Chapter 12: System Management

12.7 Windows Application Management

As you study this section, answer the following questions:

* What is the difference between a traditional desktop application and a Metro app from the Windows Store?
* How is a shortcut different from an executable file?
* What is the difference between the Program Files and the Program Files (x86) folders? Which operating systems have the Program Files (x86) folder?
* What group membership is required for a user to install or uninstall applications?
* How can older applications be configured to run on newer versions of Windows?
* Which tool should be used to schedule an application to run automatically in the future?

In this section, you will learn to:

* Install, uninstall, and repair desktop applications
* Install and uninstall apps
* Run an application as an administrator
* Configure compatibility mode for an application
* Schedule a task to run automatically

Key terms for this section include the following:

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| Application | A software program such as web browsers, word processors, games, and utilities. |
| Compatibility Mode | A feature that allows you to run an older program using settings from a previous version of Windows. |
| Program Files | The directory name of a standard folder in Microsoft Windows operating systems in which applications are typically installed. |

This section helps you prepare for the following certification exam objectives:

|  |  |
| --- | --- |
| **Exam** | **Objective** |
| TestOut PC Pro | 2.2 Given a scenario, use operating system features and utilities  2.2.4 Manage applications and processes |
| CompTIA 220-1002 | 1.5 Given a scenario, use Microsoft operating system features and tools.   * Administrative   + Computer Management   + Device Manager   + Local Users and Groups   + Local Security Policy   + Performance Monitor   + Services   + System Configuration   + Task Scheduler   + Component Services   + Data Sources   + Print Management   + Windows Memory Diagnostics   + Windows Firewall   + Advanced Security   1.6 Given a scenario, use Microsoft Windows Control Panel utilities.   * Programs and features |

12.7.1 Windows Desktop Applications

**Windows Desktop Applications**0:00-0:13

In Windows, you can install traditional desktop applications and apps from the Microsoft Store. In this lesson, we're going to focus on managing desktop applications.

**Application Compatibility**0:14-0:45

The first thing you need to do before you start installing a desktop application is verify that the application is compatible with the version and architecture of Windows you're using.

For example, if you're using a 32-bit version of Windows, then you need to install the 32-bit version of the application. If you're using a 64-bit version of Windows, then you can usually install either the 32-bit or the 64-bit version of the application. Of course, the 64-bit version will offer better performance than the 32-bit version, so you'll want to choose that one.

**Scan for Malware**0:46-1:06

Next, you should scan the installer file for malware if it was downloaded from a website. This is important because internet downloads are becoming the primary way to distribute software. Instead of going to a store and buying a boxed copy of an application, most users purchase an application from the vendor's website, download it, and install it.

**Restore Points**1:07-1:39

When you install an application, it might create a restore point for you automatically. If not, you will want to create one manually. This is a task that many system administrators skip because problems don't typically occur during an application installation.

If the application install goes okay, then everything is fine. But sometimes, installing an application causes problems. If that happens, you're going to have big problems because uninstalling the misbehaving application may not fix all of the problems that it caused, so create a restore point just to be safe.

**Installer Files**1:40-2:12

After the restore point is created, we can run the installation with the installer file. Sometimes it's an executable file, such as .exe. Sometimes it's a .msi file. You can use either of these file types to launch an installation wizard and install the application on the system. Sometimes you may see the installer distributed within a compressed archive file, such as a .zip file. If this is the case, you have to unzip the archive file first and then run the installer file that was extracted from the archive.

**Programs and Features**2:13-2:40

After a desktop application is installed, we can manage it using Programs and Features in Control Panel.

For example, you can change an installed application. Some applications, such as the program I've installed here, Jing, allow you to change or customize an application even though the installation process is already complete.

When you select Change, Windows re-launches the application's installer, so you can go through the install again and make changes to the application.

**Repair Option**2:41-3:16

You can also use Programs and Features to repair an existing software installation.

From time to time, you may run into situations where you find an application is crashing all the time, or it's behaving erratically, or maybe it just freezes up completely.

For example, some programs share DLL files. You might install a new application, and it copies a DLL file to the computer that's already on the computer, and both files have the exact same name. Along the way, the developers of the new application modified that DLL file for their application, and now it doesn't work with the old application.

**Uninstall Option**3:17-3:59

When that happens, you might have to uninstall the new application using the Uninstall feature and reinstall the old application.

There are going to be times when you need to completely remove an application from the system. For example, you may not use the application anymore, and you want to reclaim the disk space it uses.

The Uninstall feature also works well in situations when an application is misbehaving or malfunctioning and needs to be fixed, but the installer doesn't offer a repair function. In this situation, the best way to resolve the problem is to simply uninstall the application and then re-install it from scratch.

To do this, select the application you want to uninstall, and then click the Uninstall option, and it will remove the application from the system.

