

Software Distribution, Installation, Patch & Configuration Management

Handbuch zur SWDIST-Administration



SUSE LINUX GmbH, Frankfurt
F. Herschel, C. Lorenz, L. Pinne, Dr. S. Siano, A. Schmidt

Date: 2007/06/26 19:19:19

Revision: 1.26

Contents

1	Einleitung	6
1.1	Fokus dieses Dokuments	6
1.2	Ansprechpartner	6
2	Überblick	7
2.1	Funktionale Eigenschaften	7
2.2	Architektur und Infrastruktur	8
2.3	Vorbedingungen, Kompatibilität	9
2.3.1	Software	9
2.3.2	Target-Hardware	10
2.3.3	Server-Hardware und Netzwerk	10
2.4	Verzeichnisstruktur Installserver	11
2.4.1	Package Root	11
2.4.2	Document Root	12
2.4.3	Abhängigkeiten und Beschränkungen	12
2.4.4	Bootserver-Bereich	12
2.5	Arbeitsverzeichnis Configuration Management	12
2.5.1	Templates Directory	13
2.5.2	Classes Directory	13
2.5.3	Info Directory	13
2.6	Datenmodell	14
2.7	Install- und Application-Server initial füllen	14
2.8	Diagnose und Bug Reporting	14
3	Basisdaten Management on Server	16
3.1	Admin-User in APP-Server anlegen	16
3.2	Architektur in APP-Server anlegen	16
4	Software Management on Server	18
4.1	Distribution in APP-Server anlegen	18
4.2	Mediaset in Install- und APP-Server importieren	18
4.2.1	Mounten der Media-Sets in den Install-Server	19
4.2.2	Anmelden der Media-Sets im APP-Server	19
4.3	Media Set aus APP- und Install-Server löschen	20
4.3.1	Löschen des Media Set im APP-Server	21
4.3.2	Löschen des Media Set auf dem Install Server	21
4.4	RPM zu Media Set in APP- und Install-Server hinzufügen	21
4.4.1	Kopie des rpm-Paketes	22
4.4.2	Neuaufbau der Paketbeschreibungen	22
4.4.3	Neuer Import des Media Set	22
4.5	RPM aus Media Set in APP- und Install-Server löschen	22
4.5.1	Paket aus dem Installserver löschen	23
4.5.2	Neuaufbau der Paketbeschreibungen des Installservers	23
4.5.3	Neuer Import des Media Set in den APP-Server	23
4.6	Konsistenz für Mediasets prüfen	23
4.7	Stream in APP-Server anlegen	23

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



4.8	Stream aus APP- in Install-Server exportieren	24
4.9	Stream aus APP- und Install-Server löschen	25
4.9.1	Stream aus Install-Server löschen	26
4.9.2	Stream aus APP-Server löschen	26
4.10	Grace Period für Stream ändern	26
4.11	Konsistenz für Streams prüfen	27
4.12	Mediaset zu Stream hinzufügen	28
4.12.1	Mediaset auf Install Server importieren	28
4.12.2	Mediaset in APP Server importieren	28
4.13	Channel in APP-Server anlegen	28
4.14	Selectionlist zu Channel in APP-Server hinzufügen	29
4.15	Selectionlist aus Channel in APP-Server löschen	30
4.16	Selectionlist in APP-Server importieren	30
4.17	Selectionlist aus APP-Server löschen	31
4.18	Custom Patches in APP- und Inst-Server importieren	32
4.18.1	Berechnen der MD5Sum	32
4.18.2	Anmelden des Patch am APP-Server	32
4.18.3	Kopie des Patches auf dem Installationsserver	32
4.19	Patch aus Stream löschen	32
4.20	SUSE Patches in APP-Server importieren	33
4.21	Patch zu Stream hinzufügen	34
4.22	Patch aus APP-Server löschen	35
4.23	Konsistenz für Patches prüfen	36
4.24	Nützliche Befehle	37
5	Target Management on Server	38
5.1	Allgemeines	38
5.2	Target in APP-Server anlegen	38
5.3	Konfiguration für Target importieren	39
5.4	Target RPMs zeigen	40
5.5	Target Status zeigen	40
5.6	Stream für Target ändern	41
5.7	Klasse für Target ändern	41
5.7.1	Klasse in APP-Server ändern	42
5.7.2	Klasse in Alice2-Konfiguration ändern	42
5.8	Architektur für Target in APP Server ändern	43
5.9	Alice2-Konfiguration in APP-Server importieren	43
5.10	Alice2-Konfiguration aus APP-Server exportieren	43
5.11	AutoYaST Control File generieren	44
5.12	Nützliche Befehle	45
6	Target Installation & Update	46
6.1	Allgemeines	46
6.2	Target via Netzwerk installieren	48
6.2.1	System bootet von Floppy	48
6.2.2	YaST installiert	48
6.2.3	ALICE startet Konfiglauf	48
6.2.4	Finaler Reboot	48
6.3	Target RPMs und Konfiguration manuell updaten (ohne GUI)	49
6.4	Target RPMs und Konfiguration manuell updaten (mit GUI)	49
6.5	Nützliche Befehle	50



Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------

List of Tables

1.1	Ansprechpartner	6
2.1	Maschinen-Rollen	8
2.2	Hardware Install- and Configuration-Server	10
2.3	Übertragungsverfahren	10
2.4	Objekte	14
2.5	Server Log Files	14
2.6	Target Log Files	15
6.1	Server Config Files	47
6.2	Target Config Files	47

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



List of Figures

2.1	Client GUI	7
2.2	Datenfluß	8
2.3	Installserver	11
2.4	Bootserver auf Installserver	13
6.1	Configuration Update	47
6.2	Architektur	57
6.3	Datenmodell	58

1 Einleitung

Das Framework für Software Distribution, Installation, Patch & Configuration Management (SWDIST) stellt Verfahren und Komponenten bereit, mit denen automatische Installation, Software-Verteilung sowie Patch- und Release-Management für Workstations und Server auf Basis des Betriebssystems SUSE LINUX Enterprise Server (SLES)¹ zuverlässig und kostensparend durchgeführt werden können. Die vorgestellte Lösung enthält darüberhinaus Module für Configuration-Management und Technical Repository.

1.1 Fokus dieses Dokuments

Das vorliegende Dokument ist das Handbuch für Software-Verteilung und Patch-Management. Zielgruppe des Dokumentes sind die für diese Aufgaben zuständigen Mitarbeiter, Mitarbeiter der Hotline sowie Serverbetreuer. Der technische Betrieb des Install- und Configuration- Servers wird in einem weiteren Dokument beschrieben. Das Erstellen von Software-Paketen und Mediasets wird in diesem Dokument nicht beschrieben.

Das SWDIST-Framework unterstützt folgenden Aufgaben:

- Initiale Installation von Workstations und Servern
- Verteilung von Software und Konfigurationsdaten
- Verteilung von Patches und Updates

Die dabei anfallenden Arbeiten sind in diesem Dokument in folgende Bereiche gegliedert:

- Software verwalten, Tasks auf Server
- Targets verwalten, Tasks auf Server
- Targets installieren und updaten

1.2 Ansprechpartner

<i>Funktion</i>	<i>Name</i>	<i>E-Mail</i>
Kontakt SUSE	Lars Pinne	lars.pinne@novell.com

Table 1.1: Ansprechpartner

¹Sowie der Derivate bzw. kompatiblen Produkte.

2 Überblick

2.1 Funktionale Eigenschaften

Die Hauptaufgaben sind automatische Installation, Software-Verteilung sowie Patch- und Release-Management. Dabei ist es möglich, zwischen *emergency* (sofort verpflichtenden) und *optional* (zeitweilig zurückstellbaren) Updates zu unterscheiden. SUSE-Patches werden aus dem SUSE-Maintenance-Web bezogen. Patches dritter können ebenfalls verwaltet werden. Alle Software, auch Patches, werden als RPM-Pakete verwaltet. Darüber hinaus werden Konfigurationsdaten verwaltet. Konfigurationsdaten, Zustandsdaten und Software-Pakete werden zentral, jedoch getrennt gespeichert. Heterogene Target-Hardware wird automatisch erkannt und unterstützt, dabei ist pro HW-Architektur (Intel 32-bit, AMD 64-bit) eine Version des Betriebssystems erforderlich.

Es werden sowohl *pull* als auch *push* zur Verteilung unterstützt. ¹ *pull* erfolgt sowohl bei Logout als auch zeitgesteuert. Die Benutzer werden per GUI über anstehende Updates informiert (siehe Abbildung 2.1).

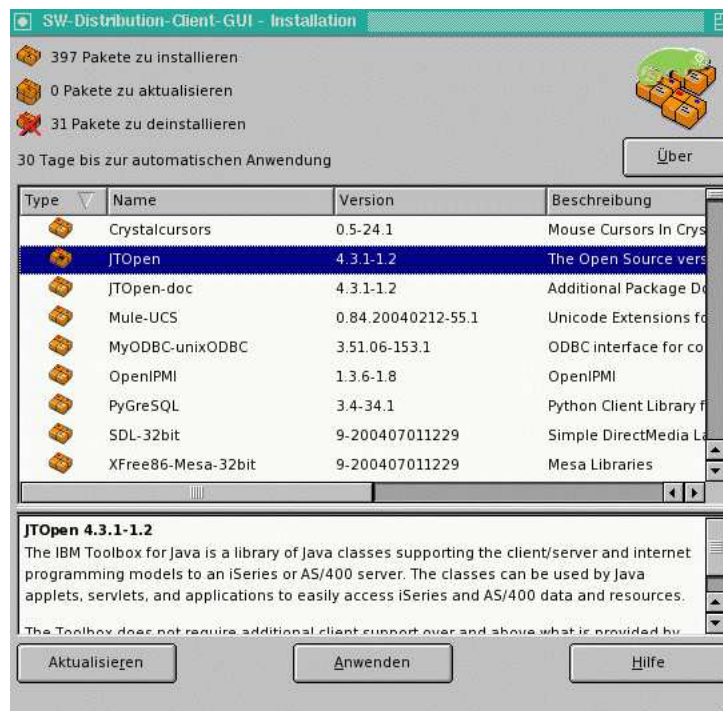


Figure 2.1: Client GUI

Zeitgesteuert (z.B. täglich) und nach jedem Update wird der Ist-Zustand von den Targets übertragen. Damit kann der letzte bekannte Ist-Zustand der Targets zentral gespeichert werden.

Die eingesetzten Verfahren stellen definierte und konsistente Softwarestände auf den Targets sicher und erstellen automatisch Software- und Hardware- Inventare. Die für die Installation erforderlichen Konfigurationsdaten der Targets werden zentral verwaltet. Die Konfigurationsdaten der Anwendungen können bei Bedarf ebenfalls zentral gehalten, jedoch nicht granular administriert werden.

¹push ist als server-getriggertes pull realisiert

Der Application Server unterscheidet die drei administrativen Rollen Superuser, Administrator und Maschine. Betriebssystemseitig sind für SWDIST-Administratoren Lese- und Schreib-Zugriffe auf das Dateisystem des Installservers im Bootserver-Bereich, im Document Root, im Package Root und ggf. im Arbeitsverzeichnis für Configuration Management erforderlich.

2.2 Architektur und Infrastruktur

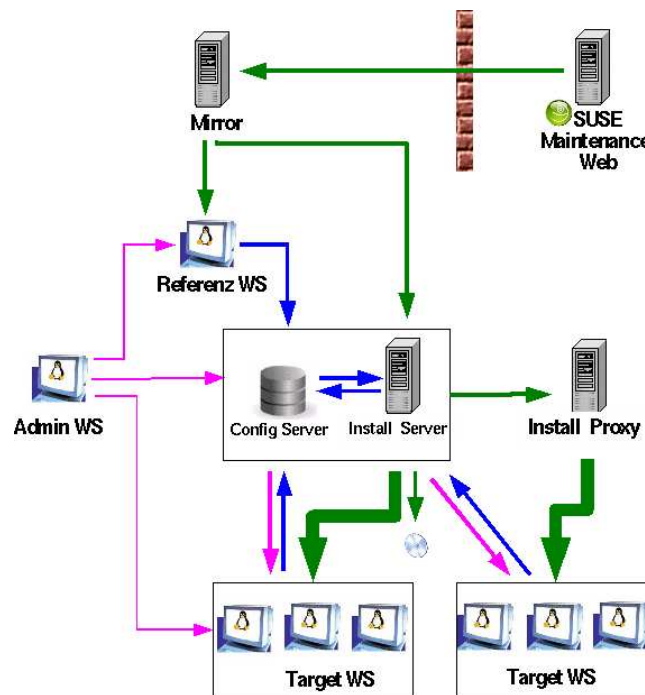


Figure 2.2: Datenfluß

Neben den zu installierenden und zu updatenden Targets werden weitere Maschinen-Rollen definiert. Diese Rollen können, abhängig von den Anforderungen und Gegebenheiten, auf reale Maschinen abgebildet werden. Die Tabelle 2.1 zeigt diese Zusammenhänge. Der modulare Aufbau ermöglicht es, das System an unterschiedliche Bedürfnisse anzupassen. Die Architektur ist im Anhang in Abbildung 6.2 detailliert dargestellt.²

<i>Funktion</i>	<i>Anzahl</i>	<i>Bemerkung</i>
SUSE Mirror	1	optional, auf Config Server
Referenz WS	min. 1	
Config Server	1	
Technical Repository	1	auf Config Server
Install Server	1	auf Config Server
Install Proxy	-	optional, abh. von Zahl Targets
Target WS	50-50000	

Table 2.1: Maschinen-Rollen

²Die maximale Anzahl Targets hängt stark von zahlreichen Parametern ab.

