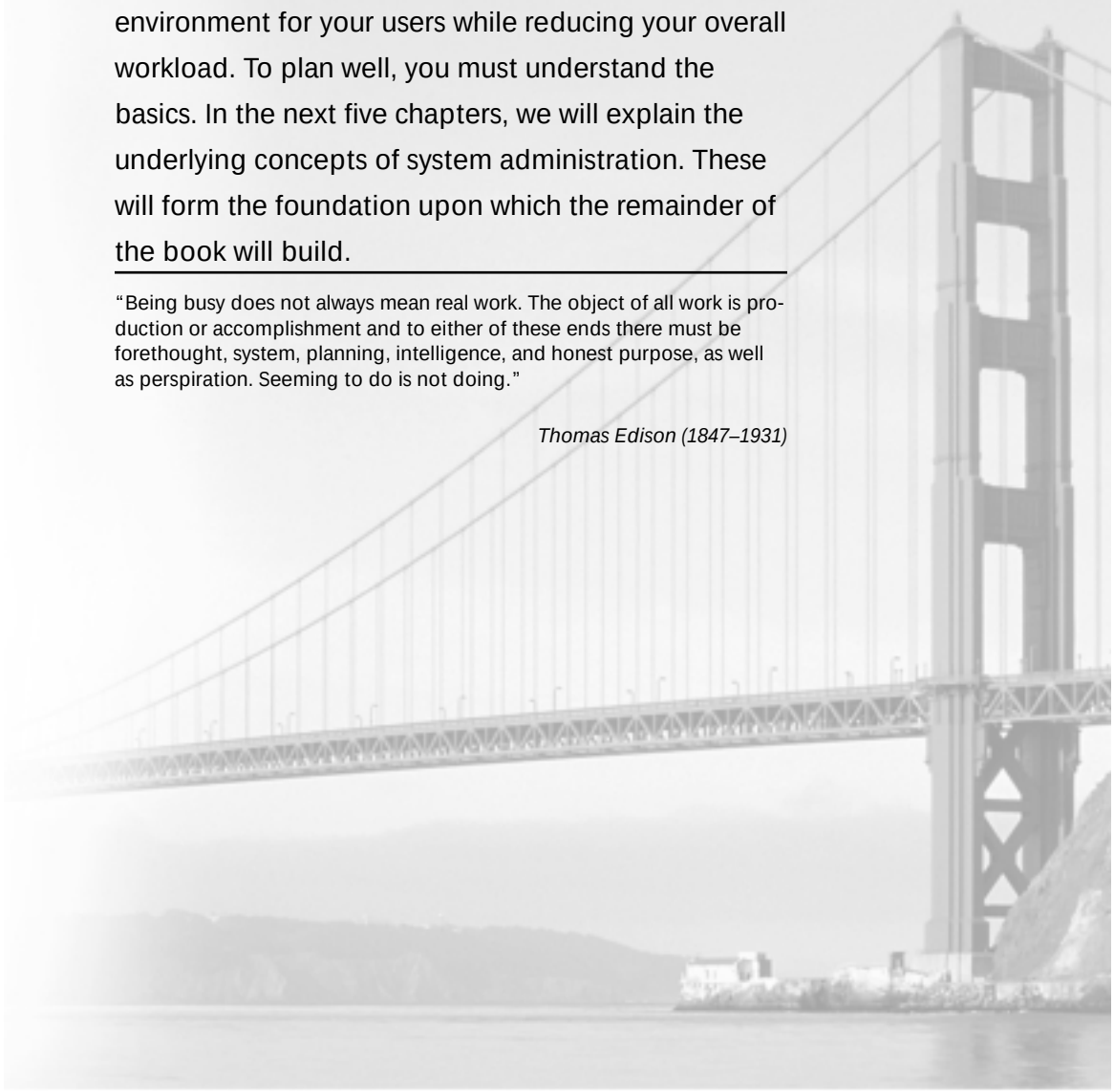
PART ONE

PLANNING

In the next few chapters, we will discuss the fundamental concepts of installation, system startup and shutdown, the filesystem, and upgrades and patches. Along with these topics, we will discuss the need for careful planning in your system and infrastructure design. Through planning, you can create a more stable environment for your users while reducing your overall workload. To plan well, you must understand the basics. In the next five chapters, we will explain the underlying concepts of system administration. These will form the foundation upon which the remainder of the book will build.

