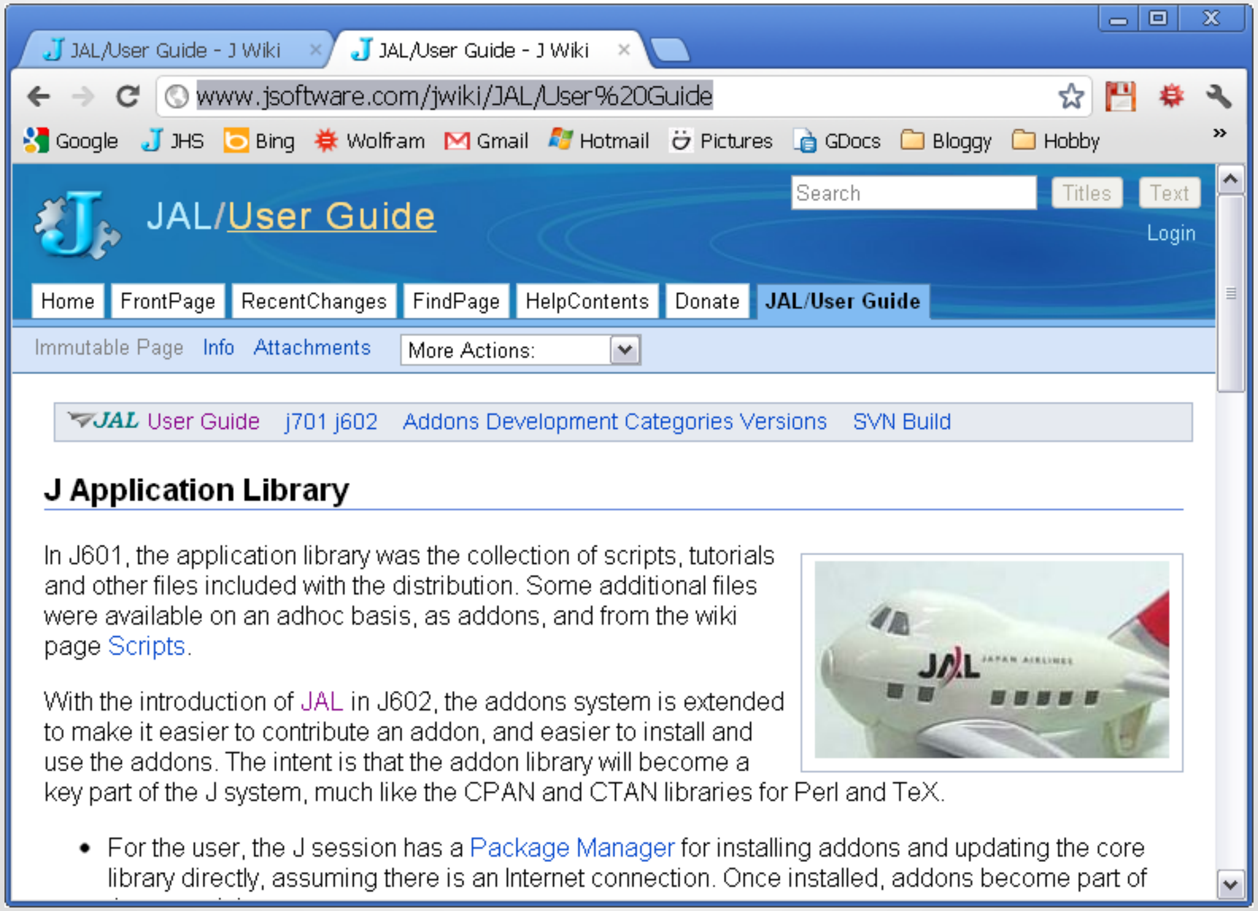
What's an "addon"?

Portable with respect to what?

Why bother with addons?

J Application Library

- http://www.jsoftware.com/jwiki/JAL/User%20Guide


JAL/User Guide - J Wiki
www.jsoftware.com/jwiki/JAL/User%20Guide
Google JHS Bing Wolfram Gmail Hotmail Pictures GDocs Bloggy Hobby
JAL/User Guide
Search Titles Text Login
Home FrontPage RecentChanges FindPage HelpContents Donate JAL/User Guide
Immutable Page Info Attachments More Actions:
JAL User Guide j701 j602 Addons Development Categories Versions SVN Build

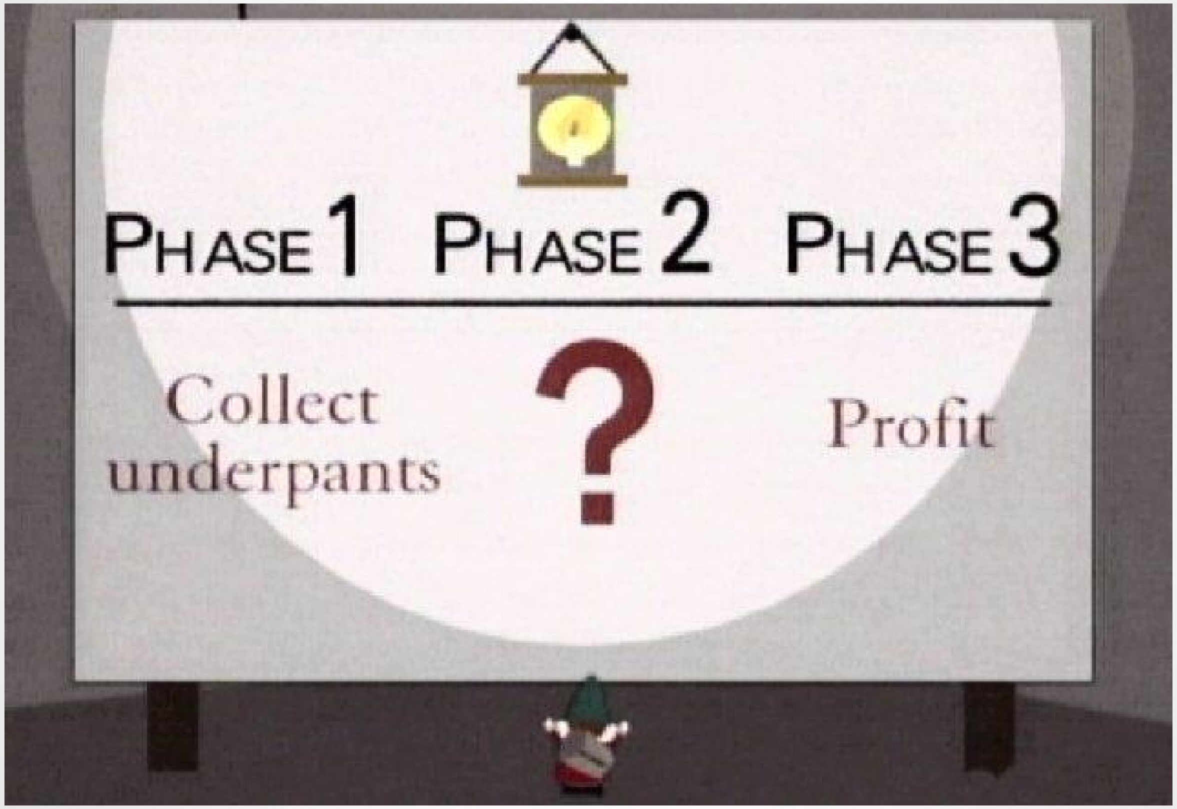
J Application Library

In J601, the application library was the collection of scripts, tutorials and other files included with the distribution. Some additional files were available on an adhoc basis, as addons, and from the wiki page Scripts.

