

19 Terminal-Dienste für Windows

Um den Administrations- und Betreuungsaufwand auf der Benutzerseite drastisch zu senken, kann man das Ausführen und Darstellen von Anwendungen so aufspalten, dass Anwendungen auf Anwendungs-Servern laufen und Benutzer mit den Anwendungen auf Terminals aus oder auf wie Terminals genutzten PCs arbeiten. Hier spricht man von Server-basierter Datenverarbeitung oder von Terminal-Diensten.

Während Kapitel 17 Terminal-Dienste für Linux-Anwendungen beschreibt und Kapitel 16 Lösungen für schlanke Endgeräte, zeigt dieses Kapitel Konzepte und kommerzielle Lösungen für den strategischen Einsatz von Linux-Servern und Linux-Endgeräten bei Terminal-Diensten für populäre Windows-Anwendungen. Da die jeweiligen Hersteller ihre kommerziellen Produkte mit Handbüchern und Installations-Support anbieten, verzichtet dieses Buch hier weitgehend auf Beschreibungen der Installation und Konfiguration.

19.1 Überblick

Zunächst gibt es zwei verschiedene Grundkonzepte für Windows-Terminal-Dienste:

Sie können von Linux-Servern angeboten werden oder von Windows-Servern mit freigeschalteten Terminal-Diensten:

Server-basierte Windows-Anwendungen lassen sich direkt von Linux-Servern mit

- dem Win4Lin Terminal-Server von NeTraverse und der Installation von Windows 95/98-Lizenzen (siehe Abschnitt 19.2) sowie
- dem CrossOver Office-Server von Codeweavers völlig ohne Windows-Lizenzen (siehe Abschnitt 19.3)

verwirklichen. Aus Benutzersicht ist deren Funktion identisch mit den Einzelplatz-Versionen, die in den Abschnitten 3 und 4 des vorangegangenen Kapitels 18 beschrieben sind. Mit den Endgeräten (Linux-Terminals, siehe Abschnitte 16.1, 16.2 und 16.4) kommunizieren diese Terminal-Server per X.11-Protokoll. Die Geschwindigkeit an den Terminals lässt sich mit Nomachine NX als Transportprotokoll steigern.

Verbreiteter sind Lösungen, die auf Windows 2000/2003-Servern mit freigeschalteten Terminal-Diensten aufsetzen. Linux-Endgeräte können Windows-Terminal-Dienste, die das proprietäre Transportprotokoll *RDP* von Microsoft verwenden,

- direkt mit freien Clients wie *rdesktop* oder kommerziellen Clients von Tarantella oder HOB nutzen,
- mit Linux-Clients für die proprietäre Independent Computing Architecture (*ICA*) von Citrix beziehen, wobei auf dem Windows-Server dann ein Metaframe-Server von Citrix das RDP durch ICA ersetzt,
- auf dem Umweg über einen Linux-basierten Middleware-Server von Tarantella beziehen, wobei die Enterprise Edition des Secure Global Desktops mit dem Windows-Server über RDP kommuniziert und letzterer selbst – wie hier schon im Abschnitt 17.3 beschrieben – mit beliebigen Endgeräten, also auch Linux-Terminals, mit dem proprietären Protokoll AIP Daten austauscht oder
- vermittelt durch Nomachine NX bekommen, wobei Nomachine NX auf einem Linux-Middleware-Server mit dem Windows-Server über das Microsoft-Protokoll RDP spricht und mit den Endgeräten (Linux- und Windows-Geräten) über das schlanke NX-Protokoll.

Käufer und Systemverwalter sollten unter Gesichtspunkten wie Sicherheit, Verfügbarkeit, Lizenzkosten und laufende Betriebskosten prüfen, welche Linux-Dienste sie auf Servern und Clients für das Nutzen von Windows-Programmen auf Linux-Endgeräten nutzen wollen.

Die folgenden Abschnitte wollen hier eine erste Orientierung erleichtern.

19.2 Win4Lin Terminal-Server

Die Win4Lin Terminal-Server sind die Server-Version der schon im Abschnitt 18.4 beschriebenen Gerätetreiber Win4Lin zum Ausführen von Windows 9.x auf Linux.

Diese Mehrplatzlösung von NeTraverse (<http://www.netraverse.com>) zum Versorgen von Linux-Arbeitsplätzen mit Windows 98 und darauf aufsetzenden Windows-Anwendungen hat die Redaktion über zwei Jahre erprobt und als stabil und betriebssicher klassifiziert. Sie setzt auf einem Linux-Terminal-Server auf, darauf für jeden Anwender oder Arbeitsplatz eine (clonbare) Windows 98-Installation und darauf für jeden Anwender oder Arbeitsplatz wie bei nativem Windows clonbare Windows-Anwendungen.

Die Installation des Terminal-Servers entspricht der in Abschnitt 18.4 beschriebenen Einzelplatz-Version. Der auffälligste Unterschied besteht im Aussehen des Installationsprogrammes.

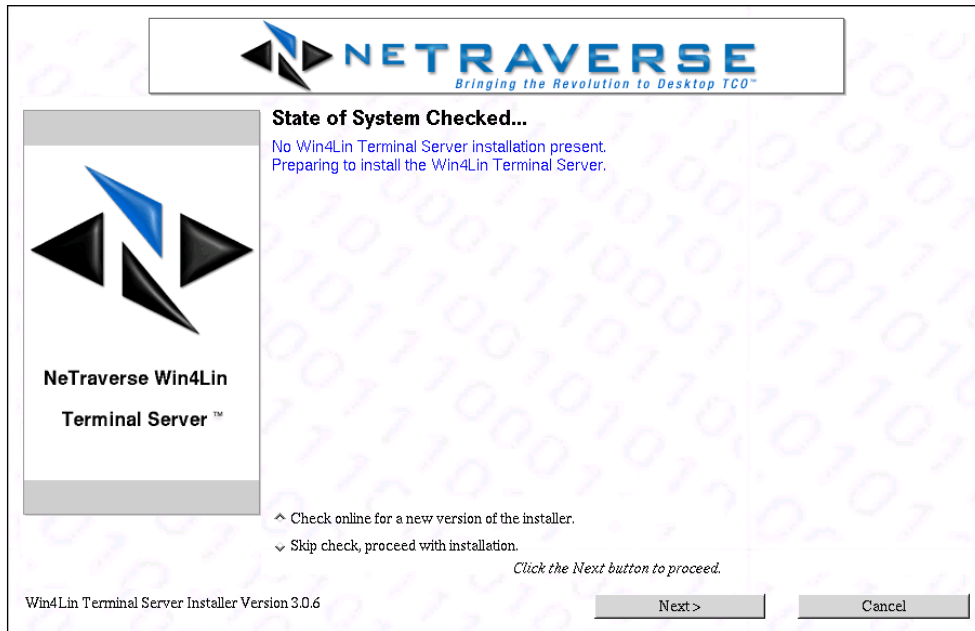


Abbildung 19.1: NeTraverse – Terminal-Server Installer

Im letzten Schritt der Installation richten Sie auch bei der Terminal-Server-Version für alle Benutzer Windows anhand der bereits eingelesenen Medien ein. Hieraus ergibt sich, dass alle Windows-Installationen mit derselben Windows-Version arbeiten müssen.

