



OpenSolaris
<User group>

Automated Installer: The Good, The Bad and The Ugly

damn right

aka: “Step-by-Step guide for AI”

Andrew Watkins

andrew@dcs.bbk.ac.uk

<http://notallmicrosoft.blogspot.com>

Birkbeck College

Department of Computer Science
and Information Systems

LOSUG

April 21st, 2010



Automated Installer (2009.06)

- Automatic installation of OpenSolaris on x86 or SPARC
- Requires a AI server and DHCP server
- Requires access to IPS repository
- Network boot or modified GRUP menu
- AI manifest file
 - ✓ disk target
 - ✓ ips repository location
 - ✓ packages
- System Configuration manifest
 - ✓ hostname, timezone, root password
- Criteria manifest, install only on given clients
 - ✓ network, MAC address
 - ✓ disk and memory size



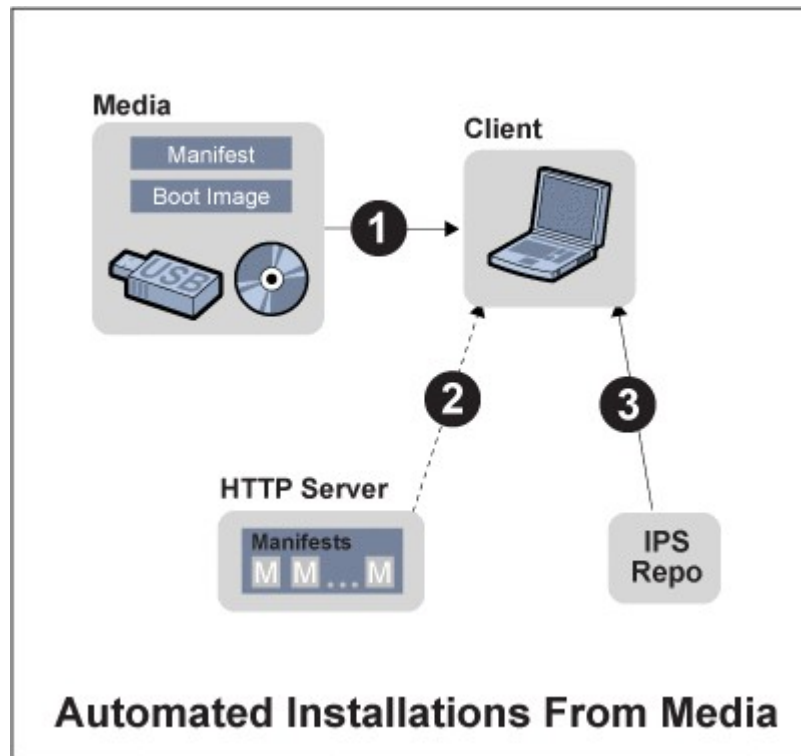
Modified Grub

```
default=0
timeout=30
min_mem64=1536
title OpenSolaris 2009.06
  debug
  verbose
  root (hd0,0,a)
  dhcp
  root (nd)
  kernel$ /i86PC.OpenSolaris-1/platform/i86pc/kernel/
  $ISADIR/unix -B
    install_media=http://193.x.x.x:5555/export/osol-0906-ai-x86,
    install_service=0906x86,
    install_svc_address=193.xx.xx.xx:46501,livemode=text -v

  module /i86PC.OpenSolaris-1/x86.microroot
```

New in Automated Installer (2010.03)

- Bootable AI Server (CD, DVD or USB)



