

11 Linux-Server für Windows-Anwendungen

Windows-Anwendungen gelten für viele Anwender und Entscheider als so populär, dass sie sich auch für Windows-Betriebssysteme als Arbeitsumgebung entscheiden. Für diese Leser stellt dieses Buch dar, wie Linux-Server einer Gruppe von Windows-Clients wichtige Serverdienste zur Verfügung stellen können. Doch ist diese Kopplung von Windows-Anwendungen und Arbeitsplatz-Betriebssystem nicht so zwangsläufig. Auch Linux-Clients können durchaus Windows-Anwendungen darstellen. Es gibt bereits mehrere kommerzielle und freie arbeitsplatzbasierte und serverbasierte Lösungsansätze, um Windows-Anwendungen auf Linux-Clients darzustellen. Ob man einen dieser Lösungsansätze wählt, dürfte u.a. von der Anzahl der Clients, welche Windows-Applikationen nutzen möchten, der Nutzungsintensität und Nutzungsdauer abhängen.

11.1 Windows-Emulatoren am Linux-Arbeitsplatz

Möchte man eine Windows-Anwendung auf einem Linux-Arbeitsplatz ablaufen lassen,

- so kann man entweder ein Windows und ein Linux voneinander unabhängig auf der Festplatte installieren und für die Windows-Anwendung gegebenenfalls dann Windows anstatt Linux booten
- oder durch Emulatoren eine Umgebung einrichten, in der Windows auf Linux läuft.

So kann man eine komplette Windows-Umgebung auf einem Linux-Betriebssystem installieren. Beide Betriebssysteme laufen dann gleichzeitig, man kann Fenster beider Betriebssysteme gleichzeitig öffnen und über eine gemeinsame Zwischenablage Daten austauschen. Solche Emulatoren erfordern mehr PC-Ressourcen, da ja zwei Betriebssysteme gleichzeitig laufen. Bei kommerzieller Software erfordert das zusätzliche Lizenzen sowie administrativen Aufwand zum Verwalten beider Betriebssystem-Umgebungen. Dafür benötigt man keine zusätzlichen Server oder eine Netzwerkinfrastruktur.

Beispiele für Windows-Emulatoren sind

- VMware 3.0 (www.vmware.de/),
- NeTraverse Win4Lin 3.0 (www.netraverse.com).
- WineX <http://www.winehq.com/>
- Bochs <http://bochs.sourceforge.net/>

11.2 Applikations-Server

Statt Windows-Anwendungen auf einzelnen Linux-Clients auszuführen, kann man hierfür eigene Applikations-Server mit Server-Betriebssystemen verwenden. Die Kommunikation zwischen Applikations-Server und Client erfolgt nun über ein Netzwerkprotokoll. So wie man gewohnt ist, von einem Windows-PC mit einem PC X-Server wie Hummingbird Exceed oder Vision Eclipse über den X11-Protokollstandard auf Unix-Anwendungen zuzugreifen, kann man von Linux-Clients aus Windows Anwendungen auf Windows-Applikations-Servern nutzen.

Diese Windows-Applikations-Server können

- Linux-Server mit bis zu 100 Windows 9x-Umgebungen (NeTraverse-Server) oder
- Microsoft Windows 2000 (Advanced) Server als Terminal-Server sein.

Während bei der NeTraverse-Lösung die Linux-Clients über den X11-Protokollstandard per Display-Umlenkung auf je ein Windows 9x auf dem Linux-Server zugreifen, verwendet Microsoft sein proprietäres Remote Desktop Protokoll (RDP) für die Verbindung der Clients zum Terminal-Server. Microsoft bietet nur Clients für seine eigenen Windows-Betriebssysteme. Das Open-source-rdesktop-Projekt entwickelt und pflegt einen RDP-Client für Linux und der Anbieter HOB (www.hob.de) RDP-Clients für fast alle denkbaren Client-Plattformen einschließlich Linux. Tarantella (www.tarantella.com) und Citrix (www.citrix.de) setzen hingegen auf serverbasierte Lösungen: Bei Tarantella vermittelt ein Tarantella-Server, der auf einem Unix- oder Linux-Server läuft, zwischen dem RDP-Protokoll und einem eigenen Adaptive Internet Protocol (AIP) für die Kommunikation mit einer Vielzahl von Clients, u.a. Linux-Clients, und bei Citrix ergänzt ein eigener Server auf dem Microsoft Windows 2000 Server dessen Terminal-Server-Funktionalität und kommuniziert mit einer Vielzahl von Clients, u.a. Linux-Clients, über seine ebenfalls proprietäre Independent Computing Architecture (ICA).

11.3 Überblick

- Im kurzen Abschnitt VMWare können Sie das Konzept und die Installation von VMWare sowie Windows 2000 Professional auf VMWare nachvollziehen.
- Der Abschnitt Tarantella geht auf das Konzept von Tarantella, die Installation, das Einrichten und Verwalten von Benutzern und Anwendungen und Fragen zum Drucken ein.

11.4 VMWare

11.4.1 Konzept von VMWare: Windows 2000 Professional in der Linux-Box

Damit Anwender gelegentlich Windows-Anwendungen nutzen können, reicht es oft aus, eine Windows-Version in einer VMWare-Box auf einem Linux-Server zu installieren. Ist das geschehen, können Anwender von X11-Terminals oder direkt vom Linux-Server aus Anwendungen für Microsoft Windows nutzen.

VMWare gibt es mittlerweile in zahlreichen Version für Workstation und Serverbetrieb. Für den zuvor beschriebenen Anwendungsfall der Darstellung von Windowsapplikationen auf Linux-Systemen unterscheidet VMWare zwischen den beiden Versionen VMWare 3.0 Workstation und der Version VMWare Guest OS Kits. Während man bei der Workstation-Version sämtliche Parameter (VMWare, Container, Windows) von Hand konfiguriert, geschieht dies bei der »Guest OS Version« automatisiert. Auch kann man die benötigte Windows-Lizenz in dieser Version gebündelt von VMWare oder einem VMWare-Distributor beziehen.

Verfolgen Sie im weiteren Text das Installieren von VMWare Workstation Version 3.0 auf einem SuSE 7.3 Professional mit Windows 2000 Professional als Gastbetriebssystem.

Das Installieren und Konfigurieren ist hier in drei Schritten beschrieben:

- Zunächst gilt es, VMWare zu installieren,
- dann dieses als Container für Windows zu konfigurieren und
- schließlich eine Windows-Version, hier im Beispiel Windows 2000, einzurichten.

Die aktuelle .tar oder .rpm-Version 3.0 von VMWare können Sie über www.vmware.de beziehen. In der SuSE-Distribution 7.3 finden Sie im Menüpunkt *Kommerzielle Software* die Version 2.04.

11.4.2 VMWare installieren

Im folgenden Beispiel wird die VMWare Workstation 3.0 als `.tar` installiert.

Dekomprimieren Sie zunächst die Installationsdatei und wechseln Sie dann in das Verzeichnis `vmware-distrib`.

```

boss:/home/software# gunzip VMwareWorkstation-3.0.0-1455.tar.gz
boss:/home/ software # tar -xf VMwareWorkstation-3.0.0-1455.tar
boss:/home/ software # cd vmware-distrib/
boss:/home/ software /vmware-distrib # ls
. .. FILES INSTALL bin doc etc install.pl installer lib man
vmware-install.pl
boss:/home/carstent/vmware-distrib #

```

Durch Aufruf der Datei `vmware-install.pl` rufen Sie die eigentliche, weitestgehend automatisierte Installation von VMWare auf. Wenn Sie andere Angaben als die in Klammern gesetzten default-Werte verwenden möchten, so können Sie diese durch einfache Eingabe nach der Frage abändern.

Die Installation fragt Sie zunächst nach einigen allgemeinen Daten zu den Speicherorten.

```

boss:/home/software/vmware-distrib # ./vmware-install.pl
Creating a new installer database using the tar2 format.
Installing the content of the package.
In which directory do you want to install the binary files?
[/usr/bin]
In which directory do you want to install the library files?
[/usr/lib/vmware]
The path "/usr/lib/vmware" does not exist currently. This script is going
to create it, including needed parent directories. Is this what you want?
[yes]
In which directory do you want to install the manual files?
[/usr/share/man]
In which directory do you want to install the documentation files?
[/usr/share/doc/vmware]
The path "/usr/share/doc/vmware" does not exist currently. This script is
going to create it, including needed parent directories. Is this what you
want?
[yes]
What is the directory that contains the init directories (rc0.d/ to
rc6.d/)?
[/etc/init.d]
What is the directory that contains the init scripts?
[/etc/init.d]

```

Danach folgen einige Optimierungsschritte, welche die Installation ebenfalls für Sie vornehmen kann.

```
the installation of VMware Workstation 3.0.0 build-1455 for Linux
completed successfully. You can decide to remove this software from your
system at any time by invoking the following command: "/usr/bin/vmware-
uninstall.pl".
```

```
Before running VMware Workstation for the first time, you need to
configure it for your running kernel by invoking the following command:
"/usr/bin/vmware-config.pl". Do you want this script to invoke the
command for you now? [yes]
Making sure VMware Workstation's services are stopped.
Stopping VMware services:
  Virtual machine monitor[32mdone
  Virtual bidirectional parallel port[32mdone
You must read and accept the End User License Agreement to continue.
Press enter to display it.
```

Danach zeigt Ihnen die Installation die Lizenzvereinbarungen, die Sie akzeptieren müssen, damit es weitergeht.

```
Trying to find a suitable vmmon module for your running kernel.

The file /lib/modules/2.4.10-4GB/misc/vmmon.o that this script was about
to install already exists. Overwrite? [yes]
The module up-2.4.10-SuSE-7.3 loads perfectly in the running kernel.
Trying to find a suitable vmnet module for your running kernel.
The file /lib/modules/2.4.10-4GB/misc/vmnet.o that this script was about
to install already exists. Overwrite? [yes]
The module up-2.4.10-SuSE-7.3 loads perfectly in the running kernel.
```

Als Nächstes müssen Sie Entscheidungen bzgl. der Netzwerkanbindung und des Zugriffs des Windows-Gastsystems auf das Linux Dateisystem definieren.

```
Do you want networking for your Virtual Machines? (yes/no/help) [yes]
Configuring a bridged network for vmnet0.
Configuring a NAT network for vmnet8.
Do you want this script to probe for an unused private subnet?
(yes/no/help)
[yes]
Probing for an unused private subnet (this can take some time).
Either your host is not connected to an IP network, or its network
configuration does not specify a default IP route. Consequently, the
subnet 192.168.224.0/255.255.255.0 appears to be unused.
```

```
Press enter to display the DHCP server copyright information.
Do you want to be able to use host-only networking in your Virtual
Machines?
[no]
Do you want this script to automatically configure your system to allow
your Virtual Machines to access the host's filesystem? (yes/no/help) y
The version of Samba used in this version of VMware Workstation is
licensed as described in the "/usr/share/doc/vmware/SAMBA-LICENSE" file.
Hit enter to continue.
Creating a host-only network on vmnet1. (this is required to share the
host's filesystem).
Configuring a host-only network for vmnet1.
Do you want this script to probe for an unused private subnet?
(yes/no/help)
[yes]
Probing for an unused private subnet (this can take some time).
Either your host is not connected to an IP network, or its network
configuration does not specify a default IP route. Consequently, the
subnet 172.16.174.0/255.255.255.0 appears to be unused.
This system appears to have a CIFS/SMB server (Samba) configured for
normal use. If this server is intended to run, you need to make sure that
it will not conflict with the Samba server setup on the private network
(the one that we use to share the host's filesystem). Please check your
/etc/samba/smb.conf file so that:
. The "interfaces" line does not contain "172.16.174.1/255.255.255.0"
. There is a "socket address" line that contains only your real host IP
  ➔ address
Hit enter to continue.
Starting VMware services:
  Virtual machine monitor[32mdone
  Virtual bidirectional parallel port[32mdone
  Virtual ethernet[32mdone
  Bridged networking on /dev/vmnet0[32mdone
  Host-only networking on /dev/vmnet1 (background)[32mdone
  Host-only networking on /dev/vmnet8 (background)[32mdone
  NAT networking on /dev/vmnet8[32mdone
You have successfully configured VMware Workstation to allow your Virtual
Machines to access the host's filesystem. Would you like to add a
username and password for accessing your host's filesystem at this time?
(yes/no/help)
[yes] y
Please specify a username that is known to your host: bernd
```

```

New SMB password:
Retype new SMB password:
Added user bernd.
Password changed for user bernd.
You have successfully configured VMware Workstation to allow your Virtual
Machines to access the host's filesystem. Your system appears to already
be set up with usernames and passwords for accessing the host's
filesystem. Would you like to add another username and password at this
time? (yes/no/help) [no] n
You can add more usernames at any time by invoking the following command
as root: "/usr/bin/vmware-smbpasswd vmnet1 -a <username>"
The configuration of VMware Workstation 3.0.0 build-1455 for Linux for
this running kernel completed successfully.
You can now run VMware Workstation by invoking the following command:
"/usr/bin/vmware".
Enjoy,
--the VMware team
boss:/home/software/vmware-distrib #

```

VMWare ist nun auf dem Linux-System eingerichtet und Sie können beginnen, es einen Container für das Gastbetriebssystem einrichten zu lassen.

11.4.3 Container konfigurieren

Container richtet man über `/usr/bin/vmware` in einem grafischen Installationsmenü ein.

Der Installations-Wizard von VMWare interessiert sich zunächst für den Lizenzschlüssel. Einen 30-Tage-Evaluierungsschlüssel können Sie unter der Adresse <http://www.vmware.de> anfordern.

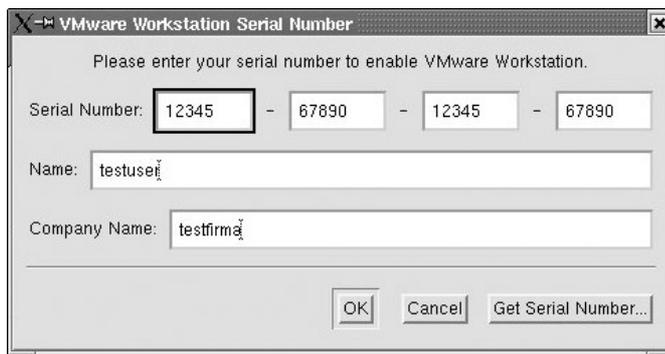


Abbildung 11.1: VMWare Lizenzschlüssel

Danach werden Sie gefragt, ob Sie VMWare neu installieren wollen oder eine bestehende Installation überarbeiten möchten.

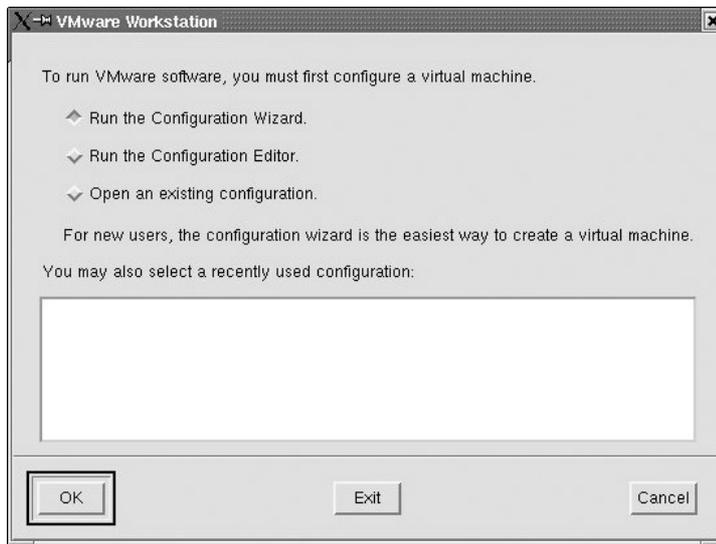


Abbildung 11.2: VMWare Start

Generell bietet VMWare zwei Methoden, Windows zu installieren. Entweder Sie installieren es manuell wie hier im Beispiel, oder Sie haben von VMWare ein Guest OS Kit erworben, welches vollautomatisch alle Parameter einstellt. Wählen Sie im folgenden Fenster die Installationsart und danach, welches Betriebssystem Sie auf VMWare-Basis installieren möchten.



