

# **IPS und Automated Installation - die nächste Generation des Solaris Provisioning in der Praxis**

**Detlef Drewanz  
Oracle Deutschland B.V. & Co. KG  
Komturstraße 18a, 12099 Berlin**

## **Schlüsselworte:**

IPS, Automated Installation, Distribution Constructor, Oracle Solaris, Solaris

## **Einleitung**

Eine wesentliche Anforderung von Kunden an die Weiterentwicklung von Solaris war eine Überarbeitung des Provisioning, d.h. des Lebenszyklus des OS von der initialen Installation über die Nach- und Deinstallation von Paketen, das Patching bis hin zu Upgrades. Mit dem Image Packaging System (IPS) und der Automated Installation (AI) wird dieser Bereich in Oracle Solaris 11<sup>1</sup> gründlich modernisiert - nicht zuletzt vor dem Hintergrund heutiger Netzwerk-Infrastrukturen. Hier werden sich für Systemadministratoren signifikante Änderungen ergeben. Im Folgenden wird die neue Architektur vorgestellt und aufgezeigt, wie deren Vorteile in der Praxis umgesetzt werden können. Die Bezeichnungen und Beispiele beziehen sich auf den Oracle Solaris 11 Entwicklungsstand von Build 149.

## **Ein kurzer Blick zurück: Paketverwaltung und Jumpstart in Oracle Solaris 10**

Oracle Solaris 10 besteht aus einer umfangreichen Distribution, die auf einer DVD ausgeliefert wird und nutzt seit vielen Jahren das bekannte SVR4 Package Format. Das Format hat sich in den vergangenen Jahren in vielen Installationen bewährt und ist durch eine hohe Flexibilität gekennzeichnet. So werden durch die Pakete und Paketverwaltungstools nicht nur einfach Dateien an bestimmte Stellen im Dateisystem kopiert, sondern es können z.B. auch Skripte während der Installations- oder De-Installationsphase ausführen (preinstall, postinstall, preremove, postremove).

Zur Behebung von Fehlern werden Patches ausgeliefert, die durch den Administrator eingespielt werden. Hierbei werden bestimmte Dateien im Dateisystem ausgetauscht. Zur Feststellung des Zustandes einer Installation müssen die Liste der aktuellen Packages und die Liste der installierten Patches gemeinsam betrachtet werden.

Die automatische Installation von Solaris erfolgt durch den Jumpstart Mechanismus, der vielfach bei Kunden im Einsatz ist. Im Grundprinzip werden durch Jumpstart die Packages einer Solaris Installation der Reihe nach automatisiert auf das Zielsystem übertragen. Zusätzlich werden bereits vorhandene Patches installiert und oft lokale Anpassungen an den Installationen durch integrierte Skripte vorgenommen. Viele Jumpstart-Installationen sind über viele Jahre gewachsenen Konstruktionen, die direkt auf das Solaris Paketformat mit all seinen Vor- und Nachteilen abgestimmt sind.

---

<sup>1</sup> Oracle plant die Verfügbarkeit von Oracle Solaris 11 für das Kalenderjahr 2011. Das erste Oracle Solaris 11 Express Release - zum Ende des Kalenderjahres 2010 erwartet - ermöglicht Kunden Zugang zu neuesten Oracle Solaris 11 Features; über einen optionalen Vertrag auch mit Support. Solaris 11 Express basiert auf Technologien früherer Solaris- und OpenSolaris-Releases. Für weitere Details siehe: <http://www.oracle.com/us/corporate/press/173478> und <http://www.oracle.com/technetwork/server-storage/solaris11>

## Im Überblick: Software Lifecycle Management in Oracle Solaris 11

Für Oracle Solaris 11 ist die Notwendigkeit eines veränderten Paketformates und einer veränderten Paketverwaltung entstanden.

- Patches sollen als Updates von Packages ausgeliefert werden und so die Integrität von OS-Installationen besser sichern.
- Die Möglichkeit der Ausführung von Skripten während der Installation von Packages bietet zwar eine hohe Flexibilität, jedoch ist diese Flexibilität eine der Ursachen für viele Probleme bei der Installation von Software oder Oracle Solaris Zonen, sowie beim Patchen und Upgraden von Installationen und Zonen.
- ZFS als Rootfilessystem bietet durch Snapshots und Clones neue Möglichkeiten bei der Installation und dem Upgrade von Systemen. Diese Möglichkeiten werden von den existierenden Tools nur bedingt ausgenutzt.
- Es werden Minimalinstallation benötigt, die lediglich nur die Pakete enthalten, die unbedingt notwendig sind.
- Abhängigkeiten zwischen den Paketen sollten fein-granuliert sein und so die Anzahl großer Softwarecluster reduziert werden.
- Die Basisinstallationen sollten einfacher und schneller ausführbar sein.
- Installationsmedien sollten unterschiedlichen Erfordernisse angepasst werden können.

