

## Monograph: Computing Omnipresence

(published jointly with Novática\*)

Guest Editors: *José-Antonio Gutiérrez de Mesa, Daniel Rodríguez-García, and Miltiadis D. Lytrasía*

*José-Antonio Gutiérrez de Mesa, Daniel*

*Rodríguez-García, and Miltiadis D. Lytras*

- 8 The Critical Mass Problem of Mobile *Ad-hoc* Networks — *Jörg Roth*
- 14 Device and Context Influence on Wireless Infotainment Access: A Real World Story — *Tacha Serif and George Ghinea*
- 20 The Problems of Printing in Server-Based Computing Environments — *Luis Bengochea-Martínez*
- 26 Using FOAF to Support Community Building — *Brian Kelly and Leigh Dodds*
- 29 Towards Ubiquitous Computing with Quality of Web Service Support — *Yannis Makripoulias, Christos Makris, Yiannis Panagis, Evangelos Sakkopoulos, Poulia Adamopoulou, Maria Pontikaki, and Athanasios Tsakalidis*
- 35 The Open Source Software vs. Proprietary Software Debate and Its Impact on Technological Innovation — *Ricardo-José Rejas-Muslera, Juan-José Cuadrado-Gallego, Javier Dolado-Cosín, and Daniel Rodríguez-García*
- 39 Localization in Ubiquitous Computing Using Acoustic Sensor Network — *Carlos-Manuel De Marziani, Jesús Ureña-Ureña, Álvaro Hernández-Alonso, Manuel Mazo-Quintas, Ana Jiménez-Martín, Juan-Jesús García-Domínguez, José-Manuel Villadangos-Carrizo, and Fernando-Javier Álvarez-Franco*
- 46 Portable System For Patient Monitoring With Wireless Technologies — *José-Manuel Rodríguez-Ascariz, Luciano Boquete-Vázquez, Ignacio Bravo-Muñoz, and Pedro Martín-Sánchez*
- 53 SIGLAS: A Practical Case of The Application of Ubiquitous Computing in Warehouse Management — *José-Julio González-Pozo and Manuel Ortega-Cantero*
- 58 Ubiquitous E-Management of Indicators — *Eladio Domínguez-Murillo, Ángel De Miguel-Artal, Beatriz Pérez-Valle, Áurea Rodríguez-Villanueva, and María-Antonia Zapata-Abad*

## UPENET (UPGRADE European NETWORK)

- 61 From **Pro Dialog** (PIPS, Poland)  
Computer Algebra Systems  
Problems with A Heart-Shaped Curve — *Leszek Jankowski and Adam Marlewski*
- 68 From **Mondo Digitale** (AICA, Italy)  
Web Engineering  
A Quality Model for Websites — *Roberto Polillo*

# The Problems of Printing in Server-Based Computing Environments

*Luis Bengochea-Martínez*

*Server-Based Computing (SBC) allows to run native applications installed in remote servers and accessing them from thin devices such as a personal computer, a PocketPC, a Smartphone or a PDA (Personal Digital Assistant). The TCP/IP (Transmission Control Protocol/Internet Protocol) connection with the server could be made through public switched telephone networks, mobile data services like GPRS (General Packet Radio Service) or UMTS (Universal Mobile Telecommunications System), Ethernet, wireless networking, or VPN (Virtual Private Networking). This technology permits the employees of a company to run the corporate applications from any location. Nevertheless, SBC environments have problems to print documents generated by those applications, firstly due to the need to install, in the servers, many printer drivers, but also due to the high bandwidth required by the printing data flow. In this article, an overview of the SBC technology is offered, showing the specific printing problems in these environments and proposing a solution based on the use of PDF (Portable Document Format) files to replace all print outputs produced by the applications executed in this way.*

**Keywords:** Free Software for Ubiquitous Computing, Server-Based Computing, Thin Client.

## 1 Introduction

Before the irruption of personal computers, all computer applications were run from physical terminals (VT220, IBM3270, etc.) connected to the enterprise mainframe or central computer either locally, or from a remote location through the use of a modem and a phone line.