**Summary**4:00-4:09

That's it for this lesson. In this lesson, we talked about the process you should follow to install a desktop application. We also talked about managing, repairing, and uninstalling desktop applications.

12.7.2 Manage Windows Desktop Applications

**Manage Windows Desktop Applications**0:00-0:15

In this demonstration, we're going to spend some time looking at desktop applications and how to manage them. This is kind of an important topic when you're dealing with Windows 10 systems because there are actually two different types of applications you can install.

**Install Desktop Applications**0:16-3:34

You can install either desktop applications or you can install apps from the Microsoft Store. For our purposes here we're going to focus just on desktop applications. We'll look at Microsoft Store apps separately.

Managing desktop applications on Windows 10 is done pretty much in the same manner as you manage desktop applications on older Windows operating systems. The first thing we want to talk about is how to install a new desktop application on the system. I've already downloaded an application from the internet into my Downloads folder and it's called Audacity. Audacity is a very good quality audio editing tool that's open source and therefore free, which is why I really like it.

So I've already downloaded it from the internet. I've already scanned it to make sure it's free of malware. At this point, I can go ahead and install it. Before I do that, I do want to point out that you will commonly see two different types of files used to install Windows desktop applications. One is an executable file such as we see here, in which case we just run the executable and the installer kicks off.

You may also see files such as this one, a .msi file. A .msi file can also be used to install a desktop application on a Windows system. A .msi file is a Windows Installer package.

Occasionally an application may be distributed in a zip file. If that's the case, then you're first going to have to extract all the files from the zip file before you can run the installation. For our purposes today, we are going to use a Windows desktop application installer that is an .exe file. To do this all I have to do is double-click the file.

Because we are installing new software on the system, we do need to give permission, we need to elevate privileges to do so because the installer is going to make changes to the system. We'll click Yes to go ahead and allow it. Now we'll walk through the Installation Wizard selecting our language, agree to the license agreement, specify the installation location, and this brings up a really important point, before we go any further.

Notice here that it's going to install the application in C:\Program Files (x86). Let's bring up a new File Explorer window and go to our C:\ drive. You'll notice that there are actually two different Program Files folders, Program Files and Program Files X86. If you're dealing with a 32-bit Windows operating system, and yes there are still a few of those floating around, you will not see this folder right here. All of your applications will go into Program Files. But if you're working on a 64-bit Windows operating system, then we will have two different Program Files folders. 64-bit applications will be installed in Program Files, while 32-bit applications will be installed in Program Files (x86).

As you can see here, Audacity must be a 32-bit program because it wants to go into Program Files (x86). I could put it somewhere else if I wanted to. There's nothing magical about putting it in the Program Files (x86) folder, it's just that it keeps things nice and organized.

I'll go ahead and click Next. We will go ahead and create a Desktop icon and run the installer by clicking Next again.

I'll give it a minute to install.

Alright, at this point my app has now been installed. Click Next.

I'll deselect the option to launch Audacity right now. We'll click Finish.

When I come to my Windows menu, you'll see that I have an icon added to the menu to launch Audacity because it was a recently added app.

That's how you install a desktop application on a Windows 10 system.

**Change, Repair, and Uninstall Applications**3:35-5:46

With that in mind, let's now talk about how you manage desktop applications. To do this, we use Programs and Features in Control Panel.

Come down here and type ‘Control Panel'. From the Control Panel main screen, you'll want to go to Programs > Programs and Features. When you do, you see a list of all the installed desktop applications on the system. Here's the Audacity application we just installed.

When I click a particular application, different options are displayed up here that allow me to do different things. For example, if I wanted to uninstall Audacity I would use this option right here.

Others might have slightly different options. For example, this CCleaner application gives me the option of either uninstalling or changing this program. The uninstall option, of course, just removes the application files from the system.

The change option will basically do a reinstall of the application. It'll run through the Wizard a second time and that will allow you to change various options used by the application. For example, you could add a particular feature to the application that wasn't included during the initial installation process or you could remove a feature that was installed during the installation process that you don't really want.

Some applications will also include a repair option as listed here. I don't believe any of these include a repair option, it doesn't look like it. A few will. If you select an application and then click the Repair option up here, the application installer will go through and examine the application's files and it will try to identify any program files that are corrupted or maybe missing. If it finds any of these types of problems, it will then replace that file with a good version.

So let's practice uninstalling an application. Let's suppose we want to uninstall Audacity now. We've used it. We're done with the project. We don't need it anymore and so we're going to uninstall it from the system. I select Audacity, I click Uninstall, and I'm prompted as to whether I really want to remove Audacity and all its components. I say "Yes, please do" and the uninstall is complete and Audacity is gone from the system. I get a warning that there are a few elements remaining but I'm not going to worry about that.

I'm going to go ahead and reinstall Audacity one more time because we're going to need it here for the last part of this lesson. I'll pause this while I do so and resume recording after it's installed.