Ausgehend von diesen und weiteren Überlegungen werden in Oracle Solaris 11 verschiedene Veränderungen an der Paketverwaltung, den Tools zur Paketverwaltung, Patching, Installation und Upgrade, sowie an der Automatischen Installation vorgenommen. Diese sind:

- Das Installationsmedium ist eine kompakte bootbare **LiveCD**. Von dieser CD kann nach dem Boot die initiale Systeminstallation vorgenommen werden. Dabei wird zur Verkürzung der Installationszeit ein auf der CD enthaltenes kompaktes Systemimage auf das System übertragen. Zusätzliche Pakete werden nach der initialen Installation eingespielt.
- Ein neuer optimierter interaktive **Installer** steht in Varianten für grafische Bedienung oder Kommandozeile zur Verfügung und erfordert nur wenige Eingaben.
- Das **Image Packaging System (IPS)** ist ein netzwerk-zentrisches Paketverwaltungs-System, das Softwarepakete in einem neuen Format in Repositories bereitstellt. Diese Pakete werden mit den Paketverwaltungstools aus dem Repository heraus direkt installiert und aktualisiert. SVR4 Pakete können nach wie vor mit traditionellen Tools installiert werden.
- Bei notwendigen Patches werden Softwarepakete aktualisiert und auf einem Repository als neue Version bereitgestellt. Das Patchen von Systemen erfolgt also durch das Einspielen **aktualisierter Pakete**.
- Für **Entwickler** stehen Tools zur Erzeugung und zur Publikation von Packages bereit. Eigene Software Repositories können erzeugt, verwaltet und gespiegelt werden.
- Es wurde ein **neues Paketverwaltungstool** zur Installation, Aktualisierung, Verwaltung und Verifikation von Paketen eingeführt. Der Mechanismus für ein komplettes OS-Upgrade wurde überarbeitet.
- Durch den **Distribution Builder** kann eine angepasste Installations-CD erstellt werden. So kann das Systemimage an die Erfordernisse der Installation angepasst werden und z.B. Minimal-Images erzeugt werden.
- Der **Automated Installer (AI)** bietet die Möglichkeit der automatisierten, angepassten Solaris Installation. Die Konfiguration von AI basiert auf xml-Manifesten.

## Das Image Packaging System (IPS)

IPS ist ein komplettes Bandbreiten-effizientes, Netzwerk-zentrisches System zur Paketverwaltung in Oracle Solaris 11 mit Funktionen zur Installation, Aktualisierung, Entfernung oder Verwaltung von Software Paketen unter Beachtung von Abhängigkeiten zwischen den Paketen.

Die Software-Pakete werden von Depot-Servern in Repositories bereitgehalten und können von Servern gespiegelt werden. Der Zugriff auf die Dateien der Softwarepakete erfolgt von Clients über das Netzwerk per http-/https-Protokoll. Die Metadaten der verfügbaren Pakete werden abgerufen und am Client angezeigt. Zur Installationen werden die Dateien der Pakete auf den Client übertragen und direkt installiert. Der Ursprung der Pakete bleibt das Repository des Depot-Servers.

Clients können jedoch gleichzeitig mit mehrere Repositories verbunden sein. So können die Pakete und Updates des Betriebssystems von denen anderer Softwareprodukte getrennt werden. Softwarehersteller können als eigene Autorität (authority) auftreten, ihre eigenen Repositories erzeugen und hier die eigene Software und Updates zur Verfügung stellen. Repositories für eigene Software-Pakete können lokal aufgesetzt werden. Bei einem Systemupgrade wird automatisch in allen aktiven Repositories nach Updates gesucht und es werden diese installiert.

Zur Paketverwaltung wird am Client das pkg(1)-Tool verwendet, für das ebenfalls GUI-Tools zur Verfügung stehen ( packagemanager(1) , pm-updatemanager(1) ).

