

Oracle9i Installation auf Debian GNU/Linux 3.0

Urs Stotz

1. September 2005

Zusammenfassung

Hier folgt eine Kurzbeschreibung, wie man Oracle9i auf Debian GNU/Linux installiert. Die Installation auf anderen Distributionen wie Redhat oder SuSE dürfte auch keine Probleme bereiten. Ich werde aber nicht darauf eingehen.

Im Gegensatz zu anderen Dokumentationen möchte ich hier aufzeigen, wie nach der Software-Installation mit dem RunInstaller, eine Basisdatenbank über die entsprechenden Shell- und SQL-Befehle erzeugt wird.

Bei dieser Dokumentation gehe ich davon aus, dass auf dem System keine weitere Instance installiert ist.

Ich habe versucht die Befehle so zu gestalten, dass sie mit copy and paste in einen Editor kopiert werden können, wo man sie überarbeitet und nachher in der Shell oder in SQL*Plus ausführt. Ich finde dieses Vorgehen bei unterschiedlichen Umgebungen praktischer als fertige Scripts. Wichtig ist, dass die Befehle welche zum Erzeugen der Instance gebraucht wurden, nachher fein säuberlich im Installationsscript stehen.

Beteiligte an dieser Dokumentation sind: Urs Stotz und Martin Furtner.

Die komplette Dokumentation bestehend aus LaTeX, Makefile, Bilder, dvi, ps, pdf html, txt und Script, lässt sich von www.napali.ch herunterladen. Die Dokumentation kann auch direkt im pdf Format heruntergeladen werden. Wenn jemand einen Fehler oder eine Ergänzung sieht, dann bin ich froh, wenn er mir dazu einen Patch schickt (stotz at gmx dot ch). Es ist aber auch möglich direkt an der Erweiterung dieser Dokumentation mitzuarbeiten und über Subversion auf das Repository zugreifen zu können. Auch dazu benötige ich ein e-Mail.

Inhaltsverzeichnis

1	Debian GNU/Linux Systemvorbereitung	4
1.1	Softwareinstallation	4
1.2	Kernel 2.4.21 installieren	4
1.3	Shared Memory	5
1.4	locales erzeugen	5
1.5	/etc/inittab anpassen	6
1.6	Oracle User und DBA Gruppe anlegen	6
1.7	Home für Oracle	7
2	Oracle Installation	7
2.1	Umgebungsvariablen	7
2.2	Installationsquelle	8
2.3	Installation mit dem Oracle Universal Installer	8
2.3.1	RunInstaller	9
2.3.2	Inventory Location	10
2.3.3	UNIX Group Name	10
2.3.4	orainstRoot.sh	10
2.3.5	File Locations	11
2.3.6	Error	11
2.3.7	Available Products	12
2.3.8	Language Selection	12
2.3.9	Installation Types	13
2.3.10	Available Product Components	13
2.3.11	Component Locations	15
2.3.12	Privileged Operating System Group	15
2.3.13	Create Database	15
2.3.14	Summary	15
2.3.15	Install	16
2.3.16	Setup Privileges	16
2.3.17	Configuration Assistant	18
2.3.18	End of Installation	18
2.3.19	Patch File Locations	19
2.3.20	Patch Set Version 9.2.0.2	19
2.3.21	Available Products	19
2.3.22	Install Patch Set	19
3	Basisdatenbank erzeugen	20
3.1	Verzeichnisse erstellen	20
3.2	initDB01.ora erstellen	21
3.3	Instance starten und Basisdatenbank erzeugen	22
3.4	Basisdatenbank mit Script erstellen	23
3.5	Basisdatenbank erstellen gescheitert	32

4	Tablespaces und Benutzer	33
4.1	Passwort von SYS und SYSTEM ändern	33
4.2	Tablespaces erstellen	34
4.3	Benutzer erstellen	34
5	Oracle Net	35
5.1	sqlnet.ora	35
5.2	listener.ora	35
5.3	tnsnames.ora	37
6	Instance mit Runlevel automatisch starten	38
6.1	dbstart und dbshut	38
6.2	/etc/init.d/listener	38
6.3	/etc/init.d/oracleDB01	39
6.4	Runlevel Einträge	40
7	Weiteres, Tips und Tricks	40
7.1	Java anpassen für oemapp, dbastudio	40

1 Debian GNU/Linux Systemvorbereitung

Voraussetzung ist eine default Installation von Debian GNU/Linux 3.0 (woody). Hilfreich dafür sind diese Links:

1. Debian GNU/Linux
2. Debian GNU/Linux Anwenderhandbuch
3. The Linux Documentation Project

1.1 Softwareinstallation

Zusätzlich zur default Installation von Debian benötigen wir weitere Pakete, die wir mit dem Befehl `apt-get` installieren können.

Um nachher den Kernel zu erstellen:

```
tux01:~# apt-get install binutils libbz2-1.0 fileutils \  
    bzip2 libc6-dev gcc make libncurses5-dev kernel-package
```

Systemtools und Bibliotheken:

```
tux01:~# apt-get install vim wget bonnie++ sysstat unzip \  
    pdksh xbase-clients xterm apt-file libstdc++2.9-glibc2.1 sudo
```

Wichtig ist die Bibliothek `libstdc++2.9-glibc2.1`, weil die Java Version, die von Oracle mitgeliefert wird, gegen die alte GNU `stdc++` library gelinkt wurde. Die Shell `ksh` (`pdksh`) wird von einigen Oracle Scripts gebraucht. Beim Installieren löst Debian die Abhängigkeit von Paketen selbstständig auf.

Dann deinstallieren wir noch unnötige Pakete:

```
tux01:~# dpkg -P portmap nfs-common nfs-kernel-server \  
    nvi ppp pppconfig pppoe pppoeconf
```

Nfs und Portmap werden auf einem Oracle-Server nicht gebraucht. Es ist zu empfehlen, nur die wirklich notwendigen Dienste zu installieren.

1.2 Kernel 2.4.21 installieren

Wir downloaden die Sourcen des Vanilla Kernels von <http://www.kernel.org>

```
tux01:~# cd /usr/src  
tux01:/usr/src# wget \  
    http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.21.tar.bz2
```

Sollte ein Proxy vorhanden sein, dann ist vor dem Befehl `wget` die Variable `http_proxy` zusetzen. Dann sieht der Befehl so aus:

```
tux01:/usr/src# http_proxy=http://http-proxy:3128 wget \  
http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.21.tar.bz2
```

Nacher entpacken wir die Sourcen und compilieren den Kernel mit dem Debian „Kernel Package Tool“ `make-kpkg`, damit wird vom Kernel ein Debian Package erzeugt, welches wir über Debian Paketverwaltung installieren und deinstallieren können.

```
tux01:/usr/src# tar xjf linux-2.4.21.tar.bz2  
tux01:/usr/src# ln -f -s linux-2.4.21 linux  
tux01:/usr/src/linux# cd linux  
tux01:/usr/src/linux# make mrproper  
tux01:/usr/src/linux# make-kpkg clean  
tux01:/usr/src/linux# make-kpkg --revision=2.4.21v1.0 \  
--config=menuconfig kernel_image
```

Das neue Kernel Package kann man mit dem Befehl `dpkg` installieren.

```
tux01:/usr/src/linux# dpkg -i \  
../kernel-image-2.4.21_2.4.21v1.0_i386.deb
```

Nach der Installation des neuen Kernels muss der Rechner rebootet werden. Konfiguration für den 2.4.21er Kernel

1.3 Shared Memory

Die vom System benutzte Shared Memory setzen wir auf auf etwa ein Viertel der physischen Memory, minimum liegt etwa bei 32 MB. Ich gehe im bei der weiteren Beschreibung von einem System mit etwa 512 MB Memory aus. Dazu muss bei der Konfiguration des Kernel die Option `CONFIG_SYSCTL=y` gewählt worden sein. So ist es möglich dynamisch Kernel Parameter und Variablen im laufenden System zu ändern, ohne den Kernel neu zu kompilieren, oder das System rebooten zu müssen.

```
tux01:~# sysctl -w kernel.shmmax=100000000  
tux01:~# echo kernel.shmmax=100000000 >> /etc/sysctl.conf
```

1.4 locales erzeugen

Wir erzeugen die locales für den „National Language data support“. In den locales von `de_CH` gibt es einen Fehler. Um das zu korrigieren müssen wir die Datei `de_CH` editieren. Am Ende einer Zeile fehlt das Zeichen `”`. Mit dem folgenden Befehl springt der `vi` an die richtige Position in der Datei:

```
tux01:~# grep d_t_fmt.*\ "$ /usr/share/i18n/locales/de_CH || \  
vi -c /d_t_fmt /usr/share/i18n/locales/de_CH
```

Am Ende dieser Zeile fügen wir " an, und speichern die Datei. Nachher lassen sich die Locales erzeugen. Als default sollte C als Systemumgebung gewählt werden.

```
tux01:~# dpkg-reconfigure locales
Which locale should be the default in the system environment
C
Generating locales...
  de_CH.ISO-8859-1... done
  de_DE.ISO-8859-1... done
  de_DE.ISO-8859-15@euro... done
  en_GB.ISO-8859-1... done
  en_IE.ISO-8859-1... done
  en_IE.ISO-8859-15@euro... done
  en_US.ISO-8859-1... done
Generation complete.
```

1.5 /etc/inittab anpassen

Ich verstehe nicht, warum es bei den Linux Distributionen default ist, dass ein shutdown ausgelöst wird, wenn jemand an der lokalen Tastatur CTRL-ALT-DEL drückt. Diesen Unsinn entferne ich immer. Ich halte das für ein unnötiges Sicherheitsloch. Am besten wird die Zeile auskommentiert:

```
tux01:~# vi -c ^ca: /etc/inittab
# ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

Nachher muss dem init Prozess das Signal HUP geschickt werden, so dass er die Konfigurationsdatei /etc/inittab neu einliest.