**Pin Applications to the Start Menu**5:47-7:09

In the last part of this demonstration, I want to show you how to optimize access to applications on the Desktop itself. Because there will be some applications that you use all the time and others that you don't use very often. We want to make it fast and easy to start those that we want to use.

One option for providing quick access to an application is to include its icon on the Desktop. We have two of those here. If I double-click Audacity's icon it loads. If we right-click on this icon and go to Properties, we see that this is not actually the application itself, this icon on the Desktop. Instead, it's just a shortcut that points to the Audacity executable. The path to that executable file is listed here under Target.

You can also add tiles for frequently used applications to your Start menu. If I click Start right here, you'll notice that Audacity is not listed in the tiles.

If I wanted to use the Start menu to access Audacity every time I wanted to use it, I'd have to click the Windows icon and locate Audacity from the list. There it is right there. Well, that's kind of a pain.

If I'm going to use Audacity all the time, then I want it to be displayed immediately when I click the Windows icon. I don't want to have to go dig for it in All Apps. One way to do that is to right-click the tile over here and then click Pin to Start. When I do, a tile appears over here on the Start menu and now whenever I click on the Windows icon, I see the tile to launch Audacity.

**Pin Applications to Quick Launch**7:10-7:30

We can also add an icon for frequently used applications to the Quick Launch area on the taskbar. This allows me to launch an application just by going to the taskbar and clicking the icon. If we go over here, locate the Audacity shortcut, and right-click, I can select, Pin to taskbar, in which case we have an icon added down here.

**Run an Application as Administrator**7:31-9:58

There's one last thing I want to talk about before we end this demonstration and that is the fact that all versions of Windows since Windows Vista implement user access control. In essence, whenever we run an application, such as coming down here and clicking Audacity, the application runs as a limited user on the system. This prevents malware from really mangling your system like it did in Windows XP and earlier.

That wasn't the case with earlier versions of Windows. If I were logged in as administrator to say a Windows XP system and I launched an application, it ran with administrator level privileges. That's no longer true.

I'm currently logged into this system as an administrative user but when I run Audacity it runs as a limited user, again to prevent malware from potentially mangling your system.

There may be times when that doesn't work on your system. There may be some applications, there won't be many, but there may be a few applications that when you run, you need elevated privileges on the system. If this is the case, you can come over to the application, right-click it, and use the Run as administrator option.

So if I click Run as administrator, I'm prompted to elevate privileges this time. I'll click Yes to go ahead and grant that request. Now, Audacity is running on the system as administrator. I probably wouldn't run an audio editor as administrator on the system. There may be other utilities on the system though that do need to have administrator level privileges in order to run properly.

This process works fine if you want to run an application once as administrator. But imagine you have some utility installed on your system that needs to run with administrator level privileges every single time it's run. Well, it's kind of a pain to have to come over here, locate its icon, right-click, and say Run as administrator. It'd be better if we could configure the application to just run as administrator every single time it launches.

There's a way to do that. First you have to locate the application's executable file. We already know that this one is installed in Program Files (x86), Audacity, and here's the executable Audacity.exe. I'll right- click it, go to Properties, go to the Compatibility tab, and there's an option right here, Run this program as an administrator. If I do that, then I don't have to manually tell it to run as administrator every single time. It'll run that way automatically. You'll still have to give permission to run with elevated privileges, but you won't have to dig through, find its icon, right-click, and select Run this program as an administrator.

**Summary**9:59-10:19

That's it for this demonstration. In this demo, we talked about managing applications. We first talked about how to install desktop applications on the system. We talked about how to change, repair, and uninstall applications from the system. We talked about how to pin applications to the Start menu and to Quick Launch, and we ended this demonstration by talking about how to run an application as administrator.

12.7.3 Desktop Application Management Facts

Be aware of the following facts about managing applications.

* Application installation involves more than just copying the executable files to the computer. Installation typically modifies the registry, creates shortcuts, and configures other settings required by the application.
* Installation of an application usually creates a tile on the Start menu and may also create a shortcut on the desktop. A *shortcut* is a pointer file that identifies the location of the executable file that runs the application.
  + Shortcuts that point to removable drives (such as CD/DVD drives) or network drives could become unavailable if the referenced drive is disconnected.
  + During install, you can often choose to add shortcuts for only the current user or all users.
  + The shortcut also identifies a directory that the application uses or references when it first starts. You can modify the directory by changing the **Start in** property for the shortcut or executable file.
* A 64-bit operating system can run both 32-bit and 64-bit applications. However, a 32-bit operating system can run only 32-bit applications.
* By default, applications are installed into the Program Files directory on the root of the system drive.
  + During installation, you typically can specify an alternate install location.
  + 64-bit operating system versions include an additional folder named Program Files (x86). 32-bit applications are installed into this folder.
* Users must have the correct permissions or rights to install applications. The ability to install applications depends on the user's group membership and the operating system:

|  |  |
| --- | --- |
| **Group** | **Permissions** |
| Users | Users who are members of only the Users group are not able to install applications. If you're signed into a limited account, you see a dialog box asking you to enter an administrative password. |
| Administrators | Only people with administrative privileges can install programs in Windows 10. If you're already logged on with an administrative account, Windows 10 asks you if you want to allow the program to make changes to the computer. |