19.2.1 Windows-Installation für die Benutzer

Wenn der Server über genügend Plattenkapazität und Ihre Organisation über genügend Lizenzen verfügt, können Sie für jeden Benutzer eine eigene Windows-Installation erstellen. Das können die Benutzer auch selbst vornehmen, indem sie unter KDE in einer Shell

```
win
```

eingeben. Wenn das System dann erkennt, dass für einen Benutzer noch kein Windows 98 installiert ist, bietet es an, dieses jetzt einzurichten. Da die Installation auf den bereits eingespielten Windows-Quelldateien aufbaut, müssen Benutzer dafür keinerlei Medien, sondern nur den Lizenzschlüssel einer Windows 9.x-Lizenz kennen. Auch hier besteht kein weiterer Unterschied zur Einzelplatz-Version.



Abbildung 19.2: NeTraverse – Windows 98-Benutzerinstallation

Bei der Installation legt Win4Lin drei Verzeichnisse im Home-Verzeichnis des Benutzers an:

| | |
|--------|---|
| win | Hier liegt die eigentliche Windows-Installation. |
| merge | Hier liegen Konfigurationseinstellungen. |
| mydata | Verzeichnis für Benutzerdaten, erreichbar als d : |

Wenn Sie die nicht besonders zeitraubende Installation für jeden Benutzer ablaufen lassen wollen, können Sie die erste Installation einfach kopieren und anschließend für jede Kopie die Voreinstellung `servername-benutzername` des Rechnernamens ändern.

Für das bequeme Sichern und Zurückspielen der Installationen liefert NeTraverse auf der CD-Rom im Verzeichnis `\misc\tools` das kleine Shell Script `Nssetemplate`. Dieses kann Installationen gepackt als sog. Templates ablegen. Die Templates kann man dann einfach wieder mit diesem Ordner in ein Benutzerverzeichnis zurückkopieren.

19.2.2 Rechnerbezogene Windows-Installation

In Bildungseinrichtungen, in denen es erheblich mehr Lernende als Computer-Arbeitsplätze gibt, kann man Windows 98 arbeitssparend für jeden Linux-Client-PC statt für jeden Benutzer einrichten.

Prüfen Sie bitte in Microsofts Lizenzbedingungen, ob dies mit den Einzel- oder Campus-Lizenzen Ihrer Organisation für Windows 98 und Windows-Anwendungen erlaubt ist. Wenn ja, gehen Sie bitte wie folgt vor:

- Installieren Sie Microsoft Windows 98 und alle erforderlichen Windows-Anwendungsprogramme für einen beliebigen Nutzer auf dem Linux-Server.
- Kopieren Sie nach der Installation das `win`-Verzeichnis in eine genügend große Partition (z. B. `/opt/win4lin/`), und benennen Sie diese so wie einen der Linux-Client-Rechner. Anschließend erstellen Sie für jeden Linux-Client eine Kopie dieses Verzeichnisses, die genauso heißt wie diese Rechner (ohne Domain-Anteil).

Da Windows »meckert«, wenn bei bestimmten Systemdateien das Datum verändert ist, müssen Sie beim Kopieren die Eigenschaften und insbesondere die Herstellungsdaten/-zeiten der Dateien bewahren. Dazu sollten Sie stets mit dem Parameter `-a` (`cp -a`) kopieren:

```
cp -a rechner1 rechner2
```

Nun benötigen Sie noch ein Skript, das

- bei der Anmeldung eines Benutzers feststellt, ob es für den Client-Rechner, an dem er mit Windows arbeiten will, eine Installation gibt und
- wenn ja, im Home-Verzeichnis des Benutzers einen Link namens `win` auf dieses Verzeichnis anlegt.

Mit »geliehenen« Rechten des Systemverwalters `root` muss sich der Benutzer dann noch das Installationsverzeichnis übereignen lassen, um danach Windows zu starten.

Das erste Perl-Skript namens `windows` können Sie im Verzeichnis `/usr/local/bin` speichern. Es ermittelt den Benutzernamen des Aufrufenden und den Namen des Rechners, von dem er es aufruft. Wenn es für den ermittelten Rechner eine Windows-Installation gibt, dann legt es im Home-Verzeichnis einen Link auf das zugehörige Verzeichnis an.

```
/usr/local/bin/windows
```

```
#!/usr/bin/perl -w
# Uwe Debacher und Bernd Burre 2004
# die Verzeichnismomanipulationen erfordern Root-Rechte

# wo liegen die Windows-Installationen
```

```

$winpfad="/opt/win4lin";

# Benutzernamen ermitteln
$uid=$<;
$username=(getpwuid($uid))[0];
if ($username=~ /^(.+)$/) {
    $username = $1;
}
if ($username eq "") {
    die "Fehler: Ich kann den Benutzernamen nicht ermitteln\n";
}

# der volle Hostname steckt in der letzten Spalte
@last= `usr/bin/last -n 1 -a $username`;
$host= (split (/s+/, $last[0]))[9];
if ($host eq "") {
    die "Fehler: lokales Login auf dem Server\n";
}

# den Domainteil verwerfen
$host = (split /\./, $host)[0];
if ($host eq "") {
    die "Fehler: ich kann den Rechnernamen nicht ermitteln\n";
}

if ($host eq "" || $host eq "wtmp") {
    die "Fehler: Ich kann den Anmelde-Rechner nicht ermitteln\n";
}

# ist fuer diesen Rechner eine Installation vorhanden??
if (!-e "$winpfad/$host") {
    die "Fehler: das Verzeichnis $winpfad/$host fehlt\n";
}

# es darf keine Datei und kein Verzeichnis win geben
if (-e "win" && !-l "win") {
    die "Fehler: Ich kann den Link win nicht anlegen, es gibt
    ↵ eine solche Datei\n";
}

# vorhande win-Links müssen weg
if (-l "win") {
    print "lösche vorhandenen Link win\n";
    unlink ("win");
}

# den Link win anlegen
print "Lege Link win an\n";

```

```
symlink("$winpfad/$host", "win");

# fuer die Links werden Rootrechte benoetigt
system "wrapper";

# Erfolg testen
if (!-o "win/autoexec.bat") {
    die "Fehler: Die Eigentumsverhältnisse stimmen nicht\n";
}

# weiter mit den normalen Rechten
print "Starte Windows\n";
exec "win"
```

Bevor das Skript den Link `win` anlegt, entfernt es einen eventuell schon vorhandenen Link. Wenn es ein Verzeichnis `win` findet, dann bricht es mit einer Fehlermeldung ab.