```
tux01:~# kill -HUP 1
```

Ab sofort hat CTRL-ALT-DEL keine Auswirkung mehr. Der Windows Admin kann also nicht ausversehen unseren Server rebooten, nur weil er irrtümlich an der Switch-Console auf dem Linx-Server gelandet ist, und sich mit CTRL-ALT-DEL beim dunkeln Bildschirm anmelden wollte (was man ihm ja nicht verübeln könnte).

1.6 Oracle User und DBA Gruppe anlegen

Die Gruppe DBA erstellt man mit dem Befehl addgroup:

```
tux01:~# addgroup --gid 2000 dba
Adding group dba (2000)...
Done.
```

Und jetzt noch den Oracle User erstellen mit adduser:

```

tux01:~# mkdir /oracle
tux01:~# adduser --home /oracle --shell /bin/bash \
  --no-create-home --uid 2000 --gid 2000 oracle
adduser: Warning: The home dir you specified already exists.
Adding user oracle...
Adding new user oracle (2000) with group dba.
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for oracle
Enter the new value, or press return for the default
    Full Name []: Oracle
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [y/n] y

```

1.7 Home für Oracle

Für den Oracle User und die Oracle Installation, habe ich eine eigene Partition von 4.5 GByte Speicherplatz vorbereitet. Diese trage ich in fstab ein, so dass sie beim Systemstart automatisch gemountet wird. Als Filesystem benutze ich ext2. Für die Oracle Installation werden gut 2 GByte Speicherplatz benötigt. Die noch zu erstellende Basisdatenbank braucht etwa 860 MByte Speicherplatz.

```

tux01:~# echo /dev/hda8 /oracle ext2 defaults 0 2 >>/etc/fstab
tux01:~# mount /oracle

```

Login Scripts nach /oracle kopieren und den Owner setzen.

```

tux01:~# cp /etc/skel/.[abc]* /oracle/
tux01:~# chown -R oracle:dba /oracle/

```

2 Oracle Installation

Der grösste Teil der Installation kann als User Oracle ausgeführt werden.

2.1 Umgebungsvariablen

Oracle braucht noch weitere Umgebungsvariablen, die vor der Installation gesetzt werden müssen.

```

tux01:~# su - oracle
oracle@tux01:~$ cat >>.bash_profile <<EOF
export LANG=C

```

```

export ORACLE_SID=DB01
export ORACLE_BASE=/oracle
export ORACLE_HOME=\$ORACLE_BASE/product/920
export ORACLE_DOC=\$ORACLE_HOME/doc
export ORACLE_TERM=xterm
export ORA_NLS33=\$ORACLE_HOME/ocommon/nls/admin/data
export PATH=\$ORACLE_HOME/bin:/usr/local/bin:\$PATH
export LD_LIBRARY_PATH=\$ORACLE_HOME/lib
export LC_ALL=en_US
export LC_LANG=en_US
export SHELL=/bin/bash
EOF
oracle@tux01:~$ . .bash_profile

```

Wichtig ist der Befehl „. .bash_profile“, dass die neuen Variablen für die aktuelle Shell aktiv werden.

2.2 Installationsquelle

Die Installationsdateien können von Oracle heruntergeladen werden. Dazu muss man sich bei Oracle registrieren und unter anderem nicht in Kuba wohnhaft sein. Der private Gebrauch von Oracle ist kostenlos. Die genauen Lizenzbestimmungen müssen bei Oracle nachgelesen werden.

Die Archive der Installationsdateien belegen 1.55 GByte. Die entpackten Installationsdateien belegen 1.72 GByte.

Wir entpacken die mit gzip und cpio gepackten Quellen:

```

oracle@tux01:~$ su - oracle
oracle@tux01:~$ cd /data
oracle@tux01:/data$ zcat lnx_920_disk1.cpio | cpio -idmv
oracle@tux01:/data$ zcat lnx_920_disk2.cpio | cpio -idmv
oracle@tux01:/data$ zcat lnx_920_disk3.cpio | cpio -idmv

```

2.3 Installation mit dem Oracle Universal Installer

Die Software wird über das Java-Tool RunInstaller installiert. Er wird Oracle Universal Installer oder OUI genannt. Dieses Programm gibt es seit der 8-er Version von Oracle und es bereitete schon öfter Probleme auf Unix wie auch Windows Systemen. Vorher war es ein konsolenbasiertes Installationsprogramm, was vollkommen ausreichte. Mit dem neuen RunInstaller (Kindergartenmodus) sollen die fähig sein ein Oraclesystem zu installieren, welche besser die Finger davon lassen würden. Für den RunInstaller lässt sich ein Respond-File erstellen, was sich sehr gut eignet, wenn immer wieder die gleiche Installation durchgeführt wird. Mit dem Respond-File kann auf User-Input verzichtet werden und die Installation kann im silent Mode automatisch ausgeführt werden. Mit dem Aufruf:

```
./runInstaller -record -destinationFile install_oracle920.rsp
```


Wird ein Respondedfile im Record-Mode erzeugt. Beim Installationspunkt „Summary page“ kann man die Installation abbrechen, und hat ein funktionierendes Respondedfile. Unter Oracle Universal Installer Concepts Guide findet man die genaue Dokumentation.

Hier machen wir das aber über die GUI vom RunInstaller.

Somit sind wir jetzt beim Bilderbuchteil dieser Dokumentation angelangt.

Achtung: Oracle 10 prüft ob die Linux Distribution eine der von Oracle unterstützten ist (RedHat, SuSE oder United Linux) und bricht die Installation ab falls dem nicht so ist. Die folgende Anweisung gaukelt dem Installer ein RedHat System vor (das File kann nach der Installation wieder entfernt werden):

```
root@tux01:~# echo redhat-3 > /etc/redhat-release
```

Tritt ein Fehler beim Starten von RunInstaller wie:

Can't open display: oder DISPLAY is not set auf, muss vorher die DISPLAY Variable auf den entsprechenden X11 Server gesetzt werden:

```
export DISPLAY=:0.0 oder export DISPLAY=host:0.0
```

Tritt der Fehler:

```
../java: error while loading shared libraries:  
libstdc++-libc6.1-1.so.2: cannot open shared object file:  
No such file or directory
```

```
mkdir lib  
locate libstdc++-libc6  
/usr/lib/libstdc++-libc6.2-2.so.3  
ln -s /usr/lib/libstdc++-libc6.2-2.so.3 lib/libstdc++-libc6.1-1.so.2  
export LD_LIBRARY_PATH='pwd'/lib:$LD_LIBRARY_PATH
```

2.3.1 RunInstaller

Vor dem Start des RunInstaller muss die Variable DISPLAY auf den entsprechenden X11 Server gesetzt werden. Da ich die Installation von meinem Arbeitsplatz aus mache und nicht direkt auf dem DB-Server, setze ich die Variable auf die IP Adresse meines Arbeitsplatzes. Nachher kann der Run-Installer gestartet werden. Der RunInstaller braucht minimum 80 MByte freien Platz in /tmp, 512 MByte Ram oder einen genügend grossen Swap.