* In some cases, users require special privileges or rights to run applications. For example, legacy applications that access the system in certain ways may require running the application as an administrator.
* Copying an application's shortcut to the C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup folder will cause that application to be automatically launched every time the system is booted. You can also use Task Scheduler to configure an application to run automatically based upon event triggers that you define.
* Many applications as they run create data files (such as documents or video files), and might also require creating temporary files. The user must have sufficient permissions to the directories where the data and temporary files are created.

12.7.4 Configure Application Compatibility

**Configure Application Compatibility**0:00-0:06

In this demonstration, we're going to look at application compatibility.

**Why Application Compatibility is an Issue**0:07-0:51

Understand that many older Windows desktop applications that were written for earlier versions of Windows may not run very well, if at all, on newer versions of Windows. All is not lost if this happens.

Essentially, in this situation, you have two options to choose from. One is to go out and buy a newer version of that application that is written to run on a later version of Windows. Honestly, that's usually the best choice, but there may be situations where that is not an option.

For example, maybe you have an in-house application that is used within your organization. It was written by an internal programmer and it was designed to run on an earlier version of Windows. The cost involved to rewrite that application to run on a newer version of Windows is probably prohibitive. Because it's a custom application, you can't just go out on the Web and pull down a newer version.

**Application Compatibility Settings**0:52-5:07

In this situation, you can use application compatibility on the newer versions of Windows to basically trick that application into thinking that it's running on an older version of Windows.

For example, we have this WinZip application currently installed on this Windows 10 system. I happen to know that this is a really old version of WinZip. It's over ten years old. I believe it was designed to run on Windows XP, and as such, it doesn't run very well on Windows 10. One thing we can do is use application compatibility to trick this application into thinking that it's running on an older version of Windows when it's really not.

To configure this, we'll right-click the application shortcut. You can either click the application shortcut here on the Desktop, or you can go to the file location and right-click the executable file itself. It works just fine either way.

We want to go over to the Compatibility tab. Here, we can configure various settings to help this application run better on a newer version of Windows. An easy way to do this is to use the option here under Compatibility Mode. We select Run this Program in Compatibility Mode for, and then select the appropriate operating system from the dropdown list.

Notice that we can go all the way back to Windows 95. We could select Windows 95, Windows 98, Windows ME, Windows XP, Vista, or Windows 7 or 8. Let's just pick Windows XP Service Pack 3 in this example.

This is a really useful feature because many applications are hard coded to check and see what version of Windows is currently installed on the system when they're being run. The idea being that the application wants to see what the version of Windows is. If you're trying to run the application on a really old version of Windows, it's probably designed to error out and say, "Sorry, I can't run on this older version of Windows."

The problem that sometimes happens is that the application developer may not have taken into account the fact that it might be running on a newer version of Windows. It may return an error just as if you were trying to run it on a really old version of Windows. It didn't recognize the version number of Windows that it's currently installed on because that version of Windows didn't even exist when the application was written.

In this situation, you can run the application in compatibility mode for a specific older version of Windows. When you do, it'll report back to that application that it's running on say, Windows XP Service Pack 3. In which case, the application says, "Hey, no problem, I'll run."

Basically, we're just faking it out. We're telling it that it's running on an old Windows XP system when it's not. It's actually running on Windows 10. In addition to using this option up here, compatibility mode, there are other settings that you can use down here to help that application that's running poorly to run a little bit better.

For example, you can tell it to run in 8-bit (256) color mode. We could tell it to run in 640 x 480 screen resolution. We could disable display scaling on high DPI images. All of these are designed to help really old, old applications to run better on newer versions of Windows.

Another thing you can do is use this option right here, Run this program as an administrator. Remember, older versions of Windows, earlier than Vista, did not use UAC, user account control. And as such, many of these older applications don't really play well on Windows Vista, 7, 8, and 10 because they do use UAC.

They were written never taking into account UAC. So, these older applications may just assume that they have a certain level of access to the system. When they don't have that access on newer versions of Windows because of UAC, they don't run properly.

If this is the case, you can come down here and try this option to run that application as administrator. But just be very careful when you do.

Remember, an application running as administrator has administrator level access to the system. That's why UAC was implemented in the first place because sometimes, an application might contain malicious code, in which case, it would be running on the system with administrator level access. Not good. So be sure you trust the application before you try to run it as administrator.