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



2.3 Vorbedingungen, Kompatibilität

2.3.1 Software

Der Install- und Application-Server läuft unter SLES8-i386. Auf dem Server ist zusätzlich zu dem SUSE-kompatiblen Installserver folgende Software installiert:

- alice-2.6.1-8.noarch.rpm
- IBMJava2-JRE_1_4-1.4.1-5.i586.rpm
- IBMJava2-SDK_1_4-1.4.1-5.i586.rpm
- jboss-3.2.5-1
- swdistappserv-1.0-278.noarch.rpm
- swdistappserv-javadoc-1.0-278.noarch.rpm
- swdistclient-1.0-283.i386.rpm
- swdistclient-admin-1.0-283.i386.rpm
- swdistclient-1.0-283.i386.rpm
- swdistclient-doc-1.0-283.i386.rpm
- youmirror-1.0-278.i386.rpm

Die Admin-Workstation läuft unter SLES8-i386. Alternativ können die administrativen Tasks auf dem Install- und Application-Server ausgeführt werden. Auf der Admin-Workstation ist folgende Software installiert:

- alice-2.6.1-8.noarch.rpm
- swdistclient-1.0-283.i386.rpm
- swdistclient-admin-1.0-283.i386.rpm
- swdistclient-cld-1.0-283.i386.rpm
- swdistclient-clgui-1.0-283.i386.rpm
- swdistclient-doc-1.0-283.i386.rpm
- swdistappserv-javadoc-1.0-278.noarch.rpm

Auf den Targets wird folgende Software zur Einbindung in das Linux Software Deployment Framework installiert:

- alice-2.6.1-8.noarch.rpm
- swdistclient-1.0-283.i386.rpm
- swdistclient-cld-1.0-283.i386.rpm
- swdistclient-clgui-1.0-283.i386.rpm

Die Targets können mit den unten genannten Betriebssystemen betrieben werden. Test und Abnahme erfolgen mit SLES9-x86_64.

- SLES8-i386 (SP3, skelcd-yast2-2004.5.11-1.x86_64)
- SLES9-i386
- SLES9-x86_64

2.3.2 Target-Hardware

Die Target-Systeme basieren auf heterogener Hardware, sowohl Intel 32-bit (i386) als auch AMD 64-bit (x86_64). Für jede Architektur ist jeweils eine Version des Betriebssystems erforderlich.

Die Hardware muß über die automatische Hardware-Erkennung des `AutoYaST2` erkannt werden. Hierbei werden vielfältige aktuelle Hardware- und Peripheriesysteme unterstützt, wobei im Produktivbetrieb eine Zertifizierung der eingesetzten Target-Systeme angestrebt werden soll.

Die Anforderungen für RAM, CPU und Festplatten werden durch das zu installierende Betriebssystem und die Anwendungen vorgegeben.

Die Erstinstallation kann entweder rein Netzwerk-basiert oder mit Bootmedien durchgeführt werden. Für eine rein Netzwerk-basierte automatische Erstinstallation ohne BootMedium ist eine PXE-fähige Netzwerkkarte erforderlich. Als Bootmedium kann entweder eine entsprechend vorbereitete Boot-Floppy (z.B. mit GRUB) oder eine Boot-CD eingesetzt werden. Pakete und Konfiguration werden für die Installation über Netzwerk geladen.

2.3.3 Server-Hardware und Netzwerk

Die Tabelle 2.2 gibt eine Übersicht.

<i>Resource</i>	<i>Minimal</i>	<i>Typical</i>
CPU	800 MHz	2 x 2.0 GHz
RAM	512 MB	2 GB
Disk	20 GB	200 GB
Network	100 Mbit/s	1000 Mbit/s

Table 2.2: Hardware Install- and Configuration-Server

Als Netzwerkumgebung wird ein geschwichtes 100 MBit LAN (Ethernet) vorausgesetzt. Für automatische Installation und Software-Verteilung werden die Dienste DNS, TFTP, DHCP, XNTP und HTTP bzw. HTTPS benötigt. Dies bedeutet, das neben den direkt zur Installation benötigten Protokollen auch Namensauflösung und Zeitsynchronisation erforderlich sind.

HTTP bzw. HTTPS wird zur Übertragung der Software-Pakete und `Alice2`-Konfigurationen eingesetzt. Für die initiale Installation wird TFTP zum Laden des Installationssystems und der `AutoYaST2` Control Files verwendet. Die Tabelle 2.3 gibt eine Übersicht.

<i>Funktion</i>	<i>auto</i>	<i>Protokoll</i>
update RPMs auf Mirror	auto	HTTP/HTTPS
install RPMs auf Referenz WS	man	HTTP/HTTPS
release RPMs, update Config	man	
update released RPMs auf Inst. Server	auto	rsync
update RPMs, Config auf Target WS	auto	HTTP/HTTPS
load install system	man	TFTP

Table 2.3: Übertragungsverfahren

2.4 Verzeichnisstruktur Installserver

Der Installserver stellt die RPM-Pakete sowohl für die initiale Installation der Targets als auch für die spätere Software-Verteilung bereit. Auf die SUSE-kompatible Verzeichnisstruktur des Installservers kann mit AutoYaST2, YaST2, YOU, Alice2, RPM oder `sdcd` zugegriffen werden.

Das Konzept der Software Streams erlaubt es, verschiedene Installations- Umgebungen gleichzeitig bereit zustellen, ohne die zugrunde liegenden Mediasets mehrfach ablegen oder verändern zu müssen.

Um dies zu erreichen, ist der Installserver in die zwei Bereiche Package Root und Document Root gegliedert, wie in Abbildung 2.3 gezeigt. Neben den abgebildeten Dateien sind noch die üblichen Dateien `directory.yast` in den Stream Directories sowie MD5SUMS in den RPM Directories erforderlich.³

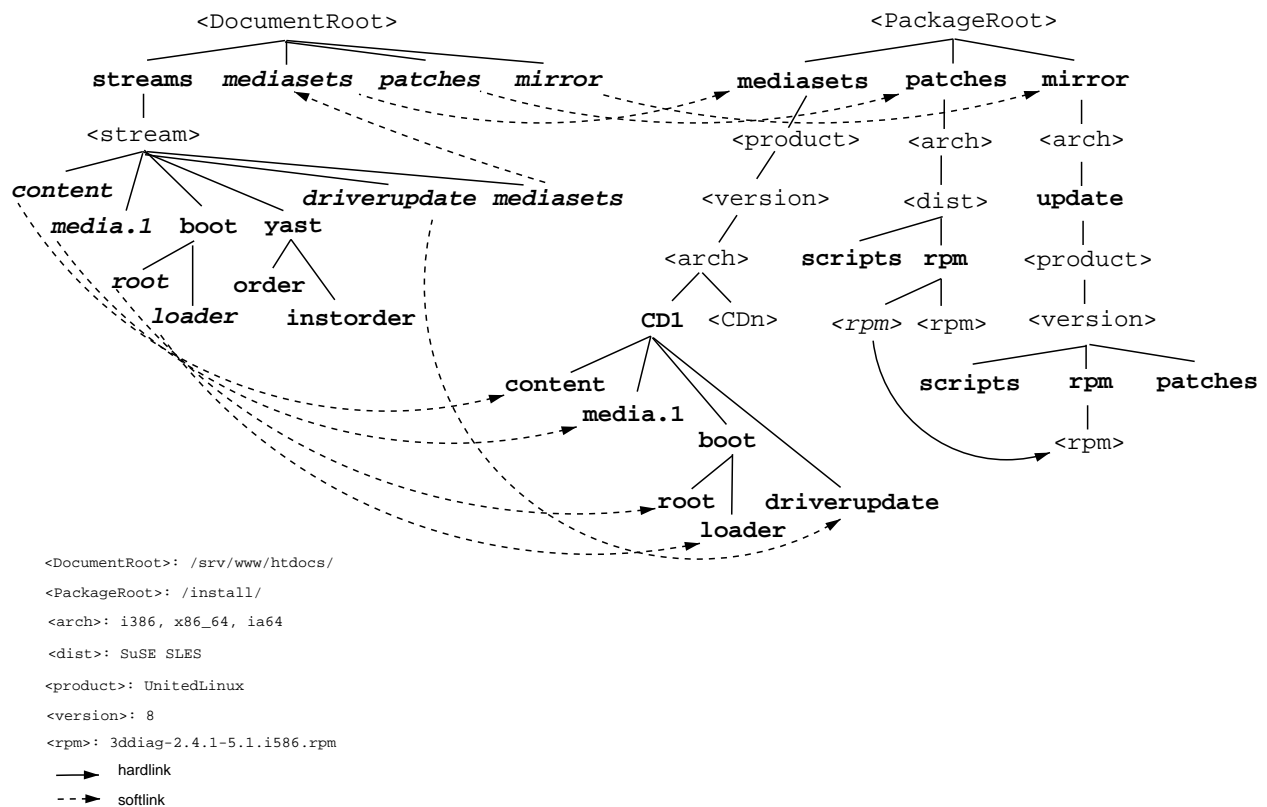


Figure 2.3: Installserver

2.4.1 Package Root

The Package Root directory is the directory to store all installation packages, patches and driverupdates as they are released by SUSE, ISVs or build by customers itself. None of the data has to be modified, once it has been copied into this directory.

³s. Manual Page `installserver.7`

Therefor this directory contains the subdirectories `mediasets`, `mirror`, `patches`. In the `mediasets` directory SUSE compliant mediasets are stored, either copied into the filesystem or loopback mounted from installation media's ISO images (eg. `UnitedLinux-1.0-SP-3-i386-RC4-CD1.iso`). In the `mirror` directory SUSE compliant patches are stored, as they could be retrieved from a YaST Online Update (YOU) server or the SUSE maintenance web. In the `patches` directory site specific or third party patches could be placed.

So, the Package Root serves as input area and physical storage.

2.4.2 Document Root

The Document Root directory is the directory, as seen by the installation targets.

Therefor this directory contains the subdirectory `streams` and symbolic links to the Package Root's subdirectories mentioned above.

In the `streams` directory site specific streams could be established, each represented by one subdirectory (eg. `SLES9-ia64-prd`). Each of this stream subdirectories contains the subdirectories `media.1`, `boot`, `yast` as well as the files `content` and eventually `driverupdate`. All of them are linked to counterparts in the respective mediaset, except the `yast` subdirectory with its files `instorder` and `order`. The files `order` and `instorder` are written for each stream while it is established.

So, the Document Root serves as output area and abstraction layer

2.4.3 Abhängigkeiten und Beschränkungen

Because the installation server substitutes SUSE compliant installation media and sources for YaST Online Update, all files and directories have to have the original's properties.

When used within the kernel's boot parameters paths should be as short as possible.

The `<arch>` architecture identifier could be one of `i386`, `x86_64`, `ia64`, `ppc`, `s390`, `s390x` only and has to be taken from the respective `CD1/content` file. The `<dist>` distribution identifier should contain alpha- numerical characters `[a-z]`, `[A-Z]`, `[0-9]` only. The `<product>` and `<version>` identifiers have to be taken from the respective `CD1/content` file.

If a mediaset is added as installation source to YaST on a target machine, YaST builds a local "cache" for package metadata. Unfortunately there is no known method to update this "cache" except removing of the respective mediaset from YaST's list of installation sources.

Changing mediasets on the server while a target installation is running could result in unpredictable effects on the target.

2.4.4 Bootserver-Bereich

Der Bootserver-Bereich ist in Abbildung 2.4 dargestellt. Wenn Control Files über die MAC-Adresse adressiert werden, ist der Dateiname mit Kleinbustaben, ohne Doppelpunkte und ohne Suffix anzulegen (z.B. `00d0592d0d40`).

2.5 Arbeitsverzeichnis Configuration Management

Auf der Admin-Workstation oder dem Install Server wird ein Arbeitsverzeichnis (`$ALICE_HOME`) genutzt, um Konfigurationsdaten anzulegen, zu ändern und weiterzuverarbeiten.

Das in den Config Files verwendete Tag-basierte Format ähnelt formal einer stark vereinfachten HTML-Notation, bei der keine verschachtelten Tags erlaubt sind. Der Tag-Inhalt entspricht weitgehend üblichen Unix/Linux Config Files. Das Format ist einerseits verlustfrei und mit angemessenem Aufwand in ein RDBMS übertragbar, andererseits ist es in der Lage, alle in einem `AutoYaST2` Control File erforderlichen Informationen abzubilden.⁴

Gespeichert werden alle Konfigurationsdaten jedoch ausschließlich über den Application Server im RDBMS. Die bei Alice verwendeten Hierarchieebenen `templates`, `classes` und `info` werden in entsprechenden Config Groups abgebildet. Die Basisdaten für die autom. Installation werden darüberhinaus als `AutoYaST2` Control File im XML-Format im Bootserver-Bereich bereitgestellt.

The ALICE working directory is separated into three sub-directories named `templates`, `classes` and `info`.

⁴s. Manual Page `alice.8`

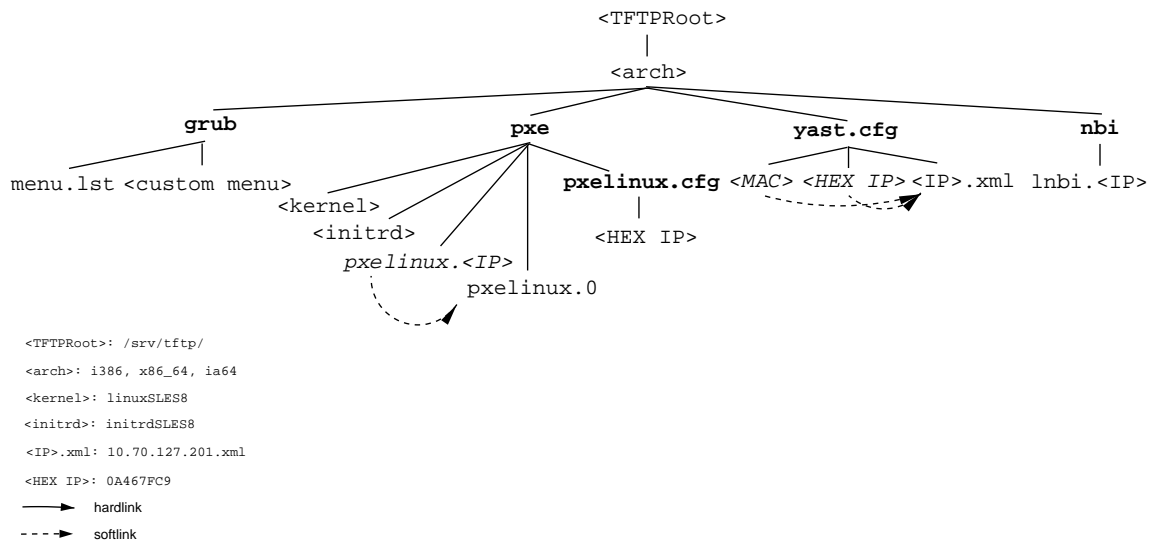


Figure 2.4: Bootserver auf Installserver

2.5.1 Templates Directory

Default files are stored in the `templates` directory. The values defined here have the lowest priority. This means they can be overloaded by values in config files of the following directories. The files are named `<section>.tcf`. Usually there is nothing to change in this directory.