"Being busy does not always mean real work. The object of all work is production or accomplishment and to either of these ends there must be forethought, system, planning, intelligence, and honest purpose, as well as perspiration. Seeming to do is not doing."

Thomas Edison (1847–1931)



CHAPTER 1

THE ART AND SCIENCE OF SYSTEM ADMINISTRATION

Way back at the beginning of the information age, when computers were enormous and data storage was tiny, system administrators were created. They were chosen for their unique ability to arbitrate between the often-petulant operating systems and the equally unique group of people trying to program them. A mystique grew up around those selected few. To most users, it appeared that only system administrators could wring compliance from the early electronic tormentors. Each machine had its own idiosyncrasies, and its own cadre of conjurors to keep it entertained. As the mechanized wonders became more powerful, they paradoxically shrank and replicated. As computers shrank in size, their capabilities, and the job of the system administrator, grew. Networking, file sharing, databases, email, Web servers, printing, and users all added to the job description. System administrators must now juggle the *science* of operating systems, their resources, and the *art* of user support and administration.

The support of computers and users in the face of ever-changing operating systems continues to challenge even the most experienced of administrators. Price wars, as well as diverse system features like application requirements and other market forces within the computer industry have created a wide range of multi-vendor environments. In particular, the mingling of UNIX and the Windows family of operating systems has become common. The combination of these factors has resulted in a shifting morass, upon which the system administrator

must build a stable environment for the users. The following chapters will guide you through the labyrinth of supporting the ever-changing needs of users and the ever-growing complexity of the infrastructure.

Understanding the Job

In this chapter, and throughout this book, we present system administration as a concept, independent of the operating system. The true basis of integrating a heterogeneous environment is understanding that system administration is a skill set that exists regardless of operating system. If you learn system administration, then you can apply the concepts to whatever operating system you are working with. Understanding the basic building blocks makes it possible to provide similar or identical services across operating systems. This book will weave this concept in with another one as *leveraged learning*. Operating systems are tools, incredibly complex, but still only tools. The understanding of how and when to use those tools is the foundation of good administration.

Most problems can be divided in multiple ways. This is certainly true with system administration. We have chosen to divide the administrator's job into three parts: *planning, integration,* and *administration*. These parts are not completely distinct. They are more like three intersecting circles. The most complex tasks exist where the three circles intersect, as shown in Figure 1.1. This book, therefore, is divided into three parts, to coincide with this grouping.

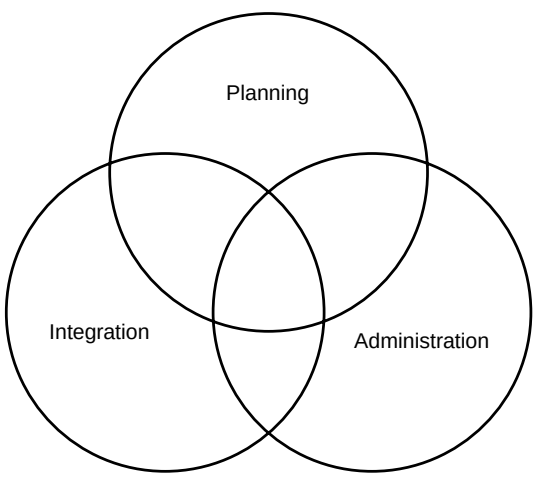


Figure 1.1 System administration areas.

PLANNING

The adage,"A stitch in time saves nine,"is certainly true when applied to system administration. Regretfully, a significant amount of your work as an administrator is done under tremendous time constraints, with anxious users standing in the wings, if not actually in your office. Planning is often sacrificed in these circumstances. In the next few chapters, we discuss the fundamental concepts of installation, system startup and shutdown, the filesystem, upgrades, and patches.

INTEGRATION

Integration, with respect to this book, is the sharing of resources between diverse operating systems. For clarity, we need to compare this with *interoperability* and *coexistence*. The three are ascending levels of the same concept. Coexistence is simply the state of cohabitation on a network. There are often applications provided to facilitate the transfer of information between diverse systems, but not without direct manipulation by the user. Integration is the next level, providing shared resources among constituent systems. The most commonly shared resources are filesystems, email, information and domain services, printing, and networking. Interoperability is the highest level of interaction, with identical application resources made available regardless of the operating system. A common example of interoperability is a database application, which must be distributed throughout a heterogeneous environment.