I'm going to turn these options off. Understand that using these options, setting them manually, is kind of a hit and miss process. You may have to, for example, come up here and say, "Okay, let's start with Windows 7. Does that work with Windows 7? Nah, it still runs cruddy. Let's try Windows Vista Service Pack 2. Nah, it still doesn't run very well."

You might have to keep going back and back and back until you finally hit an older version of Windows where the application runs. That's fine, but there's actually an easier way to do it.

**Application Compatibility Troubleshooter**5:08-6:37

You can use this option up here, Run compatibility troubleshooter. If you do this, Windows will go and detect any issues that might exist with the particular application. When you do, you get two options.

Now from experience, I know that if I try to run the program as-is, it is going to cause my system some issues. So I'm going to go ahead and pick Troubleshoot program right here.

When I tried to run this previously, it did not want to run. So if I look up here, it says, "The program worked in earlier version of Windows but won't install or run now." That is the one I am going to pick at this point. Click Next.

And then it will say, "Which version of Windows did this program work on before?" In response to that, I think that this application worked on Windows XP Service Pack 3. Click Next.

I'll click Test the program. Notice that we are prompted to elevate privileges via UAC because when this application was written, it assumed that it had full access to the system.

This is a really old application so that is most likely why I'm getting this certificate security alert. I'm going to ignore that, which normally is not a great idea, and close this window.

It looks like it is coming up just fine. I'll click use the Evaluation version, the application runs. Close the window. Click Next, and then we will save the settings for this program. The problem has been fixed. We'll click Close, and then OK.

So now, this old version of WinZip installed on this Windows 10 system will run in Windows XP Service Pack 3 application compatibility mode every time it's started.

**Summary**6:38-6:48

That's it for this demonstration. In this demo, we talked about application compatibility. We talked about why application compatibility is an issue. We then looked how to manually and automatically configuring application compatibility settings.

Test Out Logo

**TestOut PC Pro***English 6.0.4*

Home

Outline

**Actions**

Click to View Actions Menu

Copyright © 2019

TestOut Corporation

All rights reserved.

Powered by Lab Sim

Chapter 12: System Management

12.9 Digital Content Management

Copy textPrint page

As you study this section, answer the following questions:

* What is the difference between a personal software license and an enterprise (or volume) software license?
* What does the EULA contain?
* How does open source software differ from software protected by a proprietary license agreement?
* How are open source development projects funded?
* How do DRM mechanisms protect media files from illegal copy?

In this section, you will learn to:

* Select the correct software license for a specific implementation
* Explain the benefits and drawbacks of open source software

Key terms for this section include the following:

|  |  |
| --- | --- |
| **Terms** | **Definitions** |
| Digital Rights Management (DRM) | Software and hardware security limitations meant to protect digital content and prevent piracy. |
| Proprietary | Something used, made, or sold only by the particular person or company. |
| Open source | Source code for an operating system or other software whereby the source code is available for free and anyone can modify and redistribute the source code. |
| End User License Agreement (EULA) | A legal contract between a software application author or publisher and the end user of that application detailing the terms of use. |

This section helps you prepare for the following certification exam objectives:

|  |  |
| --- | --- |
| **Exam** | **Objective** |
| CompTIA 220-1002 | 4.6 Explain the processes for addressing prohibited content/activity, and privacy, licensing, and policy concepts.   * Licensing/DRM/EULA   + Open-source vs. commercial license   + Personal license vs. enterprise licenses |

12.9.1 Software Licensing

**Software Licensing**0:00-0:21

In this lesson, we're going to discuss software licensing and how it affects you as a system administrator. This topic is very important for you to understand because violating software license agreements could potentially expose your organization to litigation. If it does, it's probably going to cost you your job and you don't want that.

**Licensing Models**0:22-0:35

In today's information technology environment, there are two general software licensing models that you need to be familiar with: proprietary and open-source. The way these two models work is very different. We're going to talk about both of them.

**Proprietary Software**0:36-1:01

Let's begin by discussing the proprietary licensing model that's used by many software vendors.

Before we go any further, it's important that you understand that we will be discussing proprietary licensing in the most general of terms. Each vendor you purchase a proprietary license from should provide you with an end-user license agreement that we affectionately call the EULA. The EULA dictates the specific terms for that particular software title.

**Proprietary Development Model**1:02-2:04

Most proprietary software products, whether they're applications or whether they're an operating system, are usually developed by a for-profit organization. The software is developed as a part of a well-organized design and development effort on the part of a single software development company. Here's what usually happens in most organizations.

First, they identify a customer need.

Then, a design team is put together. They hash out a product requirements document (PRD) that specifies exactly what the product will do.

The tasks identified in the PRD are then assigned to teams of programmers who write their assigned code elements.

When that's done, the code is checked in and the product is run through a series of testing cycles.

When the product has had all of its bugs worked out, we hope, the finished product is shipped to the customer.

The customer uses that product for a period of time and usually identifies bugs that were missed during the initial testing process. They may also identify new features or functionality that they would like to see added to the product.