When the use of personal computers was generalized and applications started to be developed for them, - with development costs much smaller than those that the applications in the great systems or mainframes use to have -, the client-server paradigm became popular. In these programs, the applications are divided in several layers, the first of which, including the presentation and local data validation levels, is executed in the personal computer - "Client" -, while the other application layers execute in the central computer - "Server" - where, at the same time, several application layers could exist, speaking then about "client-server-server", etc. One server program can simultaneously serve several client programs, which, for years, allowed the companies to distribute programs between their clients to carry out different tasks (to see accounts' balances, to carry out an order, etc.).

The main problems of this paradigm were:

1. The need to distribute the "client" piece of the application the first time that it was going to be installed in the remote computers, as well as every time that modifications and improvements were carried out.
2. The different operating systems on the client side's computers require developing specific programs for each of them. This fact complicated the process of deployment and maintenance.
3. The same customer could be required to install numerous applications sent by the companies he worked with, sometimes to do identical tasks, but with different interfaces and terminology.

4. Low levels of user expertise in the installation and use of the "client" programs, together with the resolution of the incidences produced, forced companies to create and maintain costly call centers for attention and support.

Internet and its associated technologies, based on the use of Web servers to run applications that are accessed through universal browsers such as Internet Explorer or Netscape Navigator, caused the decline of the client-server technology, since they came to give solution to some of the problems previously described.

The "webization" of applications appears so as the ideal solution to implement an ubiquitous office, in which the employees - and also the company's customers - can use corporate applications regardless their location - in his/her office, at home, in an airport, - or the kind of terminal they are using - a desktop computer, a tablet PC, a PDA (Personal Digital Assistant), etc. Likewise, the arrival of Java and all its associated technologies has permitted a fast transformation of existing applications, thus favoring the wide deployment of this new way of working.

Nevertheless, a large quantity of applications used in enterprises it still remains that, for different reasons, can not be redesigned to take advantage of this new paradigm, for example:

- Legacy applications, for which the lack of knowledge or the necessary technology avoids redesigning them.

**Luis Bengochea-Martínez** graduated in Computer Science at the *Universidad Complutense de Madrid*, Spain. He has developed his professional career in the Information System Dept. of several large Spanish banks, where has led the deployment of centrally managed corporate applications, running in Server Based Computing mode, for heterogeneous and very broad computer networks. He lectured as an Associate Professor at the *Universidad de Alcalá*, Spain. Actually, he is Full Professor in the Higher Technical School of Informatics Engineering at the same university. <luis.bengochea@uah.es>

- Applications acquired from software vendors, whose license agreements oblige to use them just as they are designed.
- Applications developed by public organisms (as community aids, settlements of taxes, etc.) of obligatory use by the companies, which must be used as they were conceived.
- Applications that, due to their own nature, do not accept a design that would allow their execution on a Web server.

Most of these corporate applications, except for legacy applications or those that are executed in a mainframe, have graphical user interfaces and cannot therefore be accessed from a simple text terminal emulator. Besides, we can find other applications that are executed under different operating systems and in different environments. The solution to this problem consists in executing the applications in its natural environment and transporting to the user device, by means of an appropriate communications protocol, each and every screenshot produced by the application. At the same time, the strokes in the keyboard and the mouse events produced by the user should be also sent to the computer where the application is being run. This technology, which requires multi-user capacities on the application servers, is called Server-Based Computing or SBC.

It is in fact a hybrid solution between the use of Web applications and the traditional client-server applications, which avoids installing and maintaining pieces of software in the remote user's computers and facilitates a rapid deployment and an effective maintenance of the applications, due to its centralized character [1].

## 2 Server-Based Computing

### 2.1 The Predecessors

The X-Windows system, developed in the MIT (Massachusetts Institute of Technology) in the eighties [2], can be considered as the predecessor of SBC systems. Initially developed to be used in small workstations with access to applications running on large Unix systems, X-Windows became a *de facto* standard for remote access to operating systems such as OSF and Linux. At the beginning of that decade, hardware terminals were built for the mentioned purpose and known as X-terminals, although they were replaced by emulation programs due to the subsequent popularity of the use of desktop computers running Windows systems. The problem of X-Windows was the need of utilizing a complex protocol, with a very rich set of functionalities, that required high bandwidth, which made it useful only when the X-terminal and the applications server were both connected to a local network or to a high-speed WAN (Wide Area Network).

Besides, the X-terminal that executes a heavy X-server needs heavy computing resources, what makes impossible to use a thin client, like a PDA, precisely one of the premises for ubiquitous office. The security model of X-terminals also suggests the existence of a number of problems, since the traffic could be scanned and passwords could be extracted

for malicious access to critical enterprise resources. This is the reason why some system administrators forbid its use out of the militarized network area.