Our focus will be on integration, the middle ground between coexistence and interoperability. Specifically, we will be concerned with creating a stable environment for the user to work with. Network address resolution, user and system identification, shared filesystems, email, printing, networking, and security are the key issues in creating an integrated infrastructure. We will discuss tools and strategies for accomplishing these goals between Windows 2000 and a variety of UNIX operating systems.

ADMINISTRATION

The administrative tasks are, simply put, the most boring. However, they are easily the most critical to the users. Tasks in this part of the book deal directly with the users, or with the tools necessary to support them. Creating user accounts (and deleting them) is of obvious importance. Backups, performance-tuning, and developing user interfaces also affect the quality of the user environment. Many of the tools necessary to accomplish these tasks, and others in this book, are discussed here as well.

It is extremely easy to forget that the reason most systems and networks exist is to support some set of activities performed by users. We believe that

the ultimate test of an integrated infrastructure is its ability to meet the needs of the people who use it. No matter how nice it looks on paper, or how well it is designed in theory, if the users are not happy, then it is not right, period.

LEVERAGED LEARNING

It is amazing the number of people who believe that variations in tasks and tools constitute a unique job. For example, there is a popular misconception that Solaris administration, HPUX administration, or even Windows 2000 administration are disparate skill sets. While it is certainly true that each UNIX variant, Windows 2000, and other operating systems require specific knowledge to manage correctly, the underlying fundamentals for each are extremely similar. Accountants move from job to job regardless of the unique nature of each position. Likewise, CEOs move from company to company with no concern for the actual nature of the company. The common belief in many fields is that once you have mastered a core set of concepts, they can be applied to a wide variety of different tasks. We believe that this is the case for system administration. Continuous changes in the field have also made it an absolute necessity. Without the ability to adapt, a system administrator will rapidly become obsolete and unemployed.

The understanding of basic networking principles, account management, file management, security issues, printing, and many more of our tasks is not dependent on the specific commands used to implement them. If necessary, you can always look them up in a manual if you know what you are trying to accomplish. In this book, we have attempted to *leverage* the reader's knowledge of one operating system against the task of learning a new one. In each chapter, we present fundamental concepts first, followed by specific examples. This structure is intended to emphasize the concepts you need to master and provide substantial examples of how they are used in practice. By this process, we intend to create a mental model of how system administration is done. The danger lies in incorrectly mapping the model when applying it to a specific operating system. For this reason, we have included notes, tips, and warnings that are related to places where the mapping fails throughout the chapters.

Explaining the Job

One of the most challenging roles of the system administrator is providing sound business reasons for what they do. The advantages of adding to your infrastructure may seem obvious to you; but those same advantages might

not be clear to the person with the checkbook. Maintaining network throughput, licensing software, and adding faster hardware to cope with increased demand can easily be lost in the bottom line if your management team has a variety of responsibilities to juggle. Selling management on the pluses of new technologies and additional infrastructure can be one of the most difficult and frustrating tasks of system administration.

Managers speak the language of budgets, cost savings, value add, and customer satisfaction. These concepts are their primary responsibility. The language of the system administrator can be confusing, unless you learn to translate from bits and bytes to dollars and cents. Performing this translation does not take a business degree, just some common sense and the resolve to do some paperwork.

Information is key to helping the manager understand the business reasons for your recommendations. Organizing this information and presenting it in a clear and comprehensive way is the key to successful communication. This information, when presented in such a way as to clarify the requirements, the costs, and the pluses and minuses, is called a *business case*. A solid business case not only explains the technical needs and situation to the management, but also will provide the system administrator with peace of mind that the best solution has been found.

UNDERSTANDING THE SITUATION

Before any technical documentation is read, or any vendors are called, it is imperative that the system administrator thoroughly understands all the aspects, requirements, and underlying needs for a given project. These aspects, called *business drivers*, are the motivations for a new piece of hardware, a new software package, or the birth of a new, complicated project that may cost big dollars. Sometimes, the business driver is as simple as replacing a dead server. Other times, the business driver is very complex, such as the need to implement a new human resources management system for a global, multi-billion dollar company.