The software company receives that feedback from the customers and then the cycle starts all over again.

**Proprietary Licensing**2:05-3:12

There are several key facts you need to remember about software that uses proprietary licensing.

First, when you purchase the software, you're not actually purchasing the software. Instead, you are purchasing a license to use the software.

The software company retains ownership of the actual software. Because you're purchasing only a license and not the software itself, you're not allowed to access the software's source code and make any modifications. Software vendors usually do not make the source code available to customers to do this. In fact, you're usually not allowed by the EULA to reverse-engineer the software either. Reverse-engineering is the process of converting a binary software application back into its uncompiled source code.

The license you purchase usually permits you to install the software only on a fixed number of computers. Back in the old days, you were on your honor to abide by these limits. Today installation limits are commonly enforced by the software vendor using online software activation. If you try to use the same activation code too many times, the software will not activate and you'll not be allowed to use it.

**Types of Proprietary Licensing**3:13-4:34

When purchasing proprietary software licenses, you can usually select from two general types of licenses: personal or enterprise.

Personal licenses are aimed at home and small business customers. Usually, they allow the software to be installed on a very limited number of systems, maybe one, two, or a maximum of three systems. Because they're limited in the number of allowed installs, personal licenses are usually less expensive than other alternatives.

However, personal licenses may not be the best choice for a large organization that needs to purchase hundreds or possibly even thousands of licenses for a particular software title. Trying to purchase and manage this volume of personal licenses would be an administrative nightmare, as well as being really expensive.

To make things easier, many software vendors offer enterprise licenses that are also called volume licenses. Enterprise licenses allow the customer to install the software without restriction using the same activation code, typically until a maximum cap is reached.

Because of the volume involved, the customer is usually able to purchase an enterprise license for much less than the cost of purchasing multiple individual personal licenses. However, enterprise licenses are still usually very expensive. Because of this, they're usually not cost-effective for most home or small business users.

**Open Source Software**4:35-5:05

With this in mind, we will shift gears and talk about open-source software. Understand that for many years, proprietary licensing was the only option for purchasing new software. Today, however, you can choose to use open-source software as an alternative. Open-source software used to be exclusive to just the Linux and Unix operating systems. Today, this is no longer the case. Today, there are many open-source applications available for Windows and Macintosh operating systems as well.

**Open-source Licensing**5:06-5:52

Open-source licensing is very different from proprietary licensing. Open-source software is usually freely distributed. That means you can download, install, and use the software without paying a licensing fee.

Most open-source software is distributed under the GNU General Public License, or what we call the GPL. The GPL requires that the source code for the software remain freely available to anybody who wants it. That means you could download the source code for an application that you like, modify it, recompile it, and then use the modified version of the software. In fact, you could even post it for other users to download and use as well. You could do that as long as you make that the source code freely available to anybody who wants it.

**Open Source Business Model**5:53-7:46

Given the free nature of open-source software, you may be wondering why organizations and developers put their time, effort, and money into developing these applications if they're just going to give them away for free. Everybody likes to get a paycheck, right? Organizations that release open-source products typically use a variety of alternative options for generating revenue so they can keep on developing new products. There are many different options available.

For example, they may rely on contributions. Some open-source projects will simply ask you to contribute financially so they can keep on working.

Other open-source projects may release a base version of their software that they give away for free, but then charge you a fee for adding highly-desirable add-ons that really ratchet up what the application is capable of doing. In other words, they hook you with the free version and then charge you for the really good stuff.

Other open-source projects may provide support contracts. They may release their software for free, but then charge a fee for providing technical support. This is a really useful option for a large organization that wants to implement open-source software enterprise-wide. It ensures that any problems that come along with that software will be fixed quickly, instead of waiting for a system administrator to research the problem on the web for six or seven days trying to figure out a solution to the problem.

Other open-source projects may provide training contracts. Basically, they will give you the software for free and then provide you training on how to use it for a fee.

Some open-source projects will partner with a commercial organization. In this situation, two versions of the application will probably be created. One version is proprietary and another that is open-source. The proprietary version is typically used to finance the development of the free version.

Sometimes an open-source project will simply sell subscriptions. They may sell subscriptions for things like an online account or maybe cloud server access.

**Summary**7:47-7:53

That's it for this lesson. In this lesson, we discussed software licensing. We first reviewed the proprietary software licensing model. Then we discussed the open-source software alternative.

Test Out Logo

**TestOut PC Pro***English 6.0.4*

Home

Outline

**Actions**

Click to View Actions Menu

Copyright © 2019

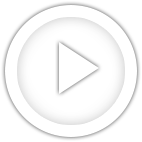
TestOut Corporation

All rights reserved.