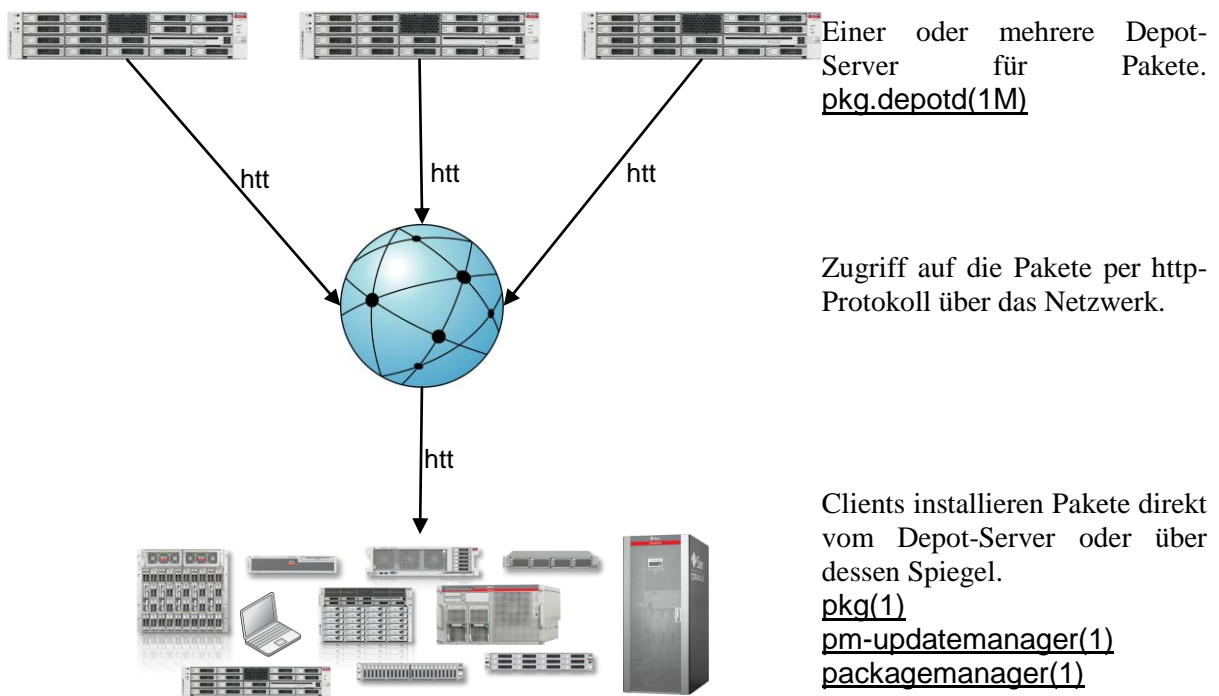


Abb. 1: Das Image Packaging System

Ein Software-Pakete ist eine Sammlung von Aktionen, Dateien, Verzeichnissen, Links, Treibern, Abhängigkeiten und wird durch einen Fault Managed Ressource Identifier (FMRI) identifiziert. Das ZFS-Paket hat z.B. den folgenden Namen:

pkg://opensolaris.org/system/file-system/zfs@0.5.11,5.11-0.149:20100924T025722Z

Die Namen von Paketen setzen sich z.B. wie folgt zusammen:

- Schema: pkg
- Authority: opensolaris.org
- Package Name: system/filesystem/zfs
- Version String
  - Component Version 5.11
  - Build Version 5.11
  - Branch Version 0.149
  - Timestamp 20100924T025722Z

Unabhängig davon, ob ein Paket von einem lokalen Repository, über einen Spiegel oder von seinem ursprünglichen Repository installiert wird, bleibt der Paketname identisch. So kann anhand des Paketnamens ermittelt werden, welche Repositories entsprechende Updates des Paketes bereitstellen.

Zur Verwaltung von Paketen wird das Kommando `pkg(1)` mit unterschiedlichen Optionen eingesetzt, das z.B. zur Installation, Deinstallation, zum Update und zur Prüfung von Paketen eingesetzt wird. Aber auch die Festlegung und Verwaltung von Repositories erfolgt mit `pkg(1)` oder dem dazugehörigen GUI-Tool `packagemanager(1)`.

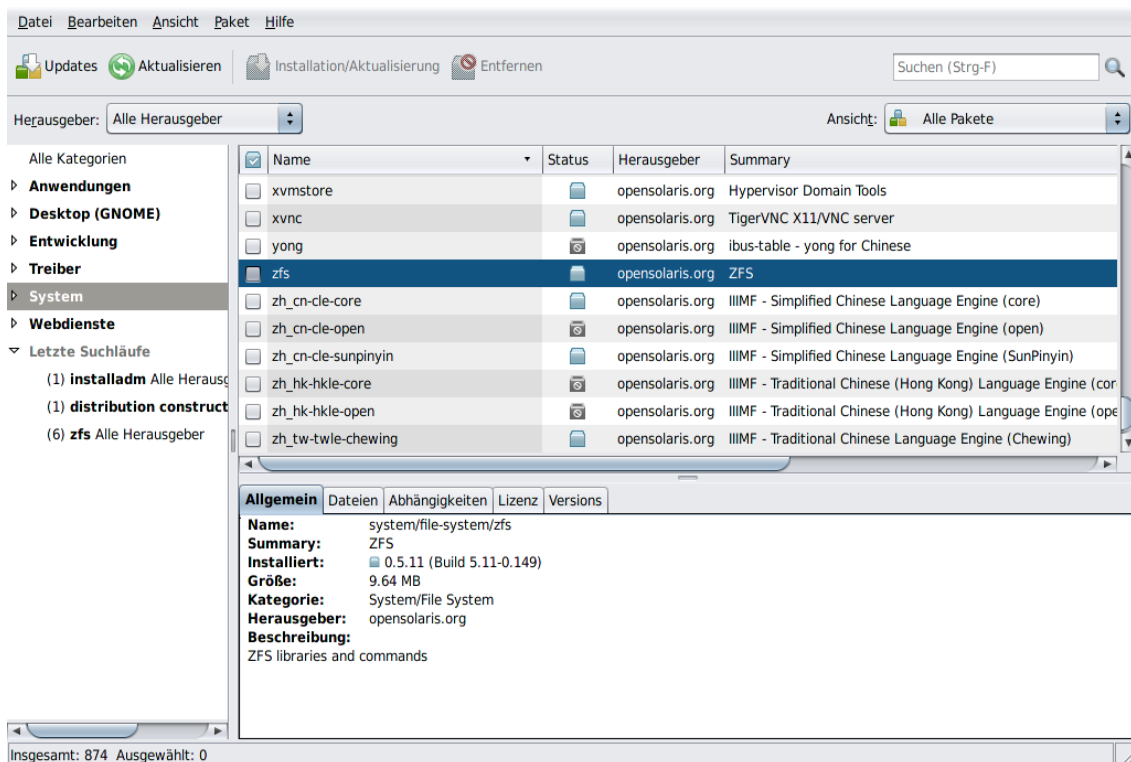


Abb. 2: `packagemanager(1)`

Es lassen sich mit einem Kommando alle Aktionen zur Paketverwaltung ausführen. Die folgende Seite zeigt einige Beispiele zur Paketverwaltung mit pkg(1).

