Let SMF handle your service

Detlef Drewanz
Systems Engineer/Ambassador Operating Systems
Sun Microsystems GmbH

Agenda

- Motivation for SMF
- SMF: Core concepts and terminology
- Creating your own service

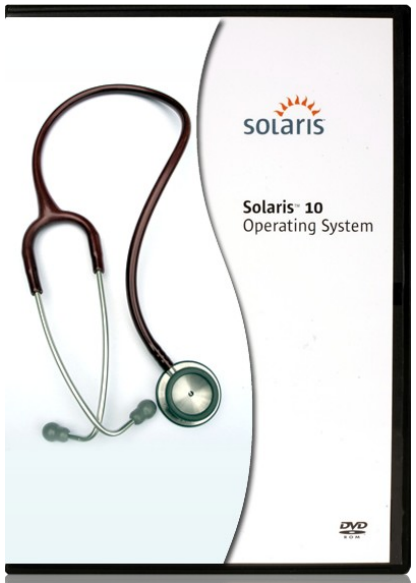
First time Developer Impressions
About SMF

- What is that manifest loading during boot ?
- Can I switch this off ?
- Where are all the /etc/init.d scripts ?
- Please give me my rc?.d back !
- What the heck is a manifest ?

SMF

- Yes, it's new
- **But:** Learn about SMF and realize new ways to develop, deploy and manage services

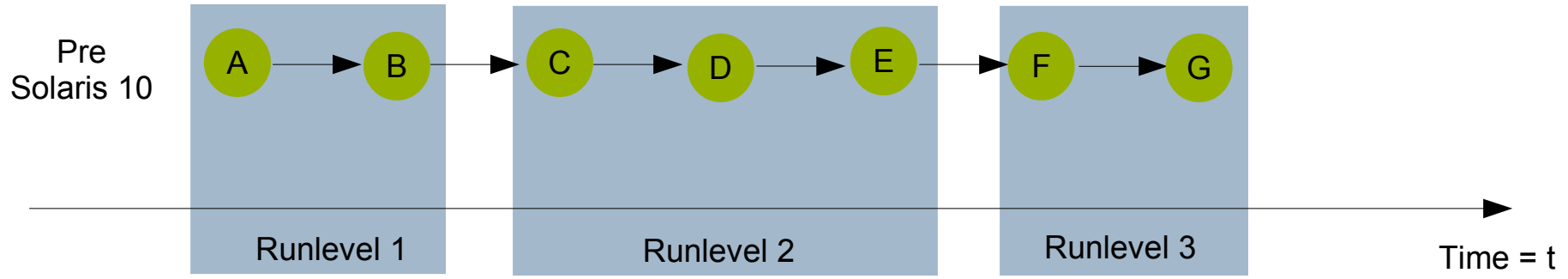
~~/etc/rc?.d~~

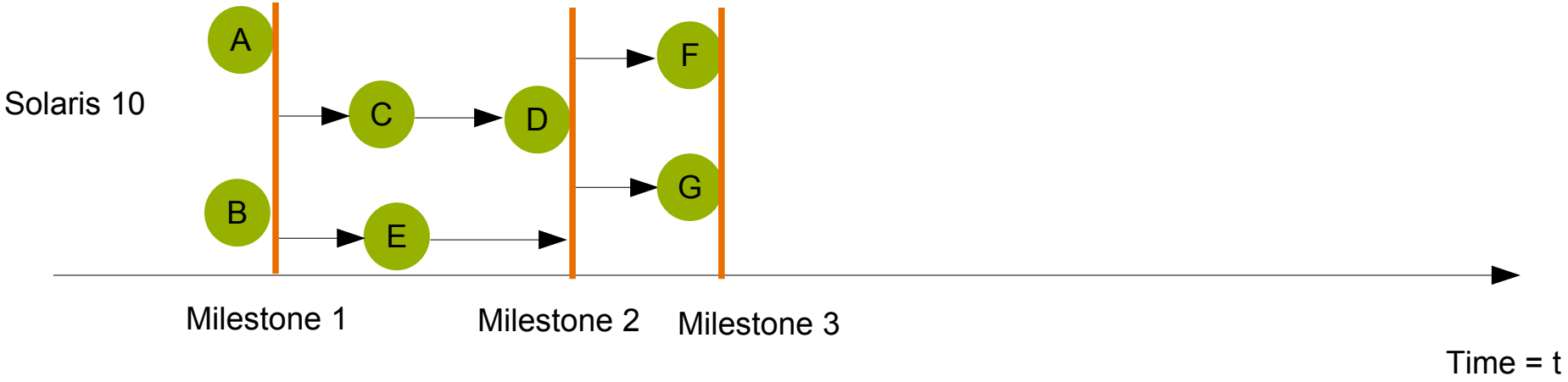


Service Management in Solaris

- Common Service Management framework
  - Start/Stop all services in Solaris from boot
  - Service dependencies
  - Parallel Service startup during boot
  - Automated restart of services
  - Securely delegate tasks to non-root users