With the introduction of JAL in J602, the addons system is extended to make it easier to contribute an addon, and easier to install and use the addons. The intent is that the addon library will become a key part of the J system, much like the CPAN and CTAN libraries for Perl and TeX.

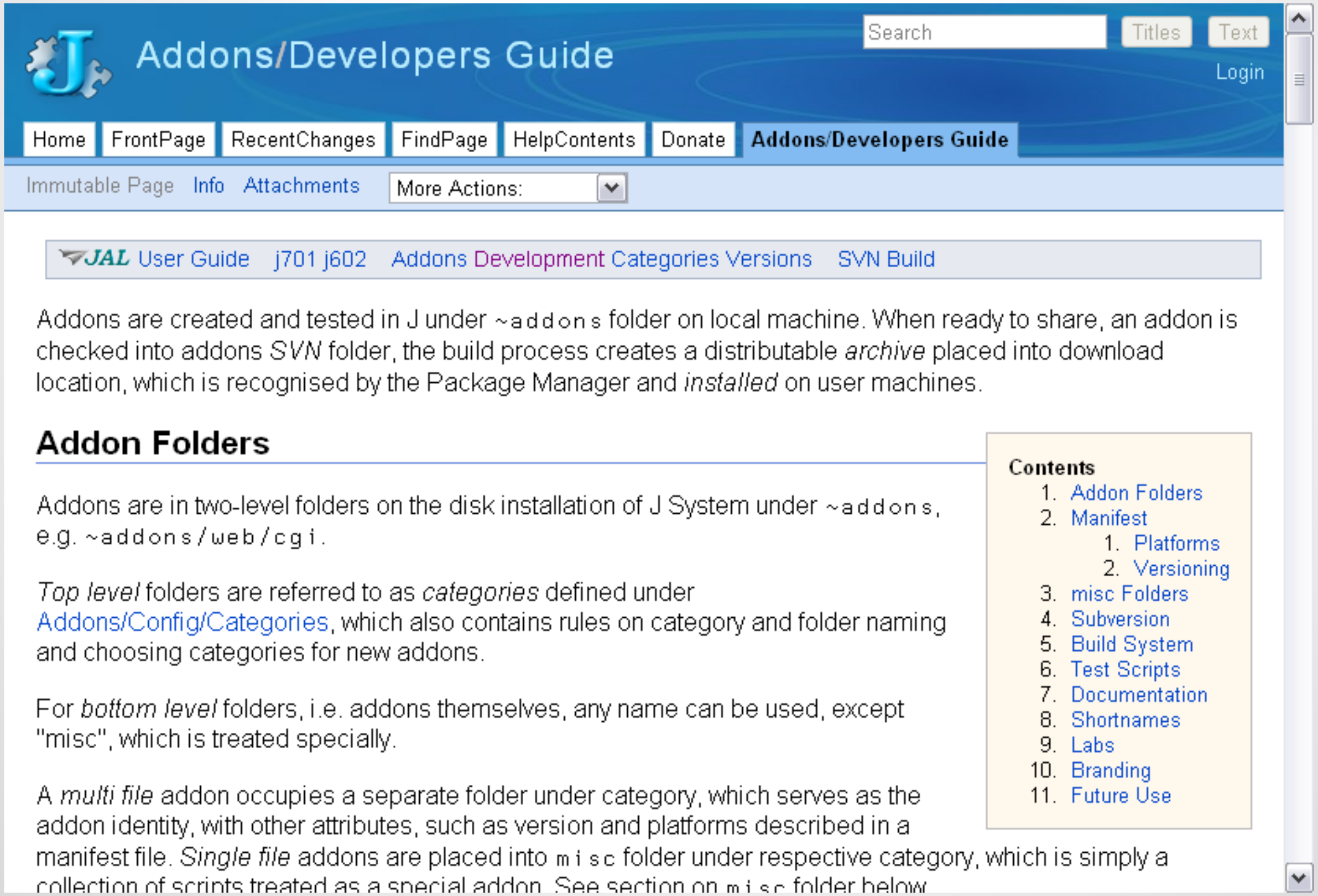
- For the user, the J session has a Package Manager for installing addons and updating the core library directly, assuming there is an Internet connection. Once installed, addons become part of

Let's write a portable addon



p1: read addon wiki guide

- http://www.jsoftware.com/jwiki/Addons/Developers%20Guide



Addons are created and tested in J under `~addons` folder on local machine. When ready to share, an addon is checked into addons *SVN* folder, the build process creates a distributable *archive* placed into download location, which is recognised by the Package Manager and *installed* on user machines.

Addon Folders

Addons are in two-level folders on the disk installation of J System under `~addons`, e.g. `~addons/web/cgi`.

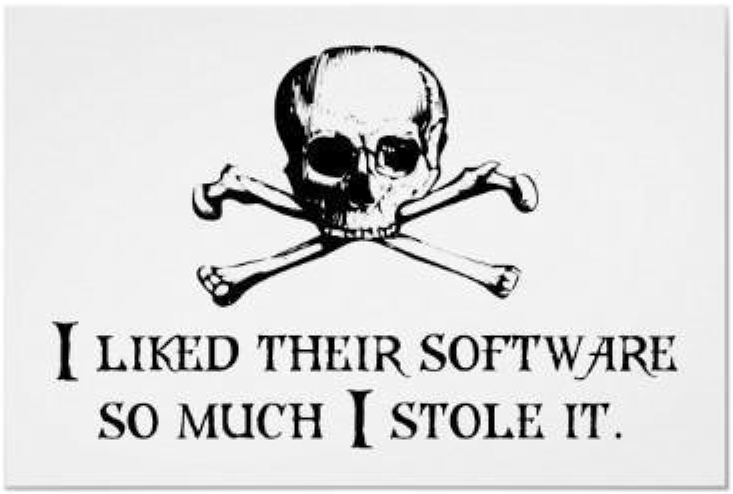
Top level folders are referred to as *categories* defined under Addons/Config/Categories, which also contains rules on category and folder naming and choosing categories for new addons.