2.5.2 Classes Directory

Class specific files (for one or multiple classes) are stored in the `classes` directory. The values defined here have medium priority. The value of a class does only take effect when you added a system to this class (e.g. if you name a class "alpes" and your machine is named "bernina.sample.suse.de" you have to add bernina to the class alpes using its configuration files). The files are named `<classname>.section.tcf`.

2.5.3 Info Directory

Host specific files (for one or multiple hosts) are stored in the `info` directory. The values in the configuration files of a host have the highest priority. This means if you have added a system to a class which configures a lot, but you want to change single values, you can overload these values here. The files are named `<hostname>.section.tcf`.

2.6 Datenmodell

Tabelle 2.4 zeigt eine vereinfachte Übersicht der Objekte, die zur Verwaltung von Software und Targets verwendet werden. Die Begriffe werden im Glossar näher erläutert. Das Datenmodell ist im Anhang in Abbildung 6.3 detailliert dargestellt.

<i>Objekt</i>	<i>Software</i>	<i>Target</i>	<i>Bedeutung</i>
event		X	Event
config		X	Target-Konfiguration
machine		X	Target
arch	X	X	Architektur
channel	X	X	Channel
patch	X	X	Patch
selectionlist	X	X	Selection List
stream	X	X	Stream
dist	X		Distribution
mediaset	X		Mediaset
packet	X		RPM-Paket

Table 2.4: Objekte

2.7 Install- und Application-Server initial füllen

Die Tabelle ?? gibt einen Überblick darüber, wie die ersten Mediasets importiert und der Application-Server initial gefüllt werden kann.

<i>No.</i>	<i>Aktion</i>	<i>Kapitel</i>	<i>Seite</i>
1	Admin-User anlegen	3.1	16
2	Architektur anlegen	3.2	16
3	Distributionen anlegen	4.1	18
4	Mediasets importieren	4.2	18
5	Patches importieren	4.20	33
6	Selectionlists importieren	4.16	30
7	Stream anlegen	4.7	23
8	Stream exportieren	4.8	24
9	Channel anlegen	4.13	28
10	Selectionlist zu Channel hinzufügen	4.14 bzw. 4.15	29 bzw. 30
11	Target anlegen	5.2	38
12	Alice2-Konfiguration importieren	5.9	43
13	Alice2-Klasse dem Target zuordnen	5.7	41
14	Alice2-Konfiguration für ein Target exportieren	5.10	43
15	AutoYaST Control File für Target erzeugen	5.11	44

Table 2.5: Daten initial bereitstellen

2.8 Diagnose und Bug Reporting

Informationen zur Funktion und Auslastung der Serverprozesse werden in Logfiles aufgezeichnet (siehe Tabelle 2.5). Diese Logfiles sollten regelmäßig auf Unregelmäßigkeiten untersucht werden, um Probleme frühzeitig zu erkennen. Auf den Targets werden ebenfalls Logfiles geschrieben (siehe Tabelle 2.6).

<i>File</i>	<i>Bedeutung</i>
/var/log/messages	allg. Log
/var/log/httpd/access_log	Apache Access Log
/var/log/httpd/error_log	Apache Error Log
/var/log/jboss/default/server.log	Application Server Log
/var/log/xinetd.log	
/var/log/ntp	
/var/log/sa/*	System Monitorig Log
/var/lib/YaST2/you/synclog/<job>-<time>	
/var/log/postgresql	optional, für lokales RDBMS PostgreSQL

Table 2.6: Server Log Files

<i>File</i>	<i>Bedeutung</i>
/var/log/messages	allg. Log
/var/log/boot.msg	Boot-Logs
/var/log/alice_log	Alice Log
/var/log/alice_state_warn.<job>	Alice Warnings
/var/log/alice_state_error.<job>	Alice Errors
/var/log/alice_state_fatal.<job>	Alice Fatal Errors
/var/log/YaST2/y2log*	YaST Log
/var/adm/autoinstall/logs/<script>.log	AutoYaST Custom Script Logs

Table 2.7: Target Log Files

Im Fehlerfalle sind die entsprechenden Richtlinien der zuständigen Stellen, Hardware-Herstellern und dritten zu beachten. Sollte ein Zusammenhang zwischen dem Fehler und der von SUSE gelieferten Software nicht auszuschließen sein, und der Fehler von der Applikationsbetreuung nicht in angemessener Zeit behoben werden können, empfiehlt SUSE folgendes Vorgehen:

- Projektverantwortliche und alle betroffenen Stellen informieren.
- Verbindung zwischen Targets, Application Server und Datenbank prüfen.
- Corefiles und Logfiles sichern.
- Releasestand der laufenden Prozesse feststellen.
- Inventar der installierten Software feststellen.
- Status und Auslastung von Hardware, Filesystemen und Netzwerken feststellen.
- Abhängigkeiten zu sonstigen Systemkomponenten feststellen.

Nach Absprache sollten diese Informationen der SUSE zur Verfügung gestellt werden. Sie bilden die Basis für das weitere Vorgehen.

3 Basisdaten Management on Server

3.1 Admin-User in APP-Server anlegen

Voraussetzungen

- App-Server konfiguriert
- Super User in APP-Server angelegt

Ergebnis

- Admin-User in APP-Server angelegt

Tools

- sdadm

Beschreibung

Mit dem sdadm Admin-Tool wird als nächstes ein Admin-Benutzer angelegt. Für weitere Informationen zu sdadm siehe die Manual-Page sdadm(8).

```
# sdadm -u super -w super
sdadm> admin add hugo biene admin,superuser
sdadm> admin list
super superuser,admin
hugo admin,superuser
sdadm> quit
```

Der Admin-Benutzer hugo mit dem Passwort biene wird angelegt.

Aktionen

- Mit dem Kommando `admin list` kann man überprüfen, ob der gewünschte Benutzer angelegt wurde.

3.2 Architektur in APP-Server anlegen

Voraussetzungen

- App-Server konfiguriert
- Admin-User angelegt (siehe [3.1](#))

Ergebnis

- Architektur in APP-Server angelegt

Tools

- sdadm

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Beschreibung

Mit dem `sdadm` Admin-Tool legt der Admin-User `hugo` die verfügbaren Architekturen an, die später installiert werden können. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
# sdadm -u hugo -w biene
sdadm> arch add noarch
sdadm> arch add i386 "" noarch
sdadm> arch add i486 "" i386
sdadm> arch add i586 "" i486
sdadm> arch add i686 "" i586
sdadm> arch add x86_64 "" i686
sdadm> arch list
sdadm> quit
```

Der Architekturbaum wurde angelegt.

Aktionen

- Mit dem Kommando `arch list` kann man überprüfen, welche Architekturen im Applikationsserver angelegt wurden.

4 Software Management on Server

4.1 Distribution in APP-Server anlegen

Voraussetzungen

- App-Server konfiguriert
- Admin-User angelegt (siehe 3.1)
- Architektur angelegt (siehe 3.2)

Ergebnis

- Distribution in APP-Server angelegt

Tools

- sdadm

Beschreibung

Wurde der Architekturbaum durch den Admin-User angelegt, kann anschließend die installierbare SUSE Distribution mit dem sdadm Admin-Tool angelegt werden. Für weitere Informationen zu sdadm siehe die Manual-Page sdadm(8).

```
# sdadm -u hugo -w biene
sdadm> dist add SLES9 i386 /patches/i386/SLES9
sdadm> dist list
SLES9 i386 /patches/i386/SLES9
sdadm> quit
```

Im obigen Beispiel wurde durch den Admin-User hugo als verfügbare Distribution der SLES9 angelegt.

Aktionen

- Mit dem Kommando `dist list` kann man überprüfen, welche SUSE Distributionen angelegt wurden.

4.2 Mediaset in Install- und APP-Server importieren

Voraussetzungen

- InstallServer installiert und konfiguriert
- Applikation Server installiert und konfiguriert
- Mediasetes als Medien (CDs, ISO-Images) vorhanden

```
/space/iso/SLES-9-i386-RC5-CD1.iso
/space/iso/SLES-9-i386-RC5-CD2.iso
/space/iso/SLES-9-i386-RC5-CD3.iso
/space/iso/SLES-9-i386-RC5-CD4.iso
/space/iso/SLES-9-i386-RC5-CD5.iso
/space/iso/SLES-9-i386-RC5-CD6.iso
```

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



```

/space/iso/SLES-9-x86-64-RC5-CD1.iso
/space/iso/SLES-9-x86-64-RC5-CD2.iso
/space/iso/SLES-9-x86-64-RC5-CD3.iso
/space/iso/SLES-9-x86-64-RC5-CD4.iso
/space/iso/SLES-9-x86-64-RC5-CD5.iso
/space/iso/SLES-9-x86-64-RC5-CD6.iso
/space/iso/SWDIST-SLES9-i386.iso
/space/iso/SWDIST-SLES9-x86_64.iso

```

Ergebnis

- Media-Set auf Install Server repliziert
- Media-Set Definitionen im APP-Server

Tools

- mount
- sdswhimp

4.2.1 Mouneten der Media-Sets in den Install-Server

Beschreibung

Es gibt verschiedene Möglichkeiten die jeweiligen Mediasets auf den Installationsserver zu kopieren. In der SUSE-Testumgebung wurden die SLES9 Installations-CDs als ISO-Images per 'loop-mount' in das Filesystem von howard eingehängt. Hierzu editiert man die Datei '/etc/fstab' nachdem man die gewünschten Mount-Points mit dem Kommando 'mkdir' auf dem Installationsserver angelegt hat. Mit dem Kommando 'mount -a -t iso9660' werden anschließend die Mediasets gemounted und auf dem Installationsserver zur Verfügung gestellt.

Da das SWDIST-Mediaset auf keiner SUSE Distribution enthalten ist, muss eine Mediaset-kompatible Verzeichnisstruktur auf dem Installationsserver integriert werden. Weitere Infomationen zur Mediaset-Struktur sind in der Manual-Page von `instserver(7)` zu finden. In der SUSE-Testumgebung wurde das SWDIST-Mediaset als ISO-Image analog zu den SLES9 Installations-CDs als ISO-Image per 'loop-mount' in das Filesystem eingehängt.

Beispiel: Die folgenden Einträge wurden der Datei '/etc/fstab' hinzugefügt:

```

/space/iso/SLES-9-i386-RC5-CD1.iso /space/mediasets/SUSE-SLES/9/i386/CD1 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-i386-RC5-CD2.iso /space/mediasets/SUSE-CORE/9/i386/CD1 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-i386-RC5-CD3.iso /space/mediasets/SUSE-CORE/9/i386/CD2 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-i386-RC5-CD4.iso /space/mediasets/SUSE-CORE/9/i386/CD3 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-i386-RC5-CD5.iso /space/mediasets/SUSE-CORE/9/i386/CD4 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-i386-RC5-CD6.iso /space/mediasets/SUSE-CORE/9/i386/CD5 iso9660 defaults,loop,ro 0 0

/space/iso/SLES-9-x86-64-RC5-CD1.iso /space/mediasets/SUSE-SLES/9/x86_64/CD1 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-x86-64-RC5-CD2.iso /space/mediasets/SUSE-CORE/9/x86_64/CD1 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-x86-64-RC5-CD3.iso /space/mediasets/SUSE-CORE/9/x86_64/CD2 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-x86-64-RC5-CD4.iso /space/mediasets/SUSE-CORE/9/x86_64/CD3 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-x86-64-RC5-CD5.iso /space/mediasets/SUSE-CORE/9/x86_64/CD4 iso9660 defaults,loop,ro 0 0
/space/iso/SLES-9-x86-64-RC5-CD6.iso /space/mediasets/SUSE-CORE/9/x86_64/CD5 iso9660 defaults,loop,ro 0 0

/space/iso/SWDIST-SLES9.iso /space/mediasets/SWDIST-SLES9/1/i386/CD1 iso9660 defaults,loop,ro 0 0

```

```
# mount -a -t iso9660
```

Aktionen

- Mit dem Kommando `df` kann anschließend überprüft werden, ob alle benötigten Mediasets auf dem Installationsserver verfügbar sind.

4.2.2 Anmelden der Media-Sets im APP-Server

Voraussetzungen

- Mediasets auf dem Installationsserver integriert.

Beschreibung

Zum Importieren von Mediasets steht das Tool `sdsrimp` zur Verfügung. Im folgenden Beispiel werden die Mediasets `SUSE-SLES`, `SUSE-CORE` und `SWDIST-SLES9` für die Distribution `SLES9` und die Architektur `i386` in den Applikationsserver importiert. Für weitere Informationen zu `sdsrimp` siehe die Manual-Page `sdsrimp(8)`.

