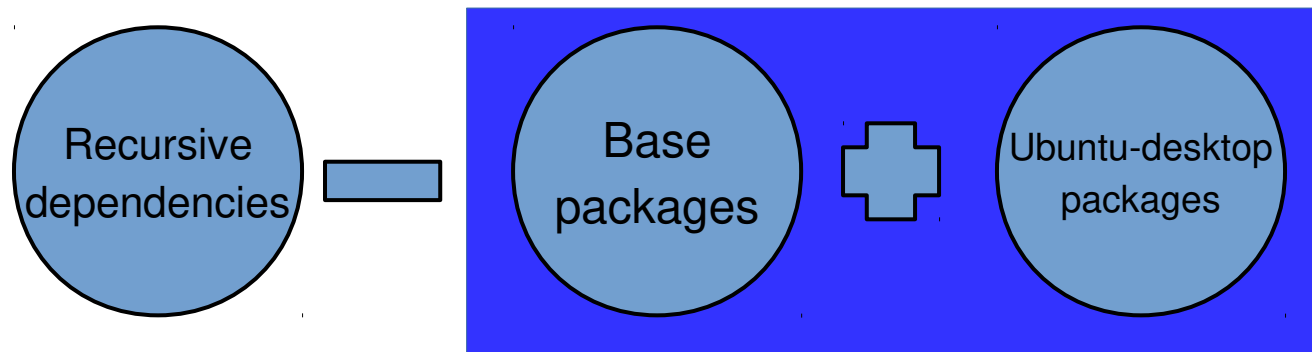


# HPE Interview Presentation

Joseph Bisch  
2016-03-10

# Armory Offline Bundle Creation Automation



```
depends = ['swig', 'libqtcore4', 'python-qt4', 'python-twisted',
          'python-psutil']
result, retval = execAndWait('apt-get -c etc/apt.conf update')
if retval != 0:
    print 'apt-get update returned error %d' % retval
    exit(1)
recursiveDepends = []
for deb in depends:
    recursiveDepends += getDepends(deb)
defaultPackages = getDefaultPackages()
recursiveDepends = [x for x in recursiveDepends if x not in defaultPackages]
# Ensure recursiveDepends only has each package once
recursiveDepends = list(set(recursiveDepends))
for deb in recursiveDepends:
    result, retval = execAndWait('apt-get -c etc/apt.conf download %s:%s' % (deb, arch))
    if retval != 0:
        print 'apt-get download returned error %d' % retval
        exit(1)
```

# Armory Offline Bundle Creation Automation

- GetDepends returns list of recursive dependencies
- GetDefaultPackages returns list of all the base packages plus the packages that make up the ubuntu-desktop meta package

```
def getDepends(deb):
    exclusions = '--no-recommends --no-suggests --no-conflicts --no-breaks --no-replaces --no-enhances --no-pre-depends'
    depends, retval = execAndWait('apt-cache -c etc/apt.conf depends --recurse %s %s | grep Depends | awk \'{print $2}\\\' * ('
    depends = depends.split('\n')
    # return list of non-virtual packages
    return [x for x in depends if x and not x[0] == '<']

def getDefaultPackages():
    desktopPackages = getDepends('ubuntu-desktop')
    # need to read suite from sources.list instead of hardcoding precise
    basePackages, retval = execAndWait('debootstrap --print-debs precise temp.tar.gz 2>/dev/null')
    basePackages = basePackages.split(' ')
    return desktopPackages + basePackages
```