For *bottom level* folders, i.e. addons themselves, any name can be used, except "misc", which is treated specially.

A *multi file* addon occupies a separate folder under category, which serves as the addon identity, with other attributes, such as version and platforms described in a manifest file. *Single file* addons are placed into `misc` folder under respective category, which is simply a collection of scripts treated as a special addon. See section on `misc` folder below

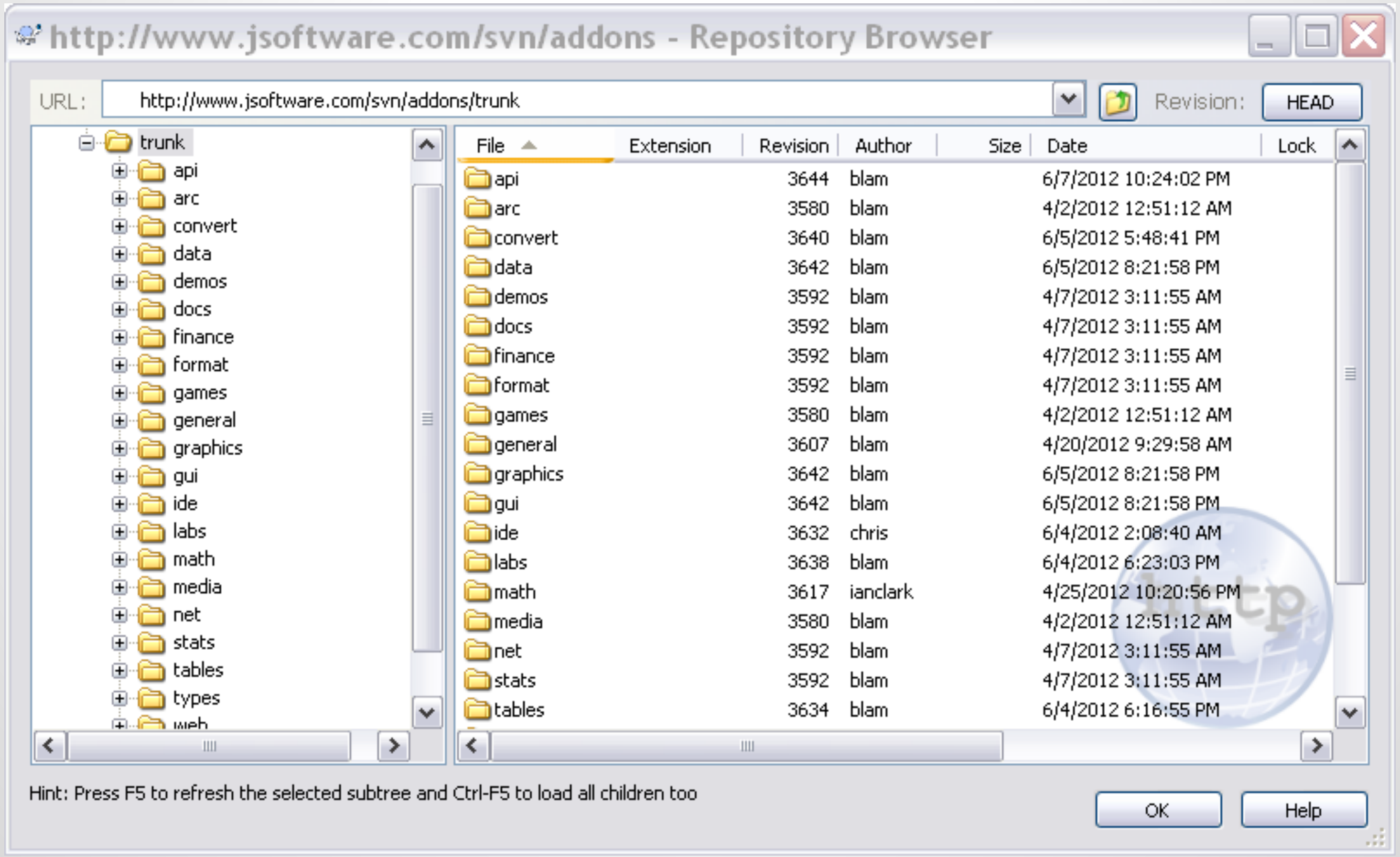
Contents

p1: steal some code



p1: addons repository

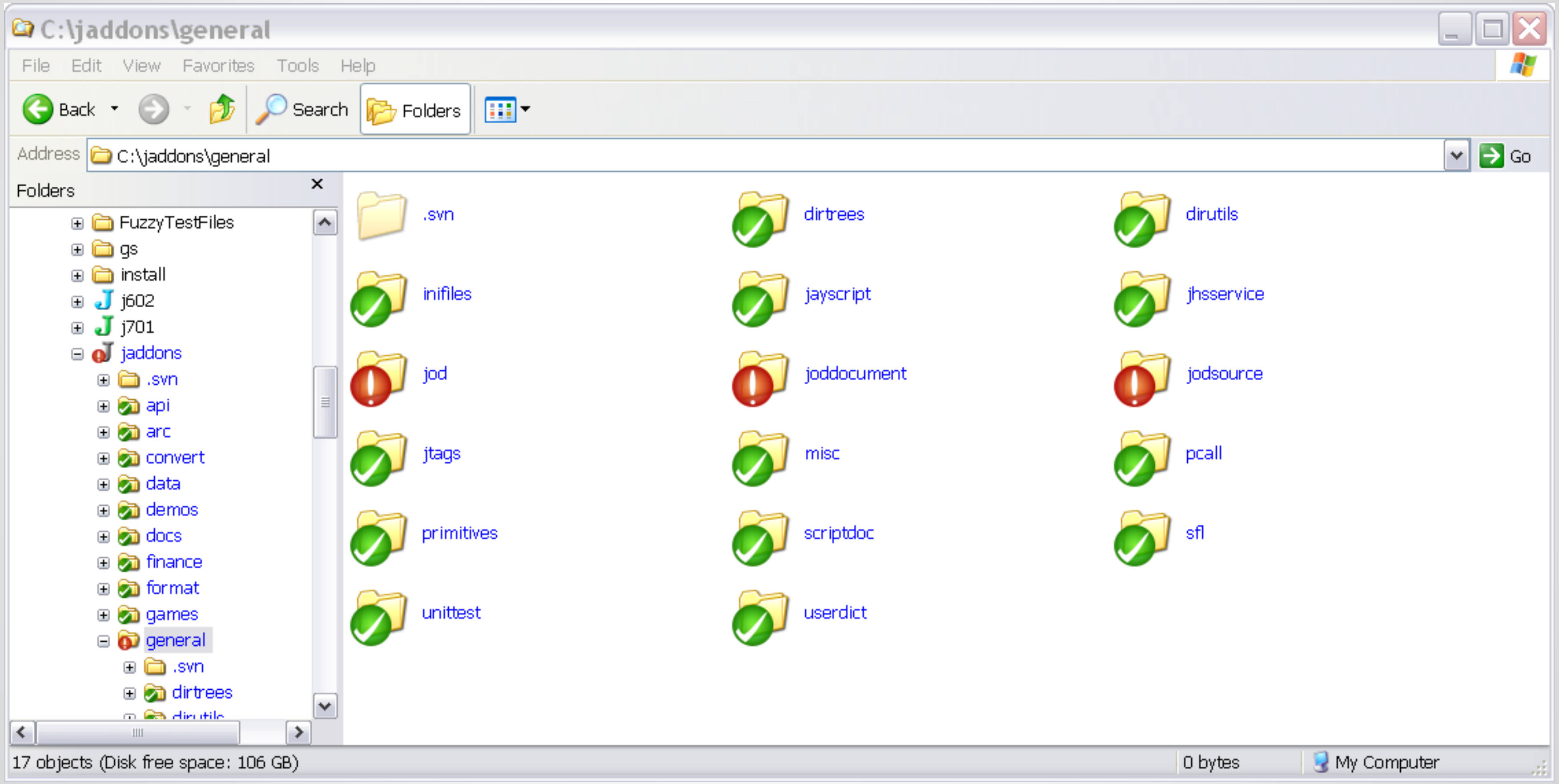
- http://www.jsoftware.com/svn/addons/trunk



