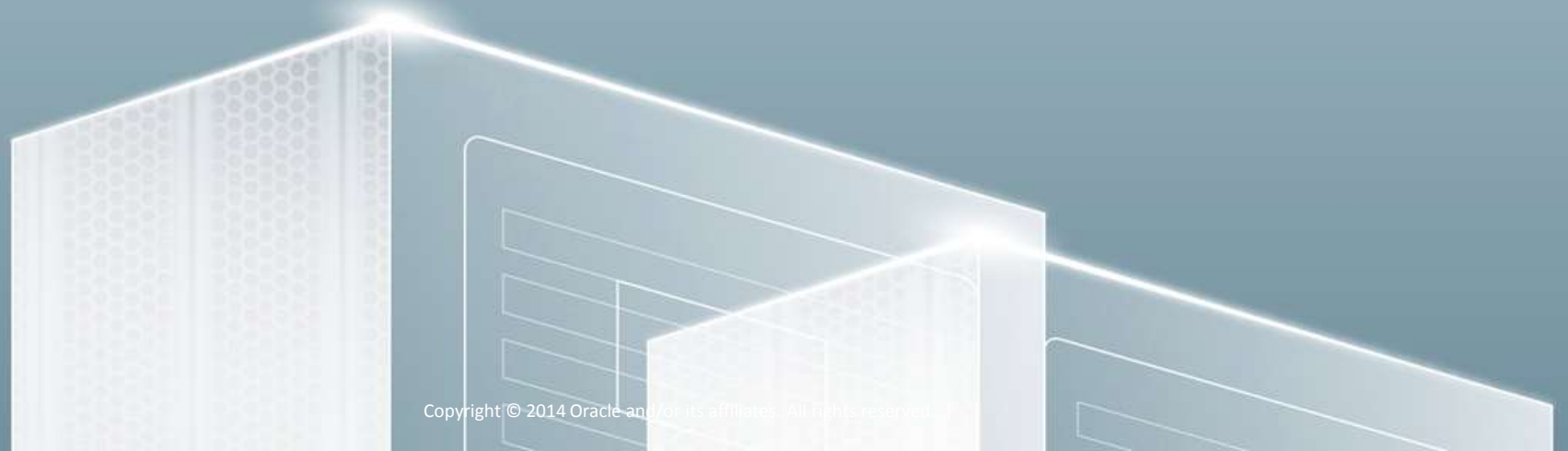

ORACLE®

Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