Einen wichtigen Teil der Arbeit erledigt das zweite Skript, das Sie über ein kleines C-Programm namens *wrapper* starten.

Das zweite Programm benötigt Root-Rechte. Da man in Perl-Skripten nicht einfach `Setuid` setzen kann, muss ein kleines C-Programm das Perl-Skript starten. Bei diesem C-Programm muss man mit

```
chmod u+s wrapper
```

das `Setuid`-Bit setzen.

Das Perl-Skript muss die Windows-Installation dem aufrufenden Benutzer übereignen, damit Windows schreibenden Zugriff auf seine eigenen Dateien bekommt. Dazu muss es den Benutzernamen ermitteln sowie den Rechner, von dem aus es aufgerufen wurde.

```
/usr/local/bin/links.pl

#!/usr/bin/perl
# (c) Uwe Debacher und Bernd Burre Dezember 2004

# den Pfad zu den Windows-Installationen bitte anpassen
$pfad="/opt/win4lin";

# wir brauchen Root-Rechte
$uid="$<";
$< = "$>";
$oldpath=$ENV{'PATH'};
$ENV{'PATH'}="/bin:/sbin:/usr/bin:/usr/sbin";

# Benutzernamen ermitteln
```

```

$username=(getpwuid($uid))[0];
if ($username=~ /^(.+)$/) {
    $username = $1;
}

# der volle Hostname steckt in der letzten Spalte
@last= `/usr/bin/last -n 1 -a $username`;
$host= (split (/s+/, $last[0]))[9];
if ($host eq "") {
    die "Fehler: lokales Login\n";
}

# den Domainteil verwerfen
$host = (split /\./, $host)[0];
if ($host eq "") {
    die "Fehler: Ich kann den Rechnernamen nicht ermitteln\n";
}

# den Pfad zum Windowsverzeichnis montieren
$pfad=$pfad . '/' . $host;
if ($pfad=~ /^(.+)$/) {
    $pfad=$1
}

# wenn es den Pfad nicht gibt, dann stimmt was nicht
if ( !-d "$pfad" ) {
    die "Fehler: $pfad nicht gefunden\n";
}

# nur können wir endlich übereignen
print "Übereigne $pfad an $username\n";
system "/bin/chown -R $username $pfad";

```

Ein großer Teil des Skriptes dient dazu, die Sicherheitsanforderungen von Perl hinsichtlich von Setuid-Programmen zu erfüllen. Perl erwartet, dass die Eingabedaten überprüft werden. Weitere Hinweise hierzu finden sich in der Manpage perlsec.

Wenn Sie gerne C-Quelltext lesen, finden Sie hier den kommentierten Code des Wrapper-Skripts.

```

/*
*/
#include <unistd.h>
#include <sys/types.h>
// compilieren: gcc -o wrapper script_suid.c
// danach:      chmod u+s wrapper
int main(argc,argv)

```



```
int argc;
char *argv[];
{
    extern char **environ;
    int c;
    c=execve("/usr/local/bin/links.pl",argv,environ);
    return c;
};
/* *** EOF *** */
```

Damit ist das System funktionsfähig, und alle Benutzer können ihre Windows 98-Umgebung starten, indem sie in einem Terminalfenster unter KDE `windows` eingeben.

Ein bisschen einmaliger Turnschuh-Sport bleibt Ihnen noch. Als Administrator müssen Sie einmal an jeden der Linux-Clients heran und den Namen für den zugehörigen virtuellen Rechner ändern, da diese nach dem Kopieren alle gleich lauten.

Mit etwas Handarbeit können Sie diese Arbeiten von einem Linux-Client aus erledigen, indem Sie die Links passend setzen und die Windows-Rechner nacheinander starten und konfigurieren.

Win4Lin verteilt die MAC-Adressen der virtuellen Netzwerkkarten automatisch jeweils in der Anmeldereihenfolge, um die Eindeutigkeit zu gewährleisten.

19.2.3 Feintuning mit KDE

Um Anwender erst gar nicht mit einem Terminal-Fenster zu konfrontieren, können Sie allen Benutzern ein Icon namens *Windows* auf ihre Desktops legen, mit dem sie den Emulator und Windows starten können.

Erstellen Sie dazu einen Desktop-Link in Form einer einfachen Textdatei.

```
[Desktop Entry]
Comment[de]=
Exec=/usr/local/bin/windows
Icon=HandOnWindowCloseIcon
MimeType=
Name[de]=Windoofs
Path=
ServiceTypes=
SwallowExec=
SwallowTitle=
Terminal=false
TerminalOptions=
Type=Application
X-KDE-SubstituteUID=false
X-KDE-Username=
```

Diese Datei muss für alle Benutzer im Verzeichnis `KDesktop` ihrer Home-Verzeichnisse stehen. Eventuell müssen Sie diese Verzeichnisse und ebenso die Verzeichnisse `.merge` und `mydata` erst noch erstellen.

Als Vorlage kopieren Sie die Verzeichnis aus dem Home-Verzeichnis des Benutzers, für den Sie Windows installiert haben, in das Verzeichnis `/etc/skel` und geben für alle Benutzer die Leserechte.

Beim Anlegen eines neuen Benutzers kopieren obige Befehle alle Dateien und Ver-

```
cp -r /home/dummyuser/KDesktop /etc/skel
chmod -R a+r /etc/skel/KDesktop
cp -r /home/dummyuser/.merge /etc/skel
chmod -R a+r /etc/skel/.merge
cp -r /home/dummyuser/mydata /etc/skel
chmod -R a+r /etc/skel/mydata
```

zeichnisse unterhalb von `/etc/skel` in dessen Home-Verzeichnis. Sind Benutzer schon angelegt, müssen Sie diese Kopiervorgänge bei der nächsten Anmeldung veranlassen.