File	Extension	Revision	Author	Size	Date	Lock
api		3644	blam		6/7/2012 10:24:02 PM	
arc		3580	blam		4/2/2012 12:51:12 AM	
convert		3640	blam		6/5/2012 5:48:41 PM	
data		3642	blam		6/5/2012 8:21:58 PM	
demos		3592	blam		4/7/2012 3:11:55 AM	
docs		3592	blam		4/7/2012 3:11:55 AM	
finance		3592	blam		4/7/2012 3:11:55 AM	
format		3592	blam		4/7/2012 3:11:55 AM	
games		3580	blam		4/2/2012 12:51:12 AM	
general		3607	blam		4/20/2012 9:29:58 AM	
graphics		3642	blam		6/5/2012 8:21:58 PM	
gui		3642	blam		6/5/2012 8:21:58 PM	
ide		3632	chris		6/4/2012 2:08:40 AM	
labs		3638	blam		6/4/2012 6:23:03 PM	
math		3617	ianclark		4/25/2012 10:20:56 PM	
media		3580	blam		4/2/2012 12:51:12 AM	
net		3592	blam		4/7/2012 3:11:55 AM	
stats		3592	blam		4/7/2012 3:11:55 AM	
tables		3634	blam		6/4/2012 6:16:55 PM	

Hint: Press F5 to refresh the selected subtree and Ctrl-F5 to load all children too

p1: setup svn working directory

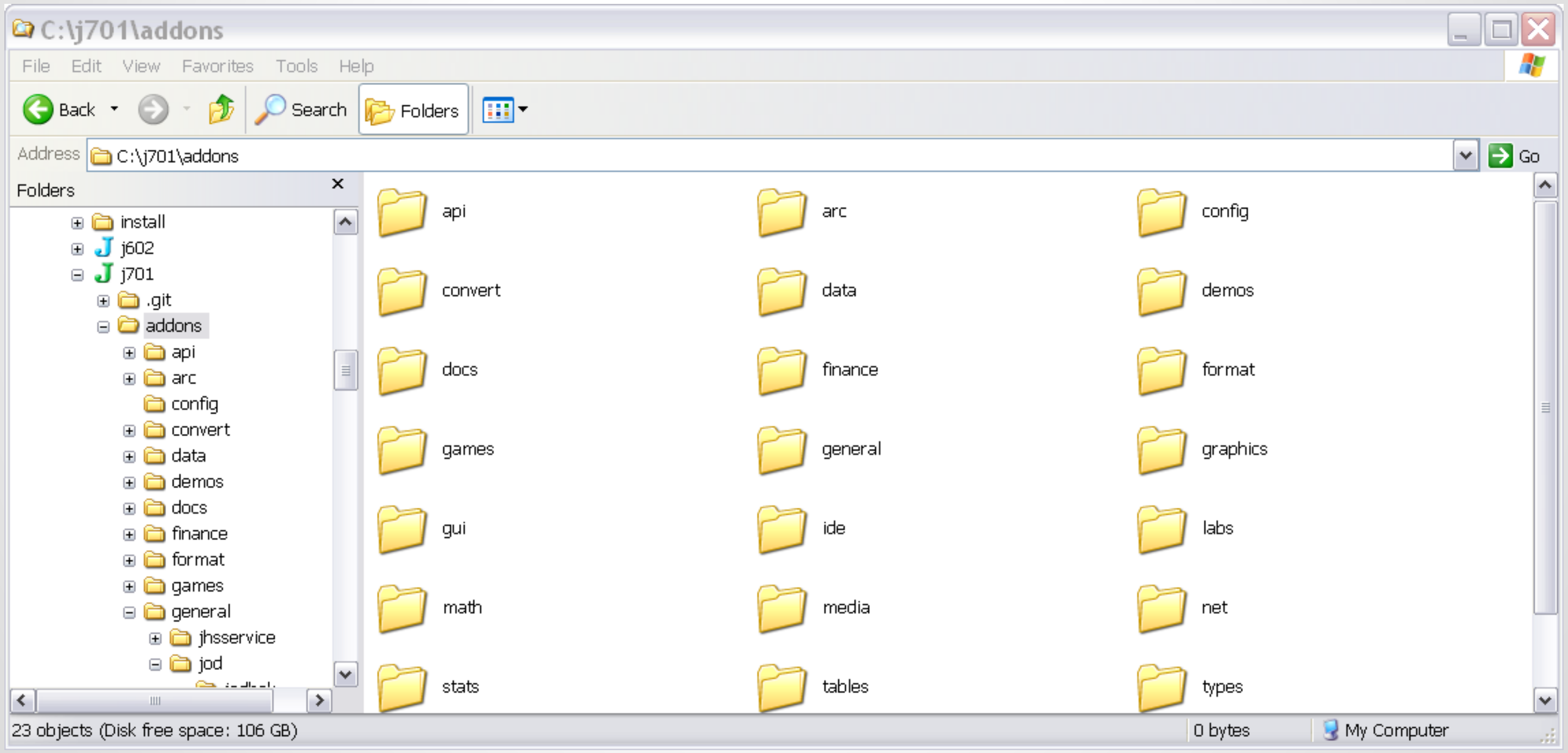


p2: addon nutshell

- After reading the wiki guide and browsing extant addons you’ll discover that “addon-izing” a J application is straightforward matter of placing it in established addon directories and providing some basic metadata scripts.