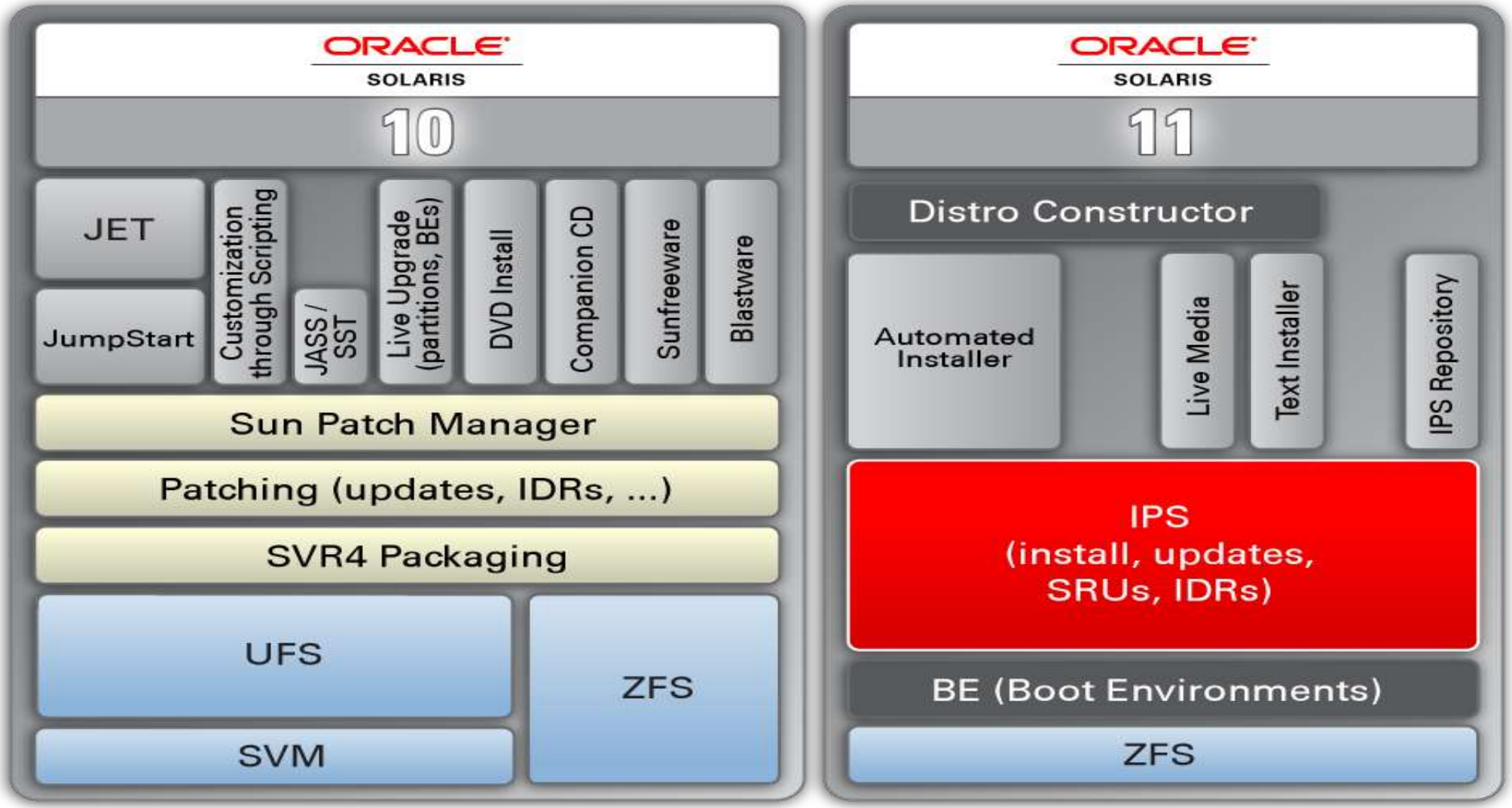
Automatische Systemkonfiguration in Solaris 11 mit First Boot Services