Dies ist eine herrlich typische Aufgaben für ein Anmelde-Skript. Bei Linux eignet sich hierfür das Skript `/etc/profile`, das Linux bei jeder Anmeldung ausführt. Individuelle Erweiterungen bringt man besser in der Datei `/etc/profile.local` unter.

```
/etc/profile.local
# Erweiterungen von Uwe Debacher und Bernd Burre für Win4Lin
#
if [ $UID -gt 999 ] ; then
    test -d ~/Desktop || cp -r /etc/skel/Desktop ~/
    test -e ~/Desktop/Windows.desktop || cp /etc/skel/Desktop/
┆ Windows.desktop ~/Desktop
    test -d ~/.merge || cp -r /etc/skel/.merge ~/
    test -d ~/mydata || cp -r /etc/skel/mydata ~/
fi
```

Um Systembenutzer und *Root* auszunehmen, kopiert das Skript nur, wenn der Benutzer eine ID größer als 999 besitzt.

Für alle anderen Benutzer richtet es die notwendigen Dateien und Verzeichnisse ein, sofern sie noch nicht vorhanden sind.

Auch bei der Nutzung des Terminal-Servers haben Sie dieselben Konfigurationsmöglichkeiten für die Windows-Installation, die Sie schon im Abschnitt 18.4.2 für die Einzelplatz-Version kennen gelernt haben.

Die Windows-Installationen innerhalb von Win4Lin sind sehr stabil. Falls Windows einmal abstürzt, dann kann man es sofort neu starten, ohne dass zeitaufwändige Festplattenüberprüfungen notwendig wären. Wenn Sie wie beschrieben mit Templates arbeiten, dann können Sie auch beschädigte Windows-Installationen sehr schnell erneuern. Diese Umgebung bietet sich daher auch zum Testen von Windows-Programmen an.

Die Redaktion lässt bei ihren Servern alle Windows-Installationen jede Nacht neu erstellen. Dadurch steht jeden Morgen ein sauberes Windows zur Verfügung.

19.3 CrossOver Office-Server-Edition

Während die im Abschnitt 18.3 beschriebenen Einplatz-Versionen von Codeweavers CrossOver Office für PCs mit oder ohne Festplatten gedacht sind, will die CrossOver Office-Server-Edition (derzeit Version 3.0.1) auf Linux-Terminal-Servern installiert werden, damit Anwender an Linux-Terminals mit Windows-Programmen arbeiten können. Lizenzen für Microsoft-Betriebssysteme benötigt man ebenso wenig wie bei den Einplatz-Versionen CrossOver Office Professional und CrossOver Office Standard. Mit den Management-Tools der Server-Edition können Systemverwalter die Auslastung des Systems und die Verwendung der Anwendungen beobachten.

Wie die Einplatz-Versionen unterstützt der CrossOver Office-Server nur eine Teilmenge von Windows-Anwendungen, wie verbreitete Office-Programme von Microsoft, IBM Lotus Notes Clients, Apple iTunes etc. Eine vollständige Liste hält Codeweavers an der Adresse http://www.codeweavers.com/site/products/cxoffice/supported_apps/ bereit. Über Evaluations-Versionen können Systemverwalter prüfen, ob ggf. weitere in ihrer Organisation eingesetzte Programme in der Praxis auch problemlos auf der Server-Edition von CrossOver Office laufen.

Über Schaltflächen auf der Seite <http://www.codeweavers.com/site/products/cxserver/> bietet Codeweavers dazu kostenlose Evaluations-Versionen und ferner Lizenzpakete mit 25, 50, 100 und 500 Client-Lizenzen zum Kauf an. Codeweavers Lizenzpolitik vertraut darauf, dass die Käufer ihrer Software stets nur so viele Anwender gleichzeitig (*concurrent*) die Software nutzen lassen wie sie Lizenzen beschafft haben. Das mitgelieferte Monitor-Tool erlaubt Systemverwaltern, die tatsächliche Nutzung zu kontrollieren und bei Bedarf die Beschaffung weiterer Lizenzen zu veranlassen. Auf eine harte Begrenzung oder Kontrolle der Nutzerzahlen verzichtet Codeweavers ausdrücklich.

Logischerweise dürfen nur Systemverwalter mit *root*-Rechten die Server-Edition auf Linux-Servern installieren und mit der deutschsprachigen Version des Server-Setup-Programms *cxsetup* arbeiten, um damit Windows-Anwendungen einzurichten.

Dieses Setup-Programm ist sehr Administrator-freundlich und damit kostensparend: Jede Windows-Anwendung müssen Systemverwalter nur einmal installieren oder updaten, unabhängig von der Zahl der Benutzer. Die Installation von Anwendungen läuft dabei genauso ab wie direkt auf PCs mit einem Windows-Betriebssystem.

Da die Server-Edition von CrossOver Office in den Heimat-Verzeichnissen der Benutzer jeweils ein Verzeichnis für benutzerspezifische Daten wie die Windows-Registry anlegt sobald diese eine Windows-Anwendung aufrufen, kommen sich die individuellen Benutzerdaten nicht in die Quere.

Klären Sie für alle Anwendungen, die Sie mit der Server-Edition von CrossOver Office installieren, ob Sie so viele Lizenzen benötigen,

- wie die Lizenz der CrossOver-Server-Edition unterstützt oder
- wie insgesamt Arbeitsplätze im Netz verfügbar sind.

19.4 Terminal-Dienste von Microsoft Windows-Servern

Anders als die beiden oben beschriebenen Konzepte, die Windows-Anwendungen auf Linux-Terminal-Servern laufen lassen und Anwendern über das X-11-Protokoll auf Linux-Terminals zur Verfügung stellen, bauen alle weiteren hier beschriebenen vier Lösungen darauf auf, dass Organisationen die Terminal-Dienste eines Microsoft Windows-Servers nutzen.