```
oracle@tux01:~$ export DISPLAY=192.168.200.93:0.0  
oracle@tux01:~$ /data/Disk1/runInstaller &
```

Welcome:

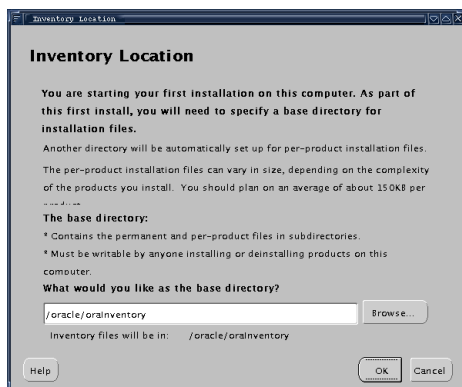


Abbildung 1: Welcome

2.3.2 Inventory Location

Hier speichert der RunInstaller, den Softwarestatus des Systems. Das heisst, es ist das Bestandesverzeichnis der installierten Software und ihres Patchlevels. Der Vorschlag `/oracle/oraInventory` kann übernommen werden. Dieser Pfad wird dann später in `/etc/oraInst.loc` gespeichert.

Inventory Location:



2.3.3 UNIX Group Name

Hier wird eingetragen welche Systemgruppe das Recht hat, die Oracle Software zu updaten. Dazu muss diese Systemgruppe Schreibrechte auf dem Oracle Verzeichnis haben. Hier wähle ich dba.

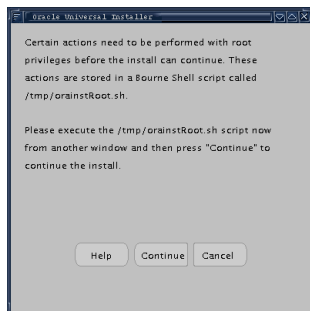
UNIX Group Name:

2.3.4 orainstRoot.sh

Jetzt erscheint eine Messagebox, ein Script müsse ausgeführt werden. Dieses Script `/tmp/orainstRoot.sh` muss man als root in einer Shell ausführen. Das



Script erstellt die Datei `/etc/oraInst.loc`, und ändert die Gruppenberechtigung von `/oracle/oraInventory`



```
tux01:/tmp# ./orainstRoot.sh
Creating Oracle Inventory pointer file (/etc/oraInst.loc)
Changing groupname of /oracle/oraInventory to dba.
```

Nachher geht es mit Continue weiter.

2.3.5 File Locations

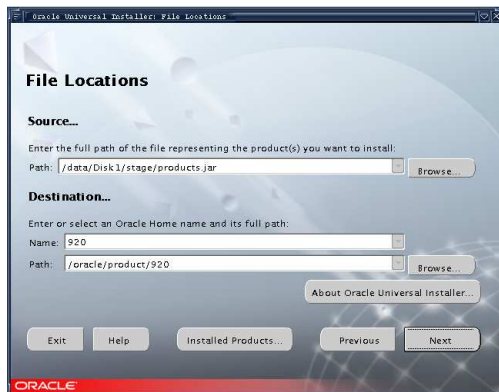
Hier muss die Installations-Datei `products.jar` als Source ausgewählt werden. Bei dieser Datei handelt es sich um die Komponentenliste, der Installationsquelle. Es wäre auch möglich einen Web-Server als Installationsquelle einzurichten.

Bei Destination wird der Name: 920 und als Installations Path: `/oracle/product/920` gewählt. Dieses Verzeichnis entspricht der System Variable `ORACLE_HOME`.

File Locations:

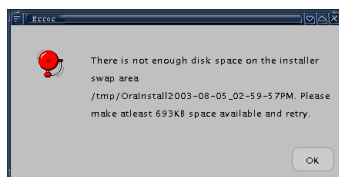
2.3.6 Error

Sollte während der Installation eine Fehlermeldung wegen zu wenig Platzes auf dem Filesystem `/tmp` gemeldet werden, kann dieses Problem so umgangen



werden. Der RunInstaller hat ein temporäres Verzeichnis unter `/tmp` angelegt. Darin speichert er temporär, die für den Installationsprozess notwendige Dateien. Der Speicherplatz belegt mindestens 75 MByte. Sollte in `/tmp` nicht mehr genügend Speicherplatz vorhanden sein, lässt sich das von RunInstaller angelegte Verzeichnis auf ein anders Dateisystem verschieben. Nachher erstellt man einen symbolischen Link in `/tmp` mit dem ursprünglichen Namen.

Error:



```
tux01:~# cd /tmp
tux01:/tmp# mv OraInstall2003-08-05_02-59-57PM /data
tux01:/tmp# ln -s /data/OraInstall2003-08-05_02-59-57PM
\ OraInstall2003-08-05_02-59-57PM
```

Jetzt kann man OK klicken, und die Installation geht weiter.

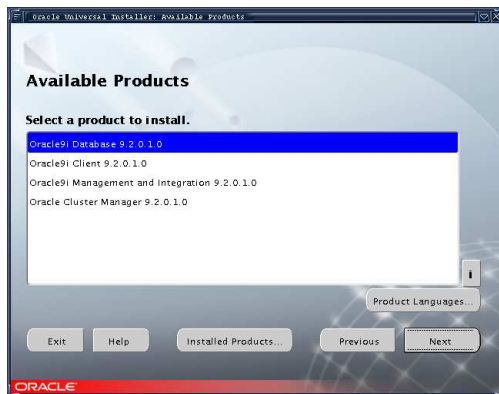
2.3.7 Available Products

Bei der Produktauswahl wähle ich Oracle9i Database 9.2.0.1.0 und klicke Next.

Available Products:

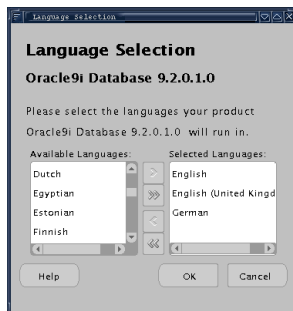
2.3.8 Language Selection

Zusätzlich zu English, wähle ich noch English UK und German.



Zur Sprachauswahl wird in der Oracle Hilfe geschrieben: „Die Sprachauswahlen gelten für die Komponente der obersten installierten Ebene. Eine Komponente der obersten Ebene entspricht dem Produkt, das in dem Dialogfeld Produktauswahl ausgewählt wurde (ein Produkt pro Installationssitzung). Die englischen Dateien werden stets für alle Dateien installiert, unabhängig von der Benutzerauswahl oder der Sprache des Betriebssystems. Wenn eine Komponente bereits mit einer anderen Spracheinstellung installiert ist, wird diese Komponente erneut installiert. Bei der Installation durchsucht OUI die Sprachliste jeder Datei/Dateigruppe, um festzustellen, welche kopiert werden muss.“

Language Selection:



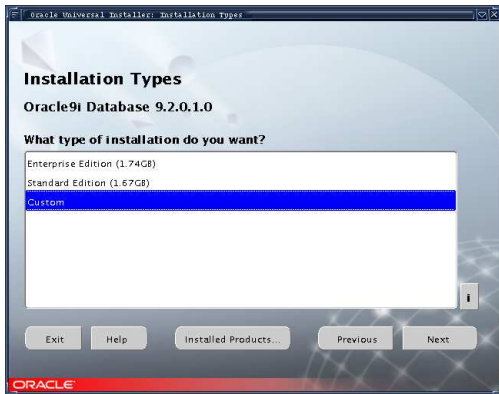
2.3.9 Installation Types

Hier empfehle ich die Wahl von Custom, weil sonst viel unnötiges Installiert wird. Man kann jederzeit weitere Komponenten nachinstallieren.

Installation Types:

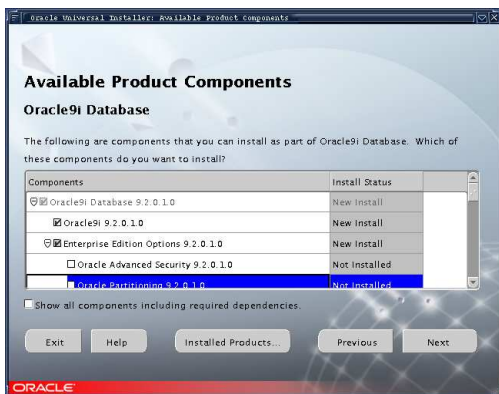
2.3.10 Available Product Components

Ich wähle die folgenden Komponenten:



- Oracle9i 9.2.0.1.0
- Enterprise Edition Options 9.2.0.1.0
- Oracle Net Services 9.2.0.1.0
 - Oracle Net Listener 9.2.0.1.0
- Oracle Enterprise Manager Products 9.2.0.1.0
 - Oracle Intelligent Agent 9.2.0.1.0
 - Enterprise Manager Client 9.2.0.1.0
 - Oracle Change Management Pack 9.2.0.1.0
 - Oracle Diagnostics Pack 9.2.0.1.0
 - Oracle Tuning Pack 9.2.0.1.0
 - Oracle Management Pack for Oracle Applications 9.2.0.1.0
 - Oracle Management Pack for Oracle Standard Edition 9.2.0.1.0
- Oracle9i Development Kit 9.2.0.1.0
 - Oracle C++ Call Interface 9.2.0.1.0
 - Oracle Call Interface (OCI) 9.2.0.1.0
 - Oracle Programmer 9.2.0.1.0
 - Oracle XML Developer's Kit 9.2.0.1.0

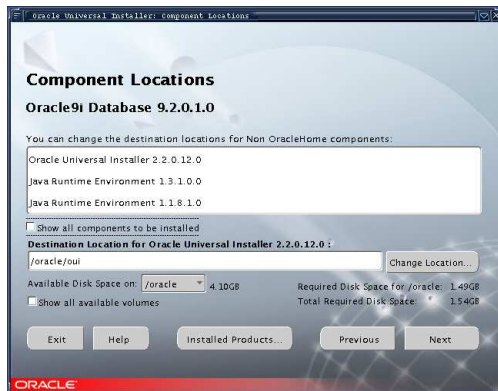
Available Product Components:



2.3.11 Component Locations

Hier kann man Next klicken.

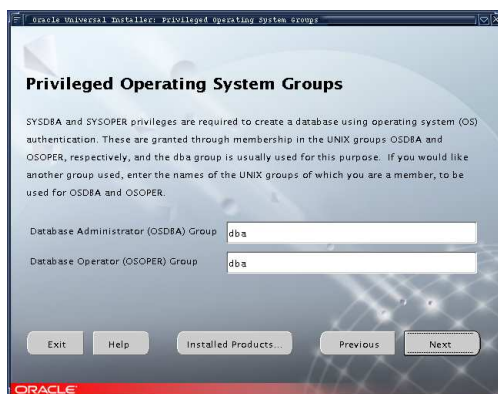
Component Locations:



2.3.12 Privileged Operating System Group

Wir haben bereits eine Systemgruppe DBA angelegt. Diese tragen, wir in beiden Feldern ein. Damit wird festgelegt, welche Systemgruppen SYSDBA- und SYSOPER-Rechte haben.

Privileged Operating System Group:



2.3.13 Create Database

Hier sagen wir No, weil wir später selber eine Basisdatenbank erzeugen.

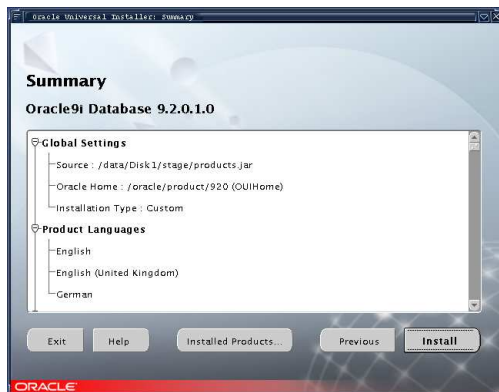
Create Database:

2.3.14 Summary

Hier bestünde die Möglichkeit, wenn man nur ein Responsefile erzeugen will, Exit zu wählen.



Wir wählen aber Install.
Summary:



Und der Installationsprozess wird gestartet.

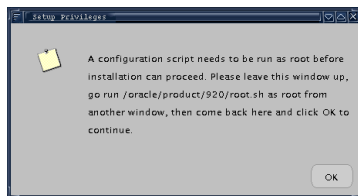
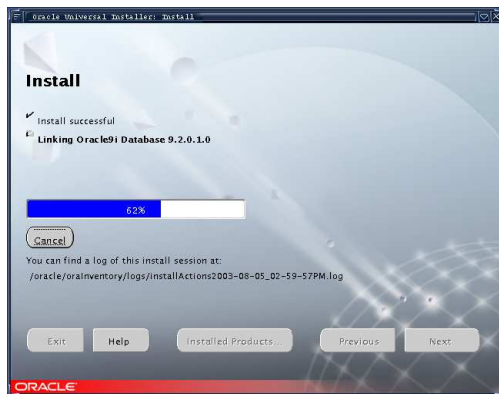
2.3.15 Install

Das Logfile der Installation wird in `/oracle/oraInventory/logs/` gespeichert, und hat einen Namen wie: `installActions"Datum-Zeit".log`
Install:

2.3.16 Setup Privileges

Während des Installationsprozesses erscheint die folgende Messagebox.
Setup Privileges:

Während der Installtion muss das Script `/oracle/product/920/root.sh` als root ausgeführt werden. Solange bleibt der Installationsprozess stehen. Da das Script `root.sh` mit absoluten Pfadangaben programmiert ist, ist es nicht sicher, ob alle Programme an dem Ort gespeichert sind, wo `root.sh` es erwartet.



Das lässt sich sehr einfach mit den folgenden Befehlen testen und korrigieren. Dazu wechseln wir in die Shell als root.