Detlef Drewanz
Master Principal Sales Consultant

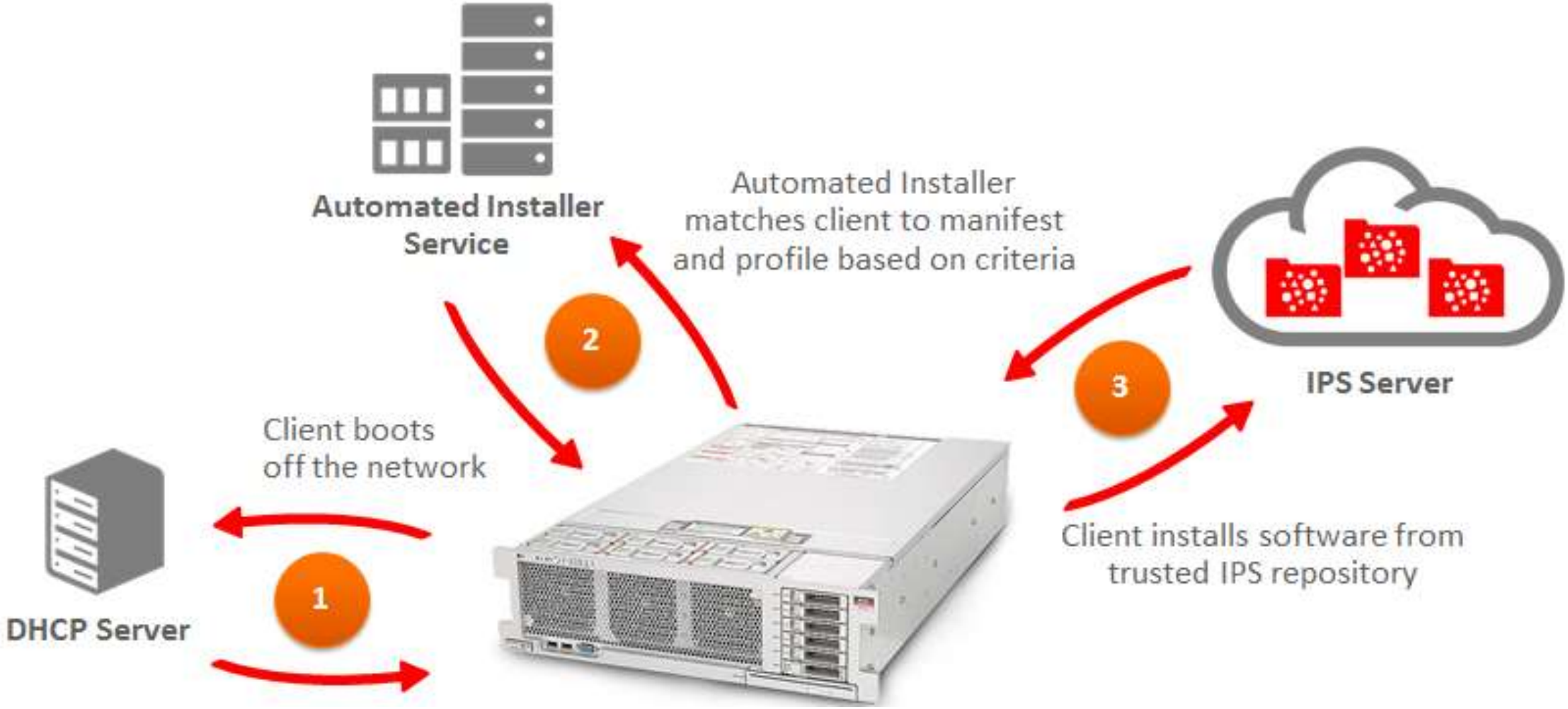


ORACLE

Intro: Oracle Solaris 11 Lifecycle Management



Intro: Automated Installer



What's next with automated System Installation ?

- Create additional user
- Network Configuration
- Customizing the system
- Additional software
- Site specific software and configuration

Options for Automatic System Configuration

- 1. Self assemblies, profiles and SMF services
  - Installed with IPS packages and used during the next boot
    - /etc/system.d/*
      - Kernel parameters
    - /etc/svc/profile/site
      - System Configuration Profile
      - Configuration of SMF services
    - /lib/svc/manifest/site
      - Create new SMF services
- 2. First Boot Service

First Boot Service (FBS)

- SMF Service
  - Activated with the first boot of a fresh installed system
  - Runs exactly once
  - Deactivates (and possibly removes) itself after job done
  - Maybe "Secondary Boot" after job done

First Boot Service (FBS) - ToDo

- Install FBS with IPS package in AI manifest
  - Create Simple IPS Package
  - Create SMF Manifest
  - Design SMF start method
  - Design Configuration files

First Boot Service (FBS) - This very easy Example

- Name the IPS package fbs
- Name the SMF service /site/fbs
- Put the package to /opt/fbs

- Actions
  - Limit the ZFS ARC Cache, depending on a condition
  - Run a script, that does something (Here: Set boot timeout to 5 seconds)

Before we begin ... some IPS Basics

Some IPS Package Basics - Parts of a Package

- Metadata and Actions
- Actions
  - Create Directories
  - Install files
  - Create dependencies
  - Set attributes
  - Create users and groups
  - Install device driver

