

Automatische Systemkonfiguration in Solaris durch First Boot Services

Detlef Drewanz
Oracle Deutschland B.V. & Co. KG
Potsdam

Schlüsselworte:

Oracle Solaris 11, IPS, AI, SMF

Einleitung

Mit der Einführung von Solaris 11 wurde eine neue Form der automatischen Installation und ein neues Paketformat IPS eingeführt. Während der Installation von Neusystemen, werden ausschliesslich Pakete installiert, jedoch keine Scripte ausgeführt. Die Art und Weise der automatischen Systemkonfiguration wird in Solaris 11 durch sogenannte First-Boot-SMF-Services realisiert. Das sind IPS Pakete, die während der automatischen Installation installiert werden, SMF Services erzeugen und diese beim ersten Booten des neu installierten Systems aktivieren und ihre Aufgabe erfüllen.

Dieser Papier zeigt an Beispielen die Schritte zur Erzeugung eines First-Boot-SMF-Services und des dazugehörigen IPS Paketes.

Überblick

In Solaris 11 werden Systeme mit dem Automated Installer (AI) installiert. Im Grunde werden dabei IPS-Pakete (IPS – Image Packaging System) installiert.

Anders als bei früheren Solaris Installationen mit Jumpstart werden keine Scripte während der Installation ausgeführt, sondern lediglich Pakete installiert. D.h. eine automatisierte Systemkonfiguration erfordert die Erzeugung von IPS Pakete und diese so bereitzustellen, dass sie während der automatisierten Installation gefunden und installiert werden.

Wenn das neu installierte System das erste Mal bootet, werden die installierten Dateien zur Systemkonfiguration verwendet. Einige Beispiele dafür:

- /etc/system ist eine Konfigurationsdatei zur Einstellung von Kernel Parametern. Neu in Solaris 11.2 ist /etc/system.d/ als Verzeichnis, in das Dateien abgelegt werden können, die Kernel Parameter setzen. Der Vorteil hierbei ist, dass lediglich eine weitere Datei abgelegt werden muss, die der Syntax in system(4) entspricht. /etc/system muss nun nicht mehr als Datei automatisiert erweitert werden.
- In /etc/svc/profile/site können SMF Profiles abgelegt werden, die zur Konfiguration von sämtlichen SMF-Services genutzt werden können. Das betrifft nicht nur die Aktivierung oder Deaktivierung von Services, sondern auch deren Konfiguration von einzelnen Eigenschaften. Da die Systemkonfiguration (Sprachen, Nameservice etc.) auch als SMF Eigenschaften abgebildet werden, erfolgt die Systemkonfiguration durch ein System Configuration Profile.

Im Idealfall lässt sich so durch das Installieren von Dateien eine vollständige automatisierte Systemkonfiguration erzeugen.

In der Realität ist jedoch zusätzlich oft noch die Ausführung von Kommandos notwendig, um Konfigurationen durchzuführen, die nicht durch Konfigurationsdateien oder SMF-Services abgebildet werden. In dem Fall wird ein SMF Service benutzt, der durch ein IPS-Paket installiert wird, beim ersten Boot der neu installierten Systems genau einmal abläuft und sich dann selbst deaktiviert. Die eigentlichen beabsichtigten auszuführenden Aktionen des SMF Services werden in der Start-Methode des SMF-Service untergebracht. Der SMF-Service kann sehr einfach installiert werden, indem durch das IPS-Paket ein entsprechendes SMF-Manifest nach `/lib/svc/manifest/site` installiert wird. Beim ersten Start des Systems wird dieses neue SMF-Manifest gefunden und daraus der neue SMF Service erzeugt. Dieser SMF Service wird sofort aktiviert und ausgeführt.

Ausgehend von diesen Bemerkungen ergeben sich die folgenden Teilaufgaben zur Realisierung von First Boot Services zur Systemkonfiguration:

- Erzeugung eines SMF Manifestes für die Einrichtung des First Boot Service
- Erzeugung eines Start-Scriptes für die Ausführung des First Boot Service
- Erzeugung eines IPS-Paketes zur Installation des First Boot Services
- Ablage des IPS-Paketes in einem eigenen IPS-Repository
- Installation des IPS-Paketes durch den AI

Erzeugung eines SMF Manifestes für die Einrichtung des First Boot Service

Zur Entwicklung von IPS Paketen wir ein Prototype Verzeichnis benötigt, welches die reale Verzeichnisstruktur der Zielinstallation abbildet.

- `/lib/svc/manifest/site` für das Manifest des SMF Service
- `/etc/svc/profile/site` für Profile zur Konfiguration von Services
- `/opt/fbs` zur Ablage des Paketes

```
$ mkdir -p fbs/proto/lib/svc/manifest/site
$ mkdir -p fbs/proto/etc/svc/profile/site
$ mkdir -p fbs/proto/opt/fbs/bin
$ mkdir -p fbs/proto/opt/fbs/lib
```

Das SMF Manifest wird durch die Nutzung eines Solaris Hilfprogrammes erzeugt. Das erspart die Suche nach Fehlern in XML-Dateien.