While these two examples may seem quite different, arriving at the correct answer will follow the same path. The first and sometimes most important step is to understand the requirements surrounding the business need. All too often, when there is some requirement for a new technological gizmo, administrators and bosses focus more on using the latest technology instead of finding the best solution. It is all too easy to be mesmerized by the latest technological gizmos, causing you to overlook issues related to the problem at hand. Staying focused on the problem, and not the technology of the solution, is the key to a successful project.

Staying focused on the problem, and not the technology of the solution, is the key to a successful project.

Sound advice is to create some sort of form, or procedure, to document exactly what the business drivers are, who is affected, and what the expectations are *before* you begin to look at any type of technological solution. Don't let your exuberance to find a solution cause you to overlook or undervalue minute details, for those details may come back and bite you later.

While everyone hates meetings, it is better to meet early in a project and determine the requirements than to meet later in the project and discuss the problems. Try to bring all the necessary parties to the table early and solicit input from a broad spectrum. This means including more than just your own team. If the project is far-reaching, try to assemble all affected parties, as their inputs may help clarify which technological path to take. Not only does this help you gather information, but it also helps the users feel like they are stakeholders in the project and part of the team. The more support you have, the easier it is to cross hurdles later in the project.

Once you have gathered all the key players and have defined the needs and requirements of the project, try to document what you have gathered. Having the business needs on paper, along with the input of the team members, can help you later in the project if you begin to lose your way (it's also a great way to cover yourself).

RESEARCHING ALTERNATIVES

Now that you have gathered and documented the business need, you should begin to look for the technological solution. Even if you believe you know the solution, you need to invest in some research. Learning the proper way to conduct this research is not difficult, if you know what resources are available to you.

- **Use the Internet—**One of the best places to research problems is on the Internet. With the thousands of newsgroups and hundreds of sites dedicated to technology, there is a high probability that another administrator has already investigated your problem and developed a solution. Most likely, you'll be able to find multiple solutions. Use this data to help form some opinions; but remember, it is best not to believe everything you find on the Internet.
- **Communicate with peers—**Isolation is one of the administrator's worst enemies. In many organizations, the administrator may be a one-person shop, managing everything including the network,

servers, and user workstations. Even if you are in a large shop, don't let yourself fall into the trap of working in a vacuum. Administrators need to get our of their environment and breed new ideas. Joining local user groups and attending large conferences help lend perspective, and generally will yield a few new ideas. At worst, the administrator has a new forum in which to vent about their day-to-day problems.

- **Read trade magazines and journals—**Everyone hates receiving junk mail; but the fact is, mountains of unsolicited paper is an occupational hazard. However, in the computing world, there are a lot of magazines and email digests you can subscribe to that only cost the few minutes it takes for you to fill out a questionnaire. While the information you provide will result in tons of vendor-related mailings, the information you receive from the publications will be highly valuable. Examples include *InfoWorld* and *PC Week*, just to name a few. Once you've visited a conference, or subscribed to one of these magazines, you will be solicited to join others. While you may only read one or two, the effort is worth the few nuggets of information you will find that hit the spot.
- **Solicit vendor input—**Vendors are those people and organizations you deal with to acquire new technology and resources. Vendors can run the gamut from being very helpful, to being very annoying and downright unlikable. One of the things you should work on is developing relationships with vendors you like. These vendors have resources that you can call on when you need help. One of the benefits of having a good vendor relationship is that vendors are all too happy to loan you equipment or software licenses, especially when you are in the process of evaluating new technology. Don't abuse this avenue, but also don't be afraid to ask for help. Vendors are out to make money, and they know that if they don't help you, there's a good chance their competition will.
- **Conduct your own evaluation—**While talking to other administrators, doing your homework, and obtaining help from vendors can help you come to the right conclusion, *you* are still your best avenue for evaluating the right solution to a problem. If you have the resources, create a test lab where you can evaluate new technology, tinker with configurations, and re-create your environments. Bottom line is: No one knows your environment better than you do, so do as much leg work as you can yourself so you are confident with the solution you present to your users. Remember, your neck is on the line, so make sure you have all the information before you decide on which path to take.