Dependencies, Dependents, Milestones





Service Management in Solaris

- Common Service Management framework
  - Start/Stop all services in Solaris from boot
  - Service dependencies
  - Parallel Service startup during boot
  - Automated restart of services
  - Securely delegate tasks to non-root users

Service Management in Solaris

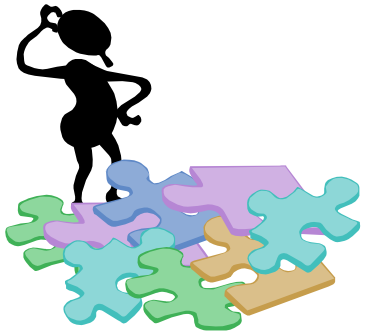
- Common Service Development and Deployment framework
  - Service Manifests
  - Handling for Upgrade and Patch

What is a Service ?

Definition:

> A long lived software object with a well-defined state, error boundary, definition of start and stop, and relationship to other services. A service is often critical to operation of system or fulfillment of business objectives.

- Define/Describe by service manifest
- A single process or a grouping of other services
- A status of a device
- A set of configuration files
- Management utilities



Service Names

How to identify a service ?

- FMRI - Fault Managed Resource ID

svc:/network/ssh:default

Descriptive name Instance

Functional categories:

Application	–	traditional daemons
Device	–	useful for dependencies
Milestone	–	similar to SVR4 run levels
Network	–	network services
Platform	–	platform specific services
System	–	platform independent system services
Site	–	site services

Service Manifests

How to describe a service ?

- Service Manifeste (XML) define Services
  - > Place into /var/svc/manifest/<category>
  - > FMRI as name
  - > Define dependencies and dependents
  - > Create methods (start/stop/refresh) with special properties

Parts of a Service

What all belongs together ? - Example ssh

/var/svc/manifest/network/ssh.xml

/usr/lib/ssh/sshd

svc:/network/ssh:default

/lib/svc/method/sshd

/var/svc/log/network-ssh:default.log

Agenda

- Motivation for SMF
- SMF: Core concepts and terminology
- Creating your own service

Create your own service: Overview

- Quickstart and Developer Documentation
  - > http://www.sun.com/bigadmin/content/selfheal/sdev_intro.html
- Create service manifest and service method
- Import manifest
- Test
- Packaging manifest, methods, binaries
- Install Service
- Enable/disable/configuration/modification of Services
- Removal of services

Create your manifest

- Service Definition - service_bundle(4)
- To create your own, maybe use+modify an existing
  - > Or use **inetconv -i** to import from /etc/inetd.conf
- Think about and place into right category
- /var/svc/manifest/<category>
  - application – traditional daemons
  - device – useful for dependencies
  - milestone – similar to SVR4 run levels
  - network – network services
  - platform – platform specific services
  - system – platform independent system services
  - site – site services

Basic structure service manifest

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='osdevcon'>
  <service
    name='site/osdevcon'
    type='service'
    version='1'>                                          General definitions
    <create_default_instance enabled='false'>
      <single_instance />
    <dependency ... >
            ...                                           Dependencies
    </dependency>
    <dependent ... >
            ...                                           Dependents
    </dependent>
    <exec_method ... >
            ...                                           Exec method
    </exec_method>
    <property_group ... >
            ...                                           Property groups
    </property_group>
    <template>
            ...                                           Template
    </template>
  </service>
</service_bundle>
```

Manifest - general definitions

- XML-file contains service description
  - Choose a service name

    <service name = 'site/osdevcon'
  - Disable the default instance on create

    <create_default_instance enabled='false' />
  - use inetd or master restarter for service

```
<restarter>
        <service_fmri value='svc:/network/inetd:default' />
</restarter>
```

Service model

Long term worker or short task to do ?

- if svc.startd is used as master restarter
- wait - wait for and restart if child exits
- contract - run forever when enabled
  - default
  - typical system daemon
- transient - no long-running processes
  - configurations services like cleanup, config loading

```
<property_group name='startd' type='framework'>
        <propval name='duration' type='astring' value='transient' />
</property_group>
```

Service methods

What to do on start/stop/refresh ?

- Exec method - see smf_method(5)

```
<exec_method
        type='method'
        name='start'
        exec='/etc/opt/svc/method/osdevcon start'
        timeout_seconds='60'>
        <method_context>
            <method_credential user='martin' group='guug'
                privileges='basic,file_dac_write' />
        </method_context>
    </exec_method>
```

- Maybe already there as rc?.d scripts
- Think about placing your method scripts !
  - > /lib/svc/method is not always a good place (see zones)

SMF security

Do I need to be root to manage my service ?

- Delegate service specific responsibilities to users
  - see rbac(5) and /etc/user_attr
  - Fixed keywords in service manifest
    - action_authorization – can execute method
    - value_authorization – can change value of existing properties
    - modify_authorization – change, add, delete properties