- Still require access to IPS repository



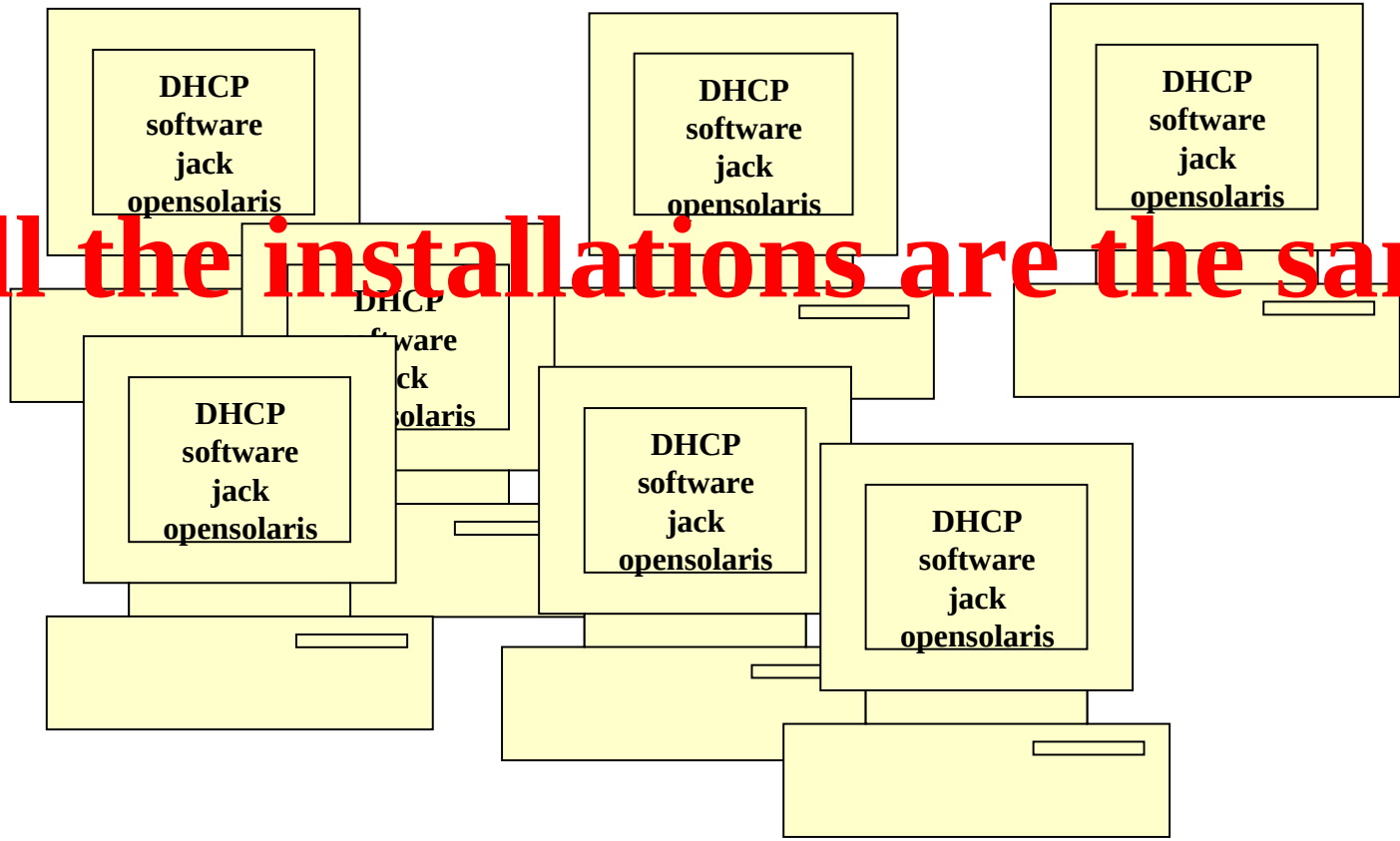
New in Automated Installer (2010.03)

- **Changes with AI server**
 - ✓ **installadm list - lot more information**
 - ✓ **create, modify extended partitions**
 - ✓ **more disk control**
(boot disk, dev id, volume id, etc)
 - ✓ **install on iSCSI**



Automated Installer (2009.06)

All the installations are the same





What is missing

➤ sysidcfg (on first boot)

- ✓ hostname, network (static, dhcp),
- ✓ Name Services
 - ✓ DNS, NIS, LDAP

➤ sys-unconfig

- ✓ hostname, IP address, DNS, NIS, LDAP

➤ finish script

- ✓ add user, add software
- ✓ your own modifications
 - ✓ sendmail.cf, cron, openoffice printers, auto_mount, gdm, etc....