```
for ora in `grep =/*bin/ root.sh | sed 's/^.*=//`
do
    basename=`basename $ora`
    deb=`which $basename`
    test -x $name || sed "s#$ora#$deb#" -i root.sh
done
```

Awk wird vom Script im Verzeichnis /bin erwartet, befindet sich aber bei Debian in /usr/bin. Das korrigieren wir und erstellen einen Link

```
tux01:/oracle/product/920# ln -s /usr/bin/awk /bin/awk
```

Nachher kann man das Script root.sh ausführen.

```
tux01:/oracle/product/920# . root.sh
Running Oracle9 root.sh script...
The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME= /oracle/product/920
```

```
Enter the full pathname of the local bin directory: [/usr/local/bin]:
Copying dbhome to /usr/local/bin ...
Copying oraenv to /usr/local/bin ...
Copying coraenv to /usr/local/bin ...
```

Creating /etc/oratab file...
Adding entry to /etc/oratab file...
Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.

Jetzt kann man die Message Box "Setup Privileges" mit OK verlassen.

2.3.17 Configuration Assistant

Perform typical configuration wählen und Next klicken. Wie erstellen die notwendigen Dateien selber über ein Script.

Configuration Assistant:



2.3.18 End of Installation

Die Oracle Software ist installiert. Als nächstes patchen wir die Software. Click auf Next Install.

End of Installation:



2.3.19 Patch File Locations

Von Metalink hab ich Patch Set Version 9.2.0.2 und 9.2.0.3 heruntergeladen. Ich entpacke Sie in /data/patches. Einen Metalink-Account erhält man nur als Kunde mit einer kommerziellen Lizenz.

```
oracle@tux01:~$ cd /data/patches/  
oracle@tux01:/data/patches$ mkdir 9202  
oracle@tux01:/data/patches$ mkdir 9203  
oracle@tux01:/data/patches$ cd 9202  
oracle@tux01:/data/patches/9202$ tar xf ../9202_lnx32_release.tar  
oracle@tux01:/data/patches/9202$ cd ../9203  
oracle@tux01:/data/patches/9203$ tar xf ../9203_lnx32_release.tar
```

2.3.20 Patch Set Version 9.2.0.2

Für die Installation von Patch Set Version 9.2.0.2 wählt man die Datei products.jar im Verzeichnis /data/patches/9202/stage, und wählt den Installation Path.

Patch Set Version 9.2.0.2:



und clickt Next.

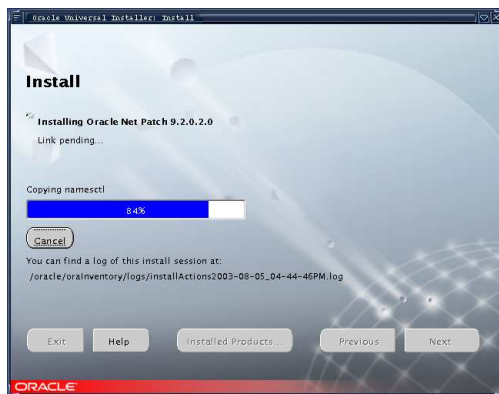
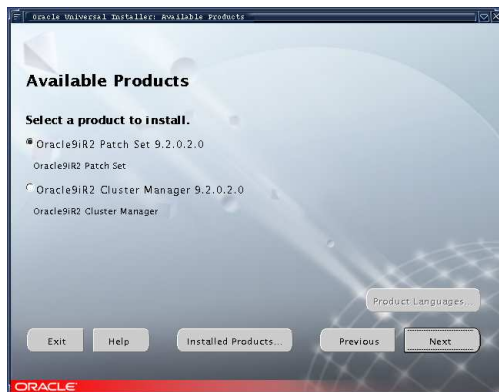
2.3.21 Available Products

Available Products:
und Next wählen.

2.3.22 Install Patch Set

Jetzt wird Patch Set Version 9.2.0.2 installiert. Nachher machen wir das Gleiche mit Patch Set Version 9.2.0.3. Wenn späten einmal ein weiteres Patch Set oder ein weiterer Patch vorhanden sein sollte, dann wird diese genau gleich installiert.

Install:



Nachher kann man mit Exit den OUI verlassen. Die Software ist installiert.

3 Basisdatenbank erzeugen

Ich erzeuge die Instance DB01. Das wird in der Shell und in SQL*Plus gemacht.

3.1 Verzeichnisse erstellen

Als User Oracle erstelle ich alle notwendigen Verzeichnisse.