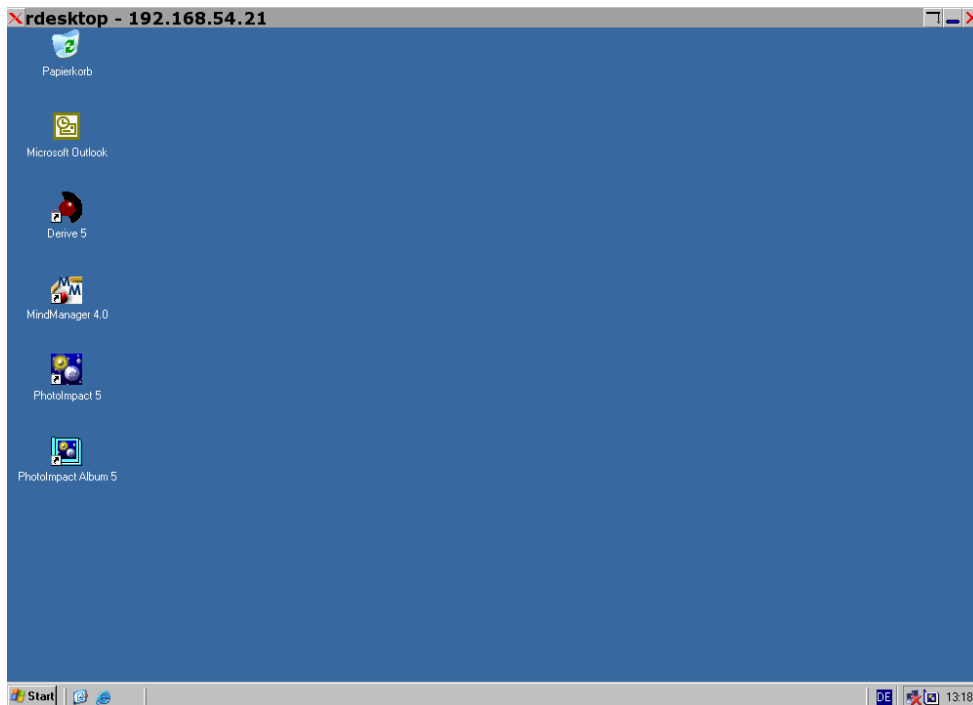
Aktuell verfügbar sind diese Windows-Terminal-Dienste auf Servern der Microsoft Windows 2000/2003-Server-Familien, jeweils mit freigeschalteten Terminal-Diensten.

Hierbei kommunizieren Clients mit den Windows-Servern über Microsofts proprietäres Remote Desktop Protocol (*RDP*).

Unter Linux kommt als Client-Programm *rdesktop* zum Einsatz, welches Sie auch auf den Installationsmedien von SuSE finden.

Das Programm *rdesktop* ist ein reines Kommandozeilen-Tool, das sich gut in Menüs integrieren lässt. Für die interaktive Nutzung sind seine Parameter etwas gewöhnungsbedürftig.

Installieren Sie daher auch gleich das Paket *tsclient* von den Installationsmedien. Dieses Frontend zu *rdesktop* liefert eine grafische Bedienoberfläche, die an die des entsprechenden Microsoft-Programmes angeglichen ist und sogar dessen Konfigurationsdateien (*.rdp*) lesen kann.

Abbildung 19.3: Windows-Terminal-Dienste mit *rdesktop*Abbildung 19.4: Grafisches Frontend für *rdesktop*

Nicht alle Windows-Anwendungen laufen überhaupt oder wenigstens nach intensiven Anpassungsarbeiten auf den Terminal-Diensten der Windows-Server. Um konkret herauszufinden, ob Microsofts Terminal-Dienste alle in Ihrer Organisation eingesetzten Anwendungen korrekt unterstützen, können Sie ein Windows Server-2003 Evaluation Kit bei Microsoft von der Adresse

```
http://www.microsoft.com/windowsserversystem/evaluation/trial/default.mspx
```

frei laden.

In *Windows 2000 Terminaldienste* von Stephan Herkert und Bernd Kretschmer, Addison Wesley, 2001, ist beschrieben, welche Anpassungsarbeiten selbst Microsoft-Programme wie Office 2000 erfordern und welche umfangreichen Skripts benötigt werden, um z. B. einen Lotus Notes-Client mit Windows-Terminal-Diensten zu verwenden.

Nur wenn die Prüfung der Anwendungscompatibilität hier erfolgreich abläuft, ist es für Ihre Organisation sinnvoll, die in den weiteren Abschnitten dieses Kapitels beschriebenen Varianten in Betracht zu ziehen.

Viele dieser Anpassungsschwierigkeiten können Sie umgehen, wenn Sie auf dem moderneren Windows 2003 und ähnlich jungen Anwendungen aufsetzen. Zusätzlich stehen Ihnen hier eine deutlich höhere Performance und einige zusätzliche Funktionen zur Verfügung.

Microsoft fordert neben den Lizenzen für Windows-Server mit freigeschalteten Terminal-Diensten zusätzlich zu den Windows-Server-Lizenzen für alle Endgeräte im Netz, die auf die Terminal-Dienste zugreifen, Terminal-Service-Zugriffs-Lizenzen (*Terminal Service Access Licenses*, TSAC) sowie Client-Zugriffslizenzen (*Client Access License*, CAL).

Die Terminal-Dienste vom Microsoft erfordern einen eigenen Lizenz-Server, der die entsprechenden CAL- und TSCAL-Lizenzen verwaltet. Wird eine Client-Lizenz benötigt, wird diese ausgestellt und am Client installiert. Der Lizenz-Server kennzeichnet diese Lizenz dann für 180 Tage als *benutzt*.

Firmen, die eine entsprechende Partnerschaft mit Microsoft haben (Citrix, Hob, Tarantella) sind an die Verwendung des Lizenz-Servers gebunden.

Verbinden sich Clients auch nur einmal, sei es nur zu »Testzwecken« auf einen Windows-Server, verbraucht das für ein halbes Jahr eine Lizenz.

Bitte prüfen Sie daher vor dem Einsatz von Microsoft Windows-Terminal-Diensten die damit verbundenen Lizenzkosten, insbesondere wenn in einem großen Netz eigentlich nur wenige Endgeräte diese Windows-Terminal-Dienste nutzen sollen.

19.5 Microsoft Windows-Terminal-Dienste mit Citrix Metaframe

Citrix hat Ende des letzten Jahrtausends für Microsofts Windows NT 3.5.1 eine Mehrbenutzererweiterung (MultiWin) und ein Übertragungsprotokoll (*Independent Computing Architecture*) entwickelt.

Die Mehrbenutzererweiterung erlaubt es, dass mehrere Benutzer von einem Client aus auf einem Windows-Server ähnlich wie auf einem lokalen PC arbeiten können.

Mit dem Übertragungsprotokoll Independent Computing Architecture (ICA) können Clients auch über schmalbandige Leitungen auf Windows-Server zugreifen.

Kurz vor der Jahrtausendwende, 1997, hat Citrix seine Mehrbenutzertechnologie an Microsoft lizenziert und Microsoft damit ermöglicht, eine spezielle Terminal-Server-Version seines Windows NT 4.0-Servers anzubieten. In die Windows 2000- und 2003-Server hat Microsoft die Technologie weiter integriert. Hier benötigt man keine spezielle Server-Version, sondern man muss die Terminal-Dienste nur freischalten (siehe vorangegangener Absatz).