Some IPS Package Basics - Controlled Installation of Contents

- variant - Predefined by Solaris - can not be self-defined
  - Install alternative package parts into an identical path
    - variant.arch — sparc, i386
    - variant.opensolaris.zone — global, nonglobal
    - variant.debug.* — true, false
- facet - Predefined by Solaris - but can be extended by self-defined facets
  - Install optional parts of a package (locales, man pages, etc.)
    - facet.locale.* — true, false
    - facet.doc.man — true, false
    - **facet.fbs.vm** — true, false

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

1. Create Prototype Directories (Sourcetree)

- Prototype tree in fbs/proto
- FBS SMF Manifest in lib/svc/manifest/site
  - Detected at boot by svc:/system/manifest-import:default
- Kernel Parameters in etc/system.d/*
- Package in opt/fbs

```
# mkdir -p fbs/proto/lib/svc/manifest/site
# mkdir -p fbs/proto/etc/system.d
# mkdir -p fbs/proto/opt/fbs
...
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

2. Package Content - fbs.sh = The SMF Method: Begin

 # vi proto/fbs/opt/fbs.sh

```
#!/bin/sh
# Wrapper Script to run First Boot Configuration Scripts
# Load SMF shell support definitions
. /lib/svc/share/smf_include.sh

# Prints out messages to console
fbsecho ()
{
        echo $1 | tee /dev/sysmsg
}

completed=$(svcprop -p config/completed site/fbs:default)
# If already done, exit with temporary disable
if [ "${completed}" = "true" ]; then
        svcadm disable svc:/site/fbs
        smf_method_exit $SMF_EXIT_OK completed "Configuration already done"
```

Create a simple FBS-IPS Package

2. Package Content - fbs.sh = The SMF Method: Actions

```
else
        # My FBS-stuff comes here
        DIR=/opt/fbs
        FBS="First Boot Service:"
        fbsecho "$FBS Waiting for another reboot ..."
        # Run *.cfg, installed depending of facet.site.*
        for ACTION in `ls ${DIR}/bin/*.cfg 2>/dev/null`
        do
                . ${ACTION}
        done
```

Create a simple FBS-IPS Package

2. Package Content - fbs.sh = The SMF Method: Finish

```
        ## temp disable to prevent from going online if we go ahead
        svcadm disable -t svc:/milestone/multi-user:default
        # Record that this script's work is done
        svccfg -s site/fbs:default setprop config/completed = true
        svcadm refresh site/fbs:default
                    ## Job Done - disable  FBS now
        fbsecho "\n $FBS cleaning up"
        svcadm disable svc:/site/fbs
         fbsecho "\n $FBS Done ...rebooting ..."
         init 6
        smf_method_exit $SMF_EXIT_OK method_completed "Configuration done”
fi
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

3. Create SMF FBS-Manifest

```
# svcbundle -s service-name=site/fbs
      -s start-method=/opt/fbs/fbs.sh \
      -s instance-property=config:completed:boolean:false > proto/fbs/lib/svc/manifest/site/fbs.xml
```

- Add Customizations
  - timeout_seconds: Might be important for long running FBS !
  - dependencies
  - dependents

Create a simple FBS-IPS Package

3. Create SMF FBS-Manifest - 1/2

```
<?xml version="1.0" ?>
<!DOCTYPE service_bundle
  SYSTEM '/usr/share/lib/xml/dtd/service_bundle.dtd.1'>
<service_bundle type="manifest" name="site/fbs">
    <service version="1" type="service" name="site/fbs">
        <!--
            We want to run after milestone name-services has been reached
           and before milestone multi-user has been reached.
        -->
        <dependency restart_on="none" type="service"
                name="name-services_dependency" grouping="require_all">
                <service_fmri value="svc:/milestone/name-services:default"/>
        </dependency>
        <dependent restart_on="none"
                name="multi-user_dependent" grouping="optional_all">
                <service_fmri value="svc:/milestone/multi-user:default" />
       </dependent>
```