```
tux01:~# su - oracle
oracle@tux01:~$ pwd
/oracle
oracle@tux01:~$ mkdir -p admin/DB01/create
oracle@tux01:~$ mkdir admin/DB01/cdump
oracle@tux01:~$ mkdir admin/DB01/adhoc
oracle@tux01:~$ mkdir admin/DB01/bdump
oracle@tux01:~$ mkdir admin/DB01/udump
```

Am Anfang erstelle ich alle Datenfiles der Instance in /oracle/oradata/DB01. Das lässt sich nachher einfach mit einigen DDL Befehlen ändern. Die Administration von Oracle ist sehr flexibel.

```
oracle@tux01:~$ mkdir -p oradata/DB01
oracle@tux01:~$ mkdir oradata/DB01/archive
```

3.2 initDB01.ora erstellen

Die Parameterdatei der Instance DB01 ist initDB01.ora. Diese Datei lässt sich mit dem cat Befehl erzeugen.

```
oracle@tux01:~$ cat >product/920/dbs/initDB01.ora <<EOF
#####
#                               Instance DB01                               #
db_name                         = DB01
instance_name                   = DB01
service_names                   = DB01
db_cache_size                   = 100M
pga_aggregate_target           = 50M
shared_pool_size                = 50M
aq_tm_processes                 = 0
log_archive_start               = FALSE
log_archive_dest                 = /oracle/oradata/DB01/archive
log_archive_format               = "T%TS%S.arc"
background_dump_dest            = /oracle/admin/DB01/bdump
user_dump_dest                  = /oracle/admin/DB01/udump
core_dump_dest                  = /oracle/admin/DB01/cdump
control_files                   = ("/oracle/oradata/DB01/control1.ctl",
                                "/oracle/oradata/DB01/control2.ctl")
oracle_trace_enable             = FALSE

#####
#                               Resources                               #
db_block_checksum               = FALSE
db_block_size                   = 8192
db_file_multiblock_read_count   = 8
db_files                        = 100
dml_locks                       = 1000
fast_start_mttr_target         = 900
java_pool_size                  = 0
max_enabled_roles               = 30
open_cursors                    = 500
parallel_automatic_tuning       = FALSE
parallel_max_servers            = 8
processes                       = 100
sga_max_size                    = 100M
undo_management                 = auto
undo_tablespace                 = undo
```

```

undo_retention          = 5400
workarea_size_policy   = auto

#####
#                      Optimizer                      #
compatible              = 9.2.0
cursor_sharing          = EXACT
cursor_space_for_time   = FALSE
optimizer_index_caching = 80
optimizer_index_cost_adj = 25
optimizer_max_permutations = 1000
optimizer_mode          = CHOOSE
timed_statistics        = TRUE

#####
#                      Logs & Dumps                    #
log_checkpoint_timeout = 10000
max_dump_file_size     = 10M

#####
#                      Misc                            #
global_names            = FALSE
os_authent_prefix       = ""
o7_dictionary_accessibility = TRUE
remote_login_passwordfile = NONE
EOF

```

3.3 Instance starten und Basisdatenbank erzeugen

Jetzt ist alles so weit vorbereitet, dass wir die Basisdatenbank erstellen können. Dazu wechseln wir nach /oracle/admin/DB01/create. Starten die Instance mit SQL*Plus und erzeugen die Basisdatenbank mit dem DDL-Befehl CREATE DATABASE.

```

oracle@tux01:~$ cd admin/DB01/create
oracle@tux01:~/admin/DB01/create$ sqlplus /nolog <<EOF
set echo on
whenever OSERROR exit 2
whenever SQLERROR exit 1

spool create1.log
set termout on
connect SYS/change_on_install as SYSDBA
startup nomount
set verify on
CREATE DATABASE DB01 CONTROLFILE REUSE
MAXINSTANCES 1
MAXLOGHISTORY 0
MAXLOGFILES 32

```

```

MAXLOGMEMBERS 4
MAXDATAFILES 1022
DATAFILE '/oracle/oradata/DB01/system01.dbf'
SIZE 200M REUSE AUTOEXTEND ON NEXT 50M MAXSIZE 2047M
EXTENT MANAGEMENT LOCAL
DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE
    '/oracle/oradata/DB01/temp01.dbf'
SIZE 200M REUSE AUTOEXTEND ON next 50M maxsize 2047M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 2M
UNDO TABLESPACE "UNDO" DATAFILE '/oracle/oradata/DB01/undo01'
    SIZE 200M REUSE AUTOEXTEND ON NEXT 50M MAXSIZE 2047M
CHARACTER SET WE8ISO8859P1
NATIONAL CHARACTER SET AL16UTF16
LOGFILE
GROUP 1 ('/oracle/oradata/DB01/redolog1a.rdo',
        '/oracle/oradata/DB01/redolog1b.rdo') SIZE 50M REUSE,
GROUP 2 ('/oracle/oradata/DB01/redolog2a.rdo',
        '/oracle/oradata/DB01/redolog2b.rdo') SIZE 50M REUSE,
GROUP 3 ('/oracle/oradata/DB01/redolog3a.rdo',
        '/oracle/oradata/DB01/redolog3b.rdo') SIZE 50M REUSE,
GROUP 4 ('/oracle/oradata/DB01/redolog4a.rdo',
        '/oracle/oradata/DB01/redolog4b.rdo') SIZE 50M REUSE;
spool off
spool create2.log
connect SYS/change_on_install as SYSDBA
whenever SQLERROR continue
@/oracle/product/920/rdbms/admin/catalog.sql;
@/oracle/product/920/rdbms/admin/catexp7.sql;
@/oracle/product/920/rdbms/admin/catblock.sql;
@/oracle/product/920/rdbms/admin/catproc.sql;
@/oracle/product/920/rdbms/admin/catoctk.sql;
@/oracle/product/920/rdbms/admin/catobtk.sql;
@/oracle/product/920/rdbms/admin/caths.sql;
@/oracle/product/920/rdbms/admin/owminst.plb;
connect SYSTEM/manager
@/oracle/product/920/sqlplus/admin/pupbld.sql;
@/oracle/product/920/sqlplus/admin/help/hlpbld.sql helpus.sql;
spool off
set echo on
whenever SQLERROR exit 1
spool create3.log
connect SYS/change_on_install as SYSDBA
shutdown immediate;
EOF

```

3.4 Basisdatenbank mit Script erstellen

Dieses Script von Martin Furter erstellt eine Basisdatenbank, aber mit sehr kleinem SYSTEM Tablespace, denn Applikations- und Benutzerdaten gehören

immer in eigene Tablespaces.

```
#!/bin/sh
#=====
#
# Copyright (C) 2003-2004 Martin Furter <mf@borg.ch>
#
# This file is part of DebOra9Howto
# (Oracle9i Installation auf Debian GNU/Linux 3.0)
#
# DebOra9Howto is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# DebOra9Howto is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with DebOra9Howto; see the file COPYING. If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#=====
#
# createOra9DB.sh
#
# A script for creating an oracle instance.
# Use 'createOra9DB.sh -h' for help.
#
#=====

ORACLE_VERSION='sqlplus '-?' | awk '/Release/{print $3}'

case "$ORACLE_VERSION" in
  9.*)
    ORACLE_VERSION=9
    ;;
  10.*)
    ORACLE_VERSION=10
    ;;
  *)
    echo "unsupported oracle version $ORACLE_VERSION"
    exit 1
    ;;
esac

help() {
```



```

echo ""
echo "Creates a new oracle database"
echo ""
echo "  usage: $0 [options]"
echo ""
echo "  options are:"
echo "    BASE=/path      oracle base directory  (env: ORACLE_BASE)"
echo "    DATA=/path     oracle data directory  (env: ORACLE_DATA)"
echo "    HOME=/path      oracle home directory  (env: ORACLE_HOME)"
echo "    SID=name        oracle SID              (env: ORACLE_SID)"
echo ""
echo "  current settings:"
if [ -n "$ORACLE_BASE" ]; then
    echo "    BASE='$ORACLE_BASE'"
fi
if [ -n "$ORACLE_DATA" ]; then
    echo "    DATA='$ORACLE_DATA'"
fi
if [ -n "$ORACLE_HOME" ]; then
    echo "    HOME='$ORACLE_HOME'"
fi
if [ -n "$ORACLE_SID" ]; then
    echo "    SID='$ORACLE_SID'"
fi
echo ""
}

while [ -n "$1" ]; do
    case "$1" in
        BASE=*)
            ORACLE_BASE='echo "$1"|sed 's/^BASE=//''
            export ORACLE_BASE
            ;;
        DATA=*)
            ORACLE_DATA='echo "$1"|sed 's/^DATA=//''
            export ORACLE_DATA
            ;;
        HOME=*)
            ORACLE_HOME='echo "$1"|sed 's/^HOME=//''
            export ORACLE_HOME
            ;;
        SID=*)
            ORACLE_SID='echo "$1"|sed 's/^SID=//''
            export ORACLE_SID
            ;;
        -h*)
            help
            exit 0
            ;;
    esac
done

```

