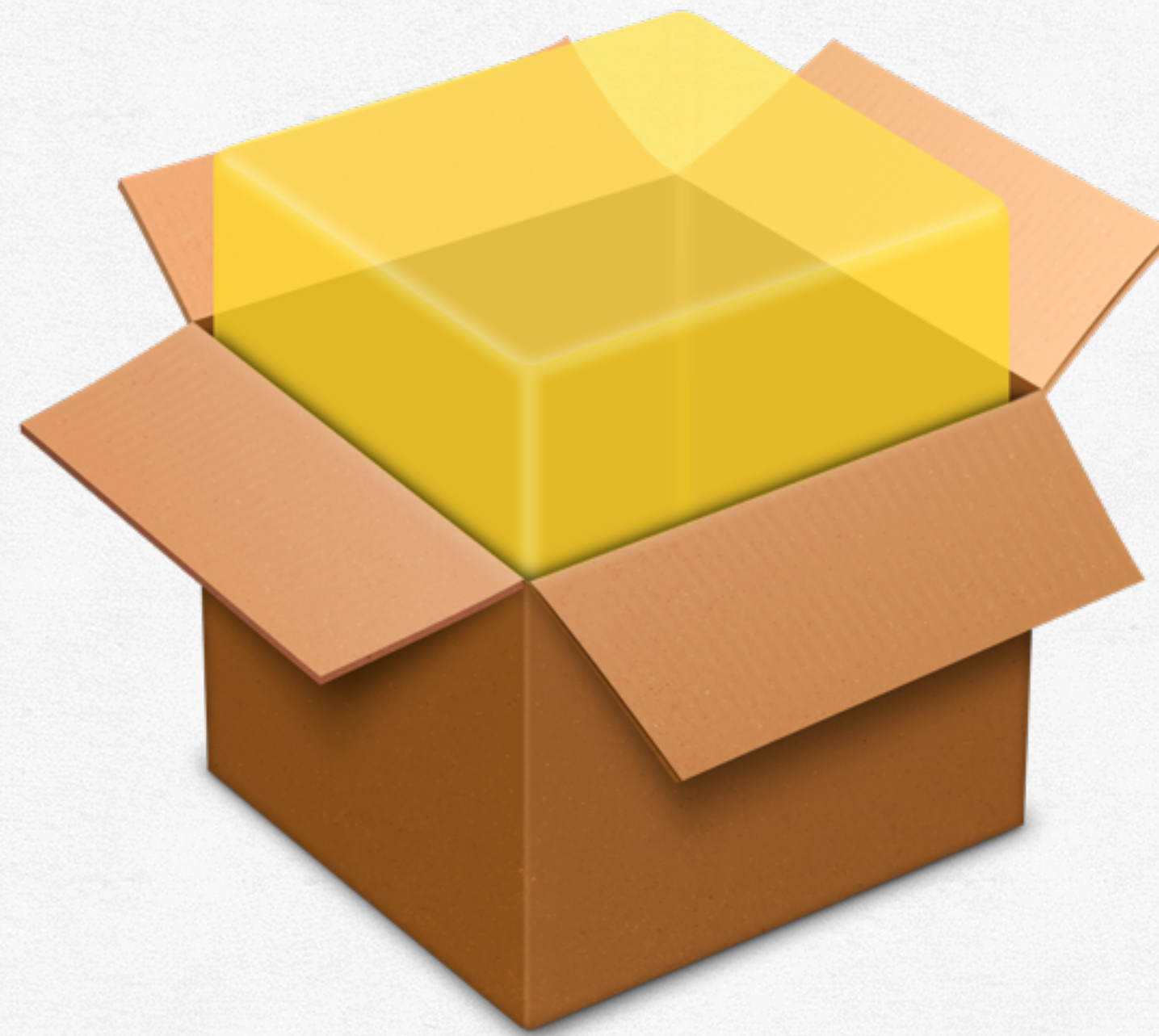


OS X Packaging



The Opinions

- They're not dead ... yet
- Why would you need to know this?
- Things that have been covered

The Statistics

- Current: 300 Vendors, 700 Titles, 1200 Packages
- 2014: 3500 Packages Built, 20000+ Delivered
- 2015: 2600 Packages Built, 11000+ Delivered so far
- What are our Packages are used with?

The References

- Distribution Definition XML Schema Reference

<https://developer.apple.com/library/mac/documentation/DeveloperTools/Reference/DistributionDefinitionRef/Chapters/Introduction.html>

- Installer JavaScript Reference

https://developer.apple.com/library/mac/documentation/DeveloperTools/Reference/InstallerJavaScriptRef/index.html#//apple_ref/doc/uid/TP40005137

- Software Delivery Legacy Guide

<https://developer.apple.com/legacy/library/documentation/DeveloperTools/Conceptual/SoftwareDistribution4/SoftwareDistribution4.pdf>

The (OS X) Tools

productbuild

pkgutil

pkgbuild

productsign

What we'll Cover

- Build a 'simple' thing
- Consider some options
- Make it complicated
- Deconstruct and Existing Package

productbuild

productbuild -- Build a product archive for the OS X Installer or the Mac App Store

In simplest (useful) form:

productbuild {**--component** component-path [install-path]} product-output-path

Demo

Minimum Viable Product

pkgutil

pkgutil -- Query and manipulate Mac OS X Installer packages and receipts.

In simple form:

pkgutil --expand pkg-path dir-path

pkgutil --flatten dir-path pkg-path

What did productbuild do?

- The Distribution
- The Package
- The Resources

What didn't productbuild do?

- The Distribution requirements
- The Package pre/post tasks

To Configure or Not to Configure?

- Will it make your life easier?
- Can it be done with a profile?

pkgbuild

pkgbuild -- Build an OS X Installer component package from on-disk files

In simplest (useful) form:

pkgbuild [options] {**--component** component-path} package-output-path

Demo

Raw Package

Demo

Add Raw Package to Distribution

Installation Requirements

- The `<domains>` tag
- The `<must-close>` tag
- etc.

Demo

Using the `<must-close>` tag in the GUI

What if...

Demo

Using the `<must-close>` tag in the CLI

Demo

Another way to Quit

productsign

productsign -- Sign an OS X Installer product archive

In simple form:

productsign --sign identity input-product-path output-product-path

Demo

Signing a Package

An Existing Package

- Check the Structure
- Check the Distribution
- Check the Payloads
- Check any pre/post tasks

Microsoft Office 2016

Structure

- Is it flat?
- Are the components in the correct place?

Distribution

- Does it use deprecated tags?
- Are tags located in the wrong place?
- What Are the Requirements?

Payloads

- What's it installing and where?

Scripts

- What does it do pre-installation?
- What does it do post-installation?
- Are there any calls to 'userland'?

BIG Demo

Let's fix it

Thank you

www.mondada.com.au