Create a simple FBS-IPS Package

3. Create SMF FBS-Manifest - 2/2

```
        <exec_method timeout_seconds="60" type="method" name="start"
            exec="/opt/fbs/fbs.sh"/>
         <exec_method timeout_seconds="60" type="method" name="stop"
            exec=":true"/>
         <property_group type="framework" name="startd">
            <propval type="astring" name="duration" value="transient"/>
         </property_group>
         <instance enabled="true" name="default">
            <property_group type="application" name="config">
               <propval type="boolean" name="completed" value="false"/>
            </property_group>
         </instance>
      </service>
   </service_bundle>
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

4. Create Metainformation for Package

```
#  more fbs.meta
set name=pkg.fmri value=fbs@1.0
set name=pkg.summary value="Site: First Boot Service"
set name=pkg.description value="Script that runs once at first boot after a new AI installation"
set name=info.classification value=\
      "org.opensolaris.category.2008:System/Administration and Configuration"
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

5. Create Package Manifest 1/3

```
# pkgsend generate proto | pkgfmt > fbs.lst
# pkgmogrify fbs.lst fbs.meta | pkgfmt > fbs.p5m
# cat fbs.p5m

set name=pkg.fmri value=fbs@1.0
set name=pkg.summary value="AI First Boot Configuration"
set name=pkg.description value="Script that runs once at first boot after a new AI installation"
set name=info.classification value="org.opensolaris.category.2008:System/Administration and Configuration"
dir  path=lib owner=root group=bin mode=0755
dir  path=lib/svc owner=root group=bin mode=0755
dir  path=lib/svc/manifest owner=root group=bin mode=0755
dir  path=lib/svc/manifest/site owner=root group=bin mode=0755
file lib/svc/manifest/site/fbs.xml path=lib/svc/manifest/site/fbs.xml owner=root group=bin mode=0644
dir  path=opt owner=root group=bin mode=0755
dir  path=opt/fbs owner=root group=bin mode=0755
dir  path=opt/fbs/bin owner=root group=bin mode=0755
file opt/fbs/fbs.sh path=opt/fbs/fbs.sh owner=root group=bin mode=0755
```

Create a simple FBS-IPS Package

5. Create Package Manifest 2/3

- Ignore directories, which has already been installed by other packages
  - Control actions for *pkgmogrify(1)*

```
#  more fbs.meta
set name=pkg.fmri value=fbs@1.0
set name=pkg.summary value="Site: First Boot Service"
set name=pkg.description value="Script that runs once at first boot after a new AI installation"
set name=info.classification value=\
      "org.opensolaris.category.2008:System/Administration and Configuration"
<transform dir path=lib -> drop>
<transform dir path=opt$ -> drop>
```

Create a simple FBS-IPS Package

5. Create Package Manifest 3/3

```
# pkgsend generate proto | pkgfmt > fbs.lst
# pkgmogrify fbs.lst fbs.meta | pkgfmt > fbs.p5m
# cat fbs.p5m

set name=pkg.fmri value=fbs@1.0
set name=pkg.summary value="Site: First Boot Service"
set name=pkg.description value="Script that runs once at first boot after a new AI installation"
set name=info.classification value="org.opensolaris.category.2008:System/Administration and Configuration"
file lib/svc/manifest/site/fbs.xml path=lib/svc/manifest/site/fbs.xml owner=root group=bin mode=0644
dir  path=opt/fbs owner=root group=bin mode=0755
dir  path=opt/fbs/bin owner=root group=bin mode=0755
file opt/fbs/bin/10-vm.cfg path=opt/fbs/bin/10-vm.cfg owner=root group=bin mode=0755
file opt/fbs/fbs.sh path=opt/fbs/fbs.sh owner=root group=bin mode=0755
```

How to control partial package installations ?