```

        --h*)
            help
            exit 0
            ;;
        *)
            echo "unknown parameter '$1'"
            exit 1
            ;;
    esac
    shift
done

# try to guess some variables
if [ -n "$ORACLE_BASE" ]; then
    ORACLE_ADMIN="$ORACLE_BASE/admin"
fi
if [ -n "$ORACLE_HOME" ]; then
    ORACLE_DBS="$ORACLE_HOME/dbs"
fi

# check needed variables
OK=Y
for VAR in ORACLE_SID ORACLE_HOME ORACLE_ADMIN ORACLE_DATA; do
    eval VAL="\${$VAR}"
    if [ -n "$VAL" ]; then
        export $VAR
    else
        echo "$VAR is not set!"
        OK=N
    fi
done

# check needed directories
if [ ! -d "$ORACLE_ADMIN" ]; then
    echo "admin dir '$ORACLE_ADMIN' doesn't exist"
    OK=N
fi
if [ ! -d "$ORACLE_HOME" ]; then
    echo "orahome dir '$ORACLE_HOME' doesn't exist"
    OK=N
fi
if [ ! -d "$ORACLE_DBS" ]; then
    echo "ora dbs dir '$ORACLE_DBS' doesn't exist"
    OK=N
fi
if [ ! -d "$ORACLE_DATA" ]; then
    echo "oradata dir '$ORACLE_DATA' doesn't exist"
    OK=N
fi
# OK ?

```

```

if [ "$OK" != "Y" ]; then
    echo "precondition failed, exiting."
    exit 1
fi

# setup variables with SID
SID_ADMIN="$ORACLE_ADMIN/$ORACLE_SID"
SID_DATA="$ORACLE_DATA/$ORACLE_SID"
SID_INITORA="$ORACLE_DBS/init$ORACLE_SID.ora"
#SID_ORAPW="$ORACLE_DBS/orapw$ORACLE_SID"

# check SID variables
if [ -d "$SID_ADMIN" ]; then
    echo "$SID_ADMIN already exists!"
    OK=N
fi
if [ -d "$SID_DATA" ]; then
    echo "$SID_DATA already exists!"
    OK=N
fi
if [ -f "$SID_INITORA" ]; then
    echo "$SID_INITORA already exists!"
    OK=N
fi
# OK ?
if [ "$OK" != "Y" ]; then
    echo "SID $ORACLE_SID already exists, exiting."
    exit 1
fi

echo "ORACLE_VERSION = '$ORACLE_VERSION'"
echo "ORACLE_SID      = '$ORACLE_SID'"
echo "ORACLE_HOME     = '$ORACLE_HOME'"
echo "ORACLE_DBS      = '$ORACLE_DBS'"
echo "ORACLE_ADMIN    = '$ORACLE_ADMIN'"
echo "ORACLE_DATA     = '$ORACLE_DATA'"
echo "SID_ADMIN       = '$SID_ADMIN'"
echo "SID_DATA        = '$SID_DATA'"
echo "SID_INITORA     = '$SID_INITORA'"
#echo "SID_ORAPW      = '$SID_ORAPW'"

# ask user if he's happy with these parameters
echo ""
OK=""
while [ "$OK" != "yes" ]; do
    echo 'create the database ? (yes|no)'
    read OK
    if [ "$OK" = "no" ]; then
        echo "aborting."
    fi
done

```

```

        exit 0
    fi
done

# create directories
mkdir $SID_ADMIN
mkdir $SID_ADMIN/create
mkdir $SID_ADMIN/cdump
mkdir $SID_ADMIN/adhoc
mkdir $SID_ADMIN/bdump
mkdir $SID_ADMIN/udump
mkdir $SID_DATA
mkdir $SID_DATA/archive

# set some variables depending on the oracle version
case "$ORACLE_VERSION" in
    9)
        ORA10SYSaux=""
        09=""
        ;;
    10)
        ORA10SYSaux=""
        SYSaux DATAFILE '$SID_DATA/sysaux01.dbf'
        SIZE 60M REUSE AUTOEXTEND ON NEXT 20M MAXSIZE 400M"
        09="# (ora9) "
        ;;
    *)
        ORA10SYSaux=""
        09=""
        exit 1
        ;;
esac

# create init.ora
cat > $SID_INITORA << EOF
#####
#                               Instance $ORACLE_SID                               #
db_name                         = $ORACLE_SID
instance_name                   = $ORACLE_SID
service_names                   = $ORACLE_SID
db_cache_size                   = 150M
pga_aggregate_target            = 50M
shared_pool_size                = 50M
aq_tm_processes                 = 0
${09}log_archive_start          = FALSE
log_archive_dest                 = $SID_DATA/archive
log_archive_format               = "T%TS%S.arc"
background_dump_dest            = $SID_ADMIN/bdump
user_dump_dest                   = $SID_ADMIN/udump

```

```

core_dump_dest          = $SID_ADMIN/cdump
control_files           = ( "$SID_DATA/control1.ctl",
                           "$SID_DATA/control2.ctl" )
${09}oracle_trace_enable = FALSE

#####
#                      Resources                      #
db_block_checksum      = FALSE
db_block_size          = 8192
db_file_multiblock_read_count = 8
db_files               = 100
dml_locks              = 1000
fast_start_mttr_target = 900
java_pool_size         = 0
${09}max_enabled_roles = 30
open_cursors           = 500
${09}parallel_automatic_tuning = FALSE
parallel_max_servers   = 8
processes              = 100
sga_max_size           = 150M
undo_management        = auto
undo_tablespace        = undo
undo_retention         = 5400
workarea_size_policy   = auto

#####
#                      Optimizer                      #
compatible             = 9.2.0
cursor_sharing         = EXACT
cursor_space_for_time  = FALSE
optimizer_index_caching = 80
optimizer_index_cost_adj = 25
${09}optimizer_max_permutations = 1000
optimizer_mode         = CHOOSE
timed_statistics       = TRUE

#####
#                      Logs & Dumps                    #
log_checkpoint_timeout = 10000
max_dump_file_size     = 10M

#####
#                      Misc                            #
global_names           = FALSE
os_authent_prefix      = ""
o7_dictionary_accessibility = TRUE
remote_login_passwordfile = NONE
EOF

```

```

# create orapw file
#rm -f $SID_ORAPW
#orapwd "file=$SID_ORAPW" password=oracle entries=10

# create create-script
cat > $SID_ADMIN/create/create.sh << CREATEEOF
#!/bin/sh

sqlplus /nolog << EOF
set echo on
set define off
whenever OSERROR exit 2
whenever SQLERROR exit 1
spool create1.log

prompt STEP: create database
set termout on
connect SYS/change_on_install as SYSDBA
startup nomount
set verify on
CREATE DATABASE $ORACLE_SID CONTROLFILE REUSE
MAXINSTANCES 1
MAXLOGHISTORY 0
MAXLOGFILES 32
MAXLOGMEMBERS 4
MAXDATAFILES 1022
DATAFILE '$SID_DATA/system01.dbf'
    SIZE 150M REUSE AUTOEXTEND ON NEXT 20M MAXSIZE 250M
    EXTENT MANAGEMENT LOCAL $ORA10SYSAUX
DEFAULT TEMPORARY TABLESPACE TEMP TEMPFILE
    '$SID_DATA/temp01.dbf'
    SIZE 20M REUSE AUTOEXTEND ON next 10M maxsize 50M
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 2M
UNDO TABLESPACE "UNDO" DATAFILE '$SID_DATA/undo01'
    SIZE 20M REUSE AUTOEXTEND ON NEXT 10M MAXSIZE 50M
CHARACTER SET WE8ISO8859P1
NATIONAL CHARACTER SET AL16UTF16
LOGFILE
    GROUP 1 ('$SID_DATA/redolog1a.rdo',
            '$SID_DATA/redolog1b.rdo') SIZE 50M REUSE,
    GROUP 2 ('$SID_DATA/redolog2a.rdo',
            '$SID_DATA/redolog2b.rdo') SIZE 50M REUSE,
    GROUP 3 ('$SID_DATA/redolog3a.rdo',
            '$SID_DATA/redolog3b.rdo') SIZE 50M REUSE,
    GROUP 4 ('$SID_DATA/redolog4a.rdo',
            '$SID_DATA/redolog4b.rdo') SIZE 50M REUSE;
prompt STEP: done.
spool off

```

```

spool create2.log
connect SYS/change_on_install as SYSDBA
whenever SQLERROR continue

prompt STEP: creates data dictionary views
@$ORACLE_HOME/rdbms/admin/catalog.sql;
prompt STEP: done.