```
sdsrimp \  
  --architecture <arch> \  
  --distribution <dist> \  
  --product <prod> \  
  --version <version> \  
  --passwd <passwd> \  
  --user <user>
```

Beispiel:

```
# sdsrimp --user hugo --passwd biene \  
  --distribution SLES9 --architecture i386 \  
  --product SUSE-SLES --version 9  
# sdsrimp --user hugo --passwd biene \  
  --distribution SLES9 --architecture i386 \  
  --product SUSE-CORE --version 9  
# sdsrimp --user hugo --passwd biene \  
  --distribution SLES9 --architecture i386 \  
  --product SWDIST-SLES9 --version 1
```

Aktionen

- Mit dem `sdadm` Kommando `mediaset list` kann man überprüfen, ob die Mediasets `SUSE-SLES`, `SUSE-CORE` und `SWDIST-SLES9` angelegt wurden.

4.3 Media Set aus APP- und Install-Server löschen

Voraussetzungen

- Keine Installationen von den betroffenen Streams am laufen
- Media-Set auf Install Server repliziert
- Media-Set Definitionen im APP-Server

Ergebnis

- Media Set auf dem Install Server gelöscht
- Media Set im APP gelöscht

Tools

- `sdadm`
- `rm`
- `umount`

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



4.3.1 Löschen des Media Set im APP-Server

Beschreibung

Mit dem `sdadm` Kommando `mediaset delete` können Mediasets die in der Datenbank des Applikationsserver mit dem Tool `sdswimp` importiert wurden (siehe 4.2), wieder gelöscht werden. Im folgenden Beispiel wird das Mediaset `SUSE-CORE` für die Distribution `SLES9` und die Architektur `i386` gelöscht. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \
-u <user> \
-p <passwd> \
mediaset delete <mediaset> <version> <dist> <arch>
```

Beispiel:

```
# sdadm -u hugo -w biene
sdadm> mediaset delete "SUSE-CORE" 9 SLES9 i386
sdadm> mediaset list
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `mediaset list` kann man überprüfen, ob das Mediaset `SUSE-CORE` für die Distribution `SLES9` und die Architektur `i386` im Application-Server gelöscht wurde.

4.3.2 Löschen des Media Set auf dem Install Server

Beschreibung

In der SUSE-Testumgebung wurden die `SLES9` Installations-CDs als ISO-Images per 'loop-mount' in das Filesystem vom Installationsserver `howard` eingehängt. Im unten beschriebenen Beispiel wird mit dem Kommando 'umount' erst das Mediaset `SUSE-CORE` für die Distribution `SLES9` und die Architektur `i386` aus dem Filesystem entfernt und anschließend mit dem 'rm' Kommando gelöscht. Anschließend sind noch die entsprechenden Einträge aus der Datei '/etc/fstab' zu löschen.

```
umount <PackageRoot>/mediasets/<mediaset>
rm <PackageRoot>/iso/<image>
```

Beispiel:

```
# umount /space/mediasets/SUSE-CORE/9/i386/CD1
# umount /space/mediasets/SUSE-CORE/9/i386/CD2
# umount /space/mediasets/SUSE-CORE/9/i386/CD3
# umount /space/mediasets/SUSE-CORE/9/i386/CD4
# umount /space/mediasets/SUSE-CORE/9/i386/CD5

# rm /space/iso/SLES-9-i386-RC5-CD2.iso
# rm /space/iso/SLES-9-i386-RC5-CD3.iso
# rm /space/iso/SLES-9-i386-RC5-CD4.iso
# rm /space/iso/SLES-9-i386-RC5-CD5.iso
# rm /space/iso/SLES-9-i386-RC5-CD6.iso
```

Aktionen

- Mit dem Kommando `df` und `ls` kann anschließend überprüft werden, ob das Mediaset aus dem Filesystem des Installationsserver entfernt, bzw. ob die dazugehörigen ISO-Dateien gelöscht wurden.

4.4 RPM zu Media Set in APP- und Install-Server hinzufügen

Voraussetzungen

- Keine laufende Installation
- Media Set besteht nicht aus IOS-Images

Ergebnis

- RPM in Install Server vorhanden
- RPM in APP- Server vorhanden

Tools

- cp
- create_package_descr
- sdswhimp

Beschreibung

Achtung! Durch Aufbau der Paketbeschreibungen können laufenden Installationen oder Updates gestört werden. Paketbeschreibungen dürfen nur aufgebaut werden, wenn keine Installationen oder Updates laufen. Nach Änderung der Paketbeschreibung sind u.U die Caches auf den Target veraltet. Als Workaround kann das betroffene Produkt aus der Liste entfernt und erneut eingetragen werden.

4.4.1 Kopie des rpm-Paketes

cp

4.4.2 Neuaufbau der Paketbeschreibungen

```
create_package_descr \  
-d datadir1 ... -d datadirN \  
-l language1 ... -l languageN \  
-o outdir
```

4.4.3 Neuer Import des Media Set

siehe [4.2](#)

4.5 RPM aus Media Set in APP- und Install-Server löschen

Voraussetzungen

- Keine Installation läuft
- Media Set besteht nicht aus ISO-Images

Ergebnis

- RPM ist nicht mehr in Media Set in APP- und Installserver enthalten

Tools

- rm
- create_package_descr
- sdswhimp

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



4.5.1 Paket aus dem Installserver löschen

`rm`

4.5.2 Neuaufbau der Paketbeschreibungen des Installservers

`create_package_descr`

4.5.3 Neuer Import des Media Set in den APP-Server

siehe [4.2](#)

4.6 Konsistenz für Mediasets prüfen

Voraussetzungen

- Mediasets in APP Server vorhanden
- Mediasets auf Install Server vorhanden

Ergebnis

- Konsistenz von APP Server und Install Server bezüglich der Mediasets ist festgestellt.

Tools

- `sdverify`

Beschreibung

Mit dem Admin-Tool `sdverify` soll überprüft werden, ob die Mediasets die in der Datenbank im Applicationserver angelegt wurden, auch auf dem Installationsserver vorhanden sind. Im folgenden Beispiel gehen wir davon aus, dass drei Mediasets (siehe Kapitel [4.2](#) auf Seite [18](#)) angelegt wurden. Für weitere Informationen zu `sdverify` siehe die Manual-Page `sdverify(8)`.

```
sdverify \  
  --user <user> \  
  --password <pwd> \  
  --mediasets
```

Beispiel:

```
howard:~ # sdverify -u hugo -w biene --mediasets  
Verifying 3 mediasets, please wait.  
Verifying mediaset SUSE SLES 9 for Distribution SLES9 (i386)  
Checking for 6 packageInstances...  
mediaset ok.
```

Aktionen

- ggf. Inkonsistenz beseitigen

4.7 Stream in APP-Server anlegen

Voraussetzungen

- Alle Media-Sets auf dem Installationsserver
- Alle Definitionen der Media-Sets im Application-Server

Ergebnis

- Stream in APP-Server angelegt

Tools

- sdadm

Beschreibung

Ein Stream beschreibt die Gesamtheit aller verfügbaren Softwarepakete für eine SUSE-Distribution die auf einem Target installiert werden können. Im folgenden Beispiel wird der Stream 'SLES9-i386-4711' für SLES9 und die Architektur i386 mit dem Admin-Tool sdadm angelegt. Für weitere Informationen zu sdadm siehe die Manual-Page sdadm(8).

```
sdadm \  
    -u <user> \  
    -w <passwd> \  
    stream add <> ...
```

Beispiel:

```
# sdadm -u hugo -w biene  
sdadm> stream add "SLES9-i386-4711" SLES9 i386 \  
    name="This is a sample stream." \  
    grace=30 \  
    order="SUSE SLES,9,SUSE CORE,9,SWDIST-SLES9,1" \  
    instorder="SUSE SLES,9,SUSE CORE,9,SWDIST-SLES9,1" \  
    links="content-> \  
        /space/mediaset/SUSE-SLES/9/i386/CD1/content,media.1-> \  
        /space/mediaset/SUSE-SLES/9/i386/CD1/media.1,boot/rescue-> \  
        /space/mediaset/SUSE-SLES/9/i386/CD1/boot/rescue,boot/root-> \  
        /space/mediaset/SUSE-SLES/9/i386/CD1/boot/root,boot/loader-> \  
        /space/mediaset/SUSE-SLES/9/i386/CD1/boot/loader,mediaset-> \  
        ../../mediaset" \  
    patches="emergency"  
sdadm> stream list  
sdadm> quit
```

Aktionen

- Mit dem sdadm Kommando `stream list` kann man überprüfen, ob der Stream z.B. 'SLES9-i386-4711' angelegt wurde.

4.8 Stream aus APP- in Install-Server exportieren

Voraussetzungen

- Stream in App-Server angelegt

Ergebnis

- Stream auf Installations-Server angelegt
- Targets können vom Stream installieren

Tools

- sdstexp

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Beschreibung

Damit ein Target via HTTP von einem Stream Software installieren kann, muss jeder Stream der in der Datenbank angelegt wurde auf dem Installationsserver exportiert werden. Im folgenden Beispiel wird der Stream 'SLES9-i386-4711' für SLES9 und die Architektur i386 mit dem Admin-Tool `sdstexp` exportiert. Für weitere Informationen zu `sdstexp` siehe die Manual-Page `sdstexp(8)`.

```
sdstexp \
  --user <user> \
  --passwd <passwd> \
  --architecture <arch> \
  --distribution <dist> \
  --streamname <stream_shortcut>
```

Beispiel:

```
# sdstexp \
  --user hugo \
  --passwd biene \
  --streamname SLES9-i386-4711 \
  --distribution SLES9 \
  --architecture i386
```

Die HTTP-Struktur wurde auf dem Installationsserver im Verzeichnis `/srv/www/htdocs/streams/` angelegt.

Aktionen

- Zur Kontrolle, ob die HTTP-Struktur auf dem Installationsserver für den Target angelegt wurde, kann die Verzeichnisstruktur z.B. mit dem `'ls'` Kommando überprüft werden. Die Ausgabe des `ls -l` Kommandos ist unten verkürzt wiedergegeben.

```
howard:/srv/www/htdocs/streams/SLES9-i386-4711 # ls -l

boot
content -> /space/mediaset/SUSE-SLES/9/i386/CD1/content
media.1 -> /space/mediaset/SUSE-SLES/9/i386/CD1/media.1
mediaset -> ../../mediaset
yast
```

4.9 Stream aus APP- und Install-Server löschen

Voraussetzungen

- Kein Target nutzt den Stream
- Stream in APP Server vorhanden
- Stream auf Install Server vorhanden

Ergebnis

- Stream ist aus APP- und Install-Server gelöscht

Tools

- `sdadm`
- `rm`

4.9.1 Stream aus Install-Server löschen

Beschreibung

Ein Stream beschreibt die Gesamtheit aller verfügbaren Softwarepakete für eine SUSE-Distribution die auf einem Target installiert werden können. Im folgenden Beispiel wird die HTTP-Struktur des Streams 'SLES9-i386-4711' für SLES9 und die Architektur i386 auf dem Installationsserver howard mit dem Kommando 'rm' gelöscht.

```
rm -rf <path>/<stream>
```

Beispiel:

```
howard:/srv/www/htdocs/streams # rm -rf SLES9-i386-4711
```

Aktionen

- Zur Kontrolle, ob die HTTP-Struktur auf dem Installationsserver für den Stream 'SLES9-i386-4711' gelöscht wurde, kann z.B. das 'ls' Kommando benutzt werden.

4.9.2 Stream aus APP-Server löschen

Beschreibung

Ein Stream der zuvor mit dem Admin-Tool `sdadm` angelegt wurde, soll wieder gelöscht werden. Im folgenden Beispiel wird der Stream 'SLES9-i386-4711' für SLES9 und die Architektur i386 mit dem Admin-Tool `sdadm` auf dem Application-Server gelöscht. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \  
-u <user> \  
-w <passwd> \  
stream delete <stream>
```

Beispiel:

```
# sdadm -u hugo -w biene  
sdadm> stream delete "SLES9-i386-4711"  
sdadm> stream list  
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `stream list` kann man überprüfen, ob der Stream z.B. 'SLES9-i386-4711' gelöscht wurde.

4.10 Grace Period für Stream ändern

Voraussetzungen

- Stream in APP Server vorhanden

Ergebnis

- Der Stream hat eine neue Grace Period

Tools

- `sdadm`

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Beschreibung

Die Parameter `grace` legt die Zeit in Tagen fest, wann Softwareänderungen 'mandatory' werden. Im folgenden Beispiel wird der Wert 'Grace Period' für den Stream 'SLES9-i386-4711' geändert. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \  
  -u <user> \  
  -w <password> \  
  stream update <stream> grace=<grace_period>
```

Beispiel:

```
# sdadm -u hugo -w biene  
sdadm> stream update SLES9-i386-4711 \  
      grace 10  
sdadm> stream list  
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `stream list` kann man überprüfen, ob der Wert für 'Grace Period' geändert wurde.
- ggf. testen (QA)

4.11 Konsistenz für Streams prüfen

Voraussetzungen

- Streams in APP Server vorhanden
- Streams auf Install Server vorhanden

Ergebnis

- Konsistenz von APP Server und Install Server bezüglich der Streams ist festgestellt.

Tools

- `sdverify`

Beschreibung

Mit dem Admin-Tool `sdverify` soll überprüft werden, ob die Streams die in der Datenbank im Applicationserver angelegt wurden, auch auf dem Installationsserver exportiert wurden. Im folgenden Beispiel gehen wir davon aus, dass der Stream 'SLES9-i386-4711' angelegt (siehe Kapitel 4.7) und exportiert (siehe Kapitel 4.8) wurde. Für weitere Informationen zu `sdverify` siehe die Manual-Page `sdverify(8)`.