- **Informationen über ein installiertes Paket**

```
# pkg info zfs
      Name: system/file-system/zfs
Zusammenfassung: ZFS
  Beschreibung: ZFS libraries and commands
    Kategorie: System/File System
      Status: Installiert
    Herausgeber: opensolaris.org
      Version: 0.5.11
Build-Release: 5.11
      Zweig: 0.149
Packaging-Datum: 24. September 2010, 02:57:22 Uhr
      Größe: 9.64 MB
      FMRI: pkg://opensolaris.org/system/file-
system/zfs@0.5.11,5.11-0.149:20100924T025722Z
```

- **In welchem Paket ist ein bestimmtes Programm enthalten ?**

```
# pkg search distribution-creator
INDEX      ACTION VALUE                                     PACKAGE
pkg.fmri   set      opensolaris.org/install/distribution-
creator    pkg:/install/distribution-creator@0.5.11-0.149
```

- **Bei der Installation von Paketen** wird je nach Pakettyp entschieden, ob ein neues Boot Environment erzeugt werden muß. Mehr Details dazu im folgenden Abschnitt.

```
# pfexec pkg install distribution-creator
      Packages to install:      1
      Create boot environment:  Nein

DOWNLOAD      PKGS      FILES      XFER (MB)
Completed          1/1      102/102     0.3/0.3$<3>
PHASE
Installationsphase          145/145
PHASE
Package State Update Phase          1/1
Image State Update Phase          2/2
PHASE
Vorhandener Index wird gelesen          8/8
Pakete werden indiziert          1/1
```

- **Deinstallation von einem Paket**

```
# pfexec pkg uninstall distribution-creator
      Packages to remove:      1
      Create boot environment:  Nein

PHASE
Entfernphase          143/143
PHASE
Package State Update Phase          1/1
Package Cache Update Phase          1/1
Image State Update Phase          2/2
PHASE
Vorhandener Index wird gelesen          8/8
Pakete werden indiziert          1/1
```

## Boot-Environments und Systemupdates

Nach der Solaris Installation liegt folgende Struktur als minimales Boot-Environment im zfs rootpool vor :

zpool rpool	
rpool/dump	zvol als Dumpdevice
rpool/swap	zvol als Swapdevice
rpool/ROOT	
rpool/ROOT/solaris	Rootfilesystem der jeweilige Version

Ein Boot-Environment (BE) ist eine bootbare Solaris-Umgebung, die alle zum Boot notwendigen Mountpunkte, Filesysteme, ZFS Datasets und non-global Zonen enthält. Das aktive BE ist das gerade gebootete BE ist.

Werden Updates von Paketen mit pkg image-update installiert, so kann dies im produktiven Betrieb erfolgen. Das reduziert erheblich geplante Ausfallzeiten von Systemen. Hier werden ähnlich wie bei LiveUpgrade von früheren Solaris Versionen, die Updates in ein neues Boot-Environment gespielt. Bei jedem Systemupdate oder dem Einspielen kritischer Updates wird also zunächst eine Kopie des aktuellen BE erzeugt und diese Kopie durch die Updates modifiziert. Die Kopie bleibt zunächst inaktiv. Dieses Verfahren ermöglicht sichere Updates, da Systemänderungen in inaktiven BE durchgeführt werden. Erst nach Fertigstellung des Systemupdates kann das neu erzeugte BE aktiviert werden und ist nach dem nächsten reboot das aktive BE. Im Fehlerfall besteht so die Möglichkeit der Fehlerkorrektur und des Fall-back auf frühere BE.

Systemupdates können einfach mit dem Kommando pkg image-update ausgelöst werden. Dabei verbindet sich das System mit seinen Repositories und prüft, welche Updates für installierte Pakete vorliegen. Fordern die Updates ein neues Boot-Environment, wird zunächst ein neues BE erzeugt. Dazu wird ein ZFS Snapshot und ein darauf aufbauender Clone des Rootfilesystems erzeugt. Darin werden die Updates installiert. Ausgehend von der obigen Struktur ergibt sich danach die folgende beispielhafte Struktur:

zpool rpool	
rpool/dump	zvol als Dumpdevice
rpool/swap	zvol als Swapdevice
rpool/ROOT	
rpool/ROOT/solaris	Rootfilesystem der alten Version
rpool/ROOT/solaris@2010-10-09-21:01:22	Snapshot
rpool/ROOT/solaris-20101001	Rootfilesystem der neuen Version