System Configuration Project

Write your own package



A complete hands free install Build 134

- A test environment (VirtualBox)
- AI Server Software
- Local copy of OpenSolaris packages
 - ✓ mirror or full.iso
- Modified AI manifest
- Your own repository for your software
- Build your own packages
 - ✓ software and SMF
- Full demo available soon after 2010.0? release
<http://notallmicrosoft.blogspot.com>



Setup up AI Server

- **static IP address (vboxnet0: 192.168.56.1)**
- **download / install AI (osol-dev-134-ai-x86.iso)**
 - # pkg install install/installadm
 - Replaces the old SUNWinstalladm-tools
- **Setup AI server and Install DHCP**
 - # installadm create-service -n 201003x86 -i 172.1.0.10 -c 5 \
-s /export/aiimages/osol-dev-134-ai-x86.iso \
/export/aiserver/osol-dev-134-ai-x86
- **Setup AI server and using existing DHCP**
 - # svcadm enable svc:/network/dns/multicast:default
 - # installadm create-service -n 201003x86 \
-s /export/aiimages/osol-dev-134-ai-x86.iso \
/export/aiserver/osol-dev-134-ai-x86

AI Server (demo)

```
$ pfexec pkg install install/installadm
```

```
$ pfexec svcadm enable svc:/network/dns/multicast:default
```

```
$ mkdir /export/aimages /export/aiserver
```

```
$ pfexec installadm create-service -n 201003x86 \
  -s /export/aimages/osol-dev-134-ai-x86.iso \
  /export/aiserver/osol-dev-134-ai-x86
```

Setting up the target image at /export/aiserver/osol-dev-134-ai-x86 ...

Registering the service 201003x86._OSInstall._tcp.local

Detected that DHCP is not set up on this server.

If not already configured, please create a DHCP macro
named dhcp_macro_201003x86 with:

```
    Boot server IP (BootSrvA) : 192.168.56.1
```

```
    Boot file      (BootFile)  : 201003x86
```

```
    GRUB Menu     (GrubMenu)   : menu.lst.201003x86
```

If you are running the Solaris DHCP Server, use the following
command to add the DHCP macro, dhcp_macro_201003x86:

```
.....
```

(e.g., if running the Solaris DHCP Server, run pntadm(1M)).

```
adding tftp to /etc/inetd.conf
```

```
Converting /etc/inetd.conf
```

```
copying boot file to /tftpboot/pxegrub.I86PC.OpenSolaris-1
```

```
Service discovery fallback mechanism set up
```



AI Server (demo)

```
$ mkdir $HOME/.VirtualBox/TFTP
```

Specify where TFTP boot server is:

```
$ VBoxManage setextradata "ai_client"  
"VBoxInternal/Devices/e1000/0/LUN#0/Config/NextServer"  
192.168.56.1
```

```
$ VBoxManage getextradata "ai_client" enumerate  
Key: VBoxInternal/Devices/e1000/0/LUN#0/Config/NextServer, Value: 192.168.56.1
```

Virtualbox TFTP search for specific filenames:
clientname.pxe & menu.lst.clientname.pxe

```
$ cd /tftpboot  
$ pfexec ln -s pxegrub.I86PC.OpenSolaris-1 ai_client.pxe  
$ pfexec ln -s menu.lst.201003x86 menu.lst.ai_client.pxe  
$ ls -l  
..          201003x86 -> pxegrub.I86PC.OpenSolaris-1  
..          I86PC.OpenSolaris-1  
..          menu.lst.201003x86  
..          pxegrub.I86PC.OpenSolaris-1  
..          rm.201003x86
```



Local copy of IPS software

- **osol-repo-0906-full.iso (7GB)**
- **osol-repo-20100?-full.iso (?GB)**
- **Create a copy of pkg.opensolaris.org**

Local copy of IPS software (demo)

```
$ pfexec zfs create -o compression=on -o atime=off  
  rpool/export/pkg  
$ pfexec lofiadm -a /export/aiimages/osol-repo-0906-  
  full.iso  
/dev/lofi/1  
$ pfexec mount -F hsfs /dev/lofi/1 /mnt  
$ pfexec rsync -aP /mnt/repo /export/pkg  
  
$ pfexec vi /export/pkg/repo/cfg_cache  
  replace  
  origins = http://pkg.opensolaris.org/release  
  with hostname or IP address of this server  
  origins = http://192.168.56.1
```


Local copy of IPS software (demo)

```
$ pfexec svccfg -s application/pkg/server setprop  
  pkg/inst_root=/export/pkg/repo
```

```
$ pfexec svccfg -s application/pkg/server setprop  
  pkg/readonly=true
```

```
$ pfexec svcadm refresh pkg/server
```

```
$ pfexec svcadm enable pkg/server
```