p2: addon directories/categories

- http://www.jsoftware.com/jwiki/Addons/Config/Categories



install
j602
j701
.git
addons
api
arc
config
convert
data
demos
docs
finance
format
games
general
jhsservice
jod

api arc config
convert data demos
docs finance format
games general graphics
gui ide labs
math media net
stats tables types

23 objects (Disk free space: 106 GB) 0 bytes My Computer

p2: addon metadata scripts

- `manifest.ijs`
- `history.ijs`
- `build.ijs`
- `test.ijs`

p2: manifest.ijs

```
NB. finance/interest manifest

CAPTION=: 'Date formatting functions'

DESCRIPTION=: 0 : 0
Various date formatting functions.
)

VERSION=: '1.0.9'

FILES=: 0 : 0
datefmt.ijs
)

RELEASE=: 'j701'
```

p2: portable do's and don't's

- No fixed paths. Use **`jpath, jpathsep, ~`**

```
jpath '~addons/misc/boo.ijs'
jpathsep '\fix/the\directory/delimiters\
/fix/the/directory/delimiters/
```

- Use short lower case ASCII path and file names with *no questionable characters*.

```
/avoid/funny/character/freenames.ijs
c:/on/windows/use/unix/style/paths
```

- Consider sticking with **`toJ`** line ends.
- There's UTF-8 and wrong!

p2: portability pitfalls

p2: J systems differ

- J is very portable but there are key differences (NANS, tolerances, et cetera) between 32/64 bit systems on various hosts.
- Portable addon code has to assume the worst and live with the lowest common denominator.
- There’s no substitute for testing.

p2: J host nouns (a)

- J 6.0x and 7.0x profile names differ

J 6.0x	J 7.0x
`PUBLIC_j_`	`Public_j_`
`BROWSER_j_`	`Browser_j_`
`PDFREADER_j_`	`PDFReader_j_`
`LOADED_j_`	`Loaded_j_`

p2: J host nouns (b)

- Identification nouns are not always present

J 6.0x	J 7.0x
IFWIN	IFWIN
IFUNIX	IFUNIX
(missing)	IFIOS
IFGTK	IFGTK
UNAME	UNAME
IFANDROID??	IFANDROID??

p2: J host nouns (c)

```
et=:3 : 0

NB.*et v-- edit text
EDTEMP et y  NB. default edit file
:
NB. write to J temp directory - created by J install
try.
  (toHOST y) write file=. jpath '~temp\' , x , IJS
  if. 0 e. wex ;:'IFJHS IFJ6 IFGTK' do.
    NB. probably on a J 6.0x system
    smopen_jijs_ file   NB. J 6.0x
  else.
    NB. open in various J7/6 editors !(*)=.  IFJHS IFGTK IFJ6 IFIOS
    if. IFJHS      do. open_jhs_ file
    elseif. IFGTK do. open_jgtk_ file
    elseif. IFJ6  do. smopen_jijs_ file   NB. J 6.0x
    elseif. IFIOS do. je_z_ file          NB. iPhone/iPad
    elseif.do. jderr ERR0262  NB. errmsg: not supported on current J system
    end.
  end.
catch. jderr ERR0255
end.
)
```

p2: dependencies

- JAL does not resolve dependencies for you

```
DEPENDS=: 0 : 0
image/platimg
web/cgi 0.1.2
xml/sax 0.2
)
```

- If your addon depends on others it inherits their portability limitations.

p2: OS/IT/Admin “issues”

- Path and file names always blow
- Fonts, screen sizes – forget about it
- Compiled dll’s, so’s et cetera differ
- Permissions – don't get me started
- IT policies – urge to kill rising
- SOX - time to hang some politicians