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

Im Ergebnis entsteht das folgende SMF Manifest. Die fettgedruckten Teile können ergänzt werden, um eine Abhängigkeit des SMF Services zwischen den Milestones `name-service` und `multi-user` zu erzwingen.

```

# <?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">
    <!--
      run between milestones name-services and multi-user
    -->
    <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>
    <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>

```

Erzeugung eines Start-Scriptes für die Ausführung des First Boot Service

SMF-Services benutzen Start-Methoden, die ausgeführt werden, wenn der Service aktiviert wird. Diese Start-Methoden können Programme oder Shell-Scripte sein. Die Start-Methode wird für den First Boot Service benutzt, um Kommandos auszuführen. Im oberen Manifest wurde als Start-Methode `/opt/fbs/fbs.sh` angegeben. Das folgende Beispiel zeigt ein mögliches Start-Script:

- Über eine Variable wird gesteuert, dass der Service nur einmal ausgeführt wird – auch wenn der Service später irrtümlich mit `svcadm enable fbs` aktiviert werden sollte.
- `/opt/fbs/bin/*.cfg` wird in alphabetischer Reihenfolge ausgeführt. Diese `*.cfg` Scripte können je nach Anforderung erstellt werden.
- Das System wird neu gestartet.

```

#!/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 $1 | tee /dev/sysmsg
}
completed=$(svccprop -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"

```

```

else
# My FBS-stuff comes here
DIR=/opt/fbs
FBS="First Boot Service:"
fbsecho "$FBS Waiting for another reboot ..."

# Run *.cfg, installed depending on facet.site.*
for ACTION in `ls ${DIR}/bin/*.cfg 2>/dev/null`
do
    . ${ACTION}
done

# 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

```

Erzeugung eines IPS-Paketes zur Installation des First Boot Services

Nachdem der First Boot Service erzeugt wurde, muss nun aus den Dateien ein IPS Paket gebaut werden. Dazu wird zunächst eine Datei mit Metainformationen benötigt.

```

# vi 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"

```

Diese Metainformationen und der bestehende Verzeichnisbaum werden zur Erzeugung des Paketmanifestes benutzt.

```

# pkgsend generate fbs/proto | pkgfmt > fbs.lst
# pkgmogrifa fbs.lst fbs.meta | pkgfmt > fbs.p5m

```

Im Ergebnis entsteht das folgende IPS Paketmanifest:

```

# 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=etc owner=root group=bin mode=0755
dir path=etc/svc owner=root group=bin mode=0755
dir path=etc/svc/profile owner=root group=bin mode=0755
dir path=etc/svc/profile/site owner=root group=bin mode=0755
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
dir path=opt/fbs/lib owner=root group=bin mode=0755
file opt/fbs/fbs.sh path=opt/fbs/fbs.sh owner=root group=bin mode=0755

```

Nach diesem Manifest würde die Installation dieses Paketes auch versuchen `/etc`, `/lib` (jeweils mit Unterverzeichnissen) und `/opt` zu erzeugen bzw. wenn die Verzeichnisse schon vorhanden sind, die Metadaten der Verzeichnisse anzupassen. Da es sich jedoch hier um durch Solaris bereits erzeugte Dateien und Verzeichnisse handelt, soll das vermieden werden. Dazu werden in die Datei mit den Metainformationen Steueranweisungen für `pkgmogrify` eingesetzt die verhindern, dass die entsprechenden Verzeichnisse in das Manifest mit übernommen werden.

```

# vi 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=etc -> drop>
<transform dir path=opt$ -> drop>

```

Eine nochmalige Erzeugung des Manifestes zeigt, dass die entsprechenden Einträge ausgeblendet werden.

Ablage des IPS-Paketes in einem eigenen IPS-Repository

Jetzt kann das neue Paket erzeugt werden. Dafür wird ein IPS-Repository z.B. mit dem Publisher `site` erzeugt. Eine Prüfung sollte ein leeres Repository zeigen.

```

# pkgrepo create /repo/site
# pkgrepo -s /repo/site add-publisher 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 info -s http://site-repo-host:8888
PUBLISHER PACKAGES STATUS UPDATED
site 0 online 2014-09-05T14:36:25.795325Z

```

Nun kann das neue Paket in das Repository übertragen werden. Nach einem Refresh des Repository zeigt sich ein Paket in dem Site Repository.

```

# pkgsend publish -s /repo/site -d fbs/proto fbs.p5m
# pkgrepo refresh -s /repo/site
# svcadm restart pkg/server:site
# pkgrepo info -s http://site-repo-host:8888
PUBLISHER PACKAGES STATUS UPDATED
site 1 online 2014-09-05T15:37:25.795325Z

```

Installation des IPS-Pakates durch den AI

Das neu erzeugte Paket kann nun in einem AI Manifest benutzt werden. Das folgende Beispiel zeigt einen Ausschnitt aus so einem Manifest.

```
...
<source>
  <publisher name="solaris">
    <origin name="http://solaris-repo-host/" />
  </publisher>
  <publisher name="site">
    <origin name="http://site-repo-host:8888/" />
  </publisher>
</source>
<software_data action="install">
  <name>pkg:/entire@0.5.11-0.175.2</name>
  <name>pkg:/group/system/solaris-large-server</name>
  <name>pkg:/fbs</name>
</software_data>
...
```

Zusammenfassung

Dieses einfache Setup zeigt die Schritte zur Erzeugung von First Boot SMF Services. Dabei werden auch Kenntnisse zur Erzeugung von IPS Paketen und Repositories benötigt. Einige Schritte sind nur für das erste gebaute Paket notwendig. Bei der Erzeugung von verschiedenen Paketen für unterschiedliche Umgebungen lassen sich viele Dinge wiederverwenden und automatisieren.

Literaturverzeichnis

- Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.2
http://docs.oracle.com/cd/E36784_01/html/E36856/index.html
- Copying and Creating Package Repositories in Oracle Solaris 11.2
http://docs.oracle.com/cd/E36784_01/html/E36805/index.html
- Installing Oracle Solaris 11.2 Systems
http://docs.oracle.com/cd/E36784_01/html/E36800/firstboot-1.html

Kontaktadressen:

Detlef Drewanz
Oracle Deutschland B.V. & Co. KG
Schiffbauergasse 14
D-14467 Potsdam
Telefon: +49 (0) 331 200 7341
E-Mail: Detlef.Drewanz@oracle.com
Internet: <http://www.oracle.com>