Create a simple FBS-IPS Package

5.1 Use of Facets

- Control partial installation of packages
- Set with "pkg change-facet" or in AI-Manifest
- **Caution**: Not defined facets are preset in IPS package operations as "true"

```
# pkg change-facet facet.site.vm="true"
# pkg facet
```

```
set name=pkg.fmri value=fbs@1.0
set name=pkg.summary value="Site: First Boot Service"
set name=pkg.description value="Script that runs once at first boot after a new AI installation"
set name=info.classification value=\
    "org.opensolaris.category.2008:System/Administration and Configuration"
<transform dir path=lib -> drop>
<transform dir path=opt$ -> drop>
<transform file path=.*10-vm.cfg -> add facet.site.vm true>
```

Create a simple FBS-IPS Package

5.1 Use of Facets

 # cat fbs.p5m

```
set name=pkg.fmri value=fbs@1.0
set name=pkg.summary value="Site: First Boot Service"
set name=pkg.description value="Script that runs once at first boot after a new AI installation"
set name=info.classification value="org.opensolaris.category.2008:System/Administration and Configuration"
file lib/svc/manifest/site/fbs.xml path=lib/svc/manifest/site/fbs.xml owner=root group=bin mode=0644
dir  path=opt/fbs owner=root group=bin mode=0755
dir  path=opt/fbs/bin owner=root group=bin mode=0755
file opt/fbs/bin/10-vm.cfg path=opt/fbs/bin/10-vm.cfg owner=root group=bin mode=0755 facet.site.vm=true
file opt/fbs/fbs.sh path=opt/fbs/fbs.sh owner=root group=bin mode=0755
```

 # cat ai-manifest.xml

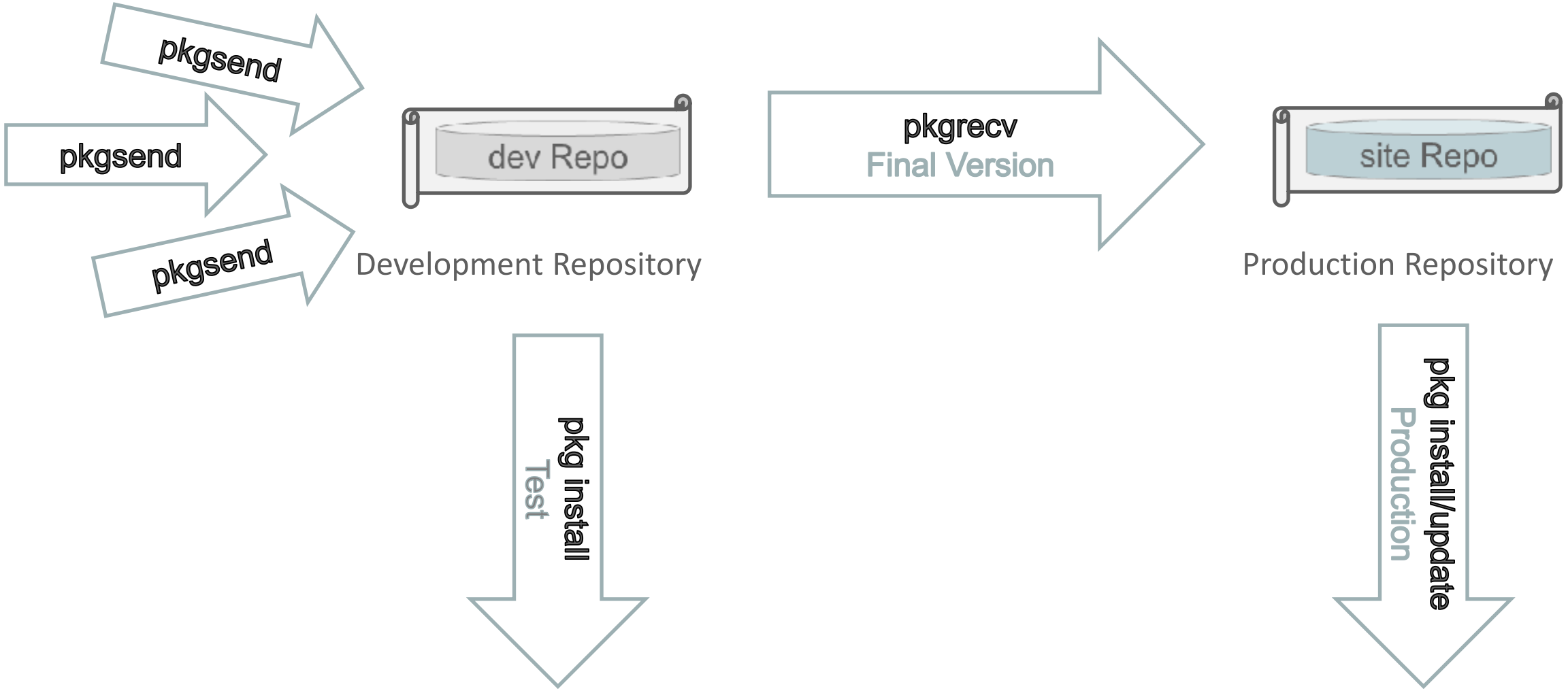
```
...
<facet set="false">facet.site.*</facet>
<facet set="true">facet.site.vm</facet>
...
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Repositories for IPS Development



Create a simple FBS-IPS Package

6. Create Development Repository