```
sdverify \  
  --user <user> \  
  --password <pwd> \  
  --streams
```

Beispiel:

```
howard:~ # sdverify -u hugo -w biene --streams  
Verifying 1 streams, please wait.  
Verifying stream SLES9-i386-4711.  
stream ok.  
Verification succeeded. Applicationserver and installserver are in sync.
```

Aktionen

- ggf. Inkonsistenz beseitigen

4.12 Mediaset zu Stream hinzufügen

Voraussetzungen

- Mediaset ist auf Install Server importiert
- Mediaset ist in APP Server importiert

Ergebnis

- Mediaset steht in Stream in richtiger Rangfolge zur Verfügung

Tools

- `make_inst_server`
- `sdsrimp`

4.12.1 Mediaset auf Install Server importieren

```
make_inst_server \
```

4.12.2 Mediaset in APP Server importieren

```
sdsrimp \  
--architecture <arch> \  
--distribution <distname> \  
--product <productname> \  
--version <productversion> \  
--passwd <passwd> \  
--user <user>
```

4.13 Channel in APP-Server anlegen

Voraussetzungen

- Distribution, Selection Lists und Pakete im Application-Server vorhanden

Ergebnis

- Channel im APP-Server angelegt
- Selectionlist zu Channel im APP-Server hinzugefügt¹

Tools

- `sdadm`

¹ Wird das `sdadm` Kommando `channel add` ohne den Parameter `selectionlist` aufgerufen, wird nur der Channel im APP-Server angelegt.

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Beschreibung

Ein Channel umfasst eine Ansammlung von Softwareauswahllisten die eine bestimmte Funktionalität z.B. Datenbankserver beschreibt. Im folgenden Beispiel wird der Channel 'base2-4711' mit dem Admin-Tool `sdadm` angelegt. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \  
-u <user> \  
-w <passwd> \  
channel add <channel> <dist> <arch> <selectionlist>,...
```

Beispiel:

```
# sdadm -u hugo -w biene  
sdadm> channel add "base2-4711" SLES9 i386 Base-System,YaST2,Minimal, \  
X11,Kde-Desktop,swdist  
sdadm> channel list  
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `channel list` kann man überprüfen, ob der Channel 'base2-4711' angelegt wurde.
- ggf. testen (QA)

4.14 Selectionlist zu Channel in APP-Server hinzufügen

Voraussetzungen

- Selectionlist, Package usw. im Application-Server vorhanden
- Channel im APP-Server angelegt

Ergebnis

- Selectionlist zu Channel im APP-Server hinzugefügt

Tools

- `sdadm`

Beschreibung

Ein Channel umfasst eine Ansammlung von Softwareauswahllisten die eine bestimmte Funktionalität z.B. Datenbankserver beschreiben. Im folgenden Beispiel werden dem Channel 'base2-4711' mit dem Admin-Tool `sdadm` die Softwareauswahllisten `Base-System`, `YaST2`, `Minimal` und `swdist` hinzugefügt. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \  
-u <user> \  
-w <passwd> \  
channel update <channel> <dist> <arch> <selectionlist>,...
```

Beispiel:

```
# sdadm -u hugo -w biene  
sdadm> channel list  
sdadm> channel update "base2-4711" SLES9 i386 Base-System,YaST2, \  
Minimal,swdist  
sdadm> channel list  
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `channel list` kann man vor und nach dem Aufruf von `channel update` überprüfen welche Softwareauswahllisten dem Channel 'base2-4711' hinzugefügt wurden.

4.15 Selectionlist aus Channel in APP-Server löschen

Voraussetzungen

- Selectionlist (siehe 4.14), Channel, Package usw. im Application-Server vorhanden

Ergebnis

- Alte Selectionlist aus Channel im APP-Server entfernt
- Neue Selectionlist zu Channel im APP-Server hinzugefügt

Tools

- `sdadm`

Beschreibung

Ein Channel umfasst eine Ansammlung von Softwareauswahllisten die eine bestimmte Funktionalität beschreiben. Mit jedem Aufruf von `channel update` und einer `<selectionlist>` wird die zuvor aktive Softwareauswahlliste (siehe 4.14) eines Channels überschrieben. Im folgenden Beispiel werden dem Channel 'base2-4711' mit dem Admin-Tool `sdadm` die Softwareauswahllisten `Base-System`, `YaST2`, `Minimal`, `X11`, `Basis-Sound`, `SuSE-Dokumentation`, `Kde-Desktop` und `swdist` hinzugefügt. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \  
  -u <user> \  
  -w <passwd> \  
  channel update <channel> <dist> <arch> <selectionlist>,...
```

Beispiel:

```
# sdadm -u hugo -w biene  
sdadm> channel list  
sdadm> channel update "base2-4711" SLES9 i386 Base-System,YaST2,Minimal, \  
  X11,Basis-Sound,SuSE-Dokumentation,Kde-Desktop,swdist  
sdadm> channel list  
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `channel list` kann man vor und nach dem Aufruf von `channel update` überprüfen welche Softwareauswahllisten dem Channel 'base2-4711' gelöscht bzw. hinzugefügt wurden.

4.16 Selectionlist in APP-Server importieren

Voraussetzungen

- Installations Server installiert und konfiguriert
- Applikation Server installiert und konfiguriert
- Selectionlist als File vorhanden
- Mediaset in App-Server importiert (siehe 4.12)

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Ergebnis

- Selectionlist im APP-Server vorhanden

Tools

- sdsлимп

Beschreibung

Die Softwareauswahllisten legen die Softwarepakete fest, die eine bestimmte Funktionalität zur Verfügung stellen. Die Datei `Kde-Desktop.sel` legt z.B. die benötigten Pakete fest, die für ein SUSE Linux System mit grafischen Desktop (KDE) installiert werden müssen.

Softwareauswahllisten werden mit dem Tool `sdsлимп` importiert. Für weitere Informationen zu `sdsлимп` siehe die Manual-Page `sdsлимп(8)`.

```
sdsлимп \
  --architecture <arch> \
  --distribution <dist> \
  --selection_list_file <path>/<file> \
  --passwd <passwd> \
  --user <user>
```

4.17 Selectionlist aus APP-Server löschen

Voraussetzungen

- Selectionlist in APP-Server vorhanden

Ergebnis

- Selectionlist aus APP-Server gelöscht

Tools

- sdadm

Beschreibung

Die Softwareauswahllisten legen die Softwarepakete fest, die eine bestimmte Funktionalität zur Verfügung stellen. Im folgenden Testfall soll die Softwareauswahlliste `Kde-Desktop.sel` gelöscht werden. Mit dem `sdadm` Kommando `selectionlist list` überprüft man vorher, welche Selectionlisten im APP-Server vorhanden sind.

```
sdadm \
  -u <user> \
  -w <passwd> \
  selectionlist delete <selectionlist> <dist> <arch>
```

Beispiel:

```
# sdadm -u hugo -w biene
sdadm> selectionlist list
sdadm> selectionlist delete Kde-Desktop SLES9 i386
sdadm> selectionlist list
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `selectionlist list` kann man überprüfen, ob die Softwareauswahlliste `Kde-Desktop.sel` gelöscht wurde.

4.18 Custom Patches in APP- und Inst-Server importieren

Voraussetzungen

- Eine oder mehrere RPMs liegen vor
- Maximal ein Script für den Patch liegt vor

Ergebnis

- Patch wird korrekt auf dem Installations-Server eingehängt
- Patch-Definition wird im APP-Server eingetragen

Tools

- `sdptimp`
- `md5sum`

4.18.1 Berechnen der MD5Sum

`md5sum`

4.18.2 Anmelden des Patch am APP-Server

Beschreibung

SUSE und Custom Patches werden mit dem Tool `sdptimp` importiert. Für weitere Informationen zu `sdptimp` siehe die Manual-Page `sdptimp(8)`.

```
sdptimp \
    --architecture <arch> \
    --distribution <dist> \
    --patch <patch> \
    --patchBaseDir <path> \
    --passwd <passwd> \
    --user <user>
```

4.18.3 Kopie des Patches auf dem Installationsserver

Voraussetzungen

- Der Schritt "Anmelden des Patch am APP-Server" muss korrekt durchlaufen sein.

Beschreibung

Kopie des Patches in den Patch/Dist Pfad

4.19 Patch aus Stream löschen

Voraussetzungen

- Stream und Patch vorhanden (siehe [4.21](#))

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Ergebnis

- Der Patch ist für Stream in APP Server nicht mehr vorhanden.
- Der Patch ist auf Target des Streams nicht mehr vorhanden.

Tools

- sdadm

Beschreibung

Mit dem `sdadm` Kommando `stream update` können Patches die in der Datenbank des Applikationsserver mit dem Tool `sdptimp` importiert wurden (siehe 4.20), einem Stream hinzugefügt werden. Mit jedem Aufruf von `stream update` mit dem Parameter `patches` wird die Patchliste überschrieben. Im folgenden Beispiel soll der in 4.21 eingefügte Patch `patch-9524` ("Recommended update for yast2-update") für die Distribution SLES9 und die Architektur i386 mit der Priorität `mandatory` aus dem Stream `'SLES9-i386-4711` wieder gelöscht werden.

Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \
  -u <user> \
  -p <passwd> \
  stream update <stream> \
  patches=<patch>,<priority>, \
  <patch>,<priority>, ...
```

Beispiel:

```
# sdadm -u hugo -w biene
sdadm> stream list
sdadm> stream update "SLES9-i386-4711" \
  patches="patch-9488,emergency"
sdadm> stream list
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `stream list` überprüft man vor und nach dem Aufruf von `stream update` welche Patches dem Stream gelöscht bzw. hinzugefügt wurden.
- ggf. testen (QA)

4.20 SUSE Patches in APP-Server importieren

Voraussetzungen

- Patch in Installserver integriert
- RPMs liegen auf dem Installserver vor
- Distribution in App-Server angelegt (siehe 4.1)

Ergebnis

- Patch-Definition wird im APP-Server eingetragen

Tools

- `sdptimp`

Beschreibung

SUSE Patches werden mit dem Tool `sdptimp` importiert. Für weitere Informationen zu `sdptimp` siehe die Manual-Page `sdptimp(8)`.

```
sdptimp \  
    --architecture <arch> \  
    --distribution <dist> \  
    --patch <patch> \  
    --patchBaseDir <path> \  
    --passwd <passwd> \  
    --user <user>
```

Beispiel:

```
# sdptimp --architecture i386 --distribution SLES9 \  
    --patch patch-9525 \  
    --patchBaseDir /space/mirror/i386/update/SUSE-CORE/9 \  
    --user hugo --passwd biene
```

```
Importing patch patch-9525  
createDistributionDetails NULL i386 SLES9  
createDistributionDetails NULL i386 SLES9  
priority is: recommended  
short description is: Recommended update for dvd+rw-tools  
added 1 package instances  
added 13 dependencies  
send soap data - please be patient.  
soap call successfull  
-----
```

4.21 Patch zu Stream hinzufügen

Voraussetzungen

- Stream und Patch vorhanden (siehe [4.7](#), [4.20](#))

Ergebnis

- Der Patch ist für Stream in APP Server vorhanden.
- Der Patch ist auf Targets des Streams vorhanden.
- Der neue Stand der Targets ist in APP-Server vorhanden.

Tools

- `sdadm`

Beschreibung

Mit dem `sdadm` Kommando `stream update` können Patches die in der Datenbank des Applikationsserver mit dem Tool `sdptimp` importiert wurden (siehe [4.20](#)), einem Stream hinzugefügt werden. Im folgenden Beispiel wird der Patch `patch-9488` ("Security update for Linux kernel") und der Patch `patch-9524` ("Recommended update for yast2-update") für die Distribution SLES9 und die Architektur i386 mit der Priorität `emergency` bzw. `mandatory` dem Stream `'SLES9-i386-4711'` hinzugefügt.

Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \  
    -u <user> \  
    -p <passwd> \  
    stream update <stream> \  
    patches=<patch>,<priority>, \  
    <patch>,<priority>, ...
```

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Beispiel:

```
# sdadm -u hugo -w biene
sdadm> stream list
sdadm> stream update "SLES9-i386-4711" \
    patches="patch-9488,emergency,patch-9524,mandantory"
sdadm> stream list
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `stream list` kann man überprüfen, ob die Patches z.B. `patch-9488` und `patch-9524` für die Distribution SLES9 und die Architektur i386 dem Stream 'SLES9-i386-4711' hinzugefügt wurden.
- ggf. testen (QA)

4.22 Patch aus APP-Server löschen

Voraussetzungen

- Patch ist aus allen Streams entfernt (siehe 4.19)

Ergebnis

- Patch ist aus APP Server gelöscht
- Patch verbleibt auf Installation-Server

Tools

- `sdadm`

Beschreibung

Mit dem `sdadm` Kommando `patch delete` können Patches die in der Datenbank des Applikationsserver mit dem Tool `sdptimp` importiert wurden (siehe 4.20), wieder gelöscht werden. Im folgenden Beispiel wird der Patch `patch-9525` für die Distribution SLES9 und die Architektur i386 ("Recommended update for dvd+rw-tools") gelöscht. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \
    -u <user> \
    -p <passwd> \
    patch delete <patch> <dist> <arch>
```

Beispiel:

```
# sdadm -u hugo -w biene
sdadm> stream list
sdadm> patch delete "patch-9525" SLES9 i386
sdadm> patch list
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `patch list` kann man überprüfen, ob der Patch z.B. `patch-9525` für die Distribution SLES9 und die Architektur i386 im Application-Server gelöscht wurde.
- ggf. Custom Patch aus Installation-Server löschen.

4.23 Konsistenz für Patches prüfen

Voraussetzungen

- Patches in APP Server vorhanden
- Patches auf Install Server vorhanden

Ergebnis

- Konsistenz von APP Server und Install Server bezüglich der Patches ist festgestellt.