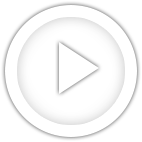
Powered by Lab Sim

12.9.1 Software Licensing





Play Video



Play with Interactive Script

12.9.2 Digital Rights Management (DRM)

Pause

0:01 / 8:06

Previous Segment

Next Segment

**CC**

**1x**

Volume

View Video in Full Screen

1. Digital Rights Management (DRM)

Copy TextPrint Page

**Interactive Script**

## Digital Rights Management (DRM) 0:00-0:54

In this lesson, we're going to discuss digital rights management (DRM). DRM is an umbrella term that encompasses many different technologies that are designed to restrict the use of copyrighted digital data.

This topic is very important for you to understand as a system administrator because you may be tempted from time to time in your job role to try to circumvent DRM technologies. For example, let's suppose your organization requires a particular application to be installed on 10 user workstations Your organization has only three licenses. In this situation you may be tempted to go on the web and search for codes that you could use to activate the other seven installations.

Doing this would be bad. It would violate the software license agreement and could potentially expose your organization to costly litigation. If it does, I can promise you that it will also probably cost you your job.

## Intellectual Property 0:55-1:46

Back in the old days, the protection of intellectual property really wasn't as much of a concern as it is today. The lack of digital technology made the infringement of intellectual property rights relatively difficult to pull off.

For example, making an illegal copy of a book would require a great deal of time and effort even if you had a photocopier. Making illegal copies of musical works was quite a bit easier, but it still required time and some audio equipment. In addition, the quality of the bootlegged work was usually considerably lower than that of the original.

That's all changed today with modern digital technology. This process is very simple to do. Copies of books, musical works, artwork, and software can be made very quickly and with a very high degree of fidelity to the original. Because digital files are very easy to copy, these bootlegged works can be widely distributed with very little effort.

## DRM Technologies 1:47-2:12

To protect the intellectual property rights of publishers, several DRM technologies have been implemented over the years. The goal of these technologies is to restrict the use of copyrighted works to only those folks who have paid the necessary licensing fees. These technologies have been used both with digital media as well as software.

With that in mind, let's review some of the key DRM technologies that you need to be familiar with, beginning with software activation.

## Software Activation 2:13-4:05

Back in the old days, most software that you purchased included a license code that you were required to enter in order to use the software after you installed it on a system. However, the verification of that license code was carried out within the application itself. As a result, it was very common for administrators, and users for that matter, to purchase a single license for a given application and then re-use that same license code over and over and over and over and over and over.

Also there were several key generation programs that were released on the Internet that you could use to generate product keys without having to pay the necessary fees to the software vendor.

To prevent this from happening today, most proprietary software vendors use a process called online software activation. Before the user can run a new application after it's installed, it first has to be activated online with a software vendor. The software vendor tracks how many times each license code has been activated and will block further activations once that license limit has been met.

For personal software licenses, the type you and I might run down to the local big box store to buy or order off the Internet, the limit is usually one to three activations. Enterprise software purchases on the other hand are usually governed by an enterprise license agreement, sometimes also called a volume license agreement. For example, an organization may purchase a 100-user license for a flat fee from the software vendor. That same license code can then be used to license all 100 installations.

Volume license agreements are usually more cost-effective for large organizations. They typically get a discount off the personal license price for doing so. However, they are usually far too expensive to be purchased by individuals or small businesses.

## Online Activation 4:06-4:41

There have been two different forms of online software activation used by software vendors. The first one is called one-time activation. The other is called persistent activation.

With one-time activation, the license code is activated once, usually right after the product is installed. Once activated, the product remains activated.

Alternatively, some products require persistent activation. With persistent activation, the license code is continuously reactivated online at a particular interval. This allows the software vendors to deactivate installations if they decide that the conditions of the license agreement have been violated in some way.

## Online Activation Drawbacks 4:42-5:51

Using online activation helps software vendors ensure that their products are used in compliance with the license agreement. However, it also has several key drawbacks.

With some online activation systems, a small change to the system can cause an application to be deactivated. For example, adding RAM to the system or maybe upgrading the CPU to a faster model could cause the activation software to think that the application has been illegally copied to a new computer system. When this happens it's a real pain. The customer usually has to contact the software vendor; try to convince them that the license agreement is still in effect and being observed; and get them to please reactivate the software.

In addition, upgrading your operating system or migrating to a new computer system can also cause software to become deactivated. Again, this situation has to be resolved with the software vendor and it requires a significant amount of time and effort to accomplish, not to mention good negotiating skills.

Online activation mechanisms can also fail if you don't have an active Internet connection. This can cause an application to deactivate when it really shouldn't be deactivated.

## DRM with Media Files 5:52-6:15

In addition to software, DRM technologies have also been commonly used with digital media files, including books, music, videos, and so on. Back in the late 2000s, most online digital media vendors implemented some type of DRM technology to encrypt the media files. Without the appropriate decryption key, the files couldn't be decrypted and played by the customer.