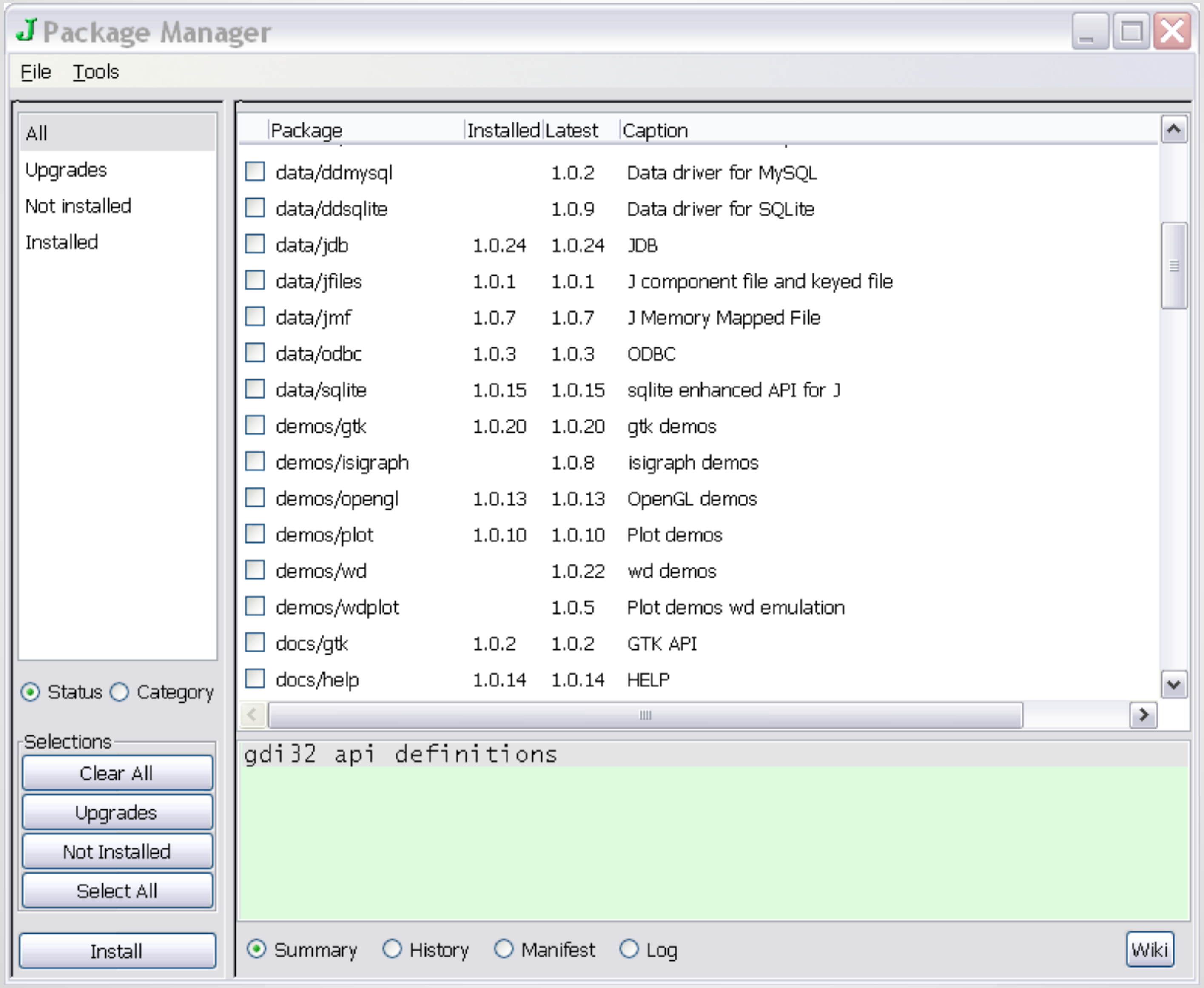
p2: build addon and commit

- Copy all files to working svn directories
- Increment **`VERSION`** noun
- Commit to JAL
- For complex addons automate this process

p2: example addon build

```
NB.*buildjoddistribution s-- full JOD distribution build

cocurrent 'base'
coerase <'AAAjodbuild999'
coclass tmploc_AAAjodbuild999_ =: 'AAAjodbuild999'
coinsert 'ijod'

ooo=: did 0
od ;:'joddev jod utils' [ 3 od ''

1 rtt 'updatejodmanifest'       NB. AAAtrash999

rs 'buildjodcompressed'         NB. AAAtrash999
rs 'buildjodtoolscompressed'    NB. AAAtrash999

1 rtt 'updatejoddistribution'   NB. AAAtrash999
1 rtt 'updatejodsourcedumps'    NB. AAAtrash999

1 rtt 'releasejod'              NB. AAAtrash999

cocurrent 'base'
coerase <tmploc_AAAjodbuild999_
```

p2: distributing addons



Package	Installed	Latest	Caption
data/ddmysql		1.0.2	Data driver for MySQL
data/ddsqlite		1.0.9	Data driver for SQLite
data/jdb	1.0.24	1.0.24	JDB
data/jfiles	1.0.1	1.0.1	J component file and keyed file
data/jmf	1.0.7	1.0.7	J Memory Mapped File
data/odbc	1.0.3	1.0.3	ODBC
data/sqlite	1.0.15	1.0.15	sqlite enhanced API for J
demos/gtk	1.0.20	1.0.20	gtk demos
demos/isigraph		1.0.8	isigraph demos
demos/opengl	1.0.13	1.0.13	OpenGL demos
demos/plot	1.0.10	1.0.10	Plot demos
demos/wd		1.0.22	wd demos
demos/wdplot		1.0.5	Plot demos wd emulation
docs/gtk	1.0.2	1.0.2	GTK API
docs/help	1.0.14	1.0.14	HELP

```
gdi32 api definitions
```

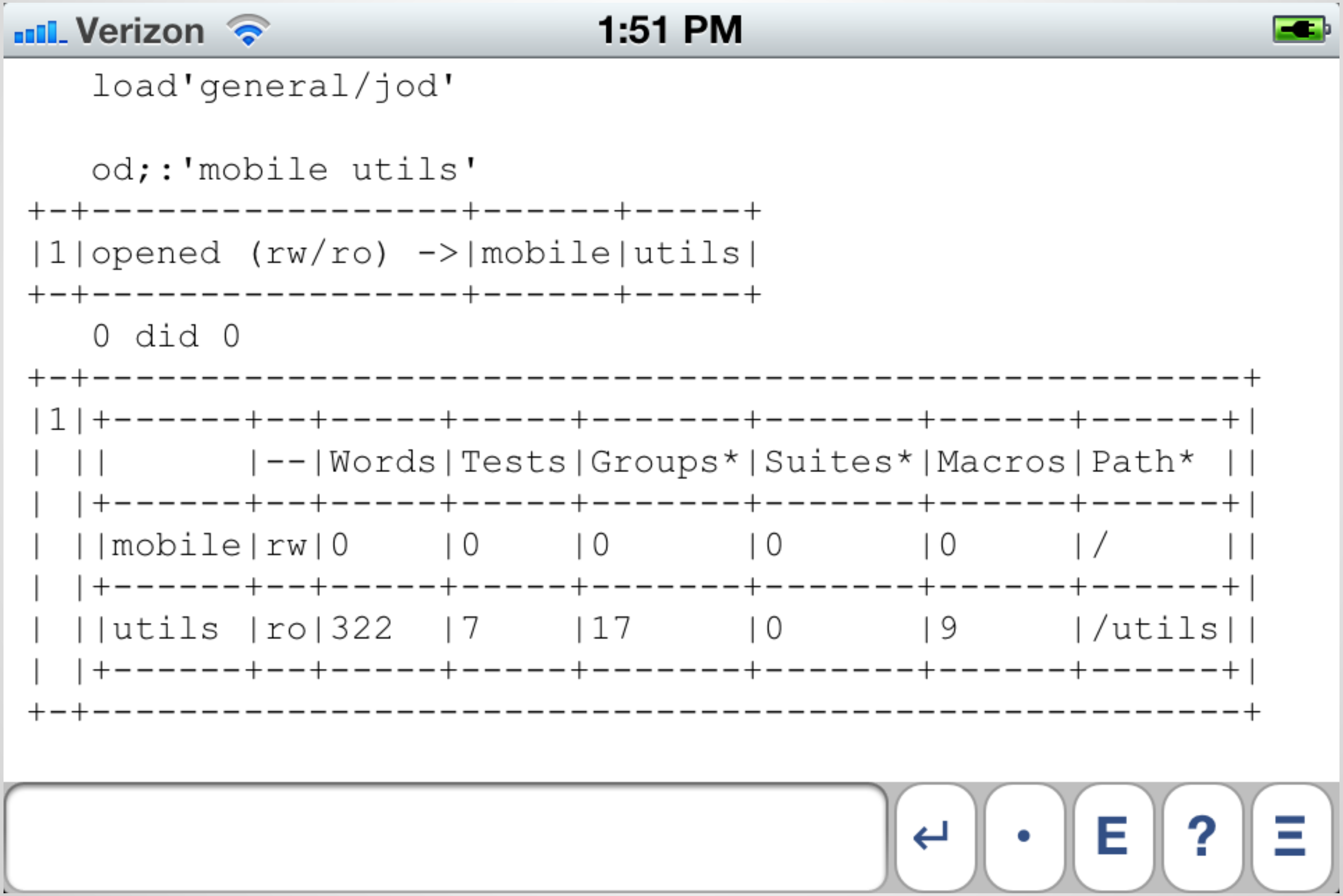
[✓ Status] Category

[✓ Summary] History Manifest Log

p2: no JAL *no problemo*

- JAL is not available on some J platforms like IOS.
- I use iZipPro and cut & paste to transfer `*.jt` files, JOD dump scripts and other files.