<http://localhost> or <http://192.168.56.1>

Local Copy of IPS Software (demo)

```
$ pfexec zfs create -o compression=on -o atime=off \  
    rpool/export/pkg
```

```
$ cd /export/pkg
```

Create repository environment (Build 134 & pkgs installed)

```
$ pfexec pkgsend -s file:///export/pkg create-repository \  
    -set-property publisher.prefix=opensolaris.org
```

```
$ pfexec pkg list -Hva | egrep "134:/install" > pkglist.txt
```

```
$ head -1 pkglist.txt
```

```
pkg://opensolaris.org/SUNWcs@0.5.11,5.11-0.134:20100302T005446Z
```

```
$ pfexec cat pkglist.txt | sed 's/pkg:\\\\opensolaris\\.org\\///' \  
    | cut -f 1 -d ' ' > packages.txt
```

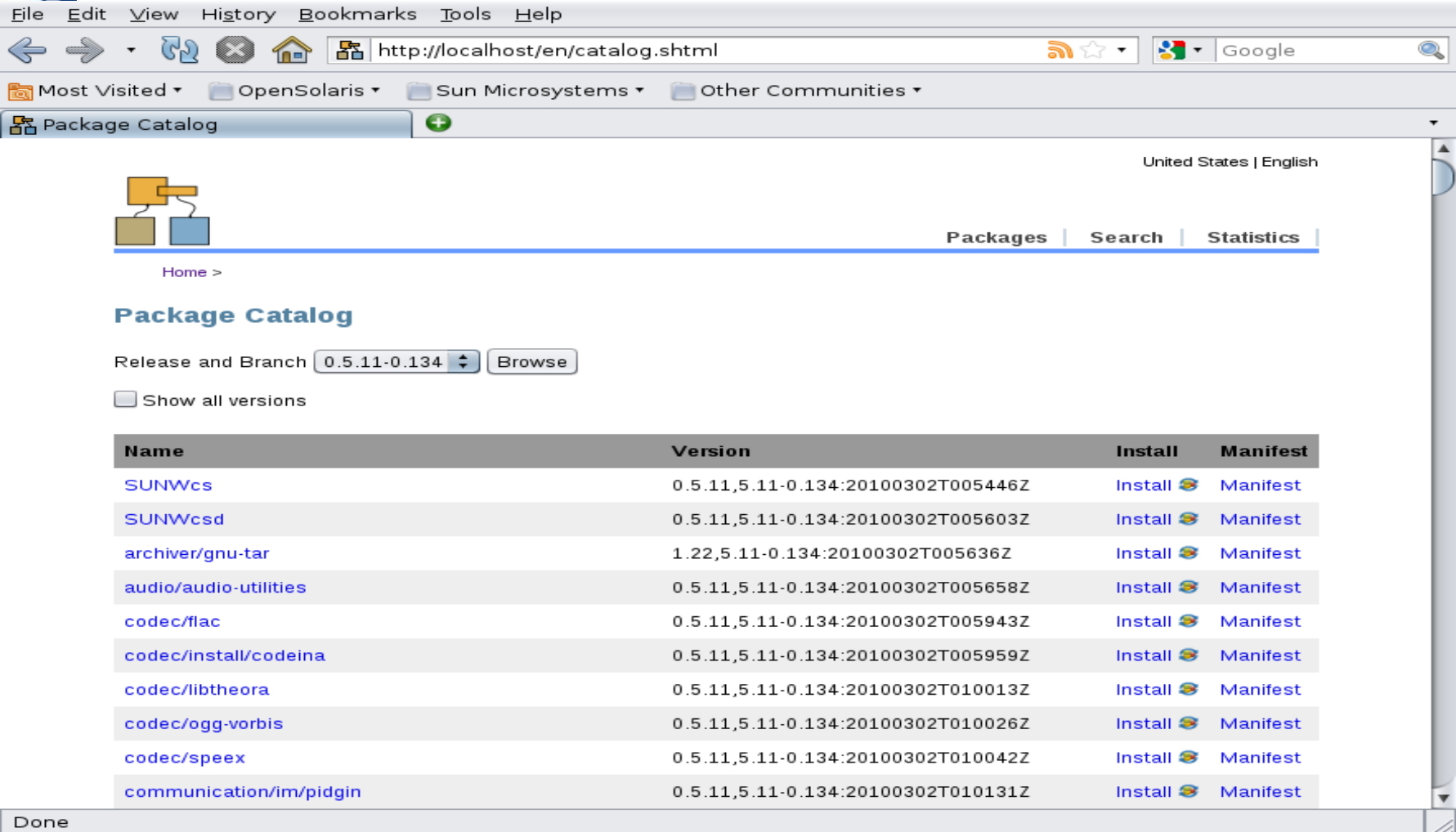
```
$ head -1 packages.txt
```

```
SUNWcs@0.5.11,5.11-0.134:20100302T005446Z
```

```
$ for i in $(cat packages.txt); do pfexec pkgrecv -s  
http://pkg.opensolaris.org/dev -d file:///export/pkg "$i"; done
```

```
$ pfexec /usr/lib/pkg.depotd -d /export/pkg -p 80
```