Abbildung 11.3: Installationsmodul

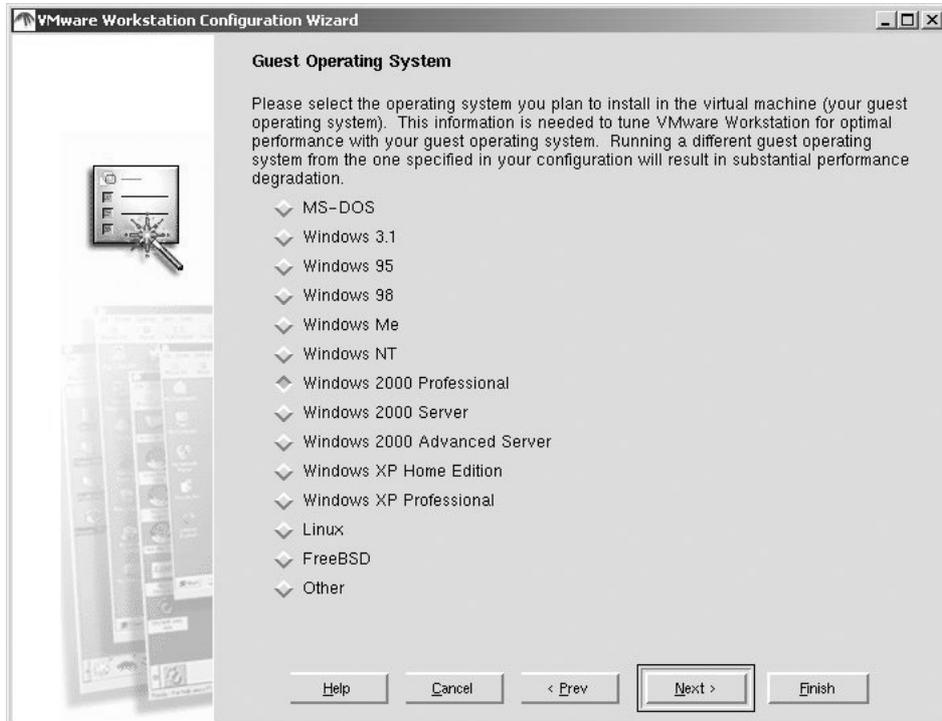


Abbildung 11.4: Wahl des Gastbetriebssystems

Nun folgen die Angaben für den Speicherort und die Speichergröße. VMWare nimmt hierfür einen Teil der Festplatte und reserviert ihn für VMWare. Diesen reservierten Bereich nennt VMWare virtual Disk. Alternativ können Sie eine eigene Festplatte für die Windows-Installation angeben.

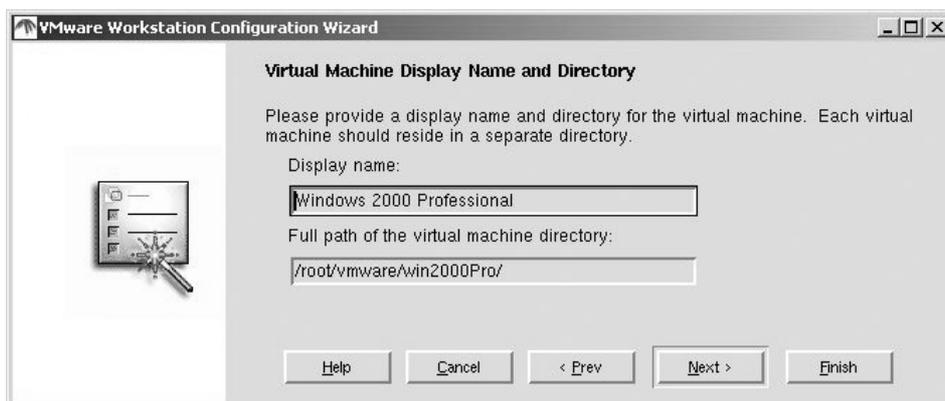


Abbildung 11.5: Zielverzeichnis für die virtuelle Festplatte

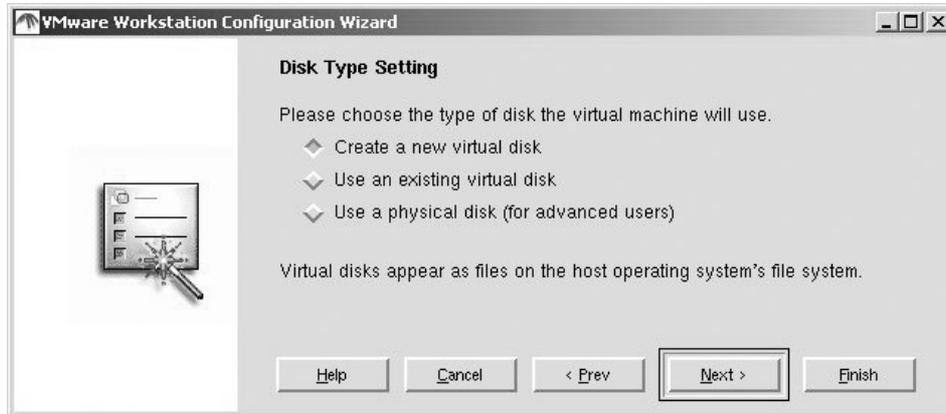


Abbildung 11.6: Wahl der Plattenrealisierung

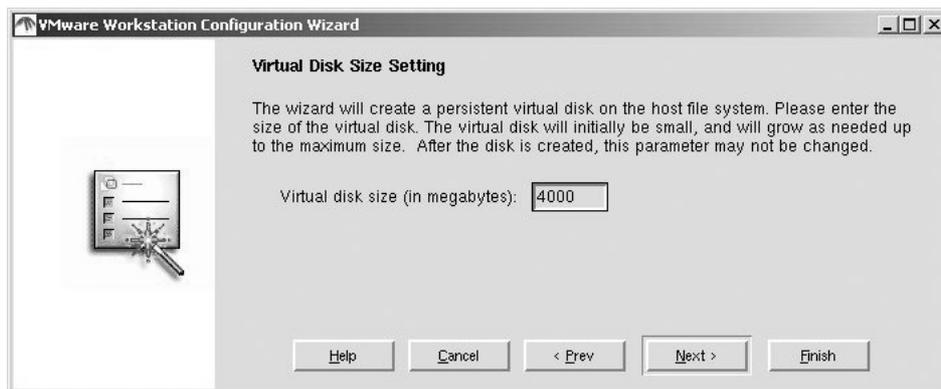


Abbildung 11.7: Größe der virtuellen Festplatte

In dem dann folgenden Fenster können Sie definieren, wie VMWare die Hardware-Komponenten (Diskettenlaufwerk, CD-Laufwerk) anbinden soll.

Zum Dank bestätigt der Wizard Ihre Entscheidungen und kündigt an, in welche Verzeichnisse er welche Konfigurationsdateien schreiben wird.

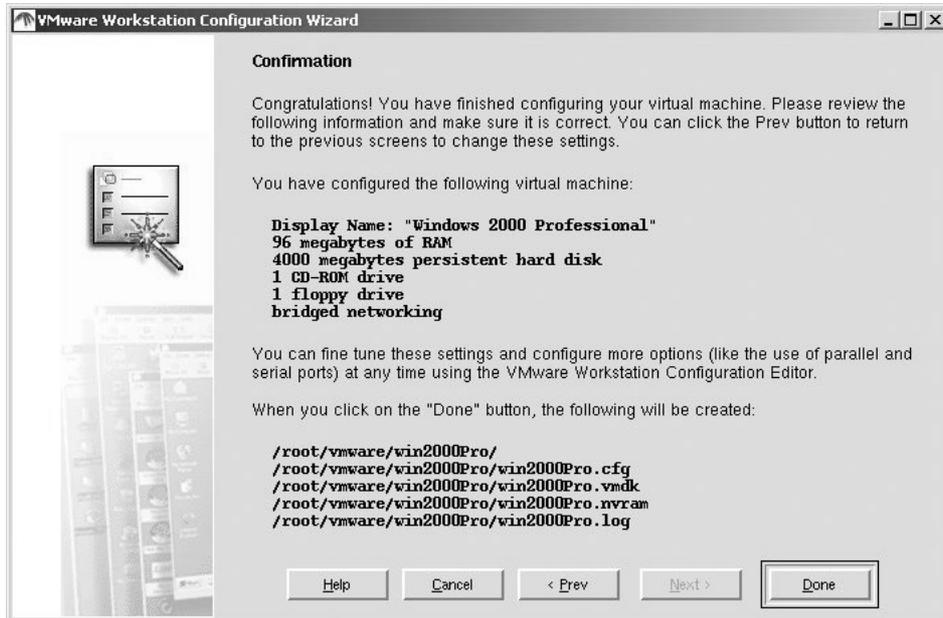


Abbildung 11.8: Bestätigung der Auswahl

Legen Sie nun die MS Windows 2000 Professional CD ein und starten Sie die Installation mit *Done*. Sie sehen nun ein virtuelles BIOS ablaufen, welches die Installation startet.

11.4.4 Windows 2000 Professional auf VMWare installieren

Nach dieser rasant schnellen Vorbereitung kann man Windows 2000 von CD installieren.

Die Begrüßung sowie alle weiteren Installationsschritte sehen wie bei der Installation auf einer physikalischen Festplatte aus.

Das Setup von Windows stellt eine neue Festplatte fest (Abbildung 11.10). Lassen Sie das Setup-Programm die von ihm erkannte Festplatte einrichten und mit dem Format NTFS formatieren.

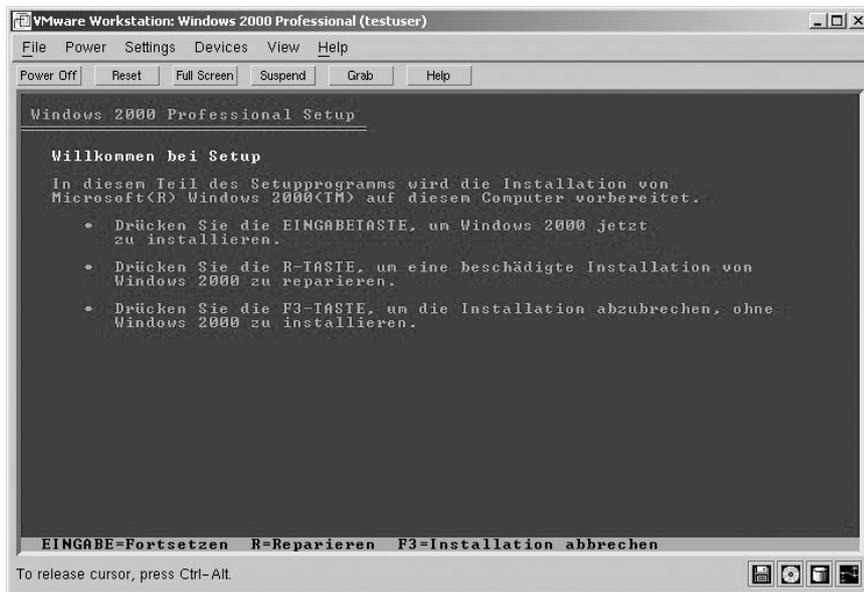


Abbildung 11.9: Auswahl im Setup

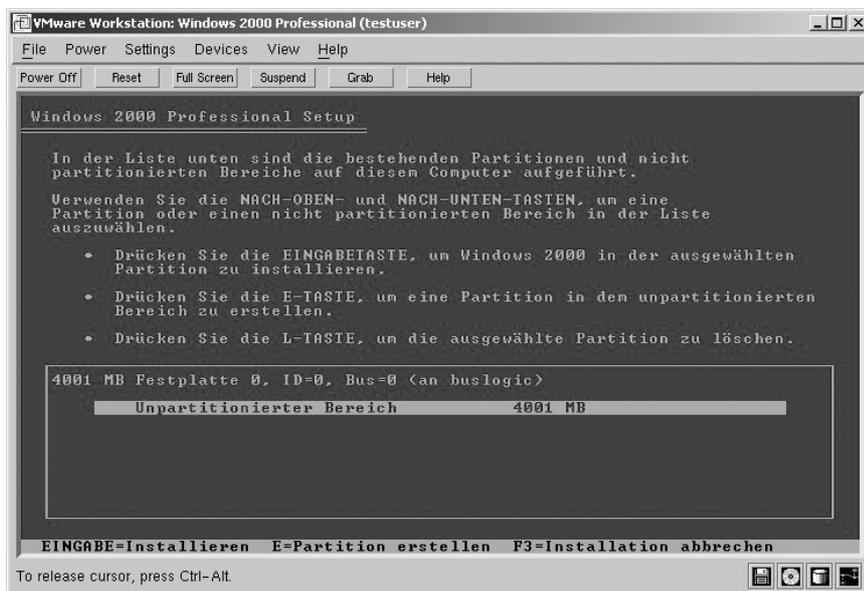


Abbildung 11.10: Festplattensetup

Das Setup von Windows 2000 kopiert danach die benötigten Dateien in die Installationsverzeichnisse und initialisiert Windows 2000.

Danach startet Windows 2000. Es setzt das Setup im Grafikmodus fort und sucht und installiert Geräte.

Als Nächstes erfasst das Setup Angaben zum Gebietsschema wie Ländereinstellungen, Zeitzone, Sprache und Tastaturlayout.



Abbildung 11.11: Gebietsschema wählen

Geben Sie bei den Benutzerinformationen Ihren Namen und den Ihrer Organisation an.



Abbildung 11.12: Benutzerinformationen eingeben

Für Benutzer von Open Source-Software mag es ungewohnt sein, im nächsten Bild eine Seriennummer anzugeben:

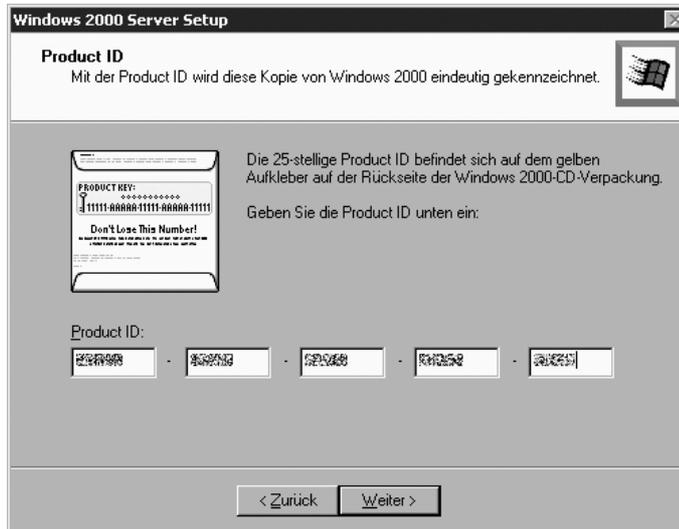


Abbildung 11.13: Seriennummer eingeben

Dann müssen Sie dem virtuellen Server einen eigenen Namen geben und den Namen und das Passwort des Systemverwalters eingeben.

