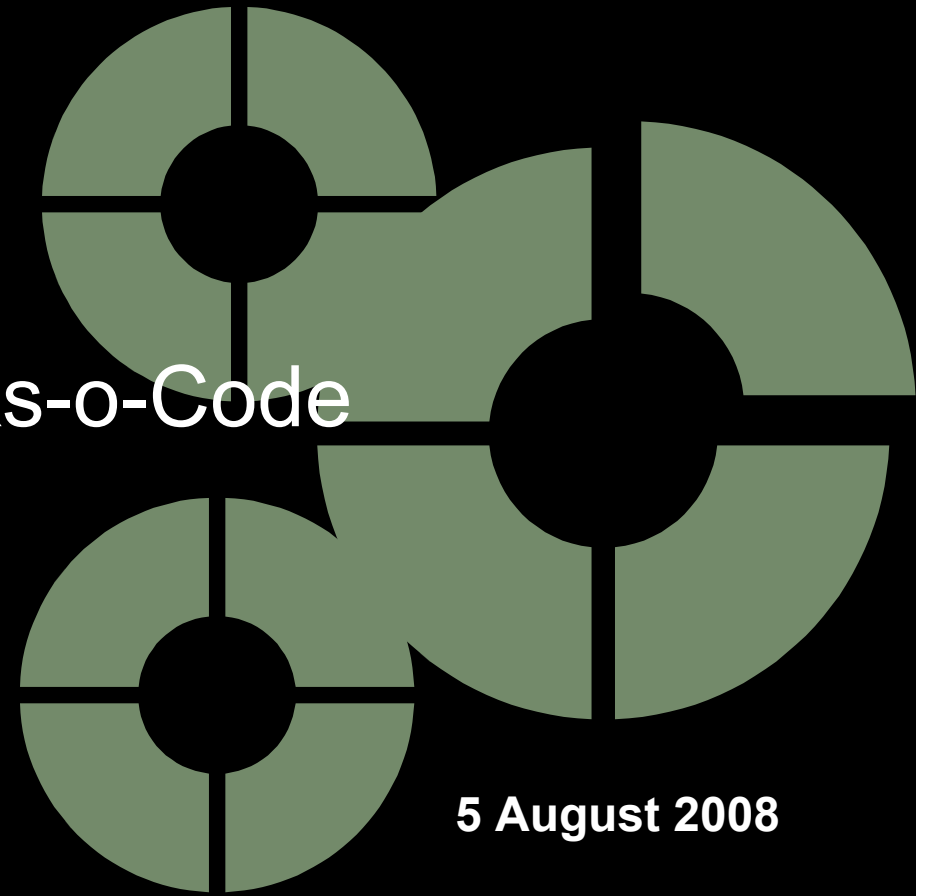


Code Scavenging Tasty Bite-Sized Chunks-o-Code



5 August 2008

krugle

The Dark Side of the Source

- › What is code scavenging?
- › Know when to say when
- › The successful scavenger

What is code scavenging?

- › Programming via copy & paste
 - » [ref to blog post]
 - » Small snippets of source code
 - » Not forking or reusing components
- › Useful for quick & dirty jobs
 - » Gluing systems together
 - » Trying out an API
 - » Programming chores
 - » Debugging aids

Know when to say when

› Reasons **not** to scavenge

- » Maintenance cost
- » Licensing issues
- » It's not a snippet
- » Component exists

› Reasons to scavenge

- » Available component size/complexity
- » Too hard/risky to refactor closest match
- » No authority to refactor closest match
- » Best option given ROI calculation

Be a good scavenger

- › Define success first
- › Know where to search
- › Know how to search
- › Practice good hygiene

Define success first

- › Crisp “I need to do xxx”
 - » Avoid preconceived constraints
 - » Is xxx really necessary?
- › Consider non-code solutions
 - » Grep, awk, sed
 - » Spreadsheets, DBs, techie text editor
 - » [interview questions]
- › Avoid unnecessary searching
 - » Ask the resident expert
 - » Check nearby references

Know where to search

- › Stuff on your hard drive
 - » Build your own snippet library
 - » Standard grep/IDE search
 - » Desktop search
- › Inside the firewall
 - » Generic search appliance
 - » Code search tools
- › OpenSource
 - » Generic search engines (aka Google)
 - » Code search engines
 - » Snippet libraries

Searching inside the firewall

- › Generic search appliances
 - » Google, OmniFind, etc.
 - » Exposing the source for crawling
- › Code search tools
 - » OpenGrok, CTAGs, codecrawler, etc.
 - » Krugle, Koders

Searching for OpenSource

- › Generic search engines (aka Google)
 - » Indirect
 - » Noisy
 - » Broad
- › Code search engines
 - » Krugle.org, Koders.com, google.com/codesearch
- › Repository sites
 - » CPAN, etc.
 - » Sometimes uncrawled
 - » Narrower focus
 - » Often limited search support
- › Snippet libraries
 - » [list of examples]
 - » O'Reilly, CodeFetch

Know how to search

- › Semantic search is hard
 - » Level of commenting sucks
 - » Comments talk about how not what
 - » Metadata typically not very useful
- › Focus on API usage
 - » Visualize how expert would do it
 - » Search on multiple API names
 - » Restrict by package names
 - » Try synonyms (equivalent APIs)

Know how to search (cont.)

- › Those pesky licenses
 - » There's lots of FUD
 - » Everyone should have basic understanding
 - » Most of the time you don't care
 - › Except if you ship your product, or
 - › It's a "pseudo open license"
- › Appropriate LOE
 - » Find container of code (project, book, etc.)
 - » Double-check reported license
 - › README, COPYING, License.txt
 - › Source headers

Know how to search (cont.)

- › Code quality is hard to assess
 - » Standard metrics don't exist
 - » No independent rating authority
 - » True assessment requires usage
- › Four quick checks
 - » Any comments at all?
 - » Any test code?
 - » Host authority
 - » Project popularity
- › Know when to give up

Practice good hygiene

- › Source attribution
 - » Host site, project/author
- › Include license information
 - » Notes on where it came from
- › Include TODO notes
 - » If I had more time I'd...

Summary

- › Define success first
- › Know where to search
- › Know how to search
- › Practice good hygiene