Während von Microsoft kaum Marketing-Aktivitäten für seine Terminal-Dienste wahrzunehmen sind, vermarktet Citrix zu Windows NT 4-Server Terminal-Server Edition, Windows 2000-Server und Windows 2003-Server sehr erfolgreich seinen Aufsatz für Windows-Server namens *Metaframe*, der u. a.

- bessere Integration der Client-Hardware,
- mehr Verwaltungsmöglichkeiten und
- statt Microsofts Protokoll *RDP* eben das Citrix-Protokoll *ICA*

bietet.

Citrix stellt eine sehr reiche Auswahl an Clients für alle nur denkbaren Plattformen zur Verfügung. Der ICA-Client für Linux integriert sich hervorragend in KDE und Gnome und ist eine gute Möglichkeit, Microsofts Windows-Terminal-Dienste mit schlanken Endgeräten wie im Kapitel 16 beschrieben zu nutzen,

Im Unix-Bereich vertreibt Citrix nur Metaframe-Versionen für HP UX, IBM AIX und Sun Solaris, aber noch nicht für Linux-Server.

Citrix lizenziert seinen Server-Aufsatz Server-basiert nach gleichzeitig zugreifenden Clients (*concurrent licences*). Die Lizenzverwaltung von Citrix Metaframe verhindert sicher, dass mehr Verbindungen als bezahlt aufgebaut werden können, so dass für Organisationen keine Lizenzrisiken durch Überschreiten der lizenzierten Nutzerzahl bestehen.

19.6 Windows-Terminal-Dienste mit Tarantella

Wie oben bereits im Abschnitt Abschnitt 17.4 (Tarantella Secure Global Desktop Enterprise Edition) beschrieben, klinkt sich die Enterprise Edition des Secure Global Desktops von einem Linux-basierten Middleware-Server aus zwischen Anwendungs-Server und Endgerät.

Ist der Anwendungs-Server ein Microsoft Windows-Server mit freigeschalteten Terminal-Diensten, so kommuniziert der Middleware-Server mit diesem über dessen Microsoft-RDP-Protokoll und mit den Endgeräten über sein eigenes AIP-Protokoll.

Hierzu hat Tarantella einen eigenen RDP-Client nach Richtlinien von Microsoft programmiert und in der EE integriert. Aus Terminal-Server-Sicht macht es daher keinen Unterschied, ob Sie einen »echten« Terminal-Server-Client verwenden oder via der SGD EE.

Tarantella vertreibt diesen Client mittlerweile als eigenes Produkt. Derzeit ist er der einzige offiziell von Microsoft unterstützte RDP Client für Linux.

Der eigentliche SGD EE-Client hat daher nur eine Verbindung via SGD EE zu dem Terminal-Server. Eine direkte Verbindung zu dem Terminal-Server gibt es nicht. Da alle Verbindungen über den EE-Server geroutet werden, können sehr viele Windows-Server angesprochen werden. Die Server-Last kann zwischen diese Servern auf Wunsch anhand verschiedener Ressourcenvariablen verteilt werden.

Das Installieren und Konfigurieren von Windows-Terminal-Diensten (Windows 2000-Server, Windows 2003) und darauf laufenden Anwendungen gestaltet sich oft aufwändig. Die folgenden Erklärungen gehen daher von einem funktionierenden Windows-Server mit freigeschalteten Terminal-Diensten aus.

Für die Zusammenarbeit zwischen EE und Windows-Terminal-Servern sollte man vorab zwei Grundeinstellungen ändern. Manche Funktionen von EE 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 Terminal-Dienste von Microsoft Windows 2000-Servern zu deaktivieren, damit EE die Anmeldung durchreichen kann, ohne dass die Benutzer sich ein zweites Mal bei Microsoft Windows anmelden müssen. Bei Windows 2003-Systemen müssen Sie hier nichts unternehmen, da Microsoft diese Standardeinstellung bereits geändert hat.
- 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, SGD EE oder die Microsoft Windows-Terminal-Server.
- Sie können beide Einstellungen auf dem Microsoft Windows-Terminal-Server im Menüpunkt *Start • All Programs • Administrative Tools • Terminal Services*

Configuration ändern. Wählen Sie dieses Programm und hier das Verzeichnis *Connections* • *RDP*.

- Durch Doppelklick öffnet die *Terminaldienstekonfiguration* vom Microsoft Windows 2003 das Fenster *Eigenschaften RDP-TCP*. Wählen Sie zunächst die Registerkarte *Logon Settings*, und deaktivieren Sie den Punkt *Always prompt for password*, damit Tarantella Kennwörter an den Microsoft Windows-Terminal-Server weiterreichen kann.
- Wechseln Sie nun auf die Registerkarte *Session*, und definieren Sie, die Microsoft Windows-Sitzung zu beenden, wenn die Verbindung getrennt wurde. Damit kontrolliert Tarantella die Benutzersitzungen.

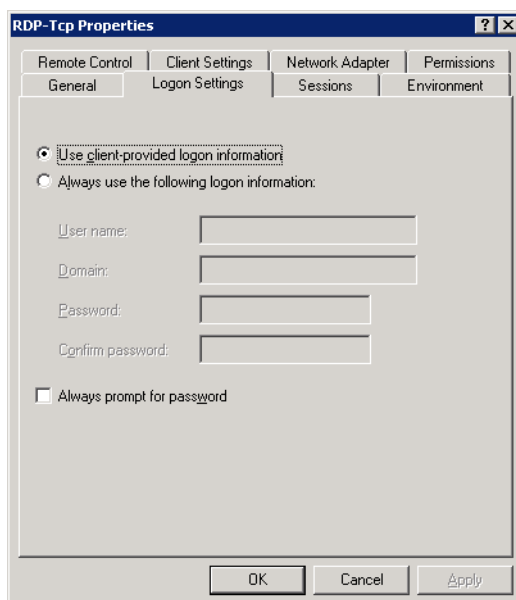


Abbildung 19.5: Einstellungen für das RDP-Kennwort

Eine Windows-Sitzung richtet man nach dem gleichen Schema X-Applikation ein (siehe Abschnitt 17.3).

Im folgenden Beispiel werden wir auf einen Server `windows.lokales-netz.de` zuzugreifen. Dieser Server ist ein Terminal-Server.

Zum Nachvollziehen legen Sie zunächst mit dem *Configuration Wizard* ein Server-Object `windows.lokales-netz.de` an und im zweiten Schritt die eigentliche Windows-Anwendung mit ihren Konfigurationsdaten.

Entscheiden Sie im ersten Schritt, ob Sie den ganzen Windows-Bildschirm oder nur eine einzelne Applikation publizieren möchten. Hier im Beispiel gilt es nur, das *WordPad* zu publizieren und daher den zweiten Punkt zu wählen.