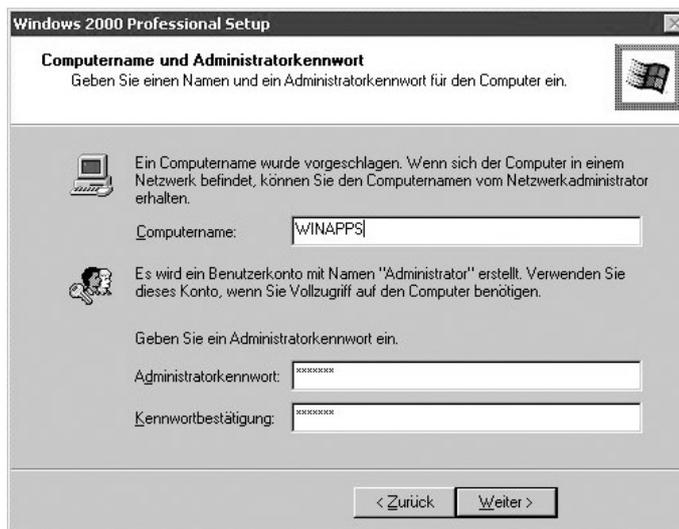


Abbildung 11.14: Namen des Servers und Admin-Kennwort eingeben

Windows 2000 installiert die virtuelle Netzwerkkarte, die es von VMWare bekommt (diese Netzwerkkarte ist völlig unabhängig von der realen Netzwerkkarte im Linux-Server), und richtet das Netzwerk ein.

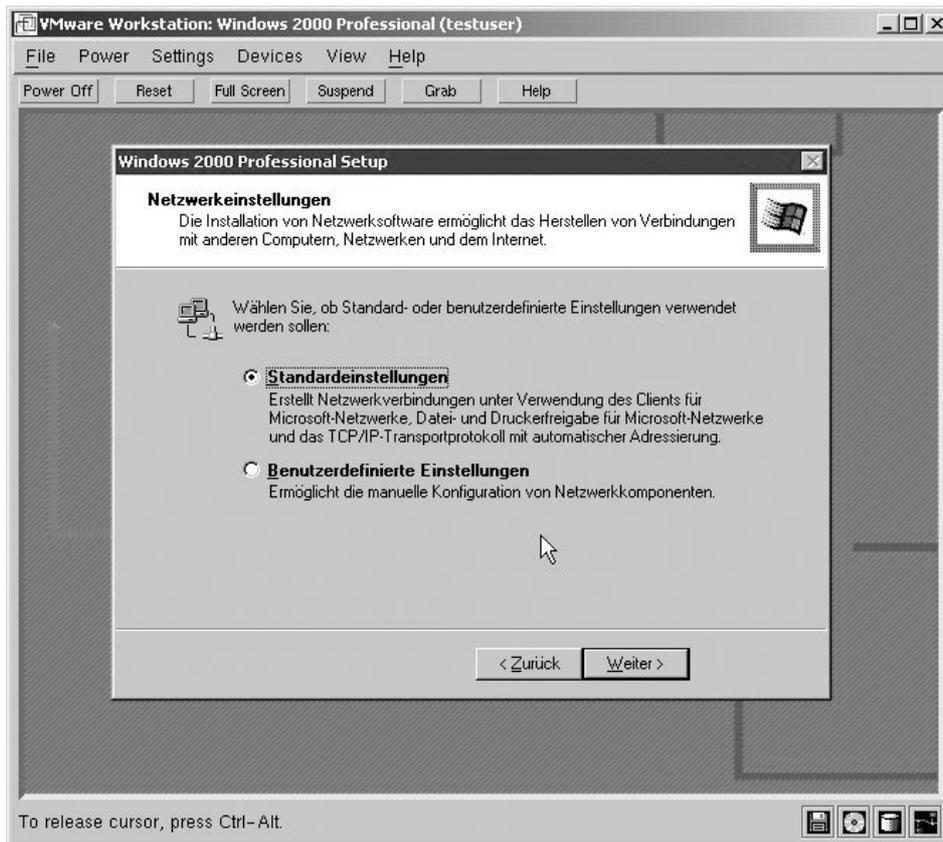


Abbildung 11.15: Microsoft-Netzwerk einrichten

Geben Sie dann an, in welche Arbeitsgruppe oder Domäne Sie den virtuellen Windows 2000 Server einfügen wollen, und geben Sie den Namen und das Passwort des Nutzers an, der den Server in die Windows-Domäne einfügen darf. Raucher können jetzt eine längere Zigarettenpause einplanen, während der Setup-Assistent die Windows 2000 Komponenten kopiert und installiert.

Sobald er fertig ist, können Sie Windows 2000 in der Box von VMWare vom Linux-Server und von jedem X11-Terminal aus wie eine eigenständige Installation nutzen und sich an populären Windows-Anwendungen wie Microsoft Office erfreuen. Starten Sie zunächst über `/usr/bin/vmware VMWare` und wählen Sie im unteren Bereich die Konfigurationsdatei, die zuvor von VMWare durch die Windows-Installation angelegt wurde.



Abbildung 11.16: Windows 2000 im VMWare-Fenster

Sie können auch zusätzliche Versionen von Windows, z.B. andere Sprachen bzw. andere Windows-Versionen, installieren. Hierzu müssten Sie wiederum die beschriebenen Installationsschritte für das neue, gewünschte Betriebssystem durchlaufen. Wenn Sie mehrere Windows-Sitzungen benötigen, können Sie auch mehrere Instanzen von Windows ausführen lassen. Dies benötigt jedoch großzügig dimensionierte Hardware-Ressourcen.

11.5 Konzept von Tarantella

Die Firma Tarantella Inc., zuvor SCO Inc., bietet seit Ende 1997 Versionen von Tarantella: Seit dem Verkauf von zwei der drei Sparten der SCO an Caldera hat sich SCOs innovativste Sparte, Visionprodukte und Tarantella, Ende 2000 in Tarantella umbenannt und sich auf eben diese Produkte fokussiert.

Die Middleware Tarantella verbindet nahezu beliebige Clients mit nahezu beliebigen Anwendungs-Servern. Statt Anwendungen lokal auf den Arbeitsplatzrechnern der Anwender zu installieren, stellt man sie diesen jetzt ganz einfach über ein LAN oder über ein WAN zur Verfügung, um Administrations-, Sicherheits- und Supportvorteile zu erzielen und die Gesamtkosten der Datenverarbeitung zu senken.

Tarantella-Server arbeiten in drei Schritten:

1. Sie verhalten sich gegenüber Mainframe (TN3270- und TN5250-Protokolle), Unix (X11, VT und ANSI-Protokolle) und Windows-Anwendungs-Servern (RDP 4 und RDP 5) wie Clients und fangen deren Benutzersitzungen ab,
2. komprimieren und verschlüsseln sie auf Wunsch in ihr eigenes Tarantella-Protokoll (AIP) und
3. übertragen sie damit an javafähige Browser oder spezielle Web-Appliances (Native Clients).

Dies ist weiter unten detaillierter beschrieben.

11.5.1 Zielgruppen für Tarantella

Mit Tarantella kann man Applikationen im Intranet und Internet für gelegentlichen oder häufigen Gebrauch bereitstellen:

- *Server-Zentriertes-Computing*: Da viele Unternehmen und öffentliche Einrichtungen die ausufernden Kosten der Betreuung von Windows-Arbeitsplätzen nicht mehr tragen wollen, stellen sie vermehrt auf server-zentrierte Datenverarbeitung um. Neue Anwendungen, Patches, Hot Fixes und Service-Packs brauchen sie dann nur noch zentral im Rechenzentrum zu installieren, laufen zu lassen und zu pflegen. Sie verwenden Tarantella als Middleware zwischen Windows-Anwendungs-Servern und Arbeitsplatzgeräten, wenn diese über ein WAN zugreifen oder sie die Last der zahlreichen Arbeitsplatzgeräte auf mehrere Windows-Terminal-Server verteilen wollen.
- *PCX-Server oder RDP-Zugriff*: Immer mehr Unternehmen und öffentliche Einrichtungen verwenden aus Sicherheits- und Kostenüberlegungen möglichst Open Source-Programme. Um ausnahmsweise von javafähigen Browsern ihrer Linux-Clients auf Windows-Programme zugreifen zu können, ohne auf Clients virenanfällige und instabile Microsoft Produkte zu installieren, stellen Sie mit Tarantella und Windows-Terminalservern die in der Übergangszeit noch gewünschten Windows-Programme bereit. Tarantella ermöglicht Anwendern, von den verschiedensten Clients aus (z.B. Windows CE) auf den verschiedensten Applikationsservern (z.B. »Gnome« auf Solaris) zu arbeiten.

- *Sicherheit:* Anwender greifen immer und ausschließlich über den Tarantella-Server auf Anwendungen und Daten zu. Dies ermöglicht es, ein sehr sicheres Netz aufzubauen. Das Öffnen zusätzlicher Ports zum Zugriff in Rechenzentren entfällt. Jede Kommunikation zwischen Client und Tarantella-Server kann zusätzlich verschlüsselt und protokolliert werden.
- *Webenabler:* Viele Firmen entwickeln nur noch Applikationen, welche in Browsern lauffähig sind (z.B. SAP). Die Bedeutung von Browsern als der zentralen Anwendung wächst. Mit Tarantella kann man verschiedenste Anwendungen innerhalb weniger Minuten »browserfähig« machen und so z.B. X11 Applikationen über Portale wie iPlanet Usern einheitlich anbieten.

11.5.2 Funktion von Tarantella

Tarantella arbeitet als Middleware zwischen Anwendungs-Servern und Clients.

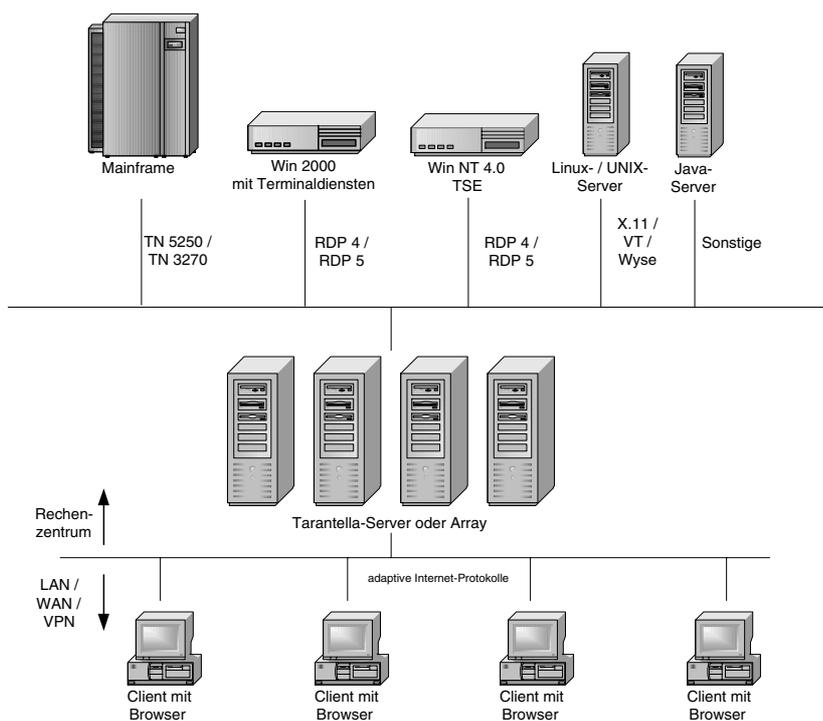


Abbildung 11.17: Die 3 Ebenen aus Tarantella-Sicht

In der Client-Ebene, der Ebene 1, arbeiten die Anwender und in der Ebene 3 die Applikations-Server, typischerweise Windows-Terminal-Server, Unix/Linux- und Mainframesysteme, Plattform-unabhängige Java-Applikations-Server sowie File-, Proxy- und Print-Server (welche manchmal auch als Ebene 4 bezeichnet wird). Zwischen der 1. und der 3. Ebene, in der Ebene 2, fangen Tarantella-Server die Protokolle der 3. Ebene auf, übersetzen diese in ihr eigenes Protokoll, das Adaptive Internet Protokoll (AIP), und stellen dessen Inhalt den Clients zur Verfügung. Tarantella muss nicht dediziert auf einem eigenen Server laufen, sondern kann auf vorhandenen Servern mitlaufen, zum Beispiel auf einem Samba-Linux-(SMB-) Server.

Tarantella kann Unix/Linux/Mainframe-Programme über X11-Protokolle oder Terminal-Emulationen leicht auffangen und in sein eigenes Protokoll AIP umsetzen. Benutzer-Sitzungen von Windows-Programmen kann es von einem speziellen Mehrbenutzer-NT (der inzwischen auslaufenden Microsoft Windows NT 4 Terminal Server Edition und von Windows 2000 Servern/Advanced Servern), der als Terminal-Server installiert ist, abfangen. Windows-Terminalserver kommunizieren mit Clients mit Microsofts proprietärem Remote Desktop Protokoll (RDP). Microsoft bietet jedoch selbst ausschließlich Windows-basierende Clients an. Aufgrund einer Vereinbarung zwischen Microsoft und Tarantella kennt Tarantella Microsofts Spezifikationen und Code zum Zugriff auf die Terminal-Server-Sitzungen. Der Unix/Linuxbasierende Tarantella-Server verhält sich gegenüber den Terminal-Servern dann ebenfalls wie ein Microsoft Terminal-Server Client. Hierbei werden einige zusätzliche Funktionen, verwirklicht, die über einen nativen RDP Client-Zugriff nicht möglich wären. So wurde der Laufwerkszugriff zwischen Terminal-Server und Client zwar von Microsoft über RDP vorgesehen, jedoch nicht integriert (die Schaltfläche hierfür kann nicht aktiviert werden). Tarantella ermöglicht dieses Feature jedoch über einen virtuellen SMB-Server, den die Tarantellainstallation automatisch einrichtet, und zwar über sein AIP-Protokoll.

Es bleiben jedoch einige grundsätzliche Einschränkungen: RDP kann derzeit nur 256 Farben und keinen Sound übertragen. Diese und einige andere Einschränkungen gelten zwangsläufig auch für den Tarantella-Server. Diese Begrenzung sollte in den Nachfolgeversionen von Windows 2000 (Microsoft Windows .NET Serverfamilie) Mitte 2002 aufgehoben sein (www.microsoft.com/windows.NETserver/evaluation/choosing/default.asp) und gemäß der Übereinkunft mit Tarantella zeitnah auch aus deren Produkt Tarantella verschwinden.

Tarantella kann mehrere Terminal-Server zu *Serverfarmen* zusammenfassen, um deren Belastung auszugleichen (Load Balancing). Dazu sollten die Terminal-Server gleich aufgebaut sein und Anwendern idealerweise über einen gemeinsamen Domain Controller oder über einen Active Directory Server die gleichen User-Accounts anbieten.

Tarantella-Server können ihr eigenes Protokoll AIP (Adaptive Internet Protokoll) verschlüsseln (OpenSSL mit – derzeit gesetzlich in den meisten Ländern (wie in Deutschland) limitiert auf – 128 Bit). Tarantella kann man für einen »Single Port Access« konfigurieren, d.h. man kann auf Tarantella über einen Standardport wie Port 80 (http) oder Port 443 (https) zugreifen (www.tarantella.com/whitepapers/firewall/), braucht also auf Firewalls keine zusätzlichen Ports zu öffnen. AIP benötigt, falls erforderlich, weniger Bandbreite als Microsofts RDP: Tarantella-Server entscheiden dynamisch nach der zur Verfügung stehenden Bandbreite, wie weit sie den Datenstrom verdichten.

Während Tarantella-Server in einem 10 Mbit-LAN praktisch nichts komprimieren, verdichten sie den Datenstrom bei langsamen WAN-Verbindungen oder wenn sich mehrere Anwender eine Leitung teilen. GSM-Verbindungen mit 14.400 Baud reichen für einen Benutzer gerade noch aus, 64 Kbit-ISDN-Leitungen bedienen bequem 2-3 User. Drucken Anwender, kann man mit Tarantella (oder bei Bedarf auch mit Zusatzprodukten wie ThinPrint) Druckaufträge packen und die von Druckaufträgen beanspruchte Bandbreite mit ThinPrint (www.thinprint.com) begrenzen.