Tools

- sdverify

Beschreibung

Mit dem Admin-Tool `sdverify` soll überprüft werden, ob die Patches die in der Datenbank im Applicationserver angelegt wurden, auch auf dem Installationsserver vorhanden sind. Im folgenden Beispiel gehen wir davon aus, dass die SUSE Patches (siehe Kapitel [4.20](#) auf Seite [33](#)) importiert wurden. Für weitere Informationen zu `sdverify` siehe die Manual-Page `sdverify(8)`.

```
sdverify \  
    --user <user> \  
    --password <pwd> \  
    --patch
```

Beispiel:

```
howard:~ # sdverify -u hugo -w biene --patch  
Verifying 110 patches, please wait.  
Verifying patch patch-9170 for Distribution SLES9 (i386)  
Checking for 1 packageInstances...  
patch ok.  
Verifying patch patch-9171 for Distribution SLES9 (i386)  
Checking for 2 packageInstances...  
patch ok.  
Verifying patch patch-9172 for Distribution SLES9 (i386)  
Checking for 1 packageInstances...  
patch ok.  
Verifying patch patch-9177 for Distribution SLES9 (i386)  
Checking for 1 packageInstances...  
patch ok.  
Verifying patch patch-9180 for Distribution SLES9 (i386)  
Checking for 1 packageInstances...  
patch ok.  
Verifying patch patch-9183 for Distribution SLES9 (i386)  
Checking for 1 packageInstances...  
patch ok.  
Verifying patch patch-9184 for Distribution SLES9 (i386)  
Checking for 7 packageInstances...  
patch ok.  
Verifying patch patch-9193 for Distribution SLES9 (i386)  
:  
:  
:
```

Aktionen

- ggf. Inkonsistenz beseitigen

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



4.24 Nützliche Befehle

Source- und Installations-Pakete auf Installserver finden

```
devsupp@tmx065:~> find /install/mediaset/suse-test/ -name "<xxx>*" 
```

```
/install/mediaset-suse-test/SuSE-SLES/8/custom/CD1/custom/src/<xxx>.src.rpm  
/install/mediaset-suse-test/SuSE-SLES/8/custom/CD1/custom/x86_64/<xxx>.x86_64.rpm
```

Anm.: Das Programm `find` durchsucht bei jedem Aufruf das gesamte Dateisystem unterhalb des gegebenen Verzeichnisses. Alternativ könnte das Programm `locate` zusammen mit `updatedb` genutzt werden. Dabei wird von `updatedb` regelmäßig, z.B. nächtlich, mittels `find` ein Index erzeugt, der von `locate` genutzt wird.

5 Target Management on Server

5.1 Allgemeines

Tabelle 2.4 gibt einen vereinfachten Überblick über die zur Abbildung der Targets verwendeten Objekte.

Die Targets werden im Application Server über die Objekte `machine` und `config` dargestellt. Die Eigenschaften des Objektes `machine` werden direkt über den Application Server mittels `sdadm` manipuliert (s. Manual Page `sdadm.8`). Die Eigenschaften des Objektes `config` werden außerhalb des Application Servers in einer `Alice Working Copy` (`$ALICE_HOME`) bearbeitet, zur Steuerung der initialen Installation wird zudem ein `AutoYaST2 Control File` im `TFTP`-Bereich abgelegt. Aktuelle Informationen zu `Alice`, insbesondere eine vollständige Referenz der Tags und Sections finden sich unter `www.suse.com/fabian`. Aktuelle Informationen zu `AutoYaST2` insbesondere eine vollständige Referenz der Tags finden sich unter `www.suse.com/nashif`.

Alle Änderungen werden zunächst im RDBMS ausgeführt und dann an die Targets übermittelt. Somit gehen sie nicht verloren, wenn Targets offline sind. Emergency-Updates werden auf den Targets sofort (d.h. innerhalb des Update Intervalls) ausgeführt, andere Änderungen durch Benutzer-Aktion auf dem Target oder nach Ablauf der Grace Period.

Wechsel zwischen Streams

Ein Wechsel zwischen Streams ist möglich. Wird zurück gewechselt, sind Patches, die nicht in beiden Streams vorhanden waren, verschwunden.

Files im TFTP-Bereich

Files im `TFTP`-Bereich (`GRUB`-Menu, `Init-RAM-Disk`, `Kernel-Image`) können gemäß der spezifischen Anforderungen durch kundenspezifische Skripten verwaltet werden (s. Abbildung 2.4).

Patches entfernen

Ein Patch wird von Targets entfernt, indem er aus dem Stream entfernt wird. Sind bei der Installation des Patches Skripte ausgeführt worden, kann es erforderlich, diese durch einen weiteren Patch rückgängig zu machen.

Indirekte Selection List

Mit `SLES 9` wurden indirekte Base Selection Lists eingeführt. So enthält die Base Selection List `Minimal.sel` keine Pakete, sondern verweist auf die Selection Lists `YaST2.sel` und `Base-System.sel`.

5.2 Target in APP-Server anlegen

Voraussetzungen

- Classes in APP Server vorhanden
- Distribution, Channel und Stream in APP Server vorhanden
- Alice-Konfiguration für Target vorhanden

Ergebnis

- Target ist in APP Server vorhanden

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Tools

- sdadm

Beschreibung

Im folgenden Beispiel wird ein zu installierendes Target `wing.lab.ffm.novell.com` mit dem Admin-Tool `sdadm` in der Datenbank angelegt und einer Architektur, einer SUSE-Distribution, einem Stream und einem Channel zugeordnet. Letztlich bestimmt der Channel `'base2-4711'` welche Software-Pakete auf dem Target installiert werden und welche Funktionalität er haben wird. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
sdadm \
    -u <user> \
    -w <password> \
    machine add <target> \
    arch=<arch> \
    dist=<dist>,<arch> \
    channel=<channel>,... \
    config=<confl>,... \
    grace=<grace_period>
```

Beispiel:

```
# sdadm -u hugo -w biene
sdadm> machine add wing.lab.ffm.novell.com \
    arch i386 \
    dist SLES9,i386 \
    stream SLES9-i386-4711 \
    grace 30 \
    channel base2-4711
sdadm> machine list
sdadm> quit
```

Aktionen

- Mit dem `sdadm` Kommando `machine list` kann man überprüfen, ob der Client `'wing.lab.ffm.novell.com'` angelegt wurde.

5.3 Konfiguration für Target importieren

Voraussetzungen

- Target in App-Server angelegt (siehe [5.2](#))
- Alice Konfiguration liegen vor
(Dateien vom Typ "`<path>/info/<hostname>.*.tcf`")

Ergebnis

- Konfiguration für Target in APP Server vorhanden

Tools

- sdacimp

Beschreibung

Alice-Konfigurationen werden mit dem Tool `sdacimp` importiert. Mit der Option `system <fqhn>`¹ werden alle systemspezifischen Alice-Konfigurationen importiert. Im folgenden Beispiel wird die Konfiguration für den Target `wing.lab.ffm.novell.com` importiert. Für weitere Informationen zu `sdacimp` siehe die Manual-Page `sdacimp(8)`.

```
sdacimp \  
  --user <user> \  
  --passwd <passwd> \  
  --alice_home <alicehome> \  
  --system <fqhn>
```

Beispiel:

```
# sdacimp \  
  --user hugo \  
  --passwd biene \  
  --alice_home wing.lab.ffm.novell.com-alice \  
  --system wing.lab.ffm.novell.com
```

5.4 Target RPMs zeigen

Voraussetzungen

- Target in APP-Server vorhanden (siehe [5.2](#))

Ergebnis

- RPMs für Target bekannt

Tools

- `sdadm`

```
sdadm \  
-u <user> \  
-w <passwd> \  
machine pkglist <target>
```

5.5 Target Status zeigen

Voraussetzungen

- Target in APP Server vorhanden (siehe [5.2](#))
-

Ergebnis

- Target Status ist bekannt

¹ fqhn = Full Qualified Host Name

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Tools

- sdadm

```
sdadm \  
-u <user> \  
-w <passwd> \  
machine list <target>
```

```
sdadm \  
-u <user> \  
-w <passwd> \  
event list machine=<target>
```

Aktionen

- ...

5.6 Stream für Target ändern

Voraussetzungen

- Stream in APP-Server vorhanden (siehe [4.7](#))
- Architektur und Distribution bleiben unverändert

Ergebnis

- Das Target gehört zum neuen Stream

Tools

- sdadm
- sdacexp
- alice2autoyast

```
sdadm \  
-u <user> \  
-w <passwd> \  
machine \  
update <target> stream=<stream>
```

Aktionen

- ...
- ggf. testen (QA)

5.7 Klasse für Target ändern

Voraussetzungen

- Klasse in APP Server vorhanden
- Target in APP Server vorhanden

Ergebnis

- Das Target gehört zur neuen Klasse

Tools

- sdacexp
- sdacimp
- sdadm
- vi

5.7.1 Klasse in APP-Server ändern

```
sdadm \  
-u <user> \  
-w <passwd> \  
machine update <target> \  
config=<classes>
```

Im folgenden Beispiel werden dem Target `wing.lab.ffm.novell.com` mit dem Admin-Tool `sdadm` die Alice-Klassen `ide`, `minimal`, `small` und `novellffm` zugeordnet. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdadm(8)`.

```
# sdadm -u hugo -w biene  
sdadm> machine update wing.lab.ffm.novell.com \  
                config="ide,minimal,small,novell"  
sdadm> machine list  
sdadm> quit
```

Mit dem `sdadm` Kommando `machine list` kann man überprüfen, ob die Alice-Klassen dem Client `wing.lab.ffm.novell.com` zugeordnet wurden.

5.7.2 Klasse in Alice2-Konfiguration ändern

```
sdacexp \  
--user <user> \  
--passwd <passwd> \  
--alice_home <path> \  
--system <target> \  
--machine_export \  
--remove  
  
vi ../info/<target>.sys.tcf  
  
sdacimp \  
--user <user> \  
--passwd <passwd> \  
--alice_home <path> \  
--system <target>
```

Aktionen

- ggf. testen (QA)

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



5.8 Architektur für Target in APP Server ändern

Voraussetzungen

- Architektur in APP Server vorhanden
- Target in APP server vorhanden

Ergebnis

- Architektur für Target in APP Server geändert

Tools

- sdadm

```
sdadm \  
    -u <user> \  
    -w <password> \  
machine update <target> arch=<arch>
```

5.9 Alice2-Konfiguration in APP-Server importieren

Voraussetzungen

- Alice Konfiguration liegt vor

Ergebnis

- ConfigGroupService: Alle Tags in den APP Server übertragen
- MachineService: Alle Alice-Klassen in den APP Server übertragen (Maschinen - Klassen Zuordnung)

Tools

- sdacimp

Beschreibung

Alice-Konfigurationen werden mit dem Tool `sdacimp` importiert. Abhängig von der gewählten Option `system`, `class` oder `default` werden alle Alice-Konfigurationen vom Typ "`<path>/info/<hostname>.*.tcf`", "`<path>/classes/<classname>.*.tcf`" oder "`<path>/templates/*.tcf`" importiert. Für weitere Informationen zu `sdacimp` siehe die Manual-Page `sdacimp(8)`.

```
sdacimp \  
    --user <user> \  
    --passwd <passwd> \  
    --alice_home <alicehome> \  
    --system <hostname> | --class <classname> | --default
```

5.10 Alice2-Konfiguration aus APP-Server exportieren

Voraussetzungen

- Alice Konfiguration in APP-Server vorhanden

Ergebnis

- Alle Tags aus APP Server übertragen

Tools

- sdacexp

Beschreibung

Alice-Konfigurationen werden mit dem Tool `sdacexp` aus dem Applikationsserver exportiert und auf dem Installationsserver abgelegt. Im folgenden Beispiel wird die Alice-Konfiguration die für den zu installierendes Target `wing.lab.ffm.novell.com` exportiert und auf dem Installationsserver abgelegt. Für weitere Informationen zu `sdadm` siehe die Manual-Page `sdacexp(8)`.

```
sdacexp \  
  --alice_home <path> \  
  --user <user> \  
  --passwd <pwd> \  
  --system <hostname>
```

Beispiel:

```
# sdacexp \  
  --user hugo \  
  --passwd biene \  
  --alice_home=/root/tests/wing.lab.ffm.novell.com-alice-exported \  
  --system wing.lab.ffm.novell.com \  
  --url http://149.44.71.13:8080/jboss-net/services/MachineService
```

Die exportierten Alice2-Dateien sind auf dem Installationsserver im Verzeichnis `/root/tests/wing.lab.ffm.novell.com-alice-exported` zu finden.

5.11 AutoYaST Control File generieren

Voraussetzungen

- Die Alice-Konfiguration für das zu installierende Target wurde mit `sdacexp` aus dem Applikationsserver exportiert und auf dem Installations-server abgelegt (siehe 5.10).
- Die MAC-Adresse des Targets ist bekannt.
- Das Verzeichnis `/srv/tftp/i386/yast.cfg/` ist auf dem Installationsserver angelegt.

Ergebnis

- Control File steht auf Boot/Install-Server zur Verfügung

Tools

- `alice2autoyast.pl`

Beschreibung

Alice-Konfigurationsdateien können mit dem Alice-Tool `alice2autoyast.pl` in "AutoYast2 Control Files" exportiert werden. Im folgenden Beispiel werden die Alice-Konfigurationsdateien (siehe 5.10) des Targets `wing.lab.ffm.novell.com` auf dem Installationsserver in AutoYast2 Control Files exportiert. Für weitere Informationen zu `alice2autoyast.pl` siehe die Manual-Page `alice2autoyast.pl(8)`.

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



```
/usr/lib/alice2/utils/alice2autoyast.pl \  
  --host <hostname> \  
  --alice_home <alicehome> \  
  --xml_file <tftp root>/<arch>/yast.cfg/<MAC Adresse>
```

Beispiel:

```
# /usr/lib/alice2/utils/alice2autoyast.pl \  
  --host wing.lab.ffm.novell.com \  
  --alice_home /root/tests/wing.lab.ffm.novell.com-alice-exported/ \  
  --xml_file /srv/tftp/i386/yast.cfg/00d0b7852462
```

5.12 Nützliche Befehle

In ALICE_HOME targetspezifischen Dateien finden

```
tmx065:~/alice # find . -name "x23204.*"
```

In ALICE_HOME alle Parameter einer Section für ein Target zeigen