p2: IOS JOD (`*.jt, *.zip`)



```
   load'general/jod'

   od;:'mobile utils'
+-+------------------+------+-----+
|1|opened (rw/ro) ->|mobile|utils|
+-+------------------+------+-----+
   0 did 0
+-+------------------------------------------------------+
|1|+------+--+-----+-----+-------+-------+------+-------+|
| ||      |--|Words|Tests|Groups*|Suites*|Macros|Path*  ||
| |+------+--+-----+-----+-------+-------+------+-------+|
| ||mobile|rw|0    |0    |0      |0      |0     |/      ||
| |+------+--+-----+-----+-------+-------+------+-------+|
| ||utils |ro|322  |7    |17     |0      |9     |/utils||
| |+------+--+-----+-----+-------+-------+------+-------+|
+-+------------------------------------------------------+
```

p2: testing addons

I DON'T ALWAYS TEST MY CODE
BUT WHEN I DO, I DO IT IN PRODUCTION
Troll.me
Anti-duckfaces at Dudelol.com

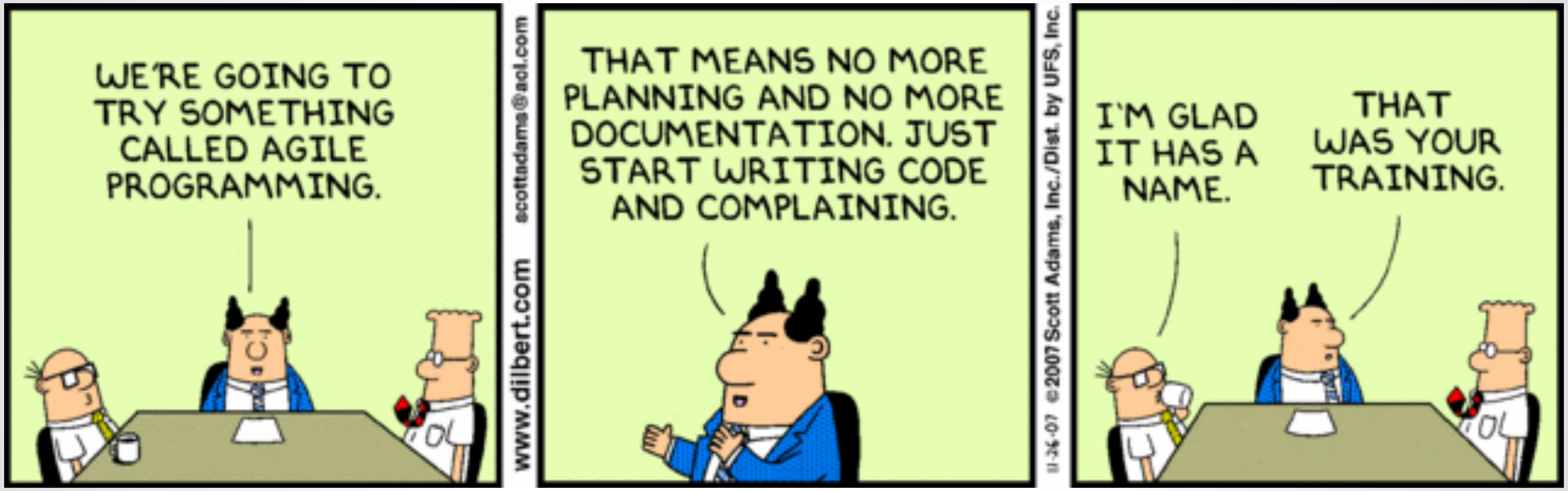
p2: testing addons

- always a good idea
- `test.ijs` when feasible
- document test process when not

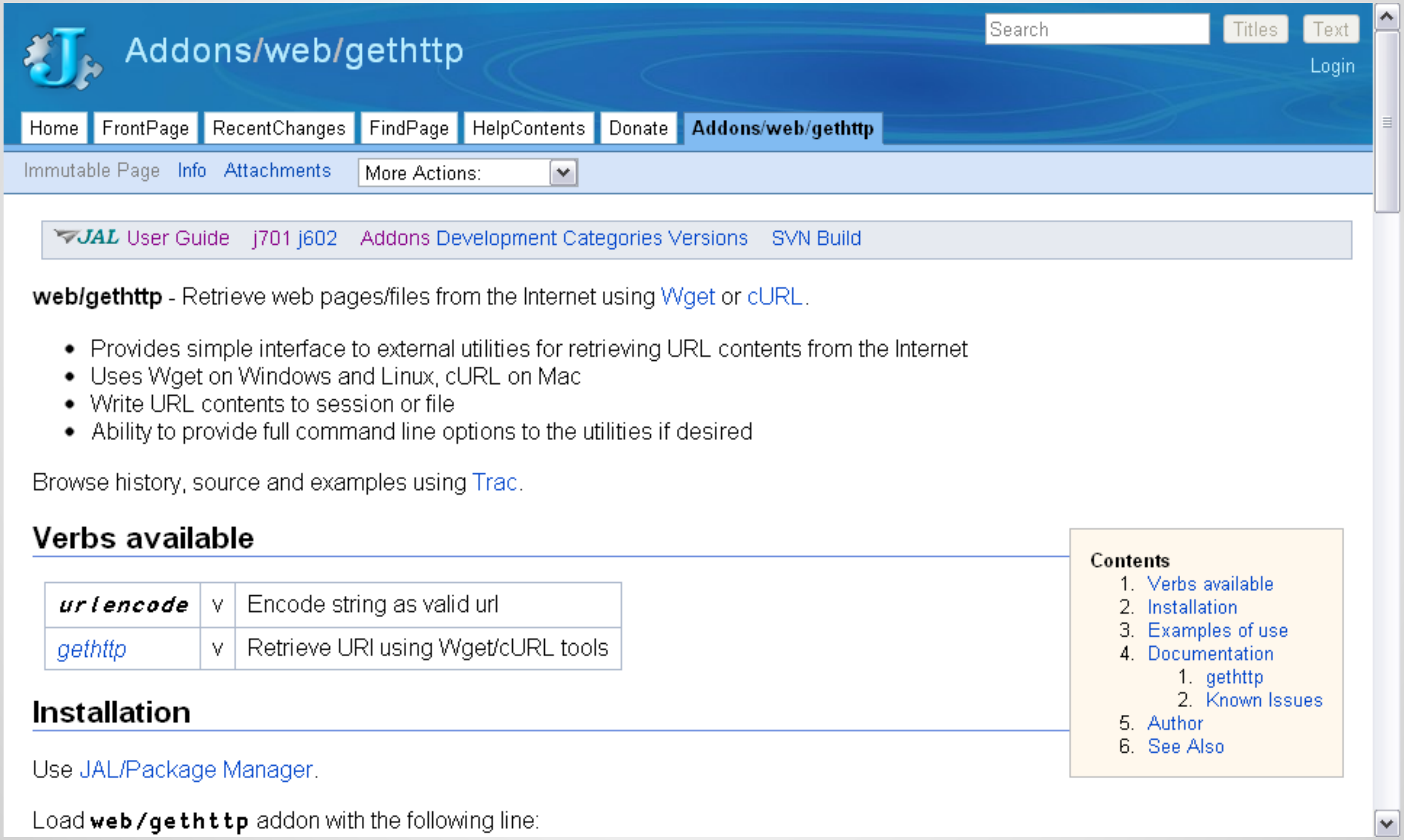
p2: JOD addon testing

```
NB.*jodbasictests s-- suite of basic JOD tests.
NB.
NB. Opens a READWRITE test dictioary (testjod0) and runs a series
NB. of basic JOD test scripts.
NB.
NB. assumes:
NB.
NB.   standard J profile
NB.   (jodtester) load script
NB.   READWRITE test dictionaries: see (createtestdictionaries)
NB.
NB. run with:
NB.
NB.   require 'general/jod'
NB.   od ;:'joddev jod utils'
NB.   3 rtt 'jodbasictests'
NB.
NB. created: 2007nov27
NB. changes: -----------------------------------------------------
NB. 07dec05 added (notopenTests001) to suite
NB. 07dec12 added (dnlMacroTexts) to suite - TESTYAMMER=:0
NB. 07dec19 (didBasic001) added
NB. 08oct07 turning on white space preservation to start tests
NB. 11nov18 updated for JOD 0.9.4+ testing
NB. 12jul05 improve run instructions
```