Die Benutzer der Tarantella-Server kann man auf unterschiedlichste Art verwalten: Unixaccounts auf den Tarantella-Servern (z.B. NIS), Authentisierung gegenüber LDAP- oder Active Directory Servern, Windows Domänen, Webservern und/oder SecureID-Servern oder mit einer eigenen Benutzerverwaltung in Tarantella selbst.

Das Session-Resume unterstützt mobile Nutzer: Benutzer können nicht ordnungsgemäß beendete Verbindungen später wieder aufnehmen. Damit sind Tunnel wie auf der ICE-Strecke von Frankfurt nach Göttingen oder Staus im Elbtunnel für mobile GSM-Nutzer nicht mehr ganz so schrecklich.

Auch in mobilen und angeblich papierlosen Büros wollen Anwender drucken. Tarantella-Server kann man als IP-Druck-Server ansprechen und so Druckaufträge ebenfalls über AIP verschlüsseln und komprimiert an Tarantella-Clients schicken.

Tarantella selbst nimmt ungefähr 100 MB Platz auf den Festplatten des Linux-Servers in Anspruch. In kleinen Installationen reichen auch ältere Server (Minimum PentiumPro oder Pentium II) mit relativ wenig Hauptspeicher (Minimum 192 MB). Um den Bedarf an Hauptspeicher und Prozessorgeschwindigkeit für einen Linux-Tarantella-Server auszurechnen, sollten Sie die Richtlinien für das Ser-

ver-Sizing in <http://www.tarantella.com/products/e3/requirements.html> lesen. Als Richtwert lassen sich mit einem Single Pentium III, 800 MHz und 1 GB Hauptspeicher ohne Schwierigkeiten 100 User gleichzeitig anbinden. Bei üppiger Hardwareausstattung sind über 1000 User pro Server möglich. Tarantella bietet unter <http://www.tarantella.com/whitepapers/> die Testergebnisse verschiedener Betriebssysteme und Hardwarelieferanten an.

Als Client kann man praktisch jedes Gerät mit javafähigem Browser verwenden. Für Clients reicht ein langsames Pentiumsystem ab Pentium 166 MHz und 48 MB Arbeitsspeicher. Beim erstmaligen Verbinden zum Tarantella-Server lädt der Browser 400 KB Javaprogramme und installiert sie. Diese Javaprogramme sollte der Client aus Performancegründen auf der Festplatte speichern können. Ältere Hardware (Pentium I mit 24 MB Arbeitsspeicher) kann man mit einem Tarantella-Native-Client verwenden. Der Tarantella Native Client ist für verschiedenste Betriebssysteme erhältlich wie: Windows 32 Bit, Windows CE, Linux Kernel 2+, Solaris 7+. Eine vollständige Liste der unterstützten Browser und Client-Betriebssysteme hat Tarantella unter der Adresse <http://www.tarantella.com/products/e3/clients.html> zusammengestellt.

Tarantella bietet derzeit zwei unterschiedliche Versionen an:

Tarantella Enterprise 3 (Versionsstand bei Drucklegung 3.20)

Diese Version von Tarantella richtet sich an große Einsatzfelder mit bis zu mehreren zehntausend gleichzeitigen (concurrent) Usern. Als Betriebssystemplattform können praktisch alle gängigen Unix-Derivate (Sparc-Solaris, AIX, HP, Unixware) sowie gängige Linux-Distributionen (SuSE, Red Hat, Caldera, Turbo Linux) verwendet werden. Eine »Serverfarm« ist sowohl auf der Applikations-Serverebene als auch auf der Tarantellaebene (»Arrays«) möglich. z.B. für Abrechnungszwecke (wie durch ASPs) kann man damit auch das Benutzerverhalten protokollieren.

Tarantella Starter für Linux (Versionsstand bei Drucklegung 3.20)

Grundsätzlich ist diese Version die gleiche Version wie *Tarantella Enterprise 3*. Sie ist jedoch lizenztechnisch auf den Einsatz für Intel-Plattformen (Unixware und Linux) und auf maximal 50 gleichzeitige Benutzer beschränkt. Arraybildungen auf Tarantellaebene sind zwar möglich, jedoch unterstützt Tarantella sie bei dieser Version für weniger als 50 User nicht (z.B. bei Anfragen im Support). Kunden können die preisgünstige Version *Starter für Linux* durch Zukauf weiterer Lizenzen (über 50 hinaus) als Enterprise 3 lizenzieren.

11.6 Tarantella installieren

Für die im Folgenden beschriebene Installation von Tarantella auf einem SuSE-basierten Linux-Server muss auf dem Linux-Server ein Web-Server laufen (z.B. Apache Web-Server). Das Netzwerk inkl. Namensauflösung sollte vor der Installation von Tarantella laufen, da nachträgliche Änderungen sehr zeitaufwändig sind. Die folgenden Beispiele basieren auf einer SuSE 7.3 Professional, Installationsmodule »KDE Standard« und zusätzlich »Einfacher Webserver«.

Sie können eine 30 Tage Demoversion von Tarantella aus dem Internet von www.tarantella.com/download fernladen. Diese Version ist bis auf das zeitliche Enddatum uneingeschränkt nutzbar und bei Bedarf von Tarantella einmalig um weitere 30 Tage verlängerbar.

Die Tarantella-Dateien gehorchen folgender Schreibweise:

Die ersten drei Buchstaben geben das Programm selbst an: `tta` steht für das Tarantella-Basis-Modul, die nächsten zwei Buchstaben geben den Prozessortyp an, z.B. `i3` für Pentiumsysteme, und die letzten Buchstaben das Betriebssystem z.B. `li` für Linux:

`Ttai3li` bedeutet Tarantella Base Pack für Pentiumsysteme mit dem Betriebssystem Linux.

Um die Installation mitzumachen, loggen Sie sich als `root` in Ihr Linux-System ein, mounten das CD-ROM-Laufwerk und rufen dann die richtige Datei mit `sh` auf. Die Installation prüft, ob Sie das richtige Paket verwenden und genügend Festplattenspeicher zur Verfügung steht. Sollten während der Installation Fehler auftreten, trägt die Installation die Fehlermeldungen in die Datei `/tmp/ttainst.log` ein:

```
boss:~ # mount /cdrom
boss:~ # cd /cdrom
boss:~ # sh ttai3li.shx
-----
Tarantella Enterprise 3
-----
Setup is verifying the shx file...
Setup is checking for disk space requirements...
Setup is installing package programs in /usr/package...
Tarantella Enterprise 3 for Intel Linux kernel 2.2+
Preparing for installation...
-----
Tarantella Software License Agreement
-----
To use Tarantella you must agree to be bound by
```

```
the terms of the Software License Agreement.
```

```
Y - I have read, and accept the terms of the license agreement
N - I do not accept the terms of the license agreement
R - Let me read the license agreement
```

```
Accept terms of Software License Agreement? [R] y
```

1. Bestätigen Sie, wie hier im Beispiel, die Lizenzvereinbarungen, und geben Sie den Installationsort, den Archivierungszeitraum für Fehlermeldungen und den Lizenzschlüssel (Standard: Evaluierung) ein.

```
-----
Setting up Tarantella Enterprise 3
-----
```

```
Tarantella Setup recommends you use the following settings:
```

```
Installation type = install 3.20.907
Installation directory = /opt/tarantella
Peer DNS name = boss.lokales-netz.de
License mode = Evaluation (30-day limit)
Archive logs every week? = yes (Sunday 03:00 hours)
```

```
Are these settings OK?
```

```
Y - Yes, install using these settings
N - No, tell me more about the options and let me change the settings
Q - Quit now
```

```
OK to use these settings? [Y]
```

2. Nach Bestätigung mit y beginnt die eigentliche Installation, die je nach Hardware etwa 5-10 Minuten dauert:

```
Tarantella Setup is recording settings...
```

```
Tarantella Setup is installing files...
```

```
...
```

```
Starting Tarantella server, please wait.....
```

```
The Tarantella server is now running.
```

```
-----
What you must do next
-----
```

```
To use Tarantella, you must first configure a web server on this host:
```

- Add a Document directory for the Tarantella document root:
Make /tarantella map to /opt/tarantella/var/docroot
- Add a Program (CGI) directory for the Tarantella cgi-bin subdirectory:
Make /tarantella/cgi-bin map to /opt/tarantella/var/docroot/cgi-bin

See the "Before you begin" section of the online documentation:
/opt/tarantella/var/docroot/help/en-us/admintocs/TOC_FUNC_TYPE.html

To get started quickly:

- Add license keys: /opt/tarantella/bin/tarantella license add <key>...
- Configure the web server as shown above.
- Go to http://<server>/tarantella
- Log in to Tarantella as root, and click Object Manager.

boss:~ #

Damit ist die Tarantella-Installation abgeschlossen. Wie an der Meldung ersichtlich, müssen Sie als Nächstes den Web-Server für eine Zusammenarbeit mit Tarantella konfigurieren.

Beachten Sie bitte ebenfalls www.tarantella.com/support für Tarantella-Updates und dort den Unterpunkt Knowhow für besondere Einstellungen und Tuningmöglichkeiten, abhängig vom verwendeten Betriebssystem.

11.6.1 Web-Server konfigurieren

Wie vom Installationsskript schon am Ende der Tarantella-Installation angezeigt, muss man am installierten Web-Server je nach Typ, Betriebssystem und Tarantella-Version noch einige Kleinigkeiten ändern.

Hier folgen Konfigurationsschritte für den Apache Web-Server. Diese basieren auf der Konfiguration des Web-Servers wie im Kapitel 6.

Tragen Sie bitte zusätzlich folgende Zeilen in /etc/httpd/httpd.conf ein:

1. Alias für Tarantella definieren:

```
#
# Aliases: Add here as many aliases as you need (with no limit).Format is
# Alias fakename realname
#
# Eintrag für Tarantella Alias (Verweist ../tarantella Anfragen)
#
Alias /tarantella /opt/tarantella/var/docroot
#
```

2. ScriptAlias für Tarantella definieren:

```
#
# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the
# client.
# The same rules about trailing "/" apply to ScriptAlias directives as to
# Alias.
#
ScriptAlias /cgi-bin/ "/usr/local/httpd/cgi-bin/"
#
# Eintrag für Tarantella ScriptAlias (Verweist cgi-bin-Anfragen)
#
ScriptAlias /tarantella/cgi-bin "/opt/tarantella/var/docroot/cgi-bin"
#
```

Nach diesen Änderungen sollten Sie sowohl den Web-Server (/etc/init.d/apache restart) also auch Tarantella neu starten. Tarantella startet man im Programmpfad (default: /opt/tarantella/bin) mit `tarantella restart`.

Wenn Anwender angemeldet sind (z.B. bei Updates), so können diese während des Restarts von Tarantella nicht weiterarbeiten. Sobald der Server jedoch wieder zur Verfügung steht (nach etwa 10 Sekunden), können die Anwender an dem Punkt weiterarbeiten, an dem sie zuvor aufhören mussten. Möchten Sie alle bestehenden Sessions neu aufsetzen, so verwenden Sie bitte `tarantella restart --kill`

Achten Sie hierbei auf eventuelle Fehlermeldungen:

```
boss:~ # cd /opt/tarantella/bin/
boss:/opt/tarantella/bin # ./tarantella restart
Tarantella services have been stopped.
Starting Tarantella server (version 3.20.907). Please wait...
Tarantella services are now available on this host.
boss:/opt/tarantella/bin #
```

11.6.2 Erste Verbindung

Tarantella bietet Clients wie bereits erwähnt zwei Zugriffsarten: per Browser oder mittels Tarantella Native Client.

Tarantella Native Client

Sie können den Tarantella Native Client für Windows 32 Bit und Linux von Ihrem Tarantella-Server von `http://boss.lokales-netz.de/tarantella/cgi-bin/install.cgi` laden und für weitere Plattformen von den Tarantella Internetseiten (`www.tarantella.com/download`) fernladen.

Das Installieren der Clients hat Tarantella als `.exe`-Datei für Windows und als komprimiertes Shellscript `.sh` für Linux automatisiert.



Abbildung 11.18: Anmeldedialog »Native Client«

Anwender können sich in der Voreinstellung mit ihren Linux-User-Accounts und den Passwörtern des Linux-Servers in Tarantella einloggen. Der Systemverwalter `root` ist auch Administrator für Tarantella, während die Rechte der *normalen* Anwender eingeschränkt sind. Verwenden Sie als Adresse den Alias, den Sie im `httpd.conf` angegeben haben, hier im Beispiel: `http://boss.lokales-netz.de/tarantella`, beim Native Client unbedingt einschließ- lich der Protokollangabe `http://`.

Tarantella über Browser

Sie können Ihren javafähigen Browser unabhängig vom Clientbetriebssystem mit dem Tarantella-Server verbinden, indem Sie im Browser die Adresse des Tarantella-Servers eingeben, hier im Beispiel `http://boss.lokales-netz.de/tarantella`. Die erforderliche Java Runtime-Umgebung bzw. eine Java Virtual Machine (Java VM) installieren aktuelle Browser automatisch; sollten sie fehlen, können Sie diese nachinstallieren.

Der Browser installiert die Java-Klassen für Tarantella je nach Bandbreite der Netzanbindung innerhalb weniger Sekunden. Sie sehen den Abschluss der Installation im Login-Screen von Tarantella. Einige Browser, z.B. die von Netscape, empfehlen nach der Installation aus Sicherheitsgründen einen Neustart des Browsers).

Der Tarantella-Server prüft bei jeder Verbindung zum Client die Aktualität der installierten Java-Plug-Ins auf dem Client. Sind diese veraltet oder nicht installiert, veranlasst Tarantella eine Neuinstallation. Die Installationsroutine fragt den Benutzer, je nach verwendetem Browser einmal oder mehrmals, ob er die erforderlichen Java-Klassen installieren möchte. Anwender brauchen daher die Clientsoftware für Tarantella auf den Anwendercomputern nicht selbst aktiv zu pflegen.

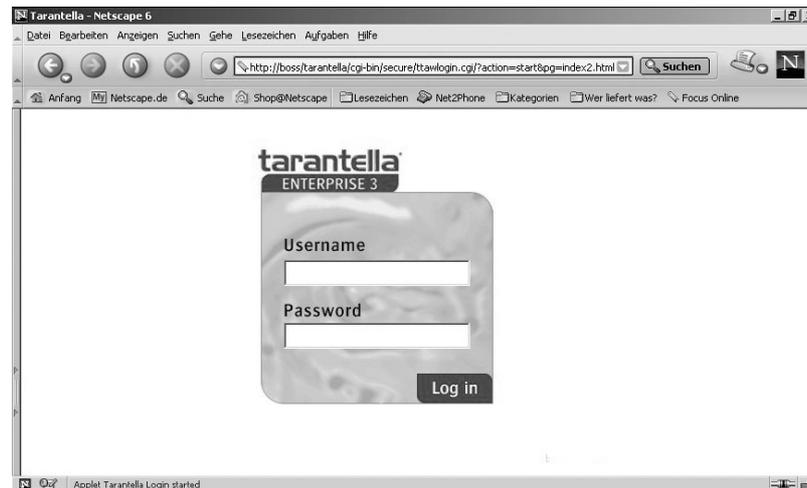


Abbildung 11.19: Tarantella Bildschirm nach Eingabe des Adress-Links

Nach dem Anmelden sehen Sie im Browser Ihres Clients zunächst den Standard-Bildschirm von Tarantella (Abbildung 11.20) mit einigen, während der Installation angelegten Anwendungen (*xclock*, *Konsole*, *KDE/Gnome* usw.) sowie einigen Hilfetools für Administratoren und Anwender (wie *Administration Guide*, *Objectmanager*, *Arraymanager* usw.).