The following step to SBC was the VNC (Virtual Network Computing) system [3], developed by the Oracle Research Laboratory. It consisted in two software pieces: the VNC Server, which should be installed in the server – that could be a simple desktop computer running Windows or Linux operating systems –, and the VNC Client, installed in the client device and based on Java (this allows the existence of different versions for all kind of thin clients). The underlying technology is a simple protocol that acts at frame buffer level and therefore is susceptible of being used on any operating system and with any transportation protocol, like TCP/IP (or even with USB - Universal Serial Bus). VNC has the additional advantage of being distributed under GNU (Gnu's Not Unix) license. It is mainly used to implement solutions for ubiquitous office where a company's employee can access his/her desktop computer (placed in the corporate headquarters), from anywhere in the world where his/her thin device can be connected to a network. It would be able to do it even from a Internet cafe or from a public console with Internet access like those installed in some airports. Once s/he takes control of his/her computer, s/he would be able to run any application as if s/he was seated in his/her workplace.

The arrival of Windows NT Server had a very significant impact on commercial computation. Numerous applications, like database systems, e-mail servers, Web servers and other, migrated from large systems such as Unix or VMS to the new NT servers. A small company, Citrix, had the idea of taking advantage of NT's multi-user capacity to make the server work not only in providing services to the network applications, but also in opening several users' sessions. In each of these sessions, a user with his/her computer works against the server, and his/her computer screen visualizes the graphical output produced by the server that, simultaneously, is receiving his/her keyboard and mouse events. The system, called Winframe (a word derived from the fact that this system was aimed at converting a NT server in a Windows mainframe) also included an optimized protocol to transport screen, keyboard and mouse information flows between the server and its clients. This protocol was called ICA (Independent Computing Architecture).

There have been other different approaches to SBC, like the launching of a new ultra light device by Sun, in September of 1999, called Sun Ray Enterprise Appliance, a device without a CPU (Central Processing Unit) and without a memory. Its only mission consisted in sending the keyboard and mouse events to the server and receiving the pixels in order to represent images on the screen. Nevertheless, it lacks an optimized communications protocol, what forces a connection with the server to be made through an Ethernet network. This kind of devices can be useful in environments where workstations are not PCs, but they cannot be considered thin clients good enough as to be part of a ubiquitous office.

### 2.2 SBC Today

When Microsoft understood the potential of the SBC solution – developed by Citrix for its operating system NT Server -, it acquired a part of the Citrix technology used in Winframe, to develop its own SBC solution, called Terminal Server. It included a new communications protocol called RDP (Remote Desktop Protocol).

At present both solutions coexist, although in both cases they are proprietary solutions, what means that licenses must be acquired, involving a cost, to use them. Microsoft's Terminal Server provides multi-user capacity to a Windows server that can be accessed from client computers running any Windows operating systems, by using the RDP protocol. This allows the applications running on the server the use of local devices, such as disks and printers, in the thin client. In order to install and use Terminal Server a CAL (Client Access License) license is required for each thin client using the SBC, except for those having Windows 2000 or a newer "professional" operating system installed.

On the other hand, Citrix's Metaframe requires having Terminal Server previously installed (for Windows servers), and provides several additional services:

- Different thin clients can be used (not only Windows-based ones), what includes Java based devices.
- It can be used on UNIX and Linux servers to provide access to applications developed in these operating systems.
- Many devices in the thin client, like disks, printers, serial or USB ports, or audio cards can be accessed from the server.
- It provides a centralized management of "farms" composed of hundreds of application servers, with load balancing.
- The ICA protocol, optimized for low bandwidth, permits to use any type of communications, from local networks to mobile GPRS (General Packet Radio Service) or

UMTS (Universal Mobile Telecommunications System) ones.

- It uses the RSA (Rivest-Shamir-Adleman) encryption to protect the information transmitted between server and client, what could be a mandatory requirement to run given applications from locations out of the enterprise internal network.