Copy of pkg.opensolaris.org



File Edit View History Bookmarks Tools Help

http://localhost/en/catalog.shtml

Most Visited OpenSolaris Sun Microsystems Other Communities

Package Catalog

United States | English

Home >

Package Catalog

Release and Branch

Show all versions

Name	Version	Install	Manifest
SUNWcs	0.5.11,5.11-0.134:20100302T005446Z	Install	Manifest
SUNWcsd	0.5.11,5.11-0.134:20100302T005603Z	Install	Manifest
archiver/gnu-tar	1.22,5.11-0.134:20100302T005636Z	Install	Manifest
audio/audio-utilities	0.5.11,5.11-0.134:20100302T005658Z	Install	Manifest
codec/flac	0.5.11,5.11-0.134:20100302T005943Z	Install	Manifest
codec/install/codeina	0.5.11,5.11-0.134:20100302T005959Z	Install	Manifest
codec/libtheora	0.5.11,5.11-0.134:20100302T010013Z	Install	Manifest
codec/ogg-vorbis	0.5.11,5.11-0.134:20100302T010026Z	Install	Manifest
codec/speex	0.5.11,5.11-0.134:20100302T010042Z	Install	Manifest
communication/im/pidgin	0.5.11,5.11-0.134:20100302T010131Z	Install	Manifest

Done



AI server

- **AI manifest**
 - **XML: defines client installation**
 - ✓ **i.e. disk partition and software**



AI manifest (example)

```
<ai_device_partitioning>
  <partition_action>create</partition_action>
  <partition_type>SOLARIS</partition_type>
  <partition_size>20</partition_size>
  <partition_size_units>GB</partition_size_units>
</ai_device_partitioning>
and
<ai_target_device>
  <target_device_iscsi_target_name>iqn.1986-03.com.sun:02: \
1533d-2d61-c198-b4eaa9e3c40a49dd</target_device_iscsi_target_name>
  <target_device_iscsi_target_ip>10.0.0.1
    </target_device_iscsi_target_ip>
  <target_device_iscsi_target_port>3260
    </target_device_iscsi_target_port>
  <target_device_iscsi_target_lun>0
    </target_device_iscsi_target_lun>  </ai_target_device>
```



AI server

- **System Configuration manifest**
 - **defines system information**
 - ✓ **timezone, root password, add user**

```
<property_group name='ai' type='application'>
  <propval name='username' type='astring' value='jack' />
  <propval name='description' type='astring' value='default user
  <propval name='userpass' type='astring' value='9Nd/cwBcNWFZg'
  <propval name='rootpass' type='astring' \
value=' $5$VgppCOxA$ycFmYW4ObRRHhtsGEygDdexk5bugqgSiaSR9niNCouC' />
  <propval name='timezone' type='astring' value='US/Pacific' />
  <propval name='hostname' type='astring' value='opensolaris' />
</property_group>
```



AI server

- **Criteria manifest**
 - **defines which client gets installed**

```
<ai_criteria name="MAC">  
  <value>0:14:4F:20:53:94</value>  
</ai_criteria>
```

or

```
<ai_criteria name="ARCH">  
  <value>i86pc</value>  
</ai_criteria>
```

or

```
<ai_criteria name="MEM">  
  <range>2048 4096</range>  
</ai_criteria>
```