Links im Bild sehen Sie gelb/orange hinterlegt das so genannte Launchpad. Über diese Leiste können Anwender alle Anwendungen durch Klicken starten. Im freien Arbeitsbereich sehen Anwender generelle Informationen oder die von ihnen aufgerufene Anwendung.

Hinweis: Da die Tarantella-Client-Darstellung vollständig durch html und Javascript gesteuert ist, kann diese nach Belieben für den Browserzugriff angepasst werden. Die folgenden Beispiele gehen immer von den »Standardansichten« aus.

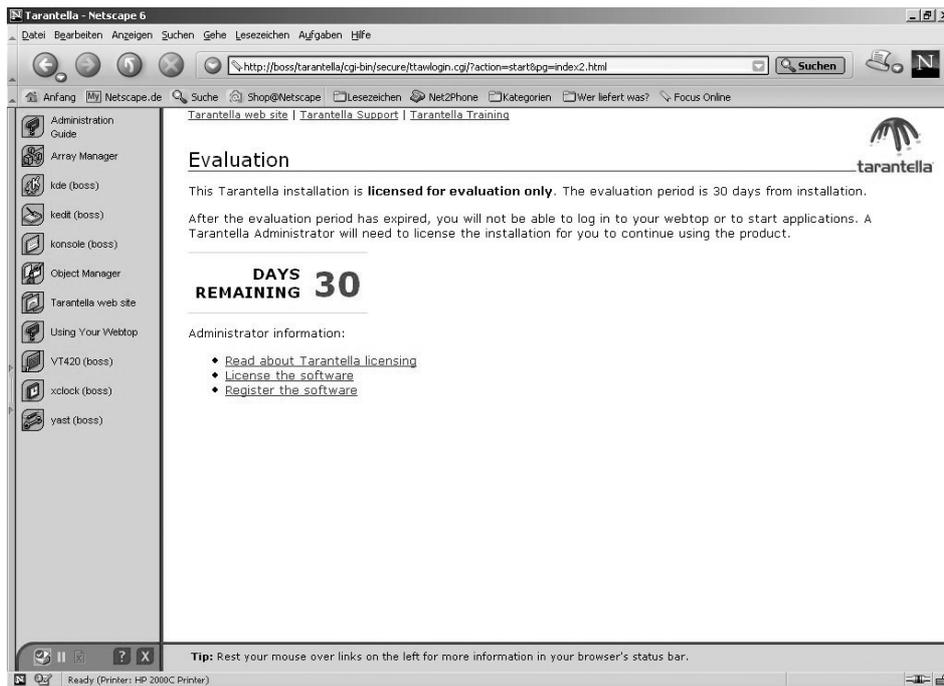


Abbildung 11.20: Tarantella-Standardbildschirm nach dem Einloggen

Für den ersten Test mit Anwendungen des Applikations-Servers per Browser reichen die bei der Tarantella-Installation automatisch eingerichteten Anwendungen aus. Anwender können diese als Vorlage für weitere Applikationen nutzen.

Rufen Sie die gewünschten Anwendungen über deren Icon auf. Der Benutzerclient baut dann, automatisiert über den Tarantella-Server, eine Verbindung zu dem Applikations-Server (hier dem Tarantella-Server selbst) auf und durchläuft dabei die Anmeldeprozedur am Applikations-Server. Standardmäßig versucht Tarantella, die Zugangsdaten in Tarantella auf die Applikation anzuwenden. Wenn diese Daten akzeptiert werden, so fragt der Tarantella-Server nicht erneut nach einem Loginnamen/Passwort für den Applikationsstart.

Tipp: Sie können beliebig viele Applikationen gleichzeitig ablaufen lassen. Solange Sie das kleine grüne Zahnrad auf dem Applikations-Icon sehen, wissen Sie, dass die Anwendung läuft.

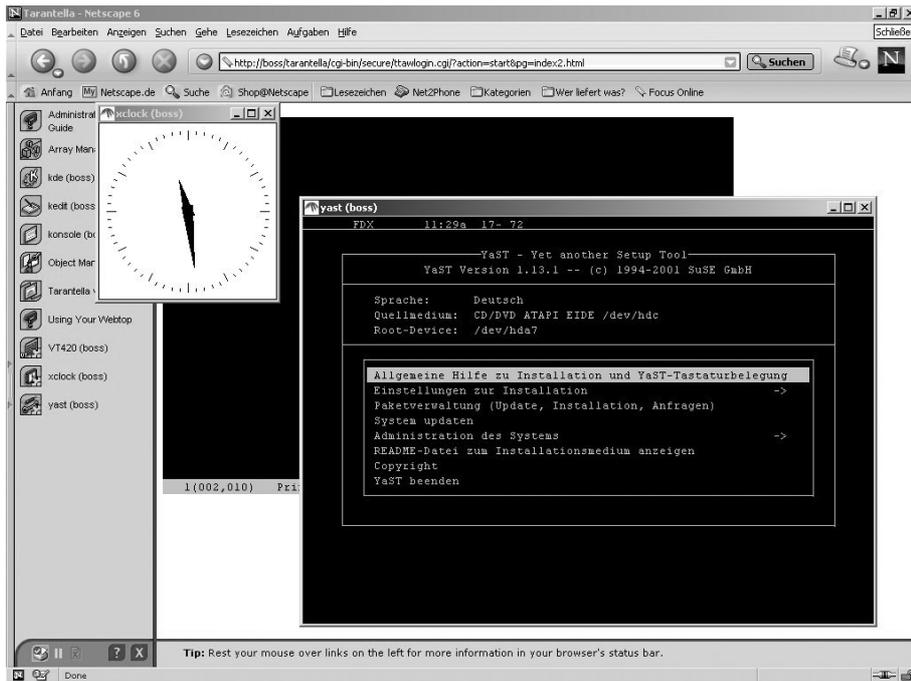


Abbildung 11.21: Tarantella mit verschiedenen Applikationen

Beenden kann man Anwendungen wie bei Windows oder wie bei Linux üblich (z.B. über *Datei • Beenden* bei Windows-Anwendungen oder z.B. über die Eingabe von `exit` im Terminalfenster). Sobald Benutzer sich aus Tarantella ausloggen, beendet Tarantella automatisch deren Anwendungen, falls Sie Tarantella nicht anders konfigurieren.

Tarantella unterscheidet die Darstellung im *Webtop* und die Darstellung als *Independent Windows/Kiosk-Modus bzw. Vollbilddarstellung* (Abbildung 11.22). Sie können den Fenstermodus bei Browserzugriff jederzeit wechseln, indem Sie bei gedrückter `[Strg]`-Taste nochmals auf das Icon der Anwendung im Launchpad klicken.

Um den lokalen Passwortcache zu umgehen, können Sie während des Klickens auf das Anwendungs-Icon (also beim Programmaufruf) die -Taste drücken. Dann fragt der Tarantella-Server Sie nach einem Usernamen und einem Kennwort für das System. Tarantella wendet dann nicht die Tarantella-Anmeldedaten oder gespeicherten Kennwörter an (Abbildung 11.23).

Anwender können ihre Anmeldedaten speichern, um diese beim nächsten Aufrufen dieser Anwendung nicht erneut eingeben zu müssen. Tarantella speichert das Passwort verschlüsselt (3DES).

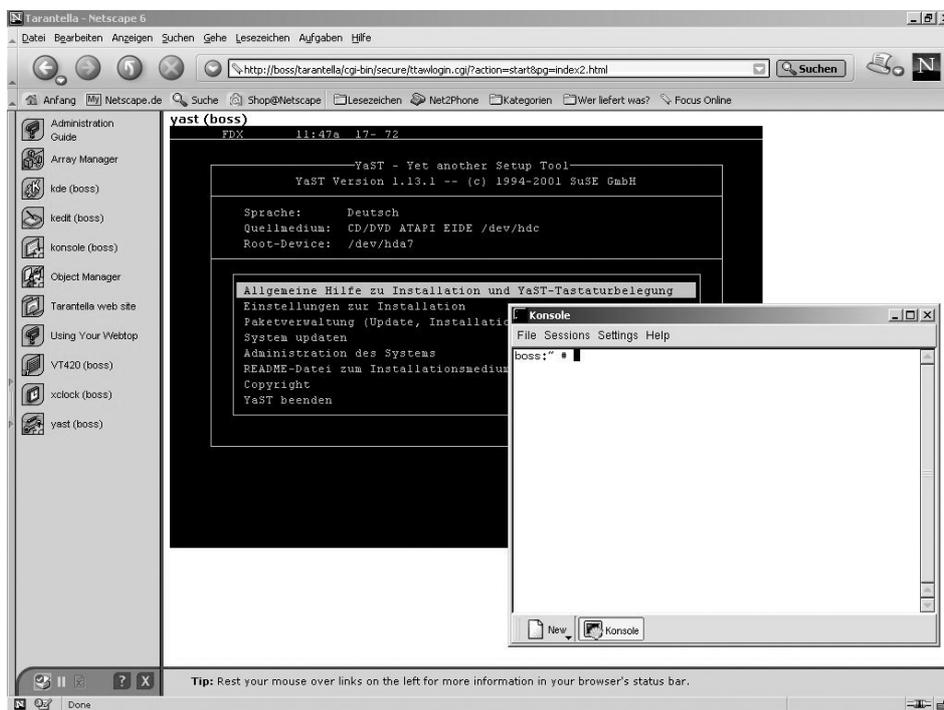


Abbildung 11.22: Tarantella mit YaST (Webtop) und Konsole (Ind. Windows)

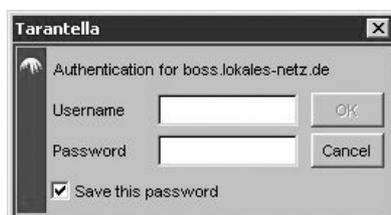


Abbildung 11.23: Username und Kennwort für eine Anwendung auf einem Server

Um sich vom Tarantella-Server abzumelden, können Sie das Kreuz am rechten unteren Rand des Launchpads betätigen. Nach einer Bestätigung gelangen Sie in den Standard-Abmeldeschirm von Tarantella. Je nach Konfiguration beendet Tarantella nun die Anwendungen oder versetzt sie in einen Schlafzustand.

11.7 Tarantella konfigurieren und administrieren

Tarantella hat beim Installieren abhängig von Ihren Eingaben ein Installationsverzeichnis angelegt. Dieses gliedert sich in drei Unterverzeichnisse:

`/bin`: Sie können Tarantella über die Command-line (CLI) oder über grafische Administrationstools konfigurieren. Ausgangspunkt für die CLI ist dieses Verzeichnis mit dem Befehl `tarantella` gefolgt von einem weiteren Parameter. Gerade beim Anlegen mehrerer Objekte kann man mit ein wenig Übung sehr viel schneller zum Ziel kommen als über das grafische Hilfetool. Verschaffen Sie sich durch die Eingabe von `tarantella help` einen Überblick:

```
boss:/opt/tarantella/bin # ./tarantella help

Usage: tarantella <command> [<command-specific args>]
Available commands:
archive           Archives the server's log files
array             Creates and manages arrays of Tarantella servers
arraymanager     Starts Array Manager
config           Edits array-wide and server-specific configuration
emulatorsession  Lists and controls emulator sessions
help             Displays this list of commands
license          Adds, lists and removes Tarantella license keys
object           Manipulates objects in the datastore
objectmanager    Starts Object Manager
passcache        Manipulates the password cache
print            Controls Tarantella printing services
query            Examines the server's log files
restart          Restarts Tarantella services
role             Configures role occupants and extra webtop links
setup            Changes Setup options, restores original objects
start            Starts Tarantella services
status           Shows the current status of Tarantella array members
stop             Stops Tarantella services
uninstall        Uninstalls Tarantella from this host
version          Displays versions of installed Tarantella packages
webcache         Manipulates the Tarantella web cache
webtopsession    Lists and controls webtop sessions
```

```
Use "tarantella <command> --help" to get help on a command.
boss:/opt/tarantella/bin #
```

/etc: In diesem Verzeichnis finden Sie Konfigurationsdateien für Tarantella.

/var: In diesem Verzeichnis befinden sich die Userdaten, Applikationskonfigurationen, Zuordnungen sowie alle Änderungen am Webtop, Startfenster usw.

Zur Minderung des Verlustrisikos empfiehlt es sich, das Verzeichnisses /var regelmäßig zu sichern und, wenn Sie viel Arbeit in die Konfigurationen gesteckt haben, das ganze Verzeichnis /opt/tarantella.

Tarantella bietet Administratoren (*root*) im angemeldeten Tarantellafenster zusätzliche Applikations-Icons wie das Tarantella-Hilfe-Dokument Administration-Guide sowie die Tarantella-Konfigurationstools Array Manager und Object Manager. Letztere vereinfachen das Einstellen von Applikationen und Usern erheblich.

Mit dem Array Manager konfiguriert man einmalig allgemeine Einstellungen wie z.B. Lizenzschlüssel, Serverfarmen (Arrays) von Tarantella-Servern, Portdefinitionen, Anmeldeverhalten usw.

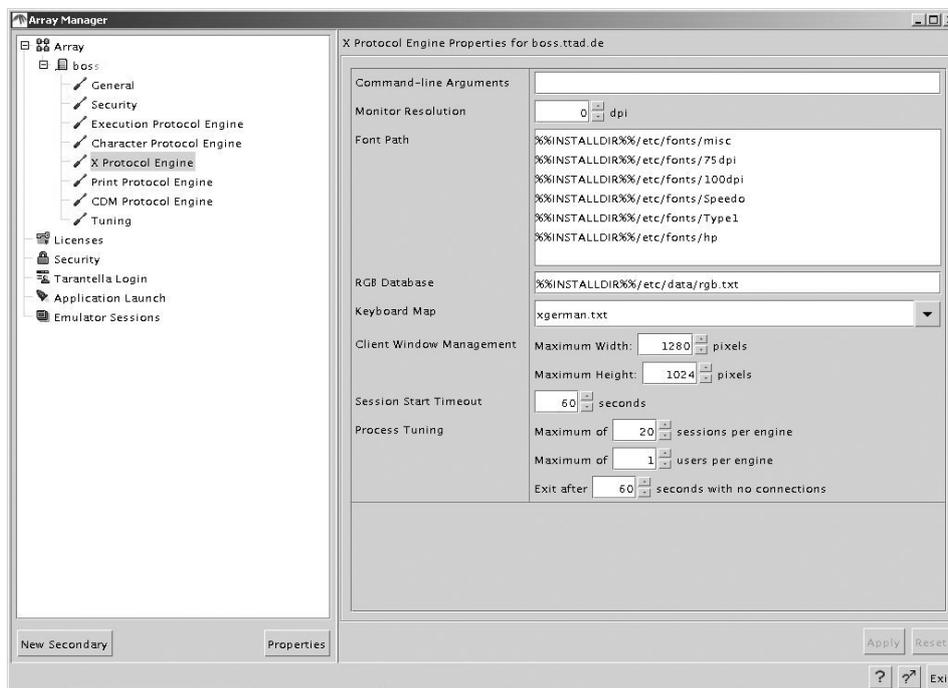


Abbildung 11.24: Array-Manager

Der Object Manager dient zum Konfigurieren von Userdaten, Anwendungen, Zuordnungen usw. Der Object Manager ist ebenfalls das zentrale Tool von Tarantella zum Managen von Usern/Sessions und wird daher sehr häufig benötigt.