Using these resources should help you find the right technical solution to almost any problem. Yet most problems require more than just technology. Once you come to the right decision, you'll need to implement the solution, train the users, and provide ongoing support to make the project successful. Choosing a technological solution that is inexpensive, but that requires many hours of training and countless long nights of support, may not be the correct solution. You need to weigh all the alternatives, comparing not just the technology, but the dollar and human costs involved, before implementing a new technology.

DELIVERING A BUSINESS CASE

So, you have looked at all the technological possibilities and you're ready to go to the boss and share your conclusions. Before you make that visit, you might want to create a document that supports your conclusions, as many a good solution has been brushed aside when the value of the technology was not made clear. You need to present a concise list of all the costs, benefits, and limitations of the solutions you investigated.

Determining Product Costs

When building a business case, if you can create a comparison that encompasses the following six elements, you will have captured the major costs, and hopefully, give the boss all the information necessary to give you the green light on your project.

- **Hardware—**While this may seem simple enough, you would be surprised at how many items in this category get overlooked when attaching cost to hardware. Few administrators will miss the cost of the server, monitor, printer, and scanner, yet many will overlook the cost of network ports, uninterruptable power supplies, and cables to tie it all together. Most commonly, administrators will overlook shared or commodity resources, such as network equipment, power strips, and cables. If you are only configuring one server, this may not be an issue. But, if you are working on a large project with a number of systems, these items can add up fast. Try to develop set formulas for recovering these costs, and make sure your manager understands the issues.
- **Software—**Software encompasses everything installed on your systems, including the operating system, virus software, system utilities, and user applications. These costs can be overlooked as customers focus on the usually larger hardware costs. While a single

missed application may only cause a small increase in overall cost, it is bad form to repeatedly ask management for more money so that you can install a critical piece of software down the line.

- **Installation—**Once you have all the parts in hand, how much time will be needed to assemble them into a working, breathing system. Is the installation straightforward, or is it a special case that will require added time or even outside resources? Determining the cost of time will be discussed in the next section,"Determining Labor Costs."
- **Ongoing support**—Now that you have the system installed, what will the day-to-day management of the solution require. Is it a simple system that will run almost by itself, or is it a complex configuration of components that will require the full-time attention of a single person? While estimating this is not an exact science, failure to ballpark this number correctly will greatly impact the lifespan of the project. Make sure your management team understands that systems require maintenance, upgrades, patches, and *backups*, and that these things cost time and money. If these costs are not favorable to your operation, then maybe you need to find a cheaper hardware or software solution, as skimping on support is always a recipe for failure.
- **Maintenance contracts—**These are the costs for ongoing support of software, or the ability to replace a piece of hardware in a short period of time. For software, maintenance is required to keep licenses intact and provide upgrades and bug fixes as the software package matures and changes over time. On the other hand, hardware contracts are akin to insurance in that they generally are not required, but if not utilized, can result in very high costs when a hardware component goes belly-up unexpectedly. Selling hardware contracts can be difficult, and the best way to explain them is to give replacement costs for key components. If they choose not to sign up, make sure they understand that you cannot guarantee that parts will be available if a failure occurs, and that you can't commit to have the system back on-line in a defined period of time.
- **Training—**Training is about the hardest cost/benefit to capture. While almost all technical types understand the benefits of a well-focused week of training, their management may have the attitude of "Go buy a book and learn all about it." Always try to include training costs when you're looking at implementing new hardware and software that doesn't fit into your current skill set, or those of your users.

Determining Labor Costs

Now that you've survived Accounting 101, and you've learned to capture all the piece-wise cost of a new project, you may ask,"How do I capture the cost of the time and effort that is put into making the project a success?"As the saying goes,"Time is money,"and as long as you're not donating your time for free Internet access, whenever you are working on a project, it is costing someone money.