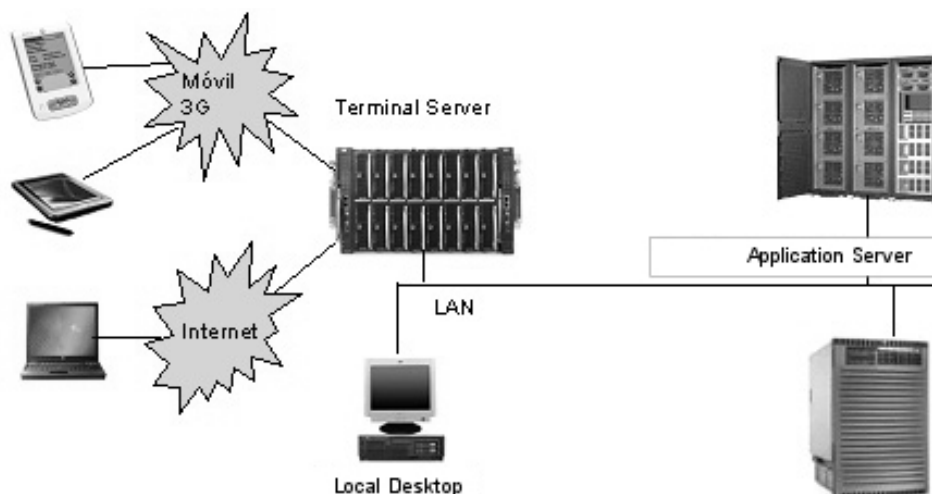
Other solutions exist, like Tarantella (before known as Santa Cruz Operation), a proprietary solution mainly oriented to provide access to Unix and Linux servers, although also supports Windows servers using Terminal Server or Metaframe.

### 3 The Mobility Challenge

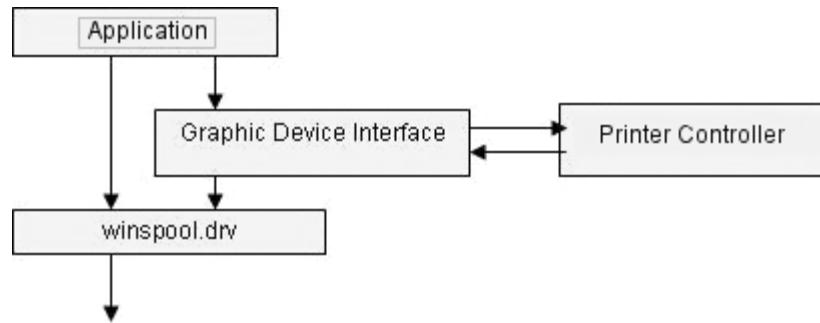
Many enterprises believe that the mobility of their employees is a competitive advantage. There are studies, as the one carried out by Volchkov in 2002 [1], showing that currently more than 10% of the employees run corporate applications from remote locations during more than a half of their worktime. Therefore, a challenge to the data processing departments consists in providing good systems for mobility, minimizing maintenance problems and guaranteeing the security.

At present, SBC technologies allow, in the client side, minimizing hardware and software maintenance costs by avoiding connections from computers with local applications, different antivirus protection, etc. On the server side, installation and maintenance of applications are carried out in a centralized form, these applications being accessed in an identical form both by employees physically located in the building and by those that are traveling [4].

Although Internet could be utilized to access the enterprise servers, a VPN (Virtual Private Network) can be established before using an application, and then utilize the same mechanisms of logon and the same passwords as those used



**Figure 1:** A Terminal Server Can Act as An Intermediate Level between All The Corporate Applications (Including Web-Based Ones) and Thin Client Devices.



**Figure 2:** The First Part of A Print Process Consists in Submitting A File to The System Spool with Instructions about how The Designed Printer Should Produce the Desired Printed Output.

in the desktop computer. SBC systems that encrypt communications provide additional confidentiality. Besides, by centralizing also the office applications, the risk for the company's security associated to loss or theft of a laptop computer is prevented.

The SBC technology appears as well as a universal solution to mobility, since it permits the use of thin clients and mobile communications with low bandwidth, and transforms the terminal servers in the entry point for all the enterprise applications, from Web applications to classical client-server applications or even specific applications on Unix or main-frame systems, as shown in Figure 1.

Of course, when very light devices with small size screens and very limited capacity of process are used (e.g. PDAs), the number and kind of corporate applications that can be run decreases.