Modified AI manifest (demo)

```
$ cd /export/aiserver/osol-dev-134-ai-x86/auto_install
```

```
$ pfexec installadm list
```

Service Name	Status	Arch	Port	Image Path
-----	-----	----	----	-----
201003x86	on	x86	46501	/export/aiserver/osol-dev-134-ai-x86

```
$ pfexec cp default.xml demo.xml
```

- change default repository
- change sc manifest. i.e. hostname
- Optional:
 - add criteria and change manifest name. i.e demo
 - add disk control. i.e. partition size
 - add extra software. i.e. openoffice
 - add another repository



Modified AI manifest (demo)

```
$ cat demo.xml
```

```
<ai_criteria_manifest>  
  <ai_criteria name="arch">  
    <value>i86pc</value>  
  </ai_criteria>  
  <ai_embedded_manifest>  
    <ai_manifest name="demo">  
      <ai_device_partitioning>  
        <partition_action>create</partition_action>  
        <partition_type>SOLARIS</partition_type>  
        <partition_size>max_size</partition_size>  
        <partition_size_units>GB</partition_size_units>  
      </ai_device_partitioning>  
    <ai_pkg_repo_default_publisher>  
      <main url="http://192.168.56.1" publisher="opensolaris.org"/>  
      <mirror url=""/>  
    </ai_pkg_repo_default_publisher>  
    <ai_pkg_repo_addl_authority>  
      <main url="http://192.168.56.1:9000" authname="local"/>  
    </ai_pkg_repo_addl_authority>  
  </ai_embedded_manifest>  
</ai_criteria_manifest>
```

Modified AI manifest (demo)

```
<ai_install_packages>
  <pkg name="entire"/>
  <pkg name="SUNWcsd"/>
  <pkg name="SUNWcs"/>
  <pkg name="babel_install"/>
<!-- You can add more packages if you want -->
<!-- <pkg name="openoffice"/> -->
  <pkg name="FINISHpkg"/>
  <pkg name="network/iscsi/initiator"/>
  <pkg name="network/iscsi/iser"/>
</ai_install_packages>
<ai_uninstall_packages>
  <pkg name="babel_install"/>
  <pkg name="slim_install"/>
</ai_uninstall_packages>
  <ai_auto_reboot>
    false
  </ai_auto_reboot>
</ai_manifest>
</ai_embedded_manifest>
```



Modified AI manifest (demo)

```
<sc_embedded_manifest name = "AI">
  <!-- <?xml version='1.0'?>
  <!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd" >
  <service_bundle type="profile" name="name">
    <service name="ai_properties" version="1" type="service">
      <instance name="default" enabled="true">
        <property_group name="ai" type="application">
          <propval name="username" type="astring" value="andrew"/>
          <!-- Using default password "jack"-->
          <propval name="userpass" type="astring" value="9Nd/cwBcNWFZg"/>
          <propval name="description" type="astring" value="default_user"/>
          <!-- Using default root password "opensolaris"-->
          <propval name="rootpass" type="astring"
value="$5$VgppC0xA$ycFmYW40bRRHhtsGEygDdexk5bugqgSiaSR9niNCouC"/>
          <propval name="timezone" type="astring" value="GB"/>
        </property_group>
      </instance>
    </service>
  </service_bundle>
-->
</sc_embedded_manifest>

</ai_criteria_manifest>
```



Modified AI manifest (demo)

```
$ pfexec /usr/sbin/installadm add -m demo.xml -n 201003x86
```

```
$ pfexec installadm list
```

Service Name	Status	Arch	Port	Image Path
-----	-----	----	----	-----
201003x86	on	x86	46501	/export/aiserver/osol-dev-134-ai-x86

```
$ pfexec installadm list -n 201003x86 -m
```

```
Manifest Criteria  
-----  
demo.xml arch = i86pc
```