p2: documenting addons


WE'RE GOING TO TRY SOMETHING CALLED AGILE PROGRAMMING.
THAT MEANS NO MORE PLANNING AND NO MORE DOCUMENTATION. JUST START WRITING CODE AND COMPLAINING.
I'M GLAD IT HAS A NAME.
THAT WAS YOUR TRAINING.
www.dilbert.com scottadams@aol.com
11-26-07 ©2007 Scott Adams, Inc./Dist. by UFS, Inc.

p2: addon wiki page



web/gethttp - Retrieve web pages/files from the Internet using Wget or cURL.

- Provides simple interface to external utilities for retrieving URL contents from the Internet
- Uses Wget on Windows and Linux, cURL on Mac
- Write URL contents to session or file
- Ability to provide full command line options to the utilities if desired

Browse history, source and examples using Trac.

Verbs available

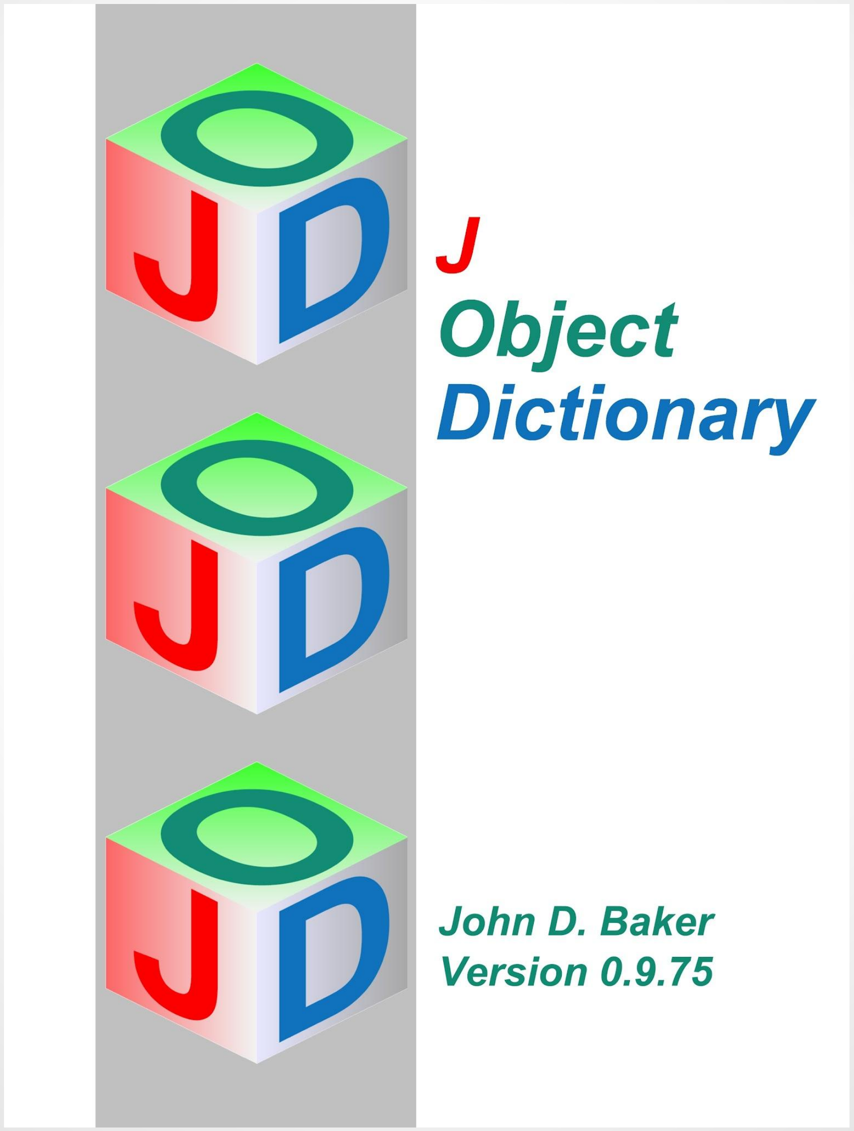
urlencode	v	Encode string as valid url
gethttp	v	Retrieve URI using Wget/cURL tools

Installation

Use JAL/Package Manager.

Load `web/gethttp` addon with the following line:

p2: manuals still rule


J
Object
Dictionary

John D. Baker
Version 0.9.75

p3: addon Profits?

- lots of fun
- solves one of your problems
- blog hits
- "we are not worthy" peer accolades
- resume padding
- another excuse to use J

p3: addon Profits?