Es können mehrere BE unter verschiedenen Namen verwaltet werden und so verschiedene alternative Softwarestände zum Betrieb bereitgehalten werden. Die Bootumgebungen werden mit dem Kommando beadm(1M) administriert. Die folgende Seite zeigt einen typischen Ablauf eines pkg image-update. und das alternativ verfügbare GUI-Tool pm\_updatemanager(1) das zur Erzeugung des Boot-Environments solaris-20101001 führte.

```
# beadm list
BE           Active Mountpoint Space  Policy Created
--           -
solaris      N      /           6.69M  static 2010-09-16 22:51
solaris-20101001 R      -          12.64G  static 2010-10-01 22:01
```

```

# pfexec pkg image-update --be-name solaris-20101001
      Packages to install:    4
      Packages to update:   983
      Create boot environment: Ja
DOWNLOAD      PKGS          FILES          XFER (MB)
Completed     987/987         11350/11350    309.6/309.6$<3>
PHASE
Entfernphase              ACTIONS
                          7333/7333
Installationsphase       7613/7613
Aktualisierungsphase     18955/18955
PHASE
Package State Update Phase 1970/1970
Package Cache Update Phase 983/983
Image State Update Phase  2/2
PHASE
Vorhandener Index wird gelesen 8/8
Pakete werden indiziert      987/987
Pakete werden indiziert      987/987
Optimizing Index...
PHASE
Pakete werden indiziert      993/993

```

Ein Klon von solaris existiert und ist aktualisiert und aktiviert worden. Beim nächsten Start wird die Startumgebung solaris-20101001 auf '/' geladen. Starten Sie neu, wenn Sie zum Wechsel auf diese aktualisierte SU bereit sind.



Abb. 2: pm\_updatemanager(1)



Alternativ kann ein Systemupdate oder eine Neuinstallation auch durch die folgende Kombination aus beadm und pkg erzeugt werden.

Hier wird zunächst ein leeres Boot-Environment erzeugt und dieses an /mnt gemounted. Danach wird vom Standard-Repository das Software-Cluster „entire“ der Build-Version 0.149 installiert. Danach kann nach dem Aktualisieren des Boot-Archives, das Boot-Environment aktiviert und gebootet werden.

```
# beadm create solaris-20101001
# beadm mount solaris-20101001 /mnt
# pkg -R /mnt install entire@0.5.11-0.149
# bootadm update-archive -R /mnt
# beadm unmount solaris-20101001
# beadm activate solaris-20101001
```

### Automated Installer (AI)

Der Automated Installer (AI) bietet die Möglichkeit der automatischen Installation ab Oracle Solaris 11. AI richtet sich an RZ-Betreiber, die mehrere Systeme sehr schnell und nach einem einheitlichen Standard ohne manuelles Eingreifen installieren wollen.

Bei der Automated Installation lädt ein Client ein Mini-Solaris per http-Protokoll von einem Install-Server (AI-Server) über das Netzwerk und installiert Software-Pakete aus einem oder mehreren IPS-Repositories. Manifeste legen die Konfiguration fest. Sollen weitere zusätzliche Anpassungen des Clients nach der Installation automatisch vorgenommen werden, so kann dies mit SMF-Services erfolgen, die genau einmal beim ersten Boot nach einer Automated Installation ausgeführt werden. Details dazu siehe in [5]. Das folgende Bild veranschaulicht das AI-Prinzip:

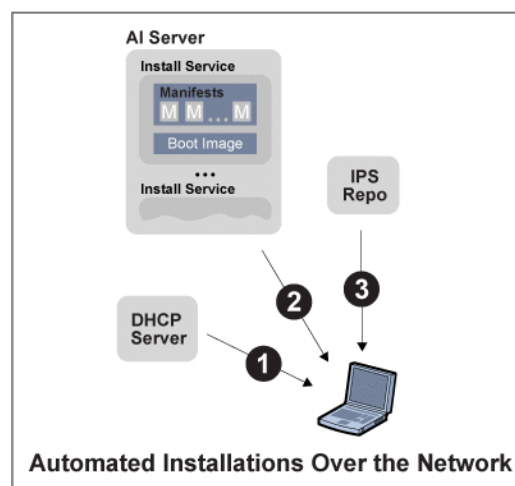


Abb. 3: Grundprinzip Automated Installer

AI wird für x86- als auch für SPARC-Clients verwendet. Zur Initiierung des Downloads des Bootimage wird bei den Clients der Netzwerkboot Mechanismus verwendet. Ein SPARC-Client muss hierzu im OpenBootPROM (OBP) den WANboot-Mechanismus unterstützen, ein x86-Client über entsprechende Erweiterungen im BIOS oder auf den Netzwerkkarten verfügen. Ist der Boot eines Clients nicht per Netzwerk möglich, so kann die Installation von einer lokalen AI-CD gestartet

werden, mit der das System zunächst gebootet wird. Die eigentliche - über Manifeste gesteuerte - Installation erfolgt danach automatisch.