Your own repository for your software

- Create another repository for local packages
- Need to decide what alterations you require to standard build



local Repository (demo)

```
$ pfexec svccfg -s pkg/server
svc:/application/pkg/server> add local
svc:/application/pkg/server> select local
svc:/application/pkg/server:local> addpg pkg application
svc:/application/pkg/server:local> addpg start method
svc:/application/pkg/server:local> setprop start/exec= astring:
    "/usr/lib/pkg.depotd
    -p %{pkg/port} -d %{pkg/inst_root}
    -t %{pkg/socket_timeout} -s %{pkg/threads}
    --log-access=%{pkg/log_access}
    --log-errors=%{pkg/log_errors}"
svc:/application/pkg/server:local>
    setprop pkg/inst_root = astring: "/export/localpkg"
svc:/application/pkg/server:local>
    setprop pkg/threads = count: 50
svc:/application/pkg/server:local>
    setprop pkg/port = count: 9000
svc:/application/pkg/server:local> exit
```

local Repository (demo)

```
$ pfexec mkdir /export/localpkg
$ pfexec pkgsend -s file:///export/localpkg
  create-repository
  --set-property publisher.prefix=local
```

```
$ pfexec svcadm refresh pkg/server:local
$ pfexec svcadm enable pkg/server:local
```

```
$ svcs pkg/server
```

STATE	STIME	FMRI
online	12:45:40	svc:/application/pkg/server:local
online	12:44:12	svc:/application/pkg/server:default

<http://localhost:9000> or <http://192.168.56.1:9000>

Build your own packages

➤ Build your own packages

✓ Software

`/etc/hosts.allow`

`/etc/hosts.deny`

`/etc/mail/sendmail.cf`

`/usr/lib/firefox/plugins/libflashplayer.so`

`/var/local/.ldapclient`

`/var/svc/manifest/system/ai-finish.xml`

`/lib/svc/method/ai-finish`

✓ SMF

***Transient* services are often configuration services, which require no long-running processes to provide service. Common transient services take care of boot-time cleanup or load of configuration properties into the kernel.**





Build your own packages (demo)

```
$ cat FINISHpkg.ips
set name=pkg.namevalue="BBKaiinstall"
set name=pkg.description value="BBK setup files for lab machines"
dir mode=0755 owner=root group=root path=/etc/mail
dir mode=0755 owner=root group=root path=/etc/mail/cf
dir mode=0755 owner=root group=root path=/etc/mail/cf/cf
dir mode=0755 owner=root group=bin path=/lib
dir mode=0755 owner=root group=bin path=/lib/svc
dir mode=0755 owner=root group=bin path=/lib/svc/method
dir mode=0755 owner=root group=root path=/usr
dir mode=0755 owner=root group=root path=/usr/lib
dir mode=0755 owner=root group=root path=/usr/lib/firefox
dir mode=0755 owner=root group=root path=/usr/lib/firefox/plugins
dir mode=0755 owner=root group=sys path=/var
dir mode=0700 owner=root group=root path=/var/local
dir mode=0755 owner=root group=sys path=/var/svc
dir mode=0755 owner=root group=sys path=/var/svc/manifest
dir mode=0755 owner=root group=sys path=/var/svc/manifest/system
file etc/hosts.allow mode=0644 owner=root group=root path=/etc/hosts.allow
file etc/hosts.deny mode=0644 owner=root group=root path=/etc/hosts.deny
file etc/mail/sendmail.cf mode=0444 owner=root group=bin path=/etc/mail/sendmail.cf
file lib/svc/method/ai-finish mode=0754 owner=root group=root path=/lib/svc/method/ai-finish
file usr/lib/firefox/plugins/libflashplayer.so mode=0755 owner=root group=bin
  path=/usr/lib/firefox/plugins/libflashplayer.so
file var/local/.ldapclient mode=0600 owner=root group=root
  path=/var/local/.ldapclient
file var/svc/manifest/system/ai-finish.xml mode=0644 owner=root
  group=sys path=/var/svc/manifest/system/ai-finish.xml
```



Build your own packages (demo)

```
$ cat var/svc/manifest/system/ai-finish.xml
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1"
<service_bundle type='manifest' name='ai-finish'>
<service
  name='system/ai-finish' type='service' version='1'>
  <property_group name='startd' type='framework'>
    <propval name='duration' type='astring' value='transient' />
  </property_group>
  <instance name='install' enabled='true'>
    <dependency name='filesystem-local' grouping='require_all'
      restart_on='none' type='service'>
      <service_fmri value='svc:/system/filesystem/local:default' />
    </dependency>
    <dependent name='ai-finish_multi-user' grouping='optional_all'
      restart_on='none'>
      <service_fmri value='svc:/milestone/multi-user' />
    </dependent>
    <exec_method type='method' name='start'
      exec='/lib/svc/method/ai-finish %i' timeout_seconds='0' />
    <exec_method type='method' name='stop'
      exec=':true'
      timeout_seconds='0' />
  </instance>
</service> </service_bundle>
```