On one hand, the nature of the application could make impossible its use in this type of devices, but, on the other, the perception that users themselves have of these devices, particularly if they are connected through a low speed network with a high latency, persuade them from running applications not specifically designed for the device. In this sense, studies have been carry out [5] to measure the relation existings between the degree of satisfaction of mobile users employing light devices and the typical metrics in a communications network (as bandwidth and latency), with the objective of identifying the limits associated to the use of very light mobile devices.

#### 4 The Problem of Printing in SBC Environments

Although almost all SBC commercial systems provide the capacity of "mapping", from the server, the local printers at the thin client - if those printers exist -, printing remains one of the weak points for the mobility solutions provided by SBC.

The problems arise, fundamentally, from the nature of the printing processes implemented in the different operating systems, most designed under the assumption that physical printers are directly connected to the computer where the application is being run or, in other cases, to a print server connected through a high bandwidth communication access.

#### 4.1 Basic Description of The Print Process

Although the description of the print process and the proposed solution that follows correspond to Windows-based operating systems [6], it is also applicable, with very light changes, to other operating systems.

When an application needs to produce a printed output, the first step is to call the Graphic Device Interface (GDI) that is responsible for creating a visible output. In order to do this, it should call the specific driver associate to the chosen printer. With the document information originated in the application, the GDI interface exchanges data with the driver to process the print work in the printer's language and, subsequently, the data to be printed are transferred to the printer's queue (see Figure 2).

The type of data generated by the print work can be RAW, that is directly interpretable by the printer, or EMF (Enhanced MetaFile), that permits a little reduction in the file size and improves the traffic between the system spool where the printed output originates and the print server the printer is really connected with (however, this is true only in the case that both elements use Windows). In the case of printers supporting the PostScript language, PostScript files (PS) could also be used.

Depending on whether it uses or not a print server, the print spooler, which is a set of DLLs (Dynamic Link Libraries) and drivers that receive, process, plan and distribute the print works, submits the work to the local printer provider or to the remote printer provider (who will in turn transfer it to the spool in the print server connected to the printer). In the case of a printer connected to the thin client and mapped by the application server, the file is submitted to the client spool where it will be processed by the local operating system as if it was a local printing.

Once the file has been placed in the spool of the server computer the printer is connected to, it is submitted to the printer monitor (if the printer has bidirectional communication, a language monitor supervises the bidirectional communication between the sender and the printer), who transfers it to the port (LPT, COM, TCP/IP port, etc.), whose monitor, at the same time, submits it to the target printer. Finally, the printer receives the work, translates each page into the appropriated format and 'physically' prints it.

### 4.2 Printing in SBC Environments

According to the discussion in the previous section, it can be noticed that the two main problems related to printing in the design of a solution for an ubiquitous office based on SBC are the following ones:

- Application servers need that the drivers of all printers connected to the client devices are installed. Each time a new local printer is connected to an employee's laptop, it will be necessary to install its driver in all SBC servers of the company.

- Since the RAW, EMF or PS files to be printed were not designed to be transmitted through a communications network, they use to have a disproportionate size that could prevent receiving them in a client computer in a reasonable time (see Table 1).

The first problem can be alleviated by implementing centralized procurement policies, in such a way that only printers that have been previously approved by the managers of the data processing department are acquired, printers whose drivers should have been tested and installed beforehand in every SBC server. Nevertheless, the huge amount of new models of printers that appear every day at the marketplace, including new improvements and different features, as well as the disparity of mobile devices that can be used to connect them, makes this solution scarcely realistic.

In SBC proprietary solutions, universal controllers are provided (usually PCL) that can be used for almost any class of printer, but with the inconvenient of being too rigid an approach.

For the second problem, the only possible solution is forbidding the transmission of RAW print files, establishing only one final format for all printed outputs – e.g., PDF (Portable Document Format) - and transmit the print files in this format to be locally printed in the client device. This solution also permits that employees with mobile devices without printing capacity can keep their printed outputs in disk for subsequent uses, send them to the printer in a business office close to their location, submitting them via e-mail to another employee or to any other person, etc.

### 5 A Free Software-based Solution

In order for any application running on a server to be able to produce printed outputs in PDF format, it is neces-

sary to define a logical printer (not connect to any real printer) that produces a PDF file every time a print job is submitted. Although proprietary solutions exist, like Adobe's Acrobat Distiller, the proposed solution is based on the use of two free tools:

- RedMon (Redirect Monitor) that permits to define ports whose monitor, instead of sending the print file to a physical device, redirects it to the standard input (stdin) of a program set by the user. It is distributed under a free public license, and is available at <<http://www.cs.wisc.edu/~ghost/redmon/>>.