```
tmx065:~/alice # /usr/lib/alice/utils/show_value \  
misc x23204.ffm.ble.de
```

In ALICE_HOME einen Parameter für ein Target zeigen

```
tmx065:~/alice # /usr/lib/alice/utils/show_value \  
misc MISC_CFG_TABLE x23204.ffm.ble.de
```

In ALICE_HOME Konfiguration eines Targets klonen

```
tmx065:~/alice # /usr/lib/alice/utils/copy_machine \  
wing.lab.ffm.novell.com amd.lab.ffm.novell.com
```

Danach sind noch targetspezifische Parameter anzupassen. Falls diese Arbeit umfangreicher ist, kann das Skript `change_netcfg` benutzt werden.

6 Target Installation & Update

6.1 Allgemeines

Auf den Targets sind zur Verteilung von Software und Patches, abgesehen vom Auslösen der initialen Installation, keine administrativen Eingriffe erforderlich. Alle Änderungen werden auf dem Application-Server eingetragen und dann automatisch auf dem Target ausgeführt. Die nachfolgend aufgeführten Use Cases 6.3 und 6.4 dienen hauptsächlich für Fehleranalyse und Testzwecke.

Die autom. Ausführung normaler Updates wird durch zwei Zeitintervalle gesteuert.

Grace Period

Die Grace Period ist die Zeitdauer, die maximal zwischen zwei Updates vergeht. Alle bis dahin aufgelaufenen Aktionen werden dann ausgeführt. D.h. eine Aktion, die kurz nach dem letzten Update eines Targets eingetragen wird, kann bis zum Ablauf der Grace Period unwirksam bleiben. Andererseits wird eine Aktion, die kurz vor Ablauf der Grace Period eingetragen wurde, kurzfristig wirksam.

Die Grace Period wird im Application Server pro Stream festgelegt, als Einheit dienen Tage. Die Grace Period kann bei Bedarf für einzelne Targets administrativ geändert werden.

Die Länge der Grace Period sollte den jeweiligen Anforderungen angepaßt werden. Die Größenordnung wird üblicherweise im Bereich von 5-30 Tagen liegen.

Polling Intervall

Das Polling Intervall ist die Zeitdauer, die zwischen zwei Anfragen des Targets an den Application Server liegt. Das Polling Intervall wird in der Target-Konfiguration festgelegt (`/etc/swdist/swdist.conf`), dabei liegt die Einheit Sekunden zugrunde.

Die Länge des Polling Intervalls sollte den jeweiligen Anforderungen angepaßt werden. Die Größenordnung wird üblicherweise im Bereich von 200-2000 Sekunden liegen.

Emergency-Updates werden sofort ausgeführt, sobald der `sdcl` sie erhält. D.h. sie werden im Mittel nach einem halben Polling Intervall ausgeführt.

Client GUI

Auf Workstations wird der Benutzer durch ein GUI über anstehende Updates informiert. Die Updates können durch den Benutzer sofort ausgeführt werden. Sollen Updates über das Ende einer normalen Grace Period hinaus zurückgestellt werden, kann der Benutzer dies bei der Hotline beantragen. Das GUI bietet eine administrativ editierbare Online-Hilfe, um entsprechende Informationen bereitzustellen. Der Hilfetext kann als HTML-Datei unter `/usr/share/doc/packages/swdist/sdcl-online-help.html` abgelegt werden.

Abbildung 6.1 zeigt das GUI mit einem anstehenden Configuration Change. Mit dem Knopf **Ausführen** werden alle anstehenden Updates sofort ausgeführt, mit dem Knopf **Aktualisieren** wird die Anzeige aktualisiert, der Knopf **Hilfe** zeigt die administrativ editierbare Online-Hilfe.

Log- und Config-Files auf Target

Tabelle 2.6 gibt einen Überblick über die wesentlichen Log-Files, Tabelle 6.2 nennt zentrale Config-Files. Beispiele der zentralen Config-Files sind in Abschnitt 6.5 wiedergegeben.

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------

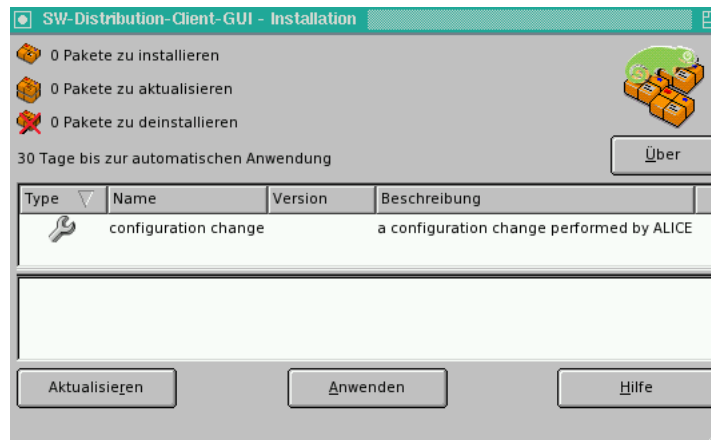


Figure 6.1: Configuration Update

<i>File</i>	<i>Bedeutung</i>
/etc/swdist/appserv/swdist-ds.xml	
/etc/jboss/default/login-config.xml	
/etc/jboss/default/log4j.xml	
/srv/www/jboss/default/deploy/jbossweb-tomcat50.sar/server.xml	

Table 6.1: Server Config Files

<i>File</i>	<i>Bedeutung</i>
/etc/swdist/swdist.conf	SWDIST Client Config
/etc/swdist/platforms	Mapping der Release-Bezeichnung
/etc/sysconfig/alice	
/etc/ntp.conf	optional, falls NTP-Server nicht per DHCP geliefert wird

Table 6.2: Target Config Files

6.2 Target via Netzwerk installieren

Voraussetzungen

- Target, Stream usw. sind in APP-Server vorhanden (siehe ??)
- Control File ist im Boot/Install -Server abgelegt (s. ??)
- Boot-Medium ist vorhanden (s. ??)
- Target-BIOS erlaubt Floppy-Boot
- DHCP-Server und Netzwerk sind verfügbar

Ergebnis

- Target ist installiert und konfiguriert

Tools

- grub
- autoyast
- sdacexp
- make_all

Beschreibung

Das Target wird per GRUB-Floppy gebootet. Anschließend wird `kernel` und `initrd` geladen. Danach startet `linuxrc` und lädt YaST. Dieser lädt anhand der MAC-Adresse das Control File und führt die darin beschriebene Grundkonfiguration und SW-Installation aus.

Nachdem

6.2.1 System bootet von Floppy

```
grub
```

6.2.2 YaST installiert

YaST, AutoYaST

6.2.3 ALICE startet Konfiglauf

```
/etc/init.d/alice  
/usr/lib/alice2/bin/make_all
```

6.2.4 Finaler Reboot

Aktionen

- Grub Floppy entfernen
- ggf. Target testen (QA)

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



6.3 Target RPMs und Konfiguration manuell updaten (ohne GUI)

Voraussetzungen

- Target in APP Server angelegt
- Konfiguration für Target in APP Server
- Target und Server online

Ergebnis

- Target RPMs und Konfiguration up to date

Tools

- sdclld

```
rcsdclld trigger
```

Aktionen

- ggf. Target RPMs testen (QA)

6.4 Target RPMs und Konfiguration manuell updaten (mit GUI)

Voraussetzungen

- Target in APP Server angelegt
- Konfiguration für Target in APP Server
- Target und Server online

Ergebnis

- Target RPMs und Konfiguration up to date

Tools

- sdclgui

```
sdclgui  
click the button "Ausführen"
```

Aktionen

- ggf. Target RPMs testen (QA)

6.5 Nützliche Befehle

OS Version, Distributions-Architektur, Service Pack anzeigen

```
dbnu002:~ # cat /etc/SuSE-release
SuSE SLES-8 (AMD64)
VERSION = 8.1
```

```
dbnu002:~ # cat /etc/UnitedLinux-release
UnitedLinux 1.0 (AMD64)
VERSION = 1.0
PATCHLEVEL = 3
```

Der Service Pack (Patch Level) steht nur in UnitedLinux-release. Die Bezeichnung der Produkte und Distributions-Architektur in den Release-Files kann von den in Mediasets und SWDIST verwendeten Standards abweichen. Der `sdclld` verwendet die Datei `/etc/swdist/platforms` für die Zuordnung.

Reboots anzeigen

```
dbnu002:~ # last | grep "boot"
reboot    system boot    2.4.19-4GB          Wed Jun 16 13:39      (01:36)
...
```

Überblick über vergangene Ereignisse

```
dbnu002:~ # last -20
dbnu002:~ # tail -200 <log_file> | less
dbnu002:~ # sar
```

Überblick über Messages des `sdclld`

```
dbnu002:~ # grep SDCLD /var/log/messages
```

Laufendes OS (Kernel) zeigen

```
dbnu002:~ # uname -a
Linux dbnu300 2.4.19-4GB #1 Mon Aug 4 23:23:04 UTC 2003 i686 unknown
```

```
dbnu002:~ # cat /proc/version
Linux version 2.4.19-4GB (root@Pentium.suse.de) (gcc version 3.2.2) #1 Mon Aug 4 23:23:04
```

```
dbnu002:~ # cat /proc/cmdline
root=/dev/hda3 vga=794
```

Installierte Pakete zeigen

```
dbnu002:~ # rpm -qa --last
```

Ein installiertes Paket zeigen

```
root@dbnu002:~> rpm -qa --last | grep -i <xxx>
```

Info, Abhängigkeiten und Inhalt eines installierten Paketes zeigen

```
root@dbnu002:~> rpm -qiRl <xxx> | less
```

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Veränderte Dateien zu einem installiertem Paket auf Target finden

```
root@dbnu002:~> rpm -V <xxx>
```

Installiertes Paket zu einer Datei auf Target finden