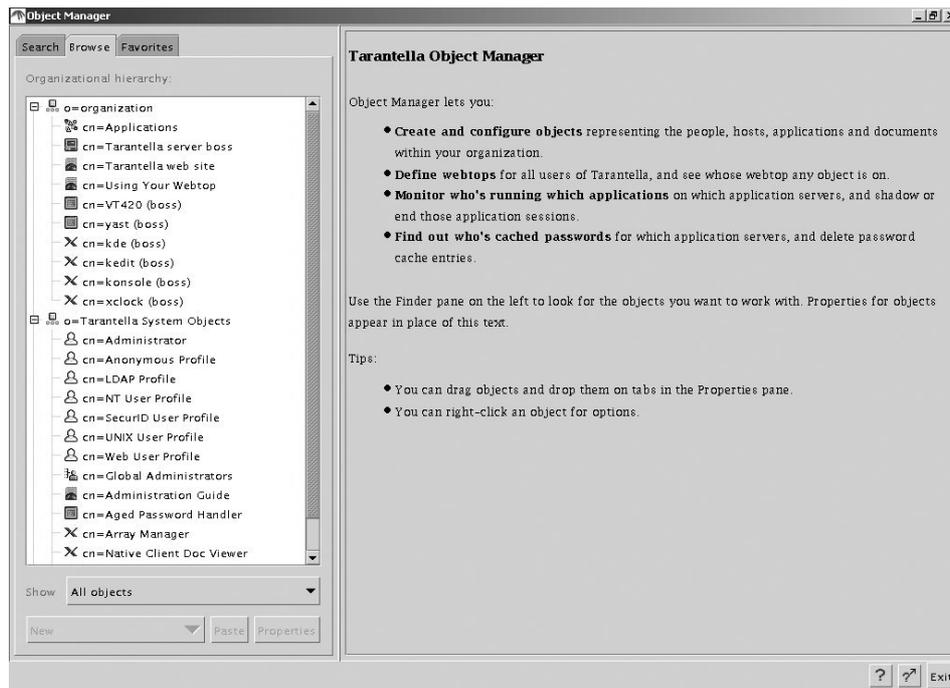


Abbildung 11.25: Object Manager

Um Darstellungsprobleme in Tarantella zu verhindern, sollten Administratoren als Standard Keyboard Layout das jeweilige Ländermapping einstellen.

Wählen Sie hierfür im Array Manager den Unterpunkt X-Protokoll Engine und ändern Sie die Einstellung in Keyboard Map von Automatic auf xgerman.txt (siehe Abb. 11.25). Diese Einstellung kann man User-bezogen ändern (z.B für Kollegen aus anderen Ländern).

Eine Auflistung der unterstützten Keyboards finden Sie unter `/opt/tarantella/etc/data/keymaps`.

```
boss:/opt/tarantella/etc/data/keymaps # ls
.          xfrenchcanadian.txt  xjapanese.txt.win
..         xgerman.txt           xlocales.txt
ansikey.txt xgreek.txt           xnorwegian.txt
merge     xicelandic.txt       xpolish_programmer.txt
```

```

uis                xitalian.txt          xportuguese.txt
vt420key.txt      xjapanese.txt          xrussian.txt
w60key.txt        xjapanese.txt.aix     xspanish.txt
xdanish.txt       xjapanese.txt.hpux    xswedish.txt
xdeswiss.txt      xjapanese.txt.linuxj  xuk.txt
xdutch.txt        xjapanese.txt.solaris xuniversal.txt
xfinnish.txt      xjapanese.txt.tru64
xfrench.txt       xjapanese.txt.uw7
boss:/opt/tarantella/etc/data/keymaps #

```

11.7.1 User für Tarantella anlegen

Die Benutzerverwaltung ist aufgrund der sehr hohen Anzahl möglicher Zugriffe (mehrere Zehntausend) sehr vielschichtig. Tarantella bietet acht verschiedene Arten einer Userauthentifizierung. Die gängigsten hierbei sind die Authentifizierung gegenüber lokal angelegten Unix-Accounts, Windows NT-Domänen, Webservern, Secure ID oder der Authentifizierung gegenüber einem Verzeichnisdienst (LDAP/Microsoft ADS-Server).

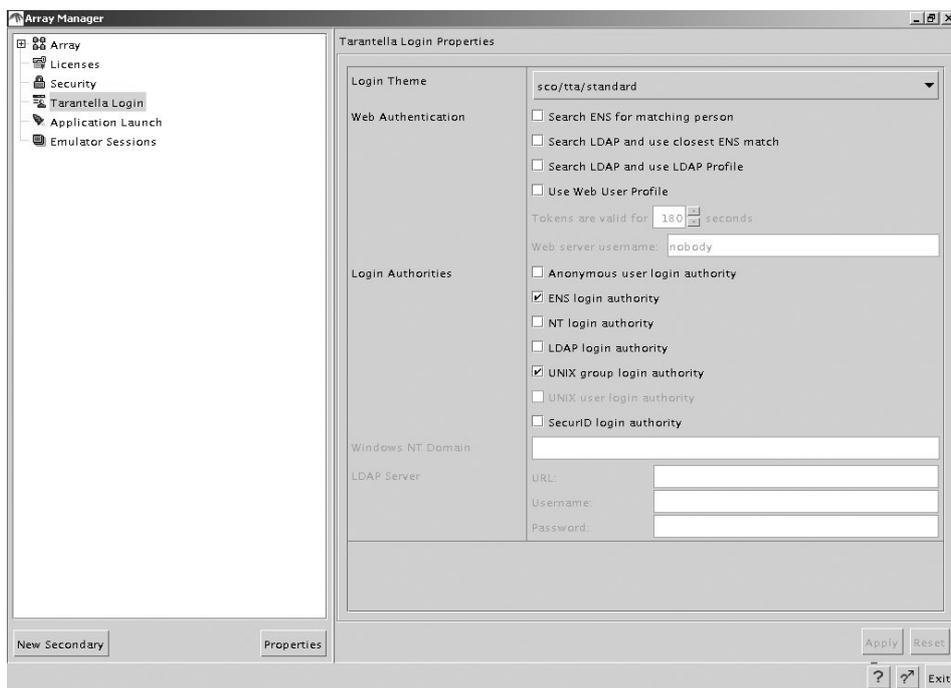


Abbildung 11.26: Userverwaltung in Tarantella

Ein dedizierter Server zur Benutzerverwaltung empfiehlt sich vor allem bei hohen Userzahlen oder wenn User auf mehrere unterschiedliche Applikations-Server zugreifen müssen. Die Userdaten werden dann nicht auf allen Applikations-Servern angelegt, sondern einmalig auf einem Verzeichnis-Server. Hier werden ebenfalls Userprofile, Mailadresse, Fileablage, Rechte usw. definiert.

Voreingestellt ist eine Authentifizierung gegenüber einem Unix-Account. Alle User mit einem Unix-Account auf dem Tarantella-Server können sich an Tarantella anmelden. Möchten Sie (für Tarantella) weitere User anlegen, so verwenden Sie am besten die Benutzerverwaltung von YaST oder YaST2. Das Linux-Kennwort ist gleichzeitig das Kennwort für Tarantella. Bitte legen Sie in YaST einen User *testuser* an und melden Sie sich mit dessen Namen an Tarantella an.

11.7.2 Applikationen zuordnen

Der Unterschied zwischen der Oberfläche des zuvor verwendeten Users *root* und dem eben eingerichteten *testuser* ist leicht zu erkennen. Dem *testuser* fehlen die administrativen Tools von Tarantella. Ein »Nicht-root«-Anwender hat weder Zugriff auf den Object-Manager noch auf den Array-Manager.

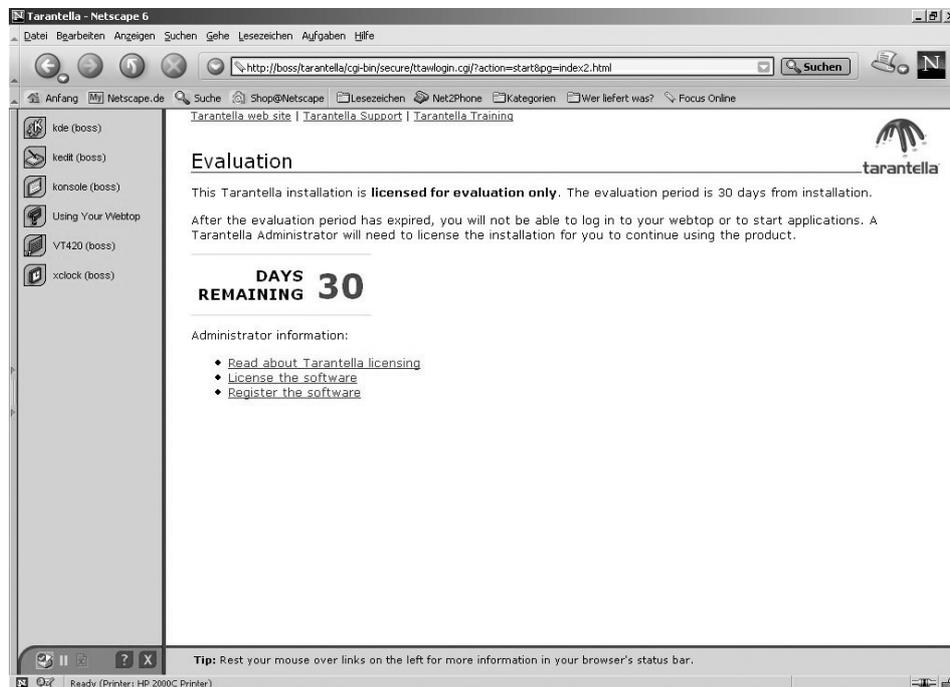


Abbildung 11.27: Anwenderdesktop

Welche Applikationen bestimmte User/Usergruppen sehen dürfen, legen Administratoren im Object-Manager fest.

Die Installation von Tarantella legt drei Grundstrukturen an. Diese sind im Object-Manager leicht als Wurzelverzeichnisse zu erkennen (siehe Abbildung 11.28)

- **Organization:** User- und Applikationenobjekte. Administratoren können zum Gliedern eigene Organisationseinheiten (OUs) bestimmen.
- **Tarantella System Objects:** Hier legt Tarantella Systeminformationen ab wie den Array-Manager und den Object-Manager und verwaltet Administratorrechte. Diesen Zweig kann man nur bedingt ändern.
- **Dc=de:** Bei Verwendung von Verzeichnisdiensten können die Strukturen, z.B. Object Units, Container, Gruppen usw., in Tarantella nachgebildet werden. Diese Gruppen können dann separiert für Tarantella konfiguriert werden. Sie werden in diesem Zweig angelegt.

Hinweis: Sowohl der Object- als auch der Array-Manger laufen auf dem Server in einem Tarantella-eigenen Java Runtime Enviroment. Aus diesem Grund gestaltet sich die Arbeit mit diesen stellenweise etwas träge. Achten Sie daher bei der Arbeit mit dem grafischen Administrationstool darauf, dass alle Änderungen wie eingestellt durch den Mausklick auf *Apply* übernommen werden.

Öffnen Sie im Object Manager den Verzeichniszweig *Tarantella System Objects*. Dort hat jedes Anmeldeverfahren für Tarantella ein eigenes voreingestelltes Benutzerprofil. Wenn ein Benutzer sich in Tarantella anmeldet, entscheidet der Tarantella-Server selbstständig, abhängig von der gewählten Authentifizierung (Unixgruppen, NT-Domänen, LDAP Profile), welches Profile und somit welche Anwendungen der Tarantella-User zugewiesen bekommt.

Wählen Sie das Profile *Unix User Profile* durch Doppelklick aus. Die anzuzeigenden Anwendungen definieren Sie im rechten Fenster auf der Registerkarte *Links*.

Um eine eigene Applikationen zuordnen zu können, markieren Sie mit der rechten Maustaste die *OU o=organization* im rechten Fenster und wählen *Remove*. Das Anzeigefenster der Registerkarte *Links* ist nun leer. Unix-Anwendern, die sich nun bei Tarantella anmelden würden, stünde keine Applikation zum Starten zur Verfügung.

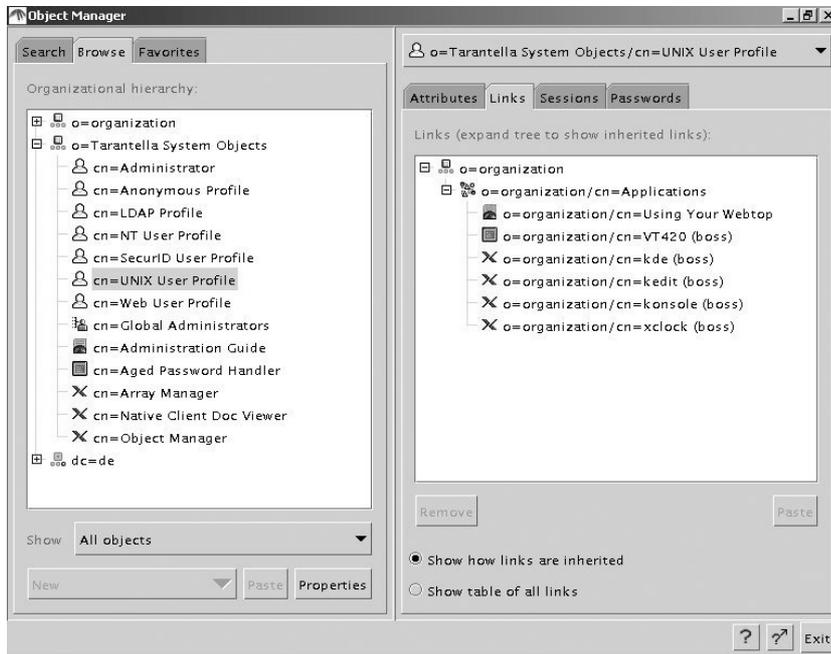


Abbildung 11.28: Profiles und Anwendungszuordnung

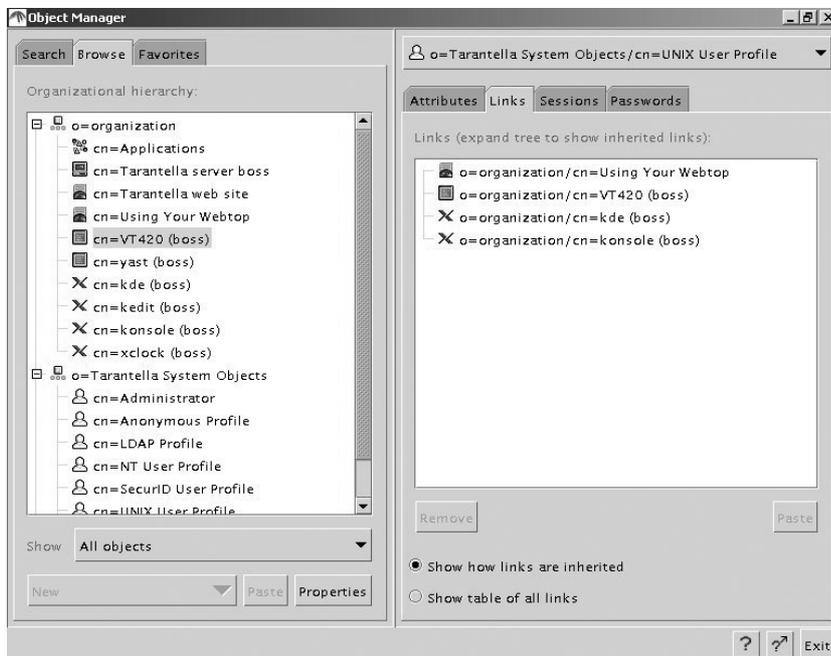


Abbildung 11.29: Definieren von Zuordnungen

Öffnen Sie nun gleichzeitig den Verzeichniszweig `o=organization`. Sie sehen die bei der Installation von Tarantella angelegten Applikationen (z.B. `xclock`, `VT420` usw.). Sie können diese Applikationen nun mit der Maus in die Registerkarte *Links* ziehen und somit die Anwendungen den *Unix Usern* unter Tarantella zur Verfügung stellen. In Abbildung 11.29 sind die Applikationen *Using your Webtop*, *VT420*, *KDE* und die *Konsole* zugewiesen. Sie können über diesen Weg pro Benutzer bis zu 512 Anwendungen bzw. Profile zuordnen.