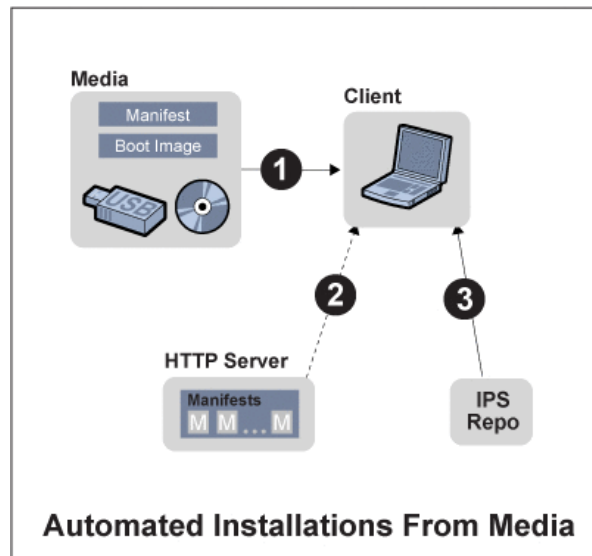


Abb. 4: Automated Installation - von einem lokalen Medium gestartet

Das folgende Bild zeigt den gesamten Ablauf einer automatischen Installation.

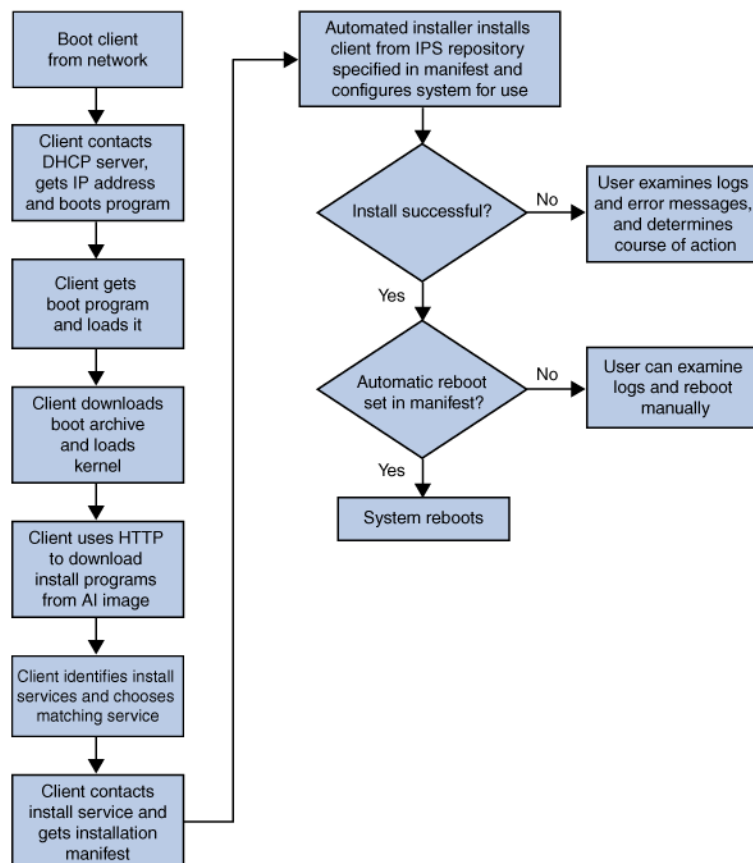


Abb. 5: Ablaufplan einer automatischen Installation

Für das Setup einer Automated Installation werden die folgenden Dinge benötigt:

- **AI Software:** Das IPS Paket installadm mit seinen Abhängigkeiten. Das Kommando installadm(1M) ist das zentrale Management-Tool für den Automated Installer.
- **Manifeste:** Zur Festlegung der zu installierenden Pakete und Einstellungen.
- **IPS-Repository:** Zur Installation der Software.
- **AI-Medium:** Zur Einrichtung des Boot-Images des Installservers oder auch. zum Standalone-Booten des Clients.

AI-Medien werden bereitgestellt, können jedoch auch selbst erzeugt werden. Dazu wird der Distribution Constructor (DC) distro\_const(1M) verwendet. DC erzeugt Images für die interaktive Installation, Boot-Images für die Automatische Installation oder Images für Virtuelle Maschinen wie Oracle VM für x86, VMware oder Oracle VirtualBox. Details siehe [7].

Die automatische Installation von Clients wird durch 2 XML-Manifeste beschrieben, die je Client oder für Gruppen von Clients festgelegt werden können.

- **AI-Manifest:** Legt die Installation fest (Disk, Partitionierung, IPS-Repository, Packages)
- **SC-Manifest:** Legt die Systemkonfiguration nach der Installation fest (Zeitzone, root-Passwort, vorkonfigurierte Nutzer)

Um bestimmte AI- und SC-Manifeste mit einem Client zu verknüpfen, werden **Criteria-Manifeste** erzeugt oder den Manifesten selbst Kriterien zugewiesen. Festlegbare Kriterien können Werte oder Bereiche der folgenden Bereiche sein:

- ARCH: Architektur i86pc/sun4u/sun4v
- CPU: i386/sparc
- IPV4: IP-Adressen
- MAC: Ethernetadressen
- MEM: Größe des ermittelten Hauptspeichers
- PLATFORM: i86pc oder Ausgabe von uname -i

Die Grundarbeitsweise des AI-Servers ist danach wie folgt:

- Zur Einrichtung eines AI-Servers wird ein DHCP-Server aktiviert, der die Installations-Clients mit IP-Adressen und Einstellungen versorgt.
- Ein konfigurierter Web-Server stellt ein Mini-Solaris System bereit, das gebootet und während der Installation benutzt wird.
- Die Clients laden per TFTP- und HTTP-Protokoll das Mini-Solaris System und booten das Solaris.
- Anhand von Standard-Einstellungen im Default-Manifest oder mit Kriterien werden die zum Client passenden Manifeste und Einstellungen ermittelt.
- Die Installation erfolgt durch den Download der Pakete aus den angegebenen Repositories.
- Nachträgliche Ergänzungen der Installation können durch eigene SMF Services erfolgen, die genau einmal beim ersten Boot nach einer Automated Installation ablaufen.