prompt STEP: create catalog v7 style export/import views
@$ORACLE_HOME/rdbms/admin/catexp7.sql;
prompt STEP: done.

prompt STEP: create catalog oracle locks
@$ORACLE_HOME/rdbms/admin/catblock.sql;
prompt STEP: done.

prompt STEP: create catalog procedural option
@$ORACLE_HOME/rdbms/admin/catproc.sql;
prompt STEP: done.

prompt STEP: create catalog oracle cryptographic toolkit
@$ORACLE_HOME/rdbms/admin/catoctk.sql;
prompt STEP: done.

prompt STEP: create catalog PL/SQL obfuscation toolkit
@$ORACLE_HOME/rdbms/admin/catobtk.sql;
prompt STEP: done.

prompt STEP: create catalog heterogeneous herVICES
@$ORACLE_HOME/rdbms/admin/cathS.sql;
prompt STEP: done.

prompt STEP: create user ovm_sys for storing metadata
@$ORACLE_HOME/rdbms/admin/owminst.plb;
prompt STEP: done.
spool off

connect SYSTEM/manager
spool create3.log

prompt STEP: install sql*plus product_user_profile tables
@$ORACLE_HOME/sqlplus/admin/pupbld.sql;
prompt STEP: done.

prompt STEP: build sql*plus help table and load English language help test
@$ORACLE_HOME/sqlplus/admin/help/hlpbld.sql helpus.sql;
prompt STEP: done.
spool off

```

```

set echo on
whenever SQLERROR exit 1
spool create4.log
prompt STEP: shutdown
connect SYS/change_on_install as SYSDBA
shutdown immediate;
prompt STEP: done.
spool off
EOF

```

```

CREATEEOF
chmod 755 $SID_ADMIN/create/create.sh

```

```

# now create the db
cd $SID_ADMIN/create
./create.sh

```

```

# and some info...

```

```

echo "=====
echo ""
echo "dont't forget to add the new SID to oratab, listener.ora and tnsnames.ora"

```

3.5 Basisdatenbank erstellen gescheitert

Sollte etwas mit beim Erzeugen der Basisdatenbank nicht funktionieren und die Oracle Prozesse der Instance sind nicht beendet, dann kann man diese beenden, indem man sich als sysdba anmeldet und mit dem Befehl shutdown die Instance herunterfährt.

```

oracle@tux01:~$ sqlplus /nolog <<EOF
conn / as sysdba
shutdown abort;
EOF

```

Wurden schon Datafiles erzeugt, so müssen die gelöscht werden.

```

oracle@tux01:~$ find /oracle/oradata/DB01/ -type f -exec rm {} \;

```

Ist die Installation gescheitert, und die Oracle Prozesse wurden nicht richtig beendet, zum Beispiel mit kill -9, dann wird der belegte Speicher in der Shared Memory und Semaphore Arrays nicht mehr frei gegeben. Um dies Speicherbereiche zu löschen, gibt es den Befehl ipcrm und zum Anzeigen ipcs. Solange dies Speicherbereiche nicht gelöscht sind, lässt sich Oracle nicht starten und gibt die Fehlermeldung:

```

ORA-01081: cannot start already-running ORACLE - shut it down first aus.

```

Hier wird kurz demonstriert, wie man diese Speicherbereiche löscht.


```

oracle@tux01:~$ ipcs

----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status
0x00000000  524288    oracle     640        4194304    0
0x00000000  557057    oracle     640        50331648   0
0x00000000  589826    oracle     640        83886080   0
0x2a0549d4  622595    oracle     640        71303168   0

----- Semaphore Arrays -----
key          semid      owner      perms      nsems      status
0x9ac10ec0  622592    oracle     640        104

----- Message Queues -----
key          msqid      owner      perms      used-bytes  messages

oracle@tux01:~$ for SHMID in `ipcs -m |grep oracle|awk '{print $2}'`;
do ipcrm shm $SHMID;
done
resource(s) deleted
resource(s) deleted
resource(s) deleted
resource(s) deleted

oracle@tux01:~$ for SEMID in `ipcs -s |grep oracle|awk '{print $2}'`;
do ipcrm sem $SEMID;
done
resource(s) deleted

oracle@tux01:~/admin/DB01/create$ ipcs

----- Shared Memory Segments -----
key          shmid      owner      perms      bytes      nattch     status

----- Semaphore Arrays -----
key          semid      owner      perms      nsems      status

----- Message Queues -----
key          msqid      owner      perms      used-bytes  messages

```

4 Tablespaces und Benutzer

4.1 Passwort von SYS und SYSTEM ändern

Nach dem Erzeugen der Basisdatenbank sollten die Standardpassworte geändert werden.

```

oracle@tux01:~$ sqlplus sys/change_on_install << EOF
ALTER USER SYS IDENTIFIED BY "sys-passwort";

```

```
ALTER USER SYSTEM IDENTIFIED BY "system-passwort";
EOF
```

4.2 Tablespaces erstellen

Applikations- und Benutzerdaten sollten immer in eigenen Tablespaces gespeichert werden. Darum erstelle ich jetzt einen Tablespace für meine Daten. Hier übernimmt Oracle den grossen Teile der Tablespaceverwaltung:

```
oracle@tux01:~$ sqlplus system/system-passwort << EOF
CREATE TABLESPACE USERDATA
  LOGGING
  DATAFILE '/oracle/oradata/DB01/USERDATA01.dbf'
  SIZE 256M REUSE
  AUTOEXTEND ON NEXT 64M MAXSIZE 1024M EXTENT MANAGEMENT LOCAL
  SEGMENT SPACE MANAGEMENT AUTO
EOF
```

Wenn man selber mehr steuern möchte oder einen Tablespace für LOBs benötigt, kann dies sinnvoll sein:

```
oracle@tux01:~$ sqlplus system/system-passwort << EOF
CREATE TABLESPACE USERDATA
  DATAFILE '/oracle/oradata/DB01/USERDATA01.dbf'
  SIZE 256M REUSE
  AUTOEXTEND ON NEXT 64M MAXSIZE 1024M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 512K;
EOF
```

Hier ist zu beachten, dass die UNIFORM SIZE passend gewählt wird. Platz für Tabellen wird nur in ganzen Vielfachen der UNIFORM SIZE alloziert. Das heisst jede Tabelle im Tablespace USERDATA braucht mindestens 512kB. Werden im Tablespace viele kleine Tabellen erzeugt, so muss dieser Wert kleiner gewählt werden, werden wenige grosse Tabellen erzeugt wählt man die UNIFORM SIZE grösser.

4.3 Benutzer erstellen

Da man bei einer Datenbank genauso wie bei einem Betriebssystem möglichst nie mit den Administratoren-Accounts arbeiten sollte, erstelle ich jetzt die benötigten Benutzer-Accounts.

```
oracle@tux01:~$ sqlplus system/system-passwort << EOF
CREATE USER URS IDENTIFIED BY "urs-passwort"
  DEFAULT TABLESPACE USERDATA
  TEMPORARY TABLESPACE TEMP
EOF
```

```

    QUOTA UNLIMITED ON USERDATA
    QUOTA UNLIMITED ON TEMP;
GRANT CONNECT TO URS;
CREATE USER MARTIN IDENTIFIED BY "martin-passwort"
    DEFAULT TABLESPACE USERDATA
    TEMPORARY TABLESPACE TEMP
    QUOTA UNLIMITED ON USERDATA
    QUOTA UNLIMITED ON TEMP;
GRANT CONNECT TO MARTIN;
EOF

```

5 Oracle Net

5.1 sqlnet.ora

sqlnet.ora ist Konfigurationsdatei für den Client oder Server. In sqlnet.ora wird die Client-Domäne gespeichert. Bei unvollständigen Dienstnamen oder Net Service Names wird diese angänhgt. Zusätzlich wird die Reihenfolge der Benennungsmethoden, die der Client bei der Auflösung eines Namens benutzen soll gespeichert.