```
# pkgrepo create devrepo
# pkgrepo -s devrepo add-publisher site
```

Create a simple FBS-IPS Package

6. Create Site Repository

```
# zfs create repo/site
# pkgrepo create /repo/site
# pkgrepo –s /repo/site add-publisher site
# pkgrepo set –s /repo/site publisher/prefix=site

# svccfg –s pkg/server add site
# svccfg -s pkg/server:site setprop pkg/port=8888
# svccfg -s pkg/server:site setprop pkg/inst_root=/repo/site

# svcadm enable pkg/server:site

# pkgrepo -s http://repo-host:8888 info
PUBLISHER PACKAGES       STATUS         UPDATED
site              -                  -                    -
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

7. Create Package

```
# pkgsend publish -d ./proto -s ./devrepo fbs.p5m
pkg://site/fbs@1.0,5.11:20140522T092819Z
PUBLISHED

# pkgrepo list -s ./devrepo
NAME (PUBLISHER)                VERSION                 IFO
fbs (site)                      1.0                     ---
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

8. Test

```
# pkg set-publisher -g ./devrepo site
# pkg publisher

PUBLISHER                 TYPE     STATUS P LOCATION
solaris                   origin   online       F http://repo-host/s11.2/
site                      origin   online       F file:///net/srv/ipsdev/devrepo/
# pkg install -nv fbs

Packages to install:        1
Estimated space available: 10.05 GB
Estimated space to be consumed: 52.53 MB
Create boot environment:       No
Create backup boot environment:       No
Rebuild boot archive:       No
Changed packages:
site
 fbs
    None -> 1.0,5.11:20140522T092819Z
```

Create a simple FBS-IPS Package

Task-List

1. Create Prototype Directories (Sourcetree)
2. Create the "real" Content (Start method, Files, ...)
3. Create SMF FBS-Manifest
4. Create Metainformation for Package
5. Create Package Manifest
6. Create Site (and Development) Repository
7. Create Package
8. Test
9. Upload Package into Site Repository

Create a simple FBS-IPS Package

9. Upload Package

- Transfer Package from Development into Production Repository

```
# pkgrecv -s ./devrepo -d /repo/site fbs@latest
# pkgrepo rebuild -s /repo/site
# svcadm restart pkg/server:site
```

- Clean-up

```
# pkgrepo remove -s ./devrepo fbs@<version> … or use …@latest
# pkgrepo rebuild -s ./devrepo
```

Summary

- First Boot Services help to automatically customize System Installations
- Preparation and planning is important
- General understanding helps a lot
- The real stuff is often just a small part

- But: Creating IPS packages and FBS is no magic

References

- Packaging and Delivering Software
  With the Image Packaging System in Oracle® Solaris 11.2

  https://docs.oracle.com/cd/E36784_01/html/E36856/index.html

- Installing Oracle® Solaris 11.2 Systems

  https://docs.oracle.com/cd/E36784_01/html/E36800/index.html

Q & A

Detlef.Drewanz@oracle.com


ORACLE®

Hardware and Software
Engineered to Work Together