Im Folgenden wird an einem einfachen Beispiel die Einrichtung eines AI-Servers gezeigt. Bei der Darstellung wird lediglich auf die wesentlichen Punkte Bezug genommen. Folgende Besonderheiten liegen vor:

- Der AI-Server wird in einer Zone (ai-zone) erzeugt.
  - Durch Bug 6852739 ist der für den AI-Server notwendige Multicast DNS-Daemon nicht in einer Zone lauffähig. Für dieses Setup wurde ein Workaround geschaffen.
  - Es wird ein Etherstub mit zwei VNIC (ai\_vnic1, ai\_test\_vnic1) erzeugt.
  - Der ai-zone ist der IP-Type exclusive und das VNIC ai\_vnic1 zugewiesen.
  - Das VNIC ai\_test\_vnic1 ist mit einer VirtualBox-Instanz ai\_test zugewiesen.
- Im AI-Server wird aus einem Repository Archiv ein lokales Repository erzeugt.
- Fortan fungiert ai-zone als Installserver und Repository Server und ai\_test als Client.

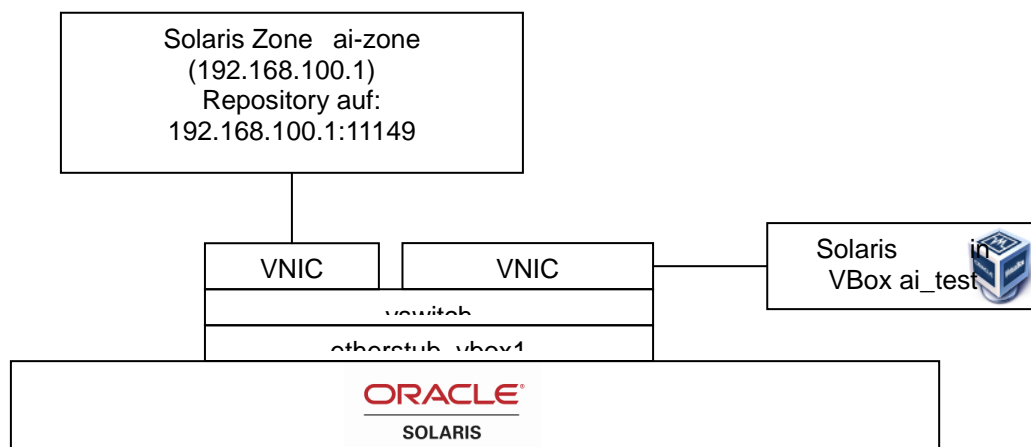


Abb. 6: Beispielaufbau AI-Server in einer Zone

1. Etherstub erzeugen  

```
global # dladm create-etherstub etherstub_vbox1
```
2. VNIC für ai-zone erzeugen  

```
global # dladm create-vnic -l etherstub_vbox1 ai_vnic1
```
3. VNIC für VirtualBox Instanz (den Test-Clients) erzeugen  

```
global # dladm create-vnic -l etherstub_vbox1 ai_test_vnic1
```
4. ai-zone Zone konfigurieren und installieren  
 (set ip-type=exclusive, add net set physical=ai\_vnic1)  
 (IP-Adresse der Zone nach Installation auf 192.168.100.1 setzen)
5. Download der Kopie des Solaris Repositories (sol-11-repo-149-full.iso) und Installation des Repository Servers in der ai-zone auf Port 11146 (siehe README zum ISO-File)
6. Package für Installserver installieren  

```
ai-zone # pkg install installadm
```

7. Verzeichnis für AI-Image und Installserver-Daten anlegen

```
ai-zone # mkdir -p /ai/img
```

8. Download AI-Image und ablegen

```
ai-zone # cp sol-11-dev-149-ai-x86.iso /ai/img
```

9. Installservice erzeugen: Hier wird ein Name nach der Version des ISO-Image gewählt. Mit dem Kommando wird gleichzeitig der Web-Server und der DHCP-Server konfiguriert, sowie das Mini-Solaris Bootimage erzeugt.

Die Clients sollen ihre Adressen ab 192.168.100.100 erhalten. Zunächst wird eine Adresse reserviert. (Dieser Weg ist sehr einfach, da er alle nötigen Schritte in diesem einen Kommando enthält. In den Handbüchern ist beschrieben, welche zusätzlichen Schritte zu gehen sind, wenn ein bereits bestehender DHCP-Service benutzt werden soll.)

```
ai-zone # installadm create-service -n 149x86 \  
-i 192.168.100.100 -c 1 \  
-s /ai/img/sol-11-dev-149-ai-x86.iso /ai/149x86
```

10. Fertig ! Der Installserver ist nun bereits nutzbar. Von der Oracle VirtualBox aus kann über das VNIC ai\_test\_vnic1 eine automatische Installation angestoßen werden.