```
root@dbnu002:~> rpm -qf /sbin/conf.d/SuSEconfig.kde.TMD
```

Glossar

Anwender Nutzer eines Targets (nicht des Management-Systems).

Architektur Die Hardwarearchitektur eines Systems, z.B. *i386* für alle Pentium-kompatiblen Systeme oder *x86_64* für die AMD64-Architektur.

Auswahlliste (selection list) In sich konsistente Liste von **Paketen** bezogen auf eine **Distribution** und eine **Architektur**. Die Auswahllisten werden aus der Distribution importiert oder durch die Systemadministratoren angelegt und gepflegt.

Channel Eine Sammlung von beliebig vielen Paketauswahlen, die zusammen eine Anwendung oder einen Anwendungsfall darstellen.

Durch die Auswahl von Channels wird die Rolle eines **Targets** definiert. Ein Target kann beliebig vielen Channels angehören.

Channel instance Die Inkarnation eines Channels für eine **Architektur** und eine **Distribution**.

Einem Channel sind **Basisauswahllisten** und **Patches** zugeordnet.

Client Ein Arbeitsplatzrechner bzw. die Instanz einer Betriebssystem-Installation. In der Regel Synonym für **Target**.

Distribution Menge von Programmen und Programmpaketen, die mit einer Betriebssystemversion ausgeliefert werden (z.B. SUSE Linux Enterprise Server 8). Eine Distribution kann für unterschiedliche **Architekturen** vorliegen.

Event Ereignis, das für ein **Target** protokolliert wird (z.B. ein Fehler bei einer Patchinstallation).

Konfiguration Information, die das Verhalten einer Software steuert.

Mediaset

Machine **Target**.

Paket (package) Abstrakte kleinste Einheit einer Software. Ein Paket kann in mehreren **Paketinstanzen** vorliegen.

Paketinstanz (package instance) Konkrete kleinste Einheit einer Software in einer konkreten Version. Beinhaltet eine Version eines Softwarepaketes. Bei SUSE-LINUX handelt es sich dabei um ein konkretes RPM-Paket.

Paketliste (package list) Liste der auf einem **Target** installierten Paketinstanzen.

Patch Einheit aus je einer Instanz mehrerer Pakete für eine bestimmte **Distribution** und eine bestimmte **Architektur**.

Produkt **Channel**

Rolle Eine Rolle beschreibt die Funktion eines **Targets** (z.B. Entwickler-Arbeitsplatz). Die Rollen existieren nicht direkt im Datenmodell. Vielmehr werden sie dort durch Software-Auswahl, nämlich **Channels**, und Konfiguration abgebildet.

Wechselt ein Target den **Stream**, bleibt die Rolle davon unberührt, sofern im neuen Stream die benötigten **Pakete** vorhanden sind.

Stream In sich homogener Stand von Paket-Versionen, bezogen auf genau eine **Distribution** und **Architektur**. Zu jeder Kombination von Distribution und Architektur kann es mehrere Streams geben (z.B. *SLES9-i386-devel* für Entwicklung und *SLES9-i386-prod* für Produktion). Der Stream beschreibt die Gesamtheit der für die zugehörigen **Targets** verfügbaren **Paketinstanzen**.

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Tag Abstrakte kleinste Einheit einer **Konfiguration**.

Target System für das (über **Channels**) Softwareinstallationen, Patches und die **Konfiguration** definiert werden.

Update Vorgang der Anwendung eines **Patches**.



Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------

Referenzierte Dokumente

1. SWDIST - Software Distribution, Installation, Patch & Configuration Management - Workshop - SUSE LINUX GmbH; L. Pinne
2. ALICE - Advanced Linux Installation and Configuration Environment
SUSE LINUX GmbH; J. Henge-Ernst, P. Hollants, F. Herschel
3. Automatic Installation and Configuration using YaST
SUSE LINUX GmbH; A. Nashif
4. SLES Option Pack: Data Center Server
SUSE LINUX GmbH; T. Franke
5. Maximum RPM
Red Hat, Inc.; E. C. Bailey, P. Nasrat, M. Saou, V. Skytt

Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------



Manual Pages

1. [alice.8](#)
2. [alice2autoyast.8](#)
3. [copy_machine.8](#)
4. [change_netcfg.8](#)
5. [create_package_descr.8](#)
6. [crontab.1](#)
7. [httpd.8](#)
8. [inetd.8](#)
9. [inetd.conf.8](#)
10. [installserver.7](#)
11. [instboot.5](#)
12. [grub.8](#)
13. [make_boot_server.8](#)
14. [make_grub_disk.8](#)
15. [make_inst_server.8](#)
16. [make_inst_dvd.8](#)
17. [md5sum.1](#)
18. [online_update.8](#)
19. [rpm.8](#)
20. [rpmlist.8](#)
21. [rsync.1](#)
22. [show_value.8](#)
23. [sdadm.8](#)
24. [sdclld.8](#)
25. [sdacexp.8](#)
26. [sdacimp.8](#)
27. [sdclld.8](#)
28. [sdptimp.8](#)



Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------

- 29. sdslimp.8
- 30. sdstexp.8
- 31. sdverify.8
- 32. sdswimp.8
- 33. swdist.conf.5
- 34. tftpd.8
- 35. wget.1
- 36. youmirror.8
- 37. youbrush.8

Konzeption

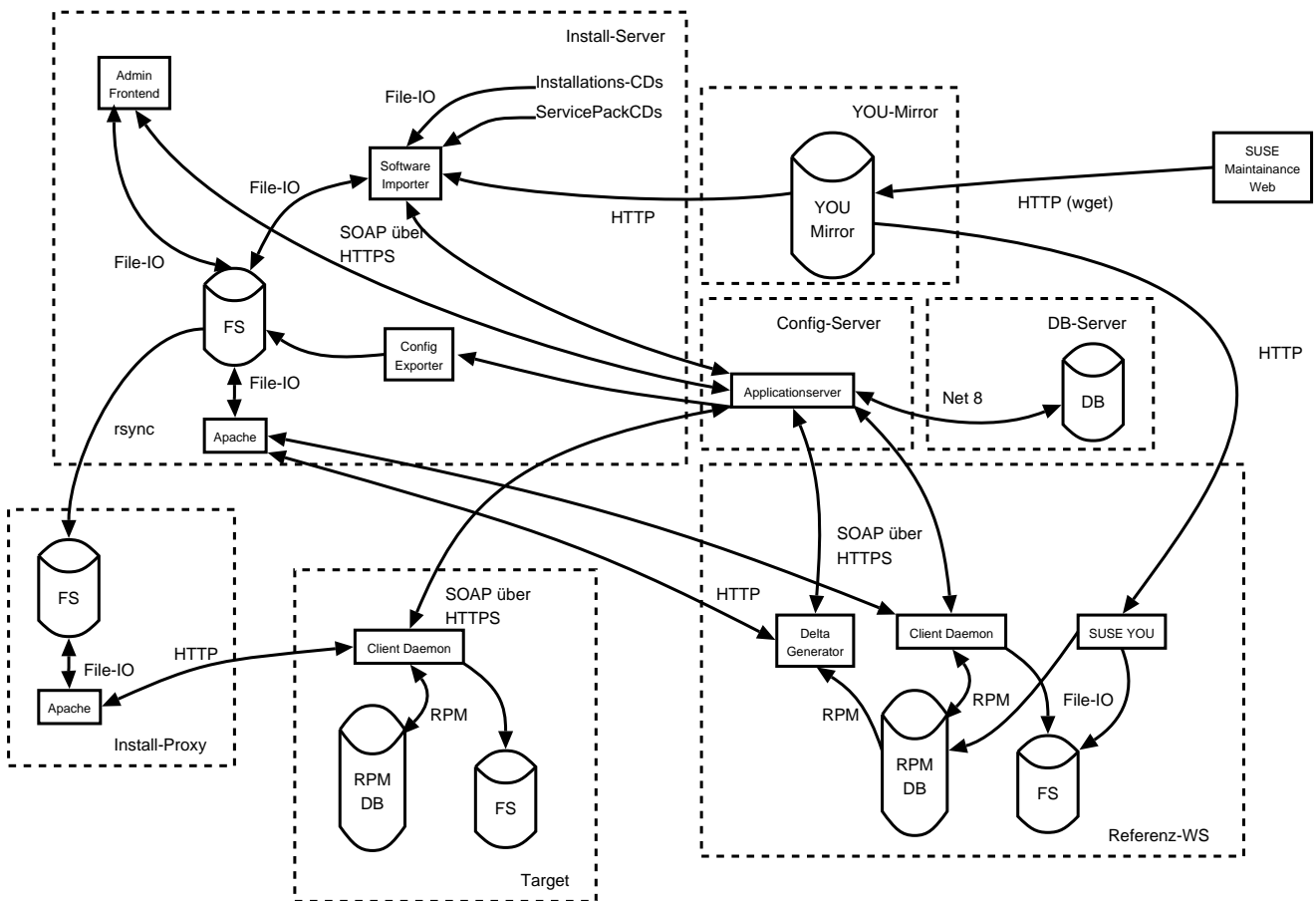


Figure 6.2: Architektur

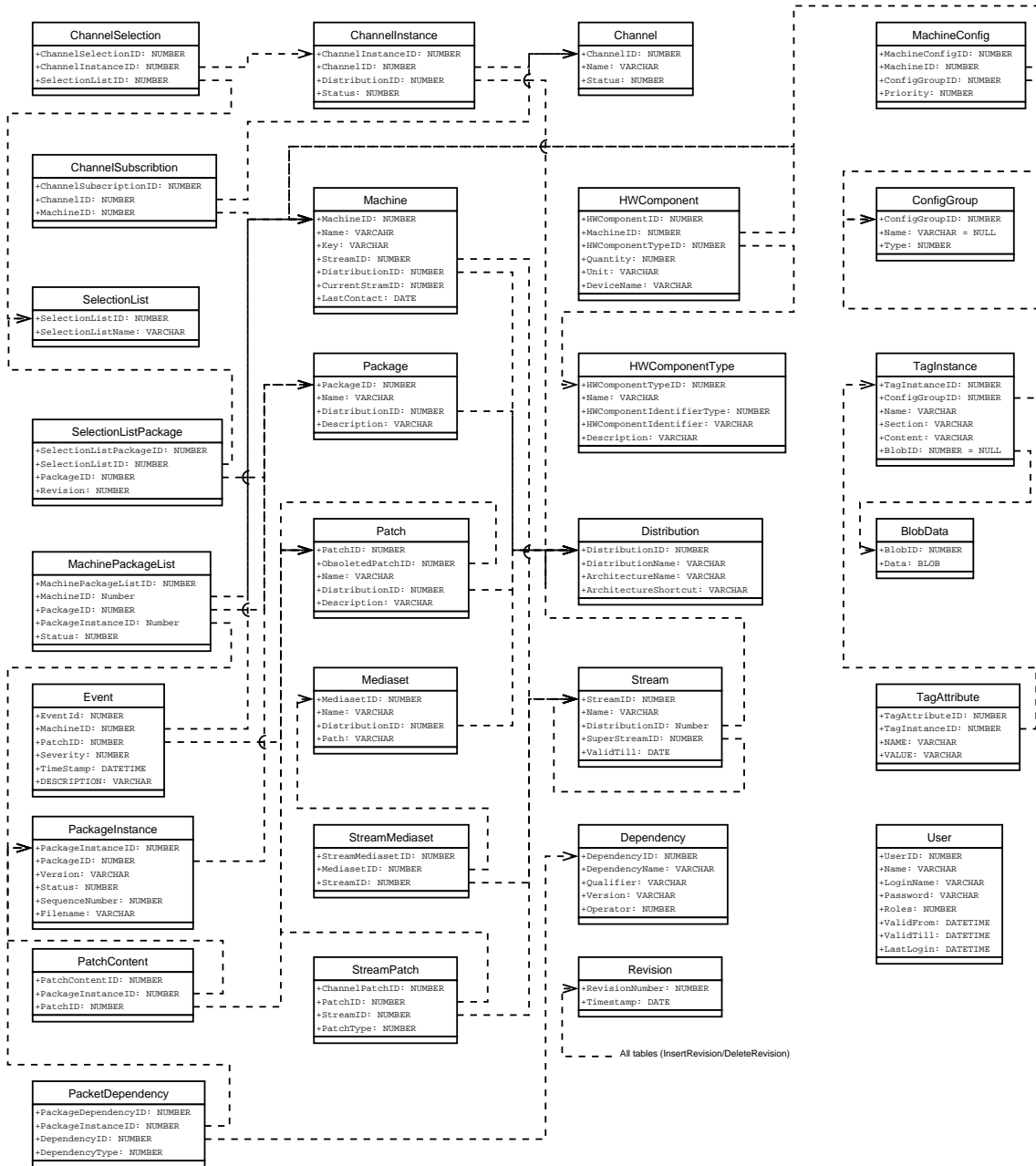


Figure 6.3: Datenmodell

Config Files

/etc/swdist/swdist.conf

```
#####
#
# This is the configuration file for the SLCL-Daemon
# (c) 2004 SUSE LINUX AG, ESCHBORN, GERMANY
#
#####
#
# userid and password.
# Note: not all services use, and only the sdcl
# requires these parameters.
# for the sdcl you need the machine name and key here.
WS_USERID = wing.lab.ffm.novell.com
WS_PASSWD = placeKeyHere
#
# The following connection paramaters to the
# applicationserver are needed by all tools.
#
# this is the hostname of the applicationserver
# Warning: if the SWDIST_PROTOCOL is https
# the hostname must be the same as in the server
# certificate.
WS_HOST = 149.44.71.13
#
# protocol and port of the applicationserver
WS_PROTOCOL = http
WS_PORT = 8080
#
# this is an URL path to the webservice
# on the applicationserver. Better leave this alone
WS_PATH = /jboss-net/services
#
# The following parameters are for the
# sdcl only
#
# the machine name
MACHINE_NAME = wing.lab.ffm.novell.com
#
# the name of the file used to determine the installed
# distribution
DIST_INFO = /etc/SuSE-release
#
# the name of the alice executable
ALICE_SCRIPT = /usr/lib/alice2/bin/make_all
#
# the name of the script to execute after the installation is done
POST_CMD = /sbin/SuSEconfig
#
# the client poll interval
POLLING_INTERVAL = 50
#
# The following parameter is used by more than one tool
#
# the name and IP address of the install server
SYS_INST_SERVER = howard.lab.ffm.novell.com
SYS_INST_SERVER_IP = 149.44.71.13
#
# The following parameters are needed for the ALICE environment
#
# the installation protocol
SYS_INST_PROTOCOL = http
#
# the OS vendor needed for ALICE
SYS_OS_VENDOR = SUSE
#
# the OS
SYS_OS = linux
#
# the following parameters are needed for the admin tools
# that run on an install server (you can ignore them on
# other systems
#
# The root directory where the software is stored
# (mirror area, mediasets, patches, private patch area).
PKG_ROOT = /space
#
# The DocuementRoot of the HTTP server
DOC_ROOT = /srv/www/htdocs
#
```



Specification SWDIST Handbuch Administration	Status DRAFT	Author SUSE LINUX GmbH
--	-----------------	---------------------------

/etc/swdist/platforms

```
#
# $Id: platforms,v 1.2 2004/12/22 20:12:35 pinne Exp $
#
# (c) 2004 SUSE LINUX GmbH
#
# Maps /etc/SuSE-release strings to distribution name and architecture
#
# Format: <str>|<dist>|<arch>
# where <str> is the string in line 1 of /etc/SuSE-release
# <dist> is the distribution name
# <arch> is the architecture shortcut
#
SuSE SLES-8 (i386)|SLES8|i386
SUSE LINUX Enterprise Server 9 (i586)|SLES9|i386
SuSE SLES-8 (x86_64)|SLES8|x86_64
SUSE LINUX Enterprise Server 9 (x86_64)|SLES9|x86_64
#
```

/etc/swdist/appserv/swdist-ds.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- ===== -->
<!-- JBoss Server Configuration -->
<!-- ===== -->
<!-- $Id: swdist-ds.xml,v 1.1 2004/12/22 20:12:35 pinne Exp -->
<!-- ===== -->
<!-- Datasource config for Oracle originally from Steven Coy -->
<!-- ===== -->

<datasources>
  <local-tx-datasource>
    <jndi-name>SwdistDS</jndi-name>
    <connection-url>jdbc:oracle:thin:@alice.lab.ffm.novell.com:1521:LSDB</connection-url>
  <!--
  Here are a couple of the possible OCI configurations.
  For more information, see http://otn.oracle.com/docs/products/oracle9i/doc\_library/release2/java.920/a96654/toc.htm

  <connection-url>jdbc:oracle:oci:@youroracle-tns-name</connection-url>
  or
  <connection-url>jdbc:oracle:oci:(description=(address=(host=youroraclehost)(protocol=tcp)(port=1521))(connect_data=(SERVICE_NAME=yourservicename)))</connection-url>

  Clearly, its better to have TNS set up properly.
  -->
  <driver-class>oracle.jdbc.driver.OracleDriver</driver-class>
  <user-name>lsd_schema</user-name>
  <password>lsd</password>
  <!-- Uses the pingDatabase method to check a connection is still valid before handing it out from the pool -->
  <valid-connection-checker-class-name>org.jboss.resource.adapter.jdbc.vendor.OracleValidConnectionChecker</valid-connection-checker-class-name>
  <!-- Checks the Oracle error codes and messages for fatal errors -->
  <exception-sorter-class-name>org.jboss.resource.adapter.jdbc.vendor.OracleExceptionSorter</exception-sorter-class-name>
  <!-- sql to call when connection is created
  <new-connection-sql>some arbitrary sql</new-connection-sql>
  -->

  <!-- sql to call on an existing pooled connection when it is obtained from pool - the OracleValidConnectionChecker is preferred
  <check-valid-connection-sql>some arbitrary sql</check-valid-connection-sql>
  -->

  </local-tx-datasource>
</datasources>
```

/etc/jboss/default/login-config.xml

/etc/jboss/default/log4j.xml

/srv/www/jboss/default/deploy/jbossweb-tomcat50.sar/server.xml