```
<property_group name='general' type='framework'>
    <!-- to start stop sshd -->
        <propval name='action_authorization' type='astring'
            value='solaris.smf.manage.ssh' />
</property_group>
```

Service dependencies

Who will I depend on ?

- Depend on other SMF managed services or a files
- A dependency can also be a milestone
- Group-status of dependencies
  - require_all – all must be online or degraded
  - require_any – at least one online or degraded
  - optional_all – all are online, disabled, degraded, or in maintenance
  - exclude_all – all are disabled, in maintenance, or not present (files)

Service dependencies (2)

What if something happens with my dependency ?

- Choices
  - none - no action
  - error - restart, if dependency had a fault
  - restart - restart, if dependency has been restarted
  - refresh- restart, if dependency has been restarted

```
<dependency
        name='mydependency'
        type='service'
        grouping='require_all'
        restart_on='none'
        <service_fmri value='svc:/my-dependency-service' />
 <dependency>
```

Service dependents

Can I tell others they should depend on me ?

- Let other services depend on my service, without changing other manifests or properties

```
<dependent
      name='mysvc'
      grouping='optional_all'
      restart_on='none'>
      <service_fmri value='svc:/my-dependent' />
<dependent>
```

Service specific authorizations

Who is able to manage this service ?

- Delegate administration of services to users
  - action_authorization – can execute method
  - value_authorization – can change value of existing properties
  - modify_authorization – change, add, delete properties
- user only need later to have this auths defined in his environment see /etc/user_attr(4)
- smf_security(5)

```
<property_group name='general' type='framework'>
    <propval name='action_authorization' type='astring'
        value='solaris.smf.manage.ssh' />
</property_group>
```

Managing inetd services

What about inetd-based services ?

- inetd managed services require special setups in the manifest
- Help create on using /etc/inetd.conf, /etc/services and inetconf
  - **inetconv -i** to import the content of inetd.conf as services
  - **inetadm -l FMRI** to list properties
  - **inetadm -m FMRI property-name=value** to modify properties

Basic structure service manifest

```
<?xml version="1.0"?>
<!DOCTYPE service_bundle SYSTEM "/usr/share/lib/xml/dtd/service_bundle.dtd.1">
<service_bundle type='manifest' name='osdevcon'>
  <service
    name='site/osdevcon'
    type='service'
    version='1'>
    <create_default_instance enabled='false'>
      <single_instance />
    <dependency ... >
            ...
    </dependency>
    <dependent ... >
            ...
    </dependent>
    <exec_method ... >
            ...
    </exec_method>
    <property_group ... >
            ...
    </property_group>
    <template>
            ...
    </template>
  </service>
</service_bundle>
```

General definitions

Dependencies

Dependents

Exec method

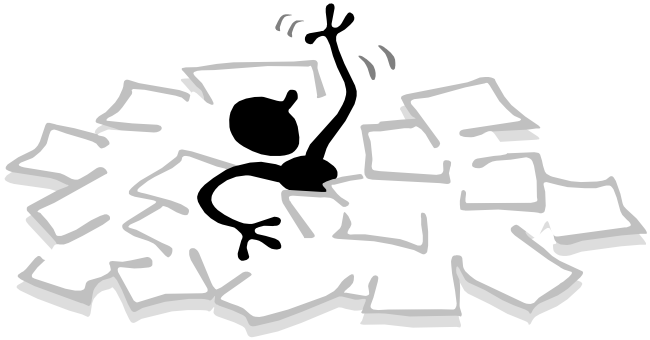
Property groups

Template

Roll your own service - Summary

- cd /var/svc/manifest
- cp <manifest>.xml <manifest-new>.xml / inetconv -i
- vi <manifest-new>.xml
- xmllint --noout <manifest-new>.xml
- svccfg validate <manifest-new>.xml
- svccfg import <manifest-new>.xml
- svcadm enable <service-new>
- svcs -lp <service-new>

More Information



- Sun Product Documentation
  - http://docs.sun.com/
- Open Solaris SMF Community
  - http://www.opensolaris.org/os/community/smf/
- Sun BigAdmin SMF Portal
  - http://www.sun.com/bigadmin/content/selfheal/
- Blogs
  - Stephen Hahn - http://blogs.sun.com/roller/page/sch
  - Liane Praza - http://blogs.sun.com/lianep/

Thank you

Detlef Drewanz
Detlef.Drewanz@Sun.COM
http://blogs.sun.com/solarium