11.7.3 Unix-Applikationen definieren

Eine neue Applikation legt man in drei Schritten an:

1. Definieren Sie den Host (Applikations-Server), auf welchem die Anwendung ablaufen soll (oder verwenden Sie den bereits angelegten).
2. Definieren Sie die Anwendung und verknüpfen Sie diese mit dem unter 1. definierten Host.
3. Weisen Sie die neue Applikation den Usern/Usergruppen zu.

Neue Objekte gleich welcher Art definiert man in Tarantella immer auf dem gleichen Weg:

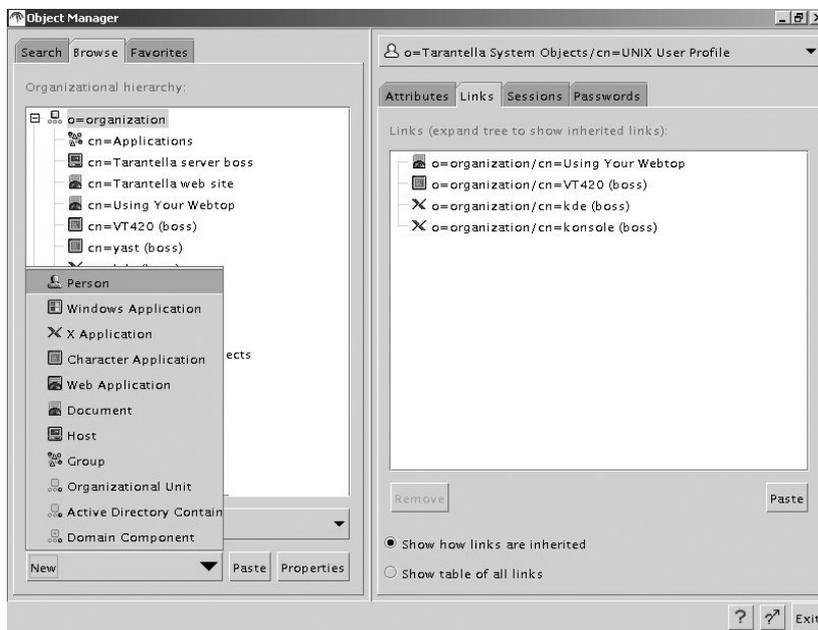


Abbildung 11.30: Neue Objekte erstellen

Wählen Sie zunächst den Verzeichniszweig, in welchem Sie das Objekt erstellen möchten, und wählen Sie dann die Schaltfläche *New*.

Erstellen Sie ein neues Hostobjekt und benennen es z.B. *Linuxserver* und bestätigen Sie Ihre Eingabe. Die rechte Fensterhälfte verlangt nun Einträge:

- *Name*: Geben Sie dem neu erstellten Objekt einen beliebigen Namen
- *Description*: Aussagekräftige Kurzbeschreibung
- *Adress*: Tragen Sie hier den vollständigen Hostnamen ein. Erfahrungsgemäß führen hier numerische IP-Adressen zu Schwierigkeiten. Testen Sie den Hostname zuvor am besten mit `ping` und `nslookup`.
- *Windows NT Domain*: Bei Verwendung von Windows-Applikations-Servern tragen Sie die Domäne oder den Rechnernamen ein. Dies entfällt bei Unix-/Linux-Server.
- *Authentication*: Wählen Sie, ob das Tarantella-Login auf den Server angewendet werden soll.

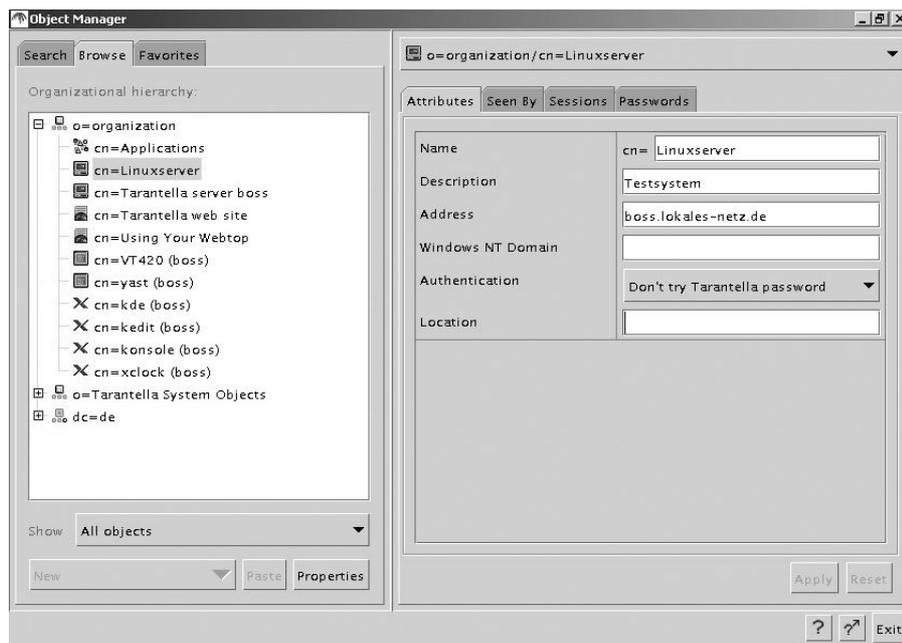


Abbildung 11.31: Neuer Host

Als zweiten Schritt zum Definieren einer Anwendung müssen Sie nun diese selbst auf dem gleichen Weg wie zuvor den Host definieren. Erstellen Sie ein neues Objekt und verwenden Sie als Objekttyp diesmal die Applikation, welche Sie veröffentlichen möchten. Zum Darstellen von X11-Anwendungen wählen Sie den Typ `X-Application` und benennen Sie diese hier.

Tarantella hat erneut die rechte Fensterhälfte verändert. Beim Bearbeiten der folgenden Einstellmöglichkeiten empfiehlt es sich, schon angelegte Objekte als Referenz zu verwenden:

- *Name*: Geben Sie dem neu erstellten Objekt einen aussagekräftigen Namen.
- *Description*: Aussagekräftige Kurzbeschreibung
- *Application*: Tragen Sie hier die Anwendung inkl. Pfad ein. Beispiele: `/opt/kde2/bin/startkde` (für eine KDE-Session), `/usr/dt/bin/startx` (CDE-Session), `/sbin/soffice` (Staroffice), `/sbin/xeyes` (Spiel) usw.
- *Connection Method*: Durch Tarantella erfolgt eine Anmeldung auf dem Applikations-Server über Zugangsprotokolle wie `ssh`, `telnet`, `rexec`, `rcmd` usw. Wenn `telnet` generell oder für den anzumeldenden User `disabled` ist (Standard bei SuSE 7.3 Professional), dann wird der Verbindungsaufbau von dritten Rechnern scheitern. Wählen Sie eine andere Konnektierungsmethode (z.B. über `ssh` oder `rexec`) oder ermöglichen Sie den Zugriff über `telnet` (definierbar in `/etc/rc.config`).
- *Resumable*: Hier legen Sie fest, was mit den Prozessen der Applikation geschehen soll, wenn diese nicht ordnungsgemäß beendet wurde (z.B. bei Verbindungsverlust einer GSM-Verbindung).
 - *Never*: Die Verbindung wird bei Verlassen des Screens beendet.
 - *Tarantella Webtop Session*: Die Verbindung bleibt so lange erhalten, wie man in Tarantella angemeldet ist.
 - *Always*: Die Prozesse der Applikation bleiben so lange erhalten, bis sie ordnungsgemäß beendet wurde (z.B. über *Datei • Beenden*)
- *Session Ends When*: Gibt an, wie der Tarantella-Server erkennt, wie ein Anwender die Tarantella-Applikation beendet.
- *Display using*: Hier können Sie festlegen, in welchem Fenstermodus die Anwendung läuft.
- *Webtop*: Das Programm wird im freien Bereich neben dem Launchpad angezeigt. Die Größe definieren Sie in den nächsten Zeilen unter *Width* und *Height*.

Dieser Modus eignet sich zur Ansicht kleiner Darstellungen wie `xclock`, `calc`, `yast`, `VT420` und `Mainframemasken`. Beispiel: Abbildung 11.21, `YaST`.

- *Independent Windows*: Das Programm läuft in einem eigenem Fenster (mit Rahmen) entsprechend den Größen in *Width* und *Height*. Dieser Modus eignet sich für Büro-Anwendungen wie `kWrite`, `MS Word`, `MS Excel`, `Star Office`. Beispiel: Abbildung 11.22, `Konsole`.
- *Client Windows Management*: X-Applikationen können das Fenstermanagement an den Tarantella-Client übergeben.

Das Fenster wird in der Größe durch die Applikation selbst vordefiniert, kann jedoch durch den Tarantella-User frei in der Größe geändert werden (durch Kleiner- bzw. Größer-Ziehen des Rahmens). Dieser Modus läuft

nicht mit Betriebssystemoberflächen, sondern eignet sich nur für einzelne X11- bzw. Javaprogramme (KWrite, xclock). Beispiel: Abbildung 11.22, xclock.

- **Kiosk:** Das Programm wird ausschließlich auf dem Client ohne Rahmen und in der maximalen Größe dargestellt. Das Betriebssystem des Tarantellaclients wird vollständig durch die Tarantella-Applikation überdeckt. Dies empfiehlt sich zur Darstellung von Betriebssystemoberflächen (KDE, Gnome, CDE, Windows-Desktop).
- **Colour Depth:** Hier können Sie die Farbtiefe bestimmen. Bei X-Applikationen beträgt diese mindestens 8 Bit (256 Farben) und höchstens 24 Bit (16 Mio. Farben). Je mehr Farben Sie verwenden, desto mehr Bandbreite benötigt die Übertragung. Der Ressourcenbedarf (Speicher/CPU-Last pro User) auf dem Tarantella-Server ändert sich dabei nur unwesentlich.
- **Webtop Icon:** Wählen Sie hier das gewünschte Icon aus.
- Das hier ausgewählte Icon wird auf dem Launchpad angezeigt und repräsentiert dem Anwender die konfigurierte Applikation.
- Definieren Sie nun wie hier im Beispiel in Abbildung 11.32 die Applikation YaST2 und bestätigen Sie Ihre Angaben mit der Schaltfläche *Apply*:

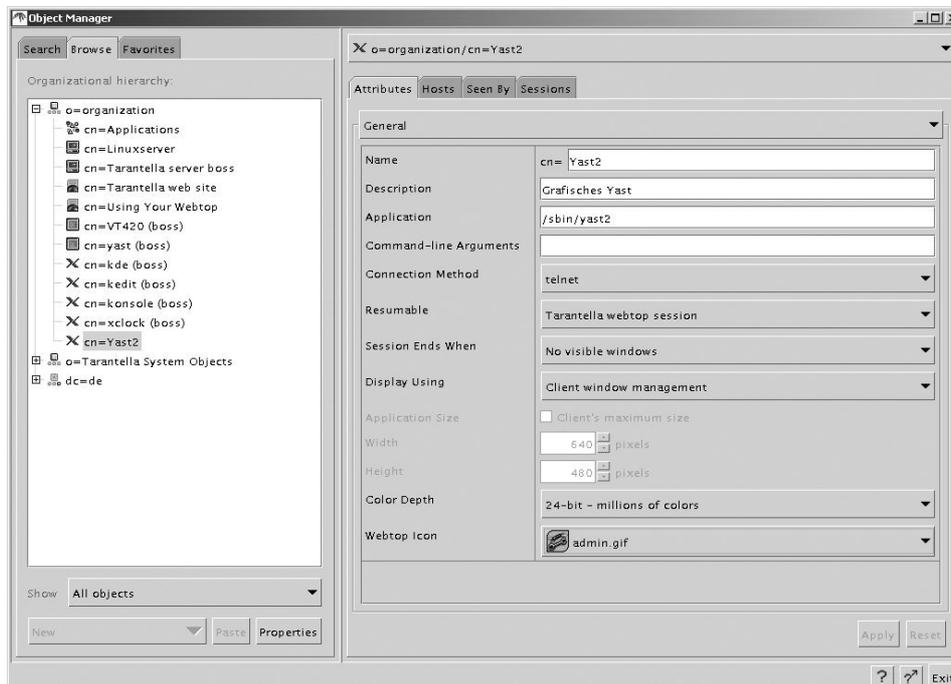


Abbildung 11.32: Neue Applikation »YaST2«

Nun ist die Applikation konfiguriert, jedoch ist noch nicht definiert, auf welchem Server sie ablaufen soll. Wechseln Sie dazu auf den Tab *Hosts* und ziehen Sie das zuvor angelegte Objekt *Linuxserver* von der linken in die rechte Fensterhälfte.

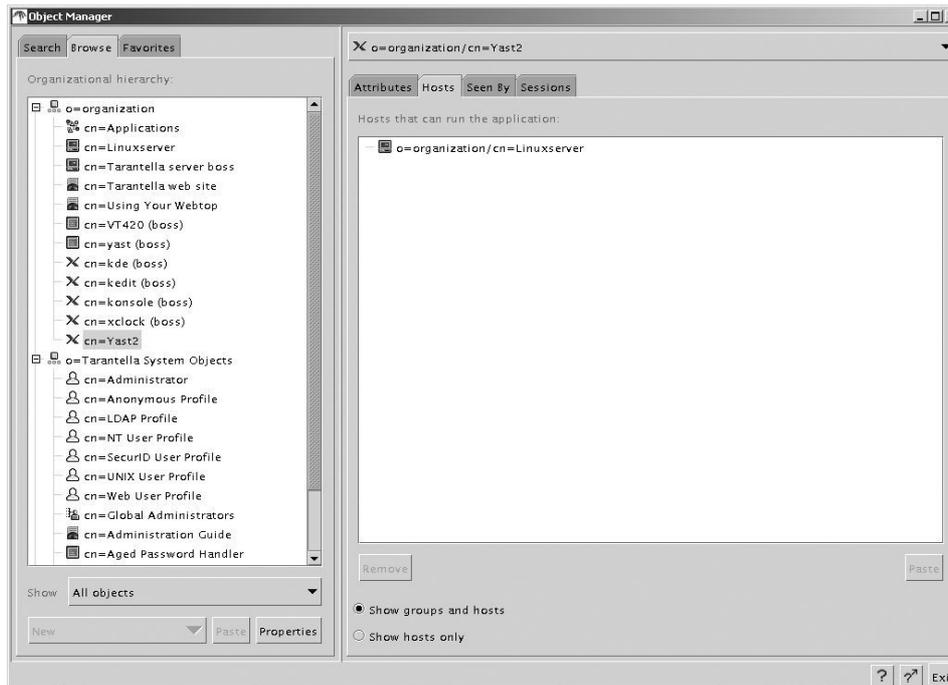


Abbildung 11.33: Zuordnung Applikation »YaST2« zu Host »Linuxserver«

Als dritten und letzten Schritt müssen Sie nun wie bereits zuvor beschrieben die Applikation für die Gruppe Unix-User verfügbar machen. Dazu ziehen Sie das neue Objekt *YaST2* auf die Registerkarte *Links* des Objekts *Unix User Profile*.

Allen Usern, die sich über die Unix-Gruppen anmelden, steht nun die neue Applikation *YaST2* zur Verfügung.