Build your own packages (demo)

```
$ cat lib/svc/method/ai-finish
#!/bin/sh -x

. /lib/svc/share/smf_include.sh

LC_ALL=C; export LC_ALL
MYHOST=`/usr/bin/hostname`
REBOOT="no"
RETVAL=$SMF_EXIT_OK

start() {
#   Setup up as LDAP client
#   modify: /etc/defaultdomain /etc/nsswitch.conf /var/ldap
#   /usr/sbin/ldapclient init \
#       proxyDN=cn=proxyagen,ou=profile,dcs=bbk,dc=ac,dc=uk \
#       domainname=bbk.ac.uk -y /var/local/.ldapclient 193.xx.xx.xx

# Add Shutdown to GDM login
if [ -s "/etc/user_attr" ]; then
    echo "gdm::::type=normal;auths=solaris.system.shutdown" >>
    /etc/user_attr
fi
}
```




Build your own packages (demo)

```
case "$1" in
'install')
    start;

    # Disable this service so that it doesn't run again.
    /usr/sbin/svcadm disable system/ai-finish

    if [ "$REBOOT" = "yes" ]; then
        echo "Rebooting"; ( sleep 30; init 6 ) &
    fi
    echo "Finished"

    if [ $? -ne 0 ] ; then
        exit $SMF_EXIT_ERR_CONFIG
    fi ;;

*)
    echo "Usage: $0 { install }"
    exit $SMF_EXIT_ERR_CONFIG ;;

esac

exit $SMF_EXIT_OK
```

Modify client to send hostname to DHCP server

```
$ cat lib/svc/method/ai-finish
....
install() {
  # Disable nwam & enable dhcp so that it sends out hostname to DHCP server
  if [ -s "/etc/nwam/llp" ]; then
    #If nwam has found a interface it will populate the above file so we will
    use it
    echo "disable nwam"

    svcadm disable svc:/network/physical:nwam
    INTERFACE=`awk ' { print $1 } ' /etc/nwam/llp`
    echo "Found interface: $INTERFACE"

    if [ -n "$INTERFACE" ]; then
      REBOOT="yes"
      echo "inet $MYHOST" > /etc/hostname.$INTERFACE

      cp /etc/default/dhcpagent /tmp
      sed -e s/#.*CLIENT_ID=/CLIENT_ID=$MYHOST/ \
        -e 's/#.*REQUEST_HOSTNAME=no/REQUEST_HOSTNAME=yes/' \
        /tmp/dhcpagent > /etc/default/dhcpagent

      touch /etc/dhcp.$INTERFACE

      echo "Starting network/physical:default"
      svcadm enable svc:/network/physical:default
    fi
  fi
}
```



Build your own packages (demo)

```
$ eval `pfexec
  pkgsend -s http://localhost:9000 open FINISHpkg@1.0-0`

$ pfexec
  pkgsend -s http://localhost:9000 include FINISHpkg.ips

$ pfexec pkgsend -s http://localhost:9000 close
PUBLISHED
pkg://local/FINISHpkg@1.0,5.11-0:20100413T152913Z
```

**Start Virtualbox client
and
wait to see if it works....**



References



My Blog

<http://notallmicrosoft.blogspot.com>



LOSUG:

<http://hub.opensolaris.org/bin/download/User+Group+losug/w-2009/ips-revisited.pdf>

<http://hub.opensolaris.org/bin/download/User+Group+losug/w-2009/AI-presentation-losug.p>



System Configuration Project

<http://hub.opensolaris.org/bin/view/Project+caiman/System+Configuration+Project>



Other:

http://blogs.sun.com/middle/entry/establishing_reduced_package_repository_for

<http://dlc.sun.com/osol/docs/content/2009.06/AIinstall/index.html>

<http://dlc.sun.com/osol/docs/content/2009.06/IMGPACKAGESYS/index.html>

<http://www.agileweboperations.com/setting-up-an-opensolaris-pkg-repository-mirror>