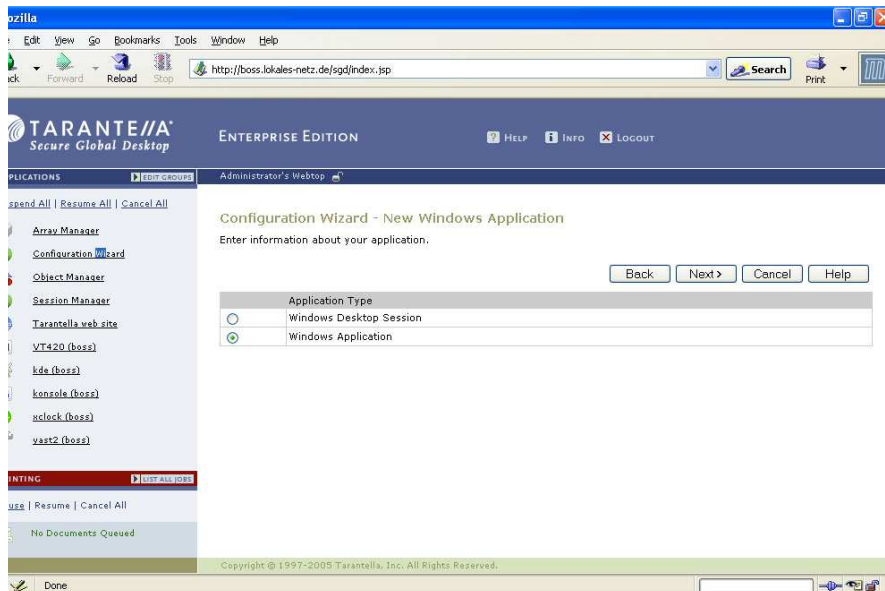


Abbildung 19.6: Neue Windows-Applikation, Teil 1

Im zweiten Schritt können Sie nun die gewünschte Applikation, die Darstellung und die Beendigung definieren.

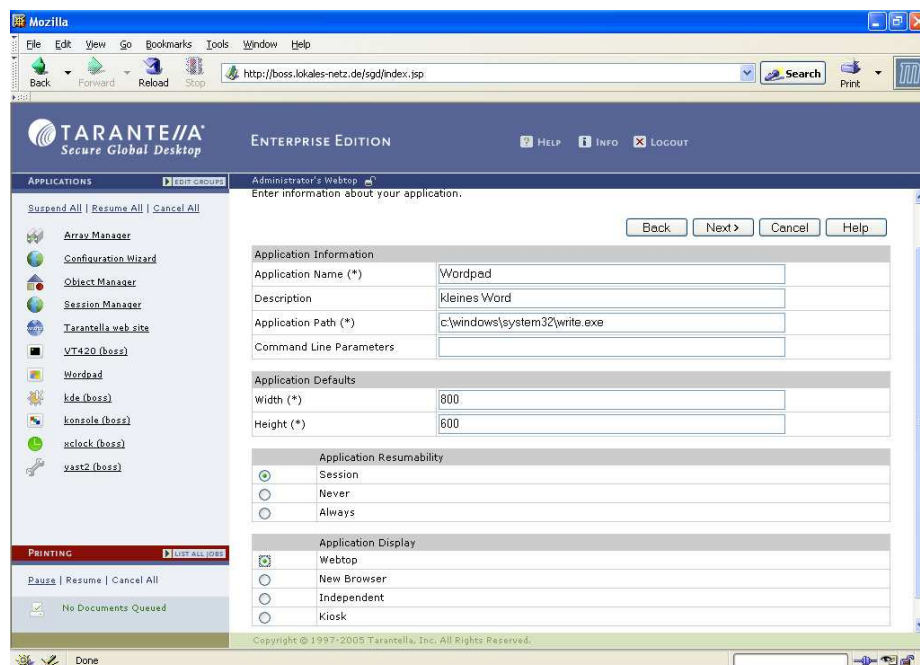


Abbildung 19.7: Neue Windows-Applikation, Teil 2

Unterschiede zwischen X- und Windows-Applikationen

Es gibt einige kleinere 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 Host-Namen 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. Die Größe eines Fensters einer Tarantella-Windows-Applikation kann man über eine neue Scaling-Funktion anpassen (*Scale to fit into Windows*).
- **Colour Depth:** Für die Verbindung zwischen Windows 2000-Terminal-Server und Tarantella-Server steht nur eine Farbtiefe von 8 Bit zur Verfügung. Bei Windows 2003-Terminal-Servern können Anwender zwischen 8 Bit, 16 Bit und 24 Bit wählen.

Melden Sie sich nun erneut als Benutzer *testuser* in Tarantella an, und starten Sie die Anwendung *WordPad*. Unter Umständen fragt Tarantella Sie hierbei zusätzlich nach Ihrem Windows-Anmeldenamen.

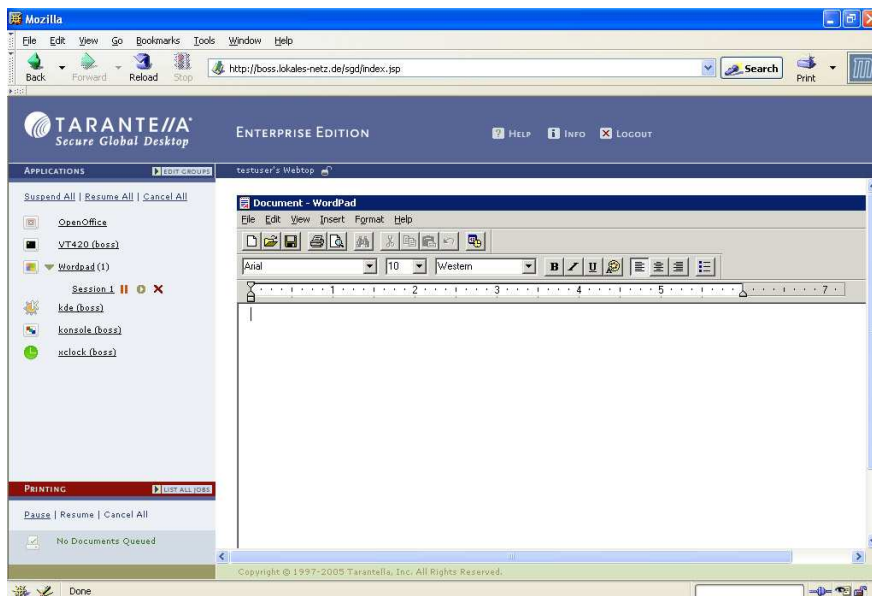


Abbildung 19.8: WordPad im Webtop

19.7 Windows-Terminal-Dienste mit Nomachine NX

Die Funktion von NX ist der Funktionsweise von Tarantella SGD EE sehr ähnlich. Auch verbinden sich Arbeitsplatzrechner nicht direkt mit einem Terminal-Server, sondern über einem Mittlerdienst auf dem NX-Server. Dieser Dienst übersetzt zunächst RDP in X11 und stellt dieses den Arbeitsplatzrechnern via NX Protokoll zur Verfügung.