- Ghostscript, a PostScript interpreter that permits to generate an output in PDF format. It is available, under GNU license, at <<http://www.cs.wisc.edu/~ghost/>>.

### 5.1 Defining A PDF Virtual Printer

Once the above mentioned products have been installed, we can define a new virtual printer for each SBC server, which will be the one to be used by the thin clients connected to that system. To do this, the following steps should be followed:

- Create a new port (RPTn:) of type "redirected port" to connect the new printer.

- Select, among all the postscript printers run by the operating system, the model that better fits our printing needs, basing the decision on criteria such as whether it permits or not the use of colors, resolution, and the like.

- Install as if we had a real printer of that model, defining the default printer parameters (paper class, resolution, color, etc.).

- Open the printer properties and modify the configuration of the RTPn: port where it is connected, using the following parameters:

- Redirect the port to program: C:\gs\gs8.50\ bin\ gswin32c.exe (or a different folder where Ghostscript was installed).

- Program arguments: -q -sDEVICE#pdfwrite -dNOPAUSE -dBATCH -dSAFER -dCompatibility Level# 1.2 -sOutputFile="%1" -\_ (where "-\_" indicates that "stdin" will be the data entry and "%1" that the output file name will be the one that the user decide before a work is printed).

- Output: choose the option "Prompt for filename" for a value to replace the default "%1".

| File format                 | Size             |
|-----------------------------|------------------|
| Original (MS Word)          | 4.052.480 bytes  |
| RAW file in PCL with 600dpi | 34.828.288 bytes |
| RAW file in PCL with 300dpi | 15.265.792 bytes |
| PS file with 600 dpi        | 9.654.261 bytes  |
| PS file with 300 dpi        | 9.631.344 bytes  |
| PDF file                    | 2.052.096 bytes  |

**Table 1.** Original Size of A Test File and The Sizes Of Print Files Generated According to The Printer Type and Resolution Used.

When a remote user opens a session in a server where a PDF printer of this kind has been set, s/he will be able to use it as a normal printer. S/he will also be able to use the PDF files produced this way in the form that s/he considers to be more convenient according to the type of device that is using and the bandwidth of the communications network to which is connected.

### 6 Conclusions

The SBC technology permits a company to establish a unique way of accessing all its corporate applications, including personal productivity ones such as Microsoft Office. This can be done by using any thin device for which a SBC client exist, from anywhere, and using any communications network.

In the case of applications that produce printed outputs, PDF virtual printers can be defined to be used by the thin clients that have no physical printer available, as well as to reduce the size of the print files that must be transmitted from the server to the client. This gives also new possibilities for using those documents in PDF format, such as sending them out via e-mail or keeping them stored in disk for subsequent processing.

### References

- [1] A. Volchkov. "Server-based computing opportunities". IT Professional Volume 4, Issue 2, March-April 2002, pp. 18-23.
- [2] M. Revett, I. Boyd, C. Stephens. "Network computing: a tutorial review". Electronics & Communication Engineering Journal. Volume 13, Issue 1, Feb 2001, pp. 5-15.
- [3] T. Richardson, Q. Stafford-Fraser, K.R. Wood, A. Hopper. "Virtual network computing". Internet Computing, IEEE. Volume 2, Issue 1, Jan.-Feb. 1998, pp. 33- 38.
- [4] D. McIntosh, R. Born. "Server-based computing: back to the future solves enterprise desktop problems". Proceedings of the 2000 IEEE International Conference on Management of Innovation and Technology, 2000. ICMIT 2000. Volume 2, 12-15 Nov. 2000, pp. 732-737.
- [5] Zhimei Jiang, H. Mason, Byoung Jo Kim, N.K. Shankaranarayanan, P. Henry.: "A subjective survey of user experience for data applications for future cellular wireless networks". Symposium on Applications and the Internet, 2001. Proceedings. 2001 8-12 Jan. 2001, pp. 167-175.
- [6] Microsoft Corp. "Windows 2000 Server Deployment Planning Guide". Chapter 16, "Deploying Terminal Services". <<http://www.microsoft.com/windows2000/techinfo/reskit/dpg>>.