Nach den Standardeinstellungen antwortet der Server auf Netzboot-Anfragen, jedoch auch auf alle anderen DHCP-Anfragen. Das ist jedoch hier kein Problem, da sich die beiden VNIC in einem separierten Etherstub befinden. Wenn trotzdem der DHCP-Dienst auf bestimmte Clients limitiert werden soll, können die folgenden Schritte genutzt werden.

11. So kann man sich die Client-Tabelle des DHCP-Servers ansehen.

```
ai-zone # pntadm -P 192.168.100.0
```

12. Der ersten Adresse kann direkt die MAC-Adresse (hier 2:8:20:cb:82:ac) des berechtigten Clients zugewiesen werden und die DHCP-Lease Time muss auf unendlich gesetzt werden.

```
ai-zone # pntadm -M 192.168.100.100 -i 01020820CB82AC \  
-f PERMANENT 192.168.100.0
```

13. Weitere Clients (hier 2:8:20:cb:45:df) können direkt nachgetragen werden und dem Install-Service 149x86 zugewiesen werden.

```
ai-zone # pntadm -A 192.168.100.101 -i 01020820CB45DF \  
-m dhcp_macro_149x86 -f PERMANENT 192.168.100.0
```

Der so installierte AI-Server arbeitet mit seinen Standards. Weitere Einstellungen können vorgenommen werden, wenn Clients direkt mit installadm create-client am Installserver registriert werden (hier ist trotzdem noch die jeweilige Adresse im DHCP-Server nachzutragen, z.B. mit Schritt 13). Weitere Anpassungen können an den AI- und SC-Manifesten vorgenommen werden. Das Standard-Manifest des 149x86-Service ist `/ai/149x86/auto_install/default.xml` und wird bei jeder Installation von diesem Service benutzt, wenn keine weiteren Kriterien genutzt werden. Die Datei ist selbsterklärend und zeigt die Einstellungsmöglichkeiten.

Zusätzlich können weitere AI- und SC-Manifeste erzeugt werden, die mit bestimmten Kriterien einem Client zugeordnet werden.

So kann z.B. ein eigenes AI\_Manifest mit einem embedded SC\_Manifest erzeugt werden, das alle Konfigurationen eines Client enthält. Dieses kann dann einem Client mit einem Kriterium (z.B. IP-Adresse=192.168.100.100) zugewiesen werden.

```
ai-zone # cd /ai/149x86/auto_install/
ai-zone # cp default.xml ai-test.xml
(... Änderungen einarbeiten, z.B. <ai_instance name="ai-test"> usw.)
ai-zone # installadm add-manifest -m ai-test.xml -n 149x86 \
        -c IPV4=192.168.100.100
```

Installadm zeigt die installierten Manifeste und Kriterien.

```
ai-zone # installadm list -m -c -n 149x86

Service Name Client Address      Arch  Image Path
-----
149x86      02:08:20:CB:82:AC x86   /ai/149x86
           02:08:20:B6:F5:92 x86   /ai/149x86

Manifest      Criteria
-----
ai-test.xml  ipv4 = 192.168.100.100
```

Über diesen Weg können flexible Anpassungen des AI-Servers für die unterschiedlichsten Anforderungen von Clients vorgenommen werden.

## **Zusammenfassung**

Das vorliegende Papier zeigt im Überblick die umfassenden Änderungen am Software-Lifecycle Management von Oracle Solaris 11. Nach wie vor sind viele Funktionalitäten in der Weiterentwicklung, jedoch zeigen sich bereits jetzt grundlegende Veränderungen gegenüber Solaris 10, die es zu meistern gilt.

## **Literaturverzeichnis**

- [1] Projekt Image Packaging System <http://hub.opensolaris.org/bin/view/Project+pkg/>
- [2] Projekt Caiman, Solaris Install Revisited <http://hub.opensolaris.org/bin/view/Project+caiman/>
- [3] Automated Installation Project [http://hub.opensolaris.org/bin/view/Project+caiman/auto\\_install/](http://hub.opensolaris.org/bin/view/Project+caiman/auto_install/)
- [4] Wiki Automated Use Cases  
<http://wikis.sun.com/display/OpenSolarisInfo/Automated+Installer+Use+Cases>
- [5] Create a First-Boot SMF Service To Run After Automated Installer Install  
<http://wikis.sun.com/display/OpenSolarisInfo/Create+a+First+Boot+Service>
- [6] Handbuch Automated Installer: `/usr/share/doc/auto-install`
- [7] Handbuch Distribution Constructor: `/usr/share/doc/distro_const/`

## **Kontaktadresse:**

### **Detlef Drewanz**

Oracle Deutschland B.V. & Co. KG  
Komturstraße 18a  
D-12099 Berlin

Telefon: +49 (0) 30-747096 856  
Fax: +49 (0) 30-747096 878  
E-Mail: [Detlef.Drewanz@oracle.com](mailto:Detlef.Drewanz@oracle.com)  
Internet: [www.oracle.de](http://www.oracle.de)