```

oracle@tux01:~$ cat > /oracle/product/920/network/admin/sqlnet.ora <<EOF
TRACE_LEVEL_CLIENT = off
log_file_client=sqlnet.trc
NAME.DEFAULT_ZONE = world
NAMES.DEFAULT_DOMAIN = world
NAMES.DIRECTORY_PATH= (TNSNAMES)
EOF

```

```

cat > /oracle/product/920/network/admin/tnsnames.ora <<EOF
DB01.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = DB01)
      (SERVER = DEDICATED)
    )
  )
)
EOF

```

5.2 listener.ora

listener.ora enthält Listener-Name und Protokolladressen über die Verbindungsanforderungen akzeptiert werden und Dienste (Datenbank- und Nicht-Datenbank-Dienste), auf die gehört wird

```

oracle@tux01:~$ cat > /oracle/product/920/network/admin/listener.ora <<EOF
LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC))
      )
      (ADDRESS_LIST =
        (ADDRESS = (PROTOCOL = TCP)(HOST = localhost )(PORT = 1521))
      )
    )
  )
)

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /oracle/product/920)
      (PROGRAM = extproc)
    )
  )
)
EOF

```

Listener starten

```
oracle@tux01:~$ lsnrctl start
```

```
LSNRCTL for Linux: Version 9.2.0.3.0 - Production on 27-AUG-2003 19:34:11
```

```
Copyright (c) 1991, 2002, Oracle Corporation. All rights reserved.
```

```
Starting /oracle/product/920/bin/tnslsnr: please wait...
```

```
TNSLSNR for Linux: Version 9.2.0.3.0 - Production
```

```
System parameter file is /oracle/product/920/network/admin/listener.ora
```

```
Log messages written to /oracle/product/920/network/log/listener.log
```

```
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))
```

```
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=tux01)(PORT=1521)))
```

```
Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
```

```
STATUS of the LISTENER
```

```
-----
```

Alias	LISTENER
Version	TNSLSNR for Linux: Version 9.2.0.3.0 - Production
Start Date	27-AUG-2003 19:34:11
Uptime	0 days 0 hr. 0 min. 0 sec
Trace Level	off
Security	OFF
SNMP	OFF
Listener Parameter File	/oracle/product/920/network/admin/listener.ora

```

Listener Log File          /oracle/product/920/network/log/listener.log
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=tux01)(PORT=1521)))
Services Summary...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

```

Listener testen

```
oracle@tux01:~$ tnsping DB01
```

```

TNS Ping Utility for Linux: Version 9.2.0.3.0
- Production on 27-AUG-2003 19:36:52

```

Copyright (c) 1997 Oracle Corporation. All rights reserved.

```

Used parameter files:
/oracle/product/920/network/admin/sqlnet.ora

```

```

Used TNSNAMES adapter to resolve the alias
Attempting to contact (DESCRIPTION = (ADDRESS_LIST =
  (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521)))
(CONNECT_DATA = (SID = DB01) (SERVER = DEDICATED)))
OK (0 msec)

```

5.3 tnsnames.ora

Die Datei tnsnames.ora enthält die Net Service Names, welchen Verbindungsdeskriptoren zugeordnet sind.

```

oracle@tux01:~$ cat >> /oracle/product/920/network/admin/tnsnames.ora <<EOF
DB01.WORLD =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = TCP)(HOST = localhost)(PORT = 1521))
    )
    (CONNECT_DATA =
      (SID = DB01)
      (SERVER = DEDICATED)
    )
  )
)
EOF

```

6 Instance mit Runlevel automatisch starten

Wir lassen die Instance automatisch über die RTnlevel von Linux starten und stoppen.

6.1 dbstart und dbshut

Die Oracle Scripte dbstart und dbshut starten und beenden die Instance. Dazu muss ein Eintrag in `/etc/oratab` ergänzt werden.

```
oracle@tux01:~$ vi /etc/oratab
DB01:/oracle/product/920:Y
```

Jetzt kann man dbstart und dbshut benutzen.

```
oracle@tux01:~$ dbstart
```

```
SQL*Plus: Release 9.2.0.3.0 - Production on Tue Aug 26 18:31:59 2003
```

```
Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.
```

```
SQL> Connected to an idle instance.
```

```
SQL> ORACLE instance started.
```

```
Total System Global Area 202445436 bytes
Fixed Size                  451196 bytes
Variable Size               83886080 bytes
Database Buffers           117440512 bytes
Redo Buffers                667648 bytes
Database mounted.
Database opened.
```

6.2 /etc/init.d/listener

Hier folgt das Script für den Listener, welches vom Init Prozess ausgeführt wird.

(Muss verbessert werden)

```
tux01:~# cat >/etc/init.d/listener <<EOF
#!/bin/sh
# Start|Stop Oracle Listener
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

case "$1" in
  start)
    su - oracle -c /oracle/product/920/bin/lsnrctl start
    ;;
  stop)
    su - oracle -c /oracle/product/920/bin/lsnrctl stop
```

```

        ;;
restart|force-reload)
    su - oracle -c /oracle/product/920/bin/lsnrctl stop
    sleep 1
    su - oracle -c /oracle/product/920/bin/lsnrctl start
    ;;
*)
    echo "Usage: /etc/init.d/listener {start|stop|restart|force-reload}"
    exit 1
    ;;
esac

exit 0
EOF

```

6.3 /etc/init.d/oracleDB01

Hier folgt das Script um die Oracle Instance DB01 zu starten.
(Muss verbessert werden)

```

tux01:~# cat > /etc/init.d/oracleDB01 <<EOF
#!/bin/sh
# Start|Stop Oracle Instance DB01

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
NAME=DB01
DESC="Oracle Instance"

case "$1" in
    start)
        su - oracle -c dbstart
        ;;
    stop)
        su - oracle -c dbshut
        ;;
    restart|force-reload)
        su - oracle -c dbshut
        sleep 5
        su - oracle -c dbstart
        ;;
    *)
        echo "Usage: /etc/init.d/oracleDB01 {start|stop|restart|force-reload}" >&2
        exit 1
        ;;
esac

exit 0
EOF

```

6.4 Runlevel Einträge

Bei Debian kann man die Runlevels über das Programm `update-rc.d` Eintragen.

```
tux01:~# update-rc.d listener start 90 2 . stop 10 0 1 2 3 4 5 6 S .
Adding system startup for /etc/init.d/listener ...
/etc/rc0.d/K10listener -> ../init.d/listener
/etc/rc1.d/K10listener -> ../init.d/listener
/etc/rc2.d/K10listener -> ../init.d/listener
/etc/rc3.d/K10listener -> ../init.d/listener
/etc/rc4.d/K10listener -> ../init.d/listener
/etc/rc5.d/K10listener -> ../init.d/listener
/etc/rc6.d/K10listener -> ../init.d/listener
/etc/rcS.d/K10listener -> ../init.d/listener
/etc/rc2.d/S90listener -> ../init.d/listener

tux01:~# update-rc.d oracleDB01 start 91 2 . stop 09 0 1 2 3 4 5 6 S .
Adding system startup for /etc/init.d/oracleDB01 ...
/etc/rc0.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc1.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc2.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc3.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc4.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc5.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc6.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rcS.d/K09oracleDB01 -> ../init.d/oracleDB01
/etc/rc2.d/S91oracleDB01 -> ../init.d/oracleDB01
```

7 Weiteres, Tips und Tricks

Ich bin froh um weitere Tips und nehme sie gerne in diesem oder anderen Abschnitten auf.

7.1 Java anpassen für oemapp, dbastudio

Mit den von Oracle mitgelieferten Javaversionen 1.3.1 und 1.1.8 kann es Probleme mit den Schriften geben:

```
oemapp dbastudio
Font specified in font.properties not found [--symbol-medium-r-normal...
...
```

Die mitgelieferten Javaprogramme lassen sich zwar starten, aber das GUI wird nur mit fehlenden Ausschnitten dargestellt. Dies lässt sich mit einer aktuellen Javaversion von SUN beheben. Hier zeige ich wie man unter Debian ein `j2sdk-1.4.2` installiert. Dazu gibt es aber auch ein Debian Howto. Als root:


```

mkdir /usr/src/j2sdk1_4_2_04
cp j2sdk-1_4_2_04-linux-i586.bin /usr/src/j2sdk1_4_2_04
cd /usr/local/lib
/usr/src/j2sdk1_4_2_04/j2sdk-1_4_2_04-linux-i586.bin
ln -s j2sdk1.4.2_04 jdk
cd /usr/src/j2sdk1_4_2_04
mkdir pkg
cd pkg
apt-get install java-common equivs
/usr/share/doc/java-common/dummy-packages/*.control .
equivs-build java-compiler-dummy.control
equivs-build java-virtual-machine-dummy.control
equivs-build java1-runtime-dummy.control
equivs-build java2-compiler-dummy.control
equivs-build java2-runtime-dummy.control
dpkg -i java-compiler-dummy_1.0_all.deb \
        java1-runtime-dummy_1.0_all.deb \
        java2-runtime-dummy_1.0_all.deb \
        java-virtual-machine-dummy_1.0_all.deb \
        java2-compiler-dummy_1.0_all.deb
update-alternatives --verbose --install /usr/bin/java java \
        /usr/local/lib/jdk/bin/java 500 --slave /usr/share/man/man1/java.1 java.1 \
        /usr/local/lib/jdk/man/man1/java.1

```

Testen:

```

java -version
java version "1.4.2_04"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_04-b05)
Java HotSpot(TM) Client VM (build 1.4.2_04-b05, mixed mode)

```

Jetzt sollten die Oracle Javaprogramme ohne Probleme funktionieren.