Tipp: Sollten User während der Konfiguration bereits in Tarantella angemeldet gewesen sein, so aktualisiert Tarantella deren Launchpad erst bei erneutem Einloggen.

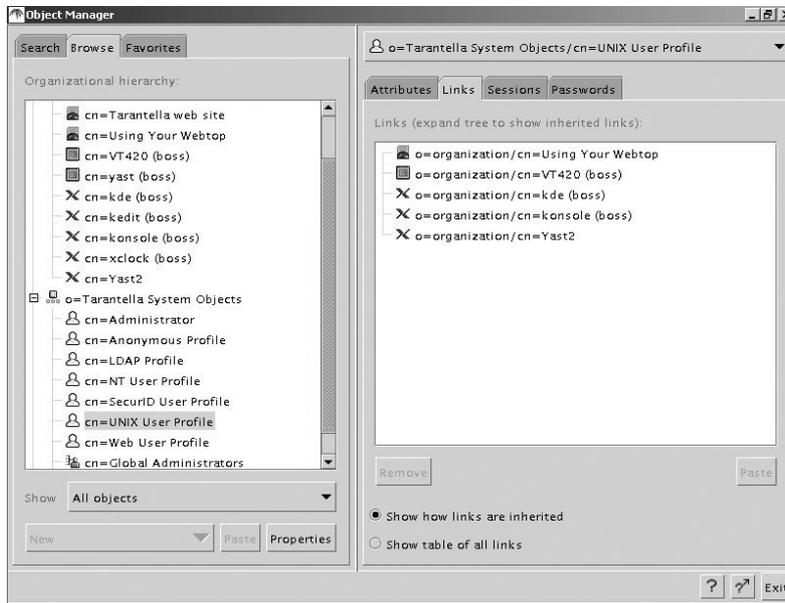


Abbildung 11.34: Zuordnung Applikation »YaST2« zu Gruppe »Unix User Profile«



Abbildung 11.35: Applikation »YaST2«

11.7.4 Windows-Anwendungen definieren

Das Installieren und Konfigurieren von Windows 2000 Terminaldiensten und darauf laufenden Anwendungen gestaltet sich oft aufwändig. Die folgenden Erklärungen gehen daher von einem funktionierenden Windows 2000 (Advanced) Server mit freigeschalteten Terminaldiensten aus.

Für die Zusammenarbeit zwischen Tarantella und Windows 2000 Terminal-Servern sollte man vorab zwei Grundeinstellungen ändern. Manche Funktionen von Tarantella können auch Microsofts Windows Terminal-Server bieten (z.B. gibt es das Session-Resume auch in Microsoft Windows). Es ist offensichtlich, dass man zusammengehörende Funktionen passend konfigurieren muss.

- So empfiehlt es sich, die Kennwortabfrage der Microsoft Windows Terminal-Server zu deaktivieren, damit Tarantella die Anmeldung durchreichen kann, ohne dass die Benutzer sich ein zweites Mal bei Microsoft Windows anmelden müssen.
- Da man das Resume-Verhalten von Sessions auch auf Microsoft Windows Terminal-Servern einstellen kann, sollte man sich entscheiden, welcher Server das Resume-Verhalten kontrollieren soll, Tarantella oder die Microsoft Windows Terminal-Server.

Sie können beide Einstellungen auf dem Microsoft Windows Terminal-Server im Menüpunkt *Start • Programme • Verwaltung • Terminaldienstkonfiguration* ändern. Wählen Sie dieses Programm und hier das Verzeichnis *Verbindungen • RDP*.

Durch Doppelklick öffnet die *Terminaldienstkonfiguration* vom Microsoft Windows 2000 Server das Fenster *Eigenschaften RDP-TCP*. Wählen Sie zunächst die Registerkarte *Anmeldeeinstellungen* und deaktivieren Sie den Punkt *Kennwort immer anfordern*, damit Tarantella Kennwörter an den Microsoft Windows Terminal-Server weiterreichen kann.

Wechseln Sie nun auf die Registerkarte *Sitzungen* und definieren Sie, die Microsoft Windows-Sitzung zu beenden, wenn die Verbindung getrennt wurde. Dies bewirkt, dass die Sitzungskontrolle bei Tarantella liegt.

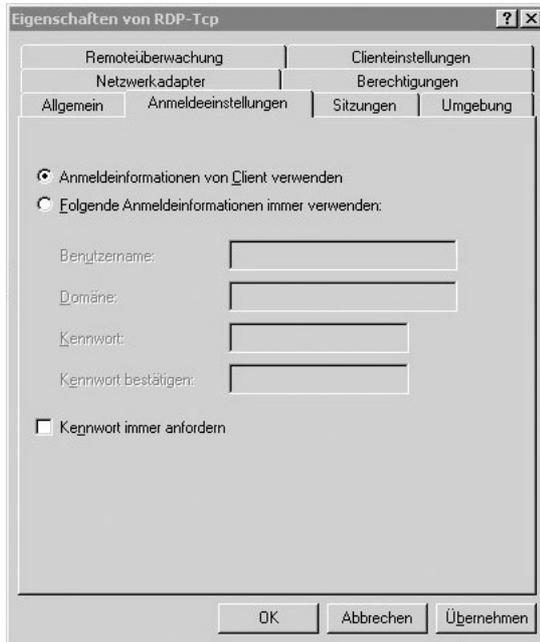


Abbildung 11.36: Einstellungen RDP TCP / Kennwort

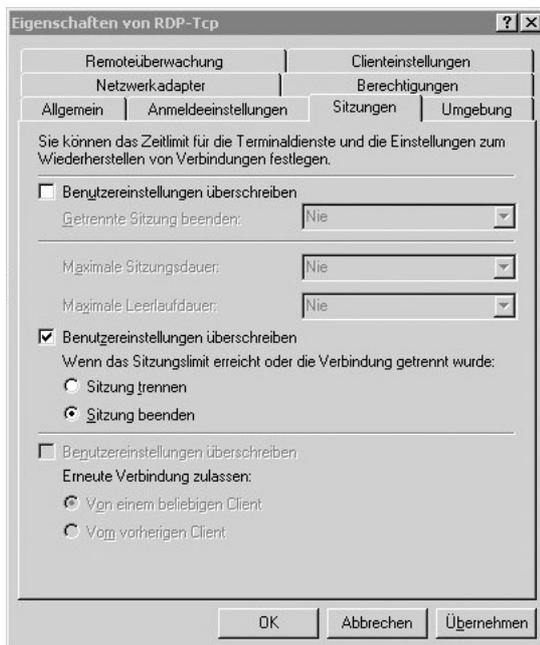


Abbildung 11.37: Einstellungen RDP TCP / Sitzungen

Konfiguration eines Windows-Objektes

Windows-Sitzungen legt man nach dem gleichen Schema wie X-Applikations-Sitzungen an.

1. Definieren Sie zunächst ein Hostobjekt mit der (Host-) Adresse des Terminal-Servers.
2. Legen Sie ein neues Objekt `Windows Application` an.
3. Weisen Sie das Objekt der gewünschte Usergruppe zu.

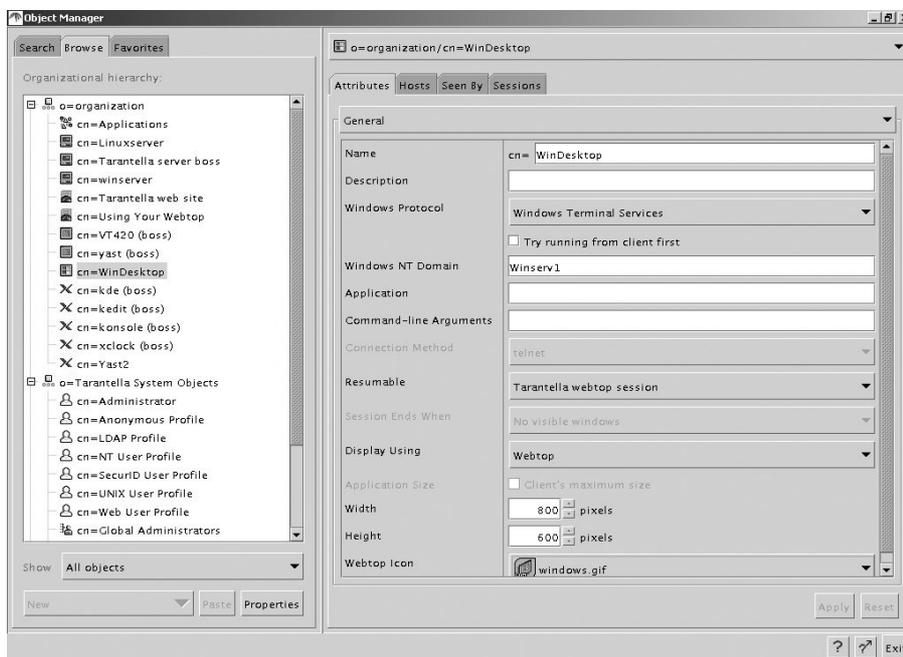


Abbildung 11.38: Konfiguration einer Windows Applikation

Unterschiede zwischen X- und Windows-Applikationen.

Es gibt einige Unterschiede zwischen X- und Windows-Applikationen und deren Konfiguration für Tarantella:

- *NT-Domain:* Tragen Sie hier bitte die NT-Domäne für den Windows-Applikations-Server ein. Sollten Anwender nicht mit einer Domäne arbeiten, so tragen Sie bitte den Hostnamen des Terminal-Servers ein.
- *Application:* Genauso wie bei X-Applikation tragen Sie hier die Anwendungspfade für Einzelapplikationen ein (z.B. c:\Programme\Microsoft Office\Office\winword.exe). Tragen Sie in dieses Feld nichts ein, stellt Tarantella den ganzen Windows Desktop dar.

- *Display Using:* Das Client Windows Management steht leider nicht für Windows-Applikationen zur Verfügung. Ein Anpassen der Größe eines Fensters einer Tarantella-Windows-Applikation ist für Windows-Applikation über Tarantella (noch) nicht möglich.
- *Connection Method:* Windows verwendet ein eigenes Protokoll. Eine Authentifizierung findet über RDP statt und kann nicht abgeändert werden.
- *Colour Depth:* Wie bereits am Anfang erwähnt, steht für die Verbindung zwischen Windows-Applikations-Server und Tarantella-Server nur eine Farbtiefe von 8 Bit zur Verfügung. Deswegen stehen auch auf der Verbindung zwischen Tarantella-Client und Tarantella-Server nicht mehr Farben zur Verfügung.

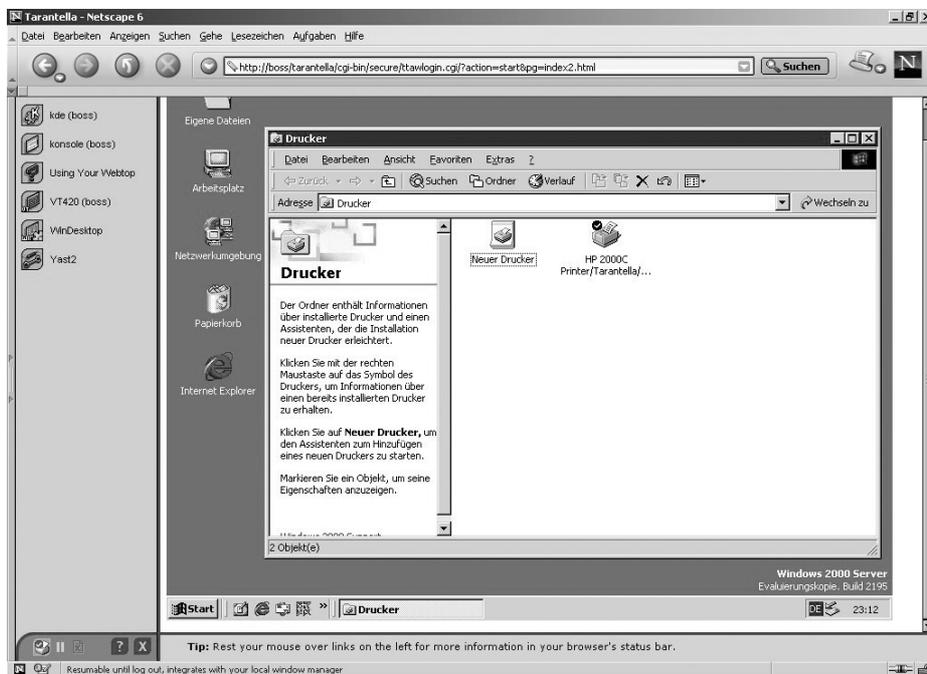


Abbildung 11.39: Windows-Desktop im Tarantella-Webtop

11.8 Drucken unter Tarantella

Beim Vermitteln von Sitzungen durch Tarantella können Sie im LAN herkömmliche Netzwerk-Druckmethoden beibehalten. Wenn Sie jedoch über WAN-Strecken drucken möchten, so empfiehlt sich der Einsatz der Druckdienste von Tarantella. Diese senden die Druckdaten ebenfalls über AIP (bei Bedarf verschlüsselt und komprimiert) an den Standarddrucker des Clients.

Tarantella selbst fungiert hierbei als zusätzlicher Druckserver, der Druckdaten empfangen und automatisiert an Clients versenden kann.

Der Tarantella-Client erkennt den Clientdrucker automatisiert. Bei einer Anmeldung an Tarantella liest eine Javaklasse (bzw. der Native Client) den Standarddrucker des Clients aus und teilt ihn dem Tarantella-Server mit.

Generell unterscheidet man das Drucken von Microsoft Windows-Terminal-Servern und von Unix/Linux-Applikations-Servern: Bei der Verwendung von Applikationen basierend auf Microsoft Windows 2000-Terminal-Servern wird bei dem Start einer Applikation der Tarantella-Client-Drucker an den Windows Server mitgeteilt und automatisch eingerichtet. Dies basiert auf der RDP-Druckeinrichtungssteuerung des Terminal-Servers.

Möchten Sie von einem Unix/Linux-Applikations-Server drucken, so müssen Sie das Script `prtinstall.en.sh` von `/opt/tarantella/bin/bin/scripts` auf den Applikations-Server kopieren und ausführen. Als einzige Eingabe brauchen Sie den Namen des Tarantella-Servers anzugeben, auf welchen gedruckt werden soll. Der eingerichtete Drucker heißt dann `tta_printer`.

```
# sh prtinstall.en.sh

DNS name of primary Tarantella server
(press [Return] to accept default [appsserver.test.com]):
Installing printer configuration file...
Installing lp/lpr "wrapper" scripts...
Local system type is SPARC Solaris.
Informing lp subsystem about local printer tta_printer...
Running: lpadmin -p tta_printer -v/dev/null -ilp_interface.en -lany -
onobanner
Running: enable tta_printer
UX:enable: WARNUNG: Ziel "tta_printer" wurde bereits aktiviert.
status=1
  (ignored)
Running: accept tta_printer
Success.
#
```

Die grundlegenden Funktionen von Tarantella haben Sie nun kennen gelernt. Die Funktionen und Konfigurationen von Tarantella sind sehr vielseitig und würden den Umfang dieses Buches bei weitem sprengen. Wenn Sie weitere Funktionen einrichten oder Tarantella individuell anpassen möchten, blicken Sie bitte in die anfangs beschriebenen weiterführenden Dokumente:

- **Admin Guide:** Den Admin Guide finden Sie als html-Version als Applikation bei Anmeldung als Tarantella-Admin in Tarantella oder im Internet unter http://www.tarantella.com/knowhow/e3.2/help/en-us/admintocs/TOC_FUNC_TYPE.html
- **Newsgroup:** www.tarantella.com/support/newsgroups
- **Know how:** <http://www.tarantella.com/knowhow/e3.2/>
- **Weiterführende Dokumentation:** <http://www.tarantella.com/knowhow/e3.2/download.html>