If you are a contractor, determining cost is easy, as your consulting company charges a fixed hourly rate that is easy to factor into your business case. However, if you are paid a salary, plus benefits in some shape or form, try a formula like the following:

Hourly cost = Salary + Benefits/(Number of working hours)

Simple enough, but now you must calculate benefits and hours worked. If you work for a large corporation or organization, your human resources representative may be able to provide you with a good working number. Otherwise, a good approximation for benefits is to take half of your salary, or 50%. For number of hours worked, a good starting point is:

*52 – Number of weeks vacation * 40 Hours per week*

Of course, no administrator works only 40 hours per week, but why advertise that you are a workaholic? For an administrator that makes $60,000 per year, with 3 weeks vacation, an hourly cost would be:

*Hourly Cost = $60,000 + $30,000/((52 – 3)*40)*
= $90,000/1960
= $45.91 per hour

While these are not the world's most accurate numbers, they are in the ballpark, and when used consistently across all comparisons in a business case, will yield the result desired, which is finding the best solution.

Now that you have your information gathered, you need to create the actual business case. We provide a simple example here. You will need to create your own format that matches the needs of your operation. If you discuss this with your management team, they may provide you with specific guidelines for what to present and how to present it.

A Sample Business Case

Acme Widget Corporation needs a new email infrastructure. While their current email system works fine, they have money in the budget to either upgrade the existing email system or purchase a totally new solution. The boss

would like to see a business case that documents the three-year cost of implementation and ownership for each choice. The boss doesn't care what technology is implemented, but wants the cheapest overall solution, with no surprises lurking in the future. While the budget is tight, the boss will shell out the cash for the right email system.

Given this information, you as the administrator can now begin your research by reading magazines, surfing the Internet, communicating with your peers, and calling in vendors to give you demos of prospective technological solutions. For the sake of space and sanity, let's assume that you do your research and are able to find all the necessary information for upgrading the current system or replacing it with a state-of-the-art, new gadget. Let's assume that the administrator makes $59,0000 per year, for a $45 per hour rate.

Table 1.1 shows the cost for upgrading the existing email system. As you can see, the up-front costs of this option are low, since you can reuse the existing server with only an inexpensive upgrade. The major cost here is the four hours of support per week required to keep the system running, coupled with the $3000 to keep the hardware and software maintenance contracts in place.

Table 1.1 Costs for Upgrading Existing Email System

Upgrade Existing Email System	Cost
Hardware: Upgrade existing server	$2000
Software: Stay with old package	$0
Installation: 4 hrs to install new hardware	$180
Ongoing Support: 4 hrs per week for 156 weeks	$28080
Maintenance Costs: $1000/year for 3 years	$3000
Training: None	$0
Total 3-Year Cost	$34880

Table 1.2 shows the costs for implementing a new email system. While the start-up costs are high because a new server and new software package are required, what stands out is that now the administrator is only spending two hours per week over the next three years supporting the new application. As you can see, this is a significant saving over the old system. Also note that the training costs for both the administrator and the users is captured under "Training". This item is often overlooked, and can bring signifi-

Table 1.2 Costs for Implementing New Email System

Implement New Email System	Cost
Hardware: Purchase new server	$6000
Software: Purchase new software	$2000
Installation: 16 hrs to install server and software	$720
Ongoing Support: 2 hrs per week for 156 weeks	$14040
Maintenance Costs: $500 /year for 3 years	$1500
Training: $1200 for the administrator, $50/user for 100 users	$7200
Total 3-Year Cost	$31460

cant cost to a given solution. If the training cost per user had doubled, or the number of users was higher, the tables could have been turned, and the older email system would have been more cost-efficient.

So, you now have the solutions and the costs. Package up your results to include an easy-to-read summary and a short explanation of how you arrived at your numbers. The boss will love this, and may even give you a raise! (Don't go back and revise your hourly cost, as you could start a vicious circle!) Hopefully, your boss is reasonable and will agree with your assessment, thereby giving you the green light to start your project. Nothing is better than embarking on a new project knowing the boss, as well as any other involved parties, are in agreement from the outset, giving you time to focus on producing a high-quality product, on time and on budget.

Summary

The following chapters provide the concepts, skills, and tools necessary to create a stable environment for the user across a multi-vendor infrastructure. The most important thing to take from this book is an understanding of the fundamental principles that form the foundation of system administration. In each chapter, we discuss concepts, provide examples, and discuss policies and procedures where applicable.

As you use this book, focus on the similarities among the operating systems. Differences will always exist, but they are easier to identify and handle when recognized as exceptions to the functional model, and not the definition of our occupation.