In Server-Farmen von mehreren Windows-Servern kann man die Arbeitslast mit vorgelagertem Server-Diensten oder rotierend ausgleichen.

Applikation legt man bei NX wie im Abschnitt 17.3 detailliert beschrieben Client-seitig an. Dazu startet man dort einen *Configuration Wizard* :

Legen Sie hierzu eine neues Profil *Windows* an, und tragen Sie als Host den Server-Namen des NX-Servers ein.

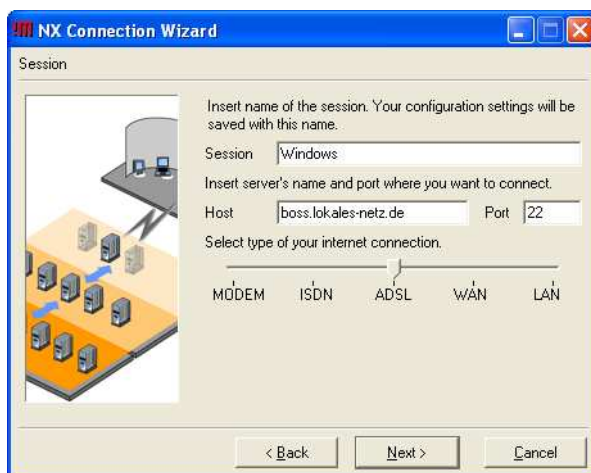


Abbildung 19.9: Einrichten einer Windows-Session

In einem zweiten Schritt wählen Sie als Verbindungsmethode statt *KDE/X11* eben *Windows*. Der NX-Client öffnet daraufhin ein neues Fenster, in dem Sie den Terminal-Server und die Anmeldedaten eintragen und eine Anwendung oder den Windows-Desktop auswählen können.

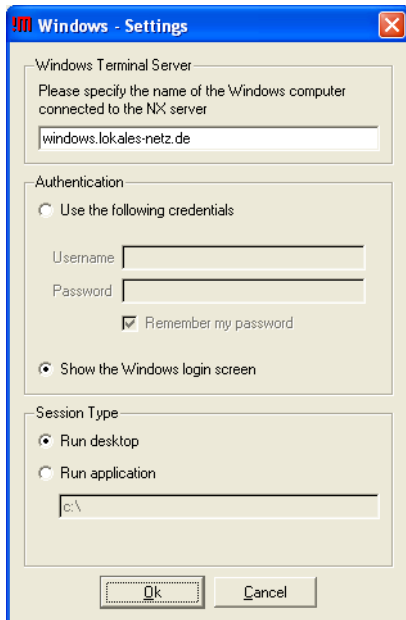


Abbildung 19.10: Konfiguration der Windows-Session

Nachdem Sie im Fenster *Windows-Settings* alles konfiguriert haben, gelangen Sie zum Fenster *Connection Wizard* zurück. Sie können nun die von Ihnen gewünschte Ansicht definieren.



Abbildung 19.11: Einrichten einer Windows Session2

Sie können diese Applikation wie im Kaptitel 17.2 beschrieben starten, indem Sie das Profil *Windows* auswählen. Bei der ersten Anmeldung sind hierfür die Anmeldeinformationen des NX-Servers erforderlich. Erst im zweiten Schritt müssen Sie – erkennbar durch die Standard-Windows-Anmeldung – die Anmeldeinformation für den Terminal-Server angeben.

Die möglichen Konfigurationen zu Windows-Sessions sind in diesem Release noch recht spärlich. Nomachine vermittelt auf seiner Web-Seite im Bereich Feature requests an der Adresse <http://www.nomachine.com/fr/index.php> einen Eindruck der geplanten Erweiterungen im nächsten Release.

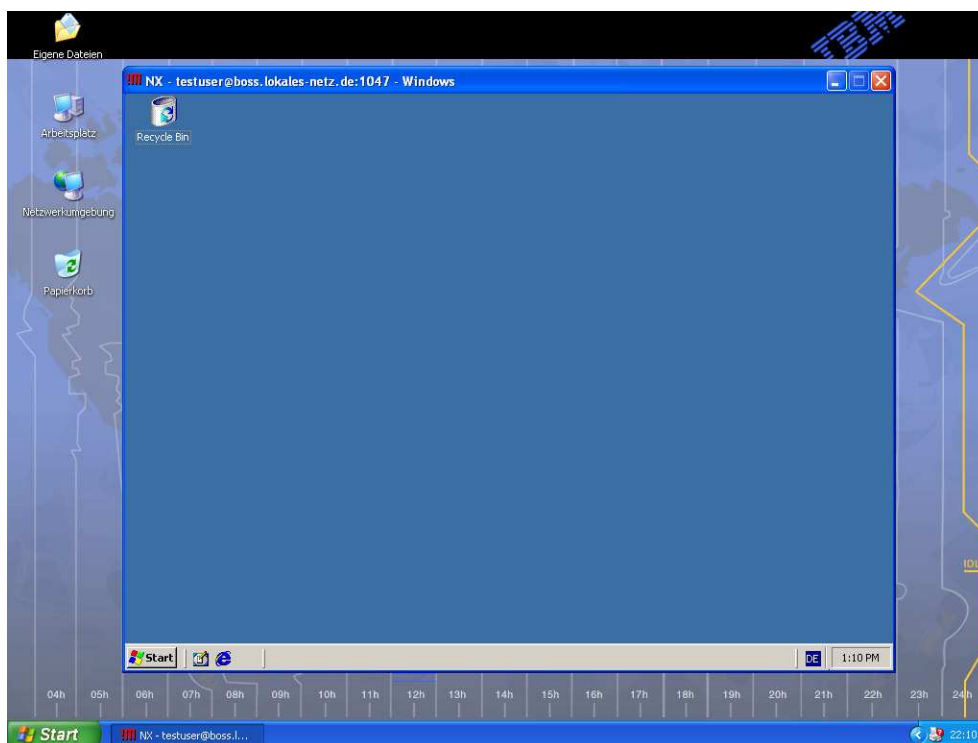


Abbildung 19.12: Windows-Desktop via NX