## Shortcomings of DRM with Media Files 6:16-7:34

In recent years, however, there's been a movement away from DRM on the part of many large digital media vendors. This has occurred because of several key shortcomings experienced with DRM.

The first one is a lack of portability. DRM-protected digital files from a given vendor sometimes would play only using software or hardware from that same vendor. For example, if you purchased a music file from one vendor and then tried to play it on a third-party music player, it would fail.

There was also a lack of backup support. DRM protection frequently made it such that protected digital files could not be backed up.

There was the issue with encryption being broken. There were many applications available on the Internet that would go through and strip the DRM protection from the digital files you purchased. With DRM protection stripped, the file was playable on any player that you wanted to use.

Obsolescence could cause older digital files to no longer be playable on newer hardware.

There was the problem of management overhead on the part of the digital media vendor. Trying to manage the DRM for legally purchased digital files required an excessive amount of management overhead on the part of the digital media vendor. A small system error on the vendor's network could cause customers to lose access to the digital files that they legally purchased.

## DRM Alternatives 7:35-7:51

To address these shortcomings, many vendors have adopted several alternatives to DRM.

First, they simply encourage their customers to not illegally share digital files. Second, they make digital files so inexpensive and so easy to access that it really doesn't make sense to make illegal copies.

## Summary 7:52-8:00

That's it for this lesson. In this lesson, we discussed digital rights management. We first looked at how DRM is implemented with computer software and then we reviewed how it's been used with digital media files.

12.9.3 Digital Content Management Facts

PC administrators need to be familiar with the following digital content management issues:

* Software licensing
* Digital Rights Management (DRM)

### Software Licensing

It is important to understand how software licensing works because violating software license agreements could potentially expose your organization to litigation. There are two general software licensing models that you need to be familiar with:

|  |  |
| --- | --- |
| **License Type** | **Description** |
| Proprietary | The proprietary licensing model is used by many software vendors. Each vendor you purchase a proprietary license from should provide an End User License Agreement (EULA) that dictates the specific terms for that particular software title.  There are several key facts that you need to remember about proprietary software licensing:   * When you purchase this type of software, you are not purchasing the software itself. Instead, you are purchasing a license to *use* the software. * You are not allowed to access the software's source code and make modifications. Usually, the source code is not made available to customers. A EULA does not typically allow you to reverse engineer the software to recreate the source code. * The license usually permits you to install the software only on a fixed number of computers. Installation limits are commonly enforced by the software vendor using online software activation. If you try to use the same activation code too many times, the software will not activate and can't be used.   Two different, general types of licenses are usually offered by software vendors:   * Personal licenses are intended for home and small business customers. Usually, they allow the software to be installed on only one to three systems. Because they are limited in the number of allowed installs, personal licenses are usually less expensive than other alternatives. However, personal licenses many not be the best choice for large organizations, which may need to purchase hundreds or even thousands of licenses for a given software title. * Enterprise licenses (which are also sometimes called volume licenses) are intended for medium and large organizations. Enterprise licenses allow the customer to install the software without restriction using the same activation code (typically until a maximum cap is reached). Because of the volume involved, the customer is usually able to purchase an enterprise license for much less than the cost of purchasing individual personal licenses. Enterprise licenses are usually too expensive for most home or small business users. |
| Open Source | Open source software used to be exclusive to Linux and Unix operating systems. However, many open source applications are now available for Windows and Mac operating systems. Open source licensing is very different from proprietary licensing:   * Open source software is usually freely distributed. You can typically download, install, and use the software without paying a license fee. * Most open source software is distributed under the GNU General Public License (GPL), which requires that the source code for the software to be freely distributable to anyone who wants it. This means you can download the source code for an application, modify it, recompile it, and then use the modified version of the software. In fact, you could even post it for others to use as long as you make your source code freely available as well.   Organizations that release open source applications typically use a variety of means to generate revenue so they can keep developing new products:   * Contributions. Some open source projects ask you to contribute financially if you use their software. * Added functionality. Some open source projects release a base version of their software for free, but then charge a fee for highly desirable add-ons. * Support contracts. Some open source projects release their software for free, but then charge a fee for technical support. * Training contracts. Like support contracts, some open source projects also provide training for a fee. * Partnerships. Sometimes an open source project will partner with a commercial organization. In this situation, two versions of an application will be created, one that is proprietary and one that is open source. The proprietary version is typically used to finance the development of the free version. * Subscriptions. Sometimes an open source project will sell subscriptions for online accounts or server access. |

### Digital Rights Management

System administrators are frequently responsible for ensuring that intellectual property rights are observed on the computers and mobile devices they are responsible for. To protect the intellectual property rights of publishers, several Digital Rights Management (DRM) technologies have been implemented over the years. The goal of these technologies is to restrict the use of copyrighted works to only those who have paid the necessary licensing fees:

|  |  |
| --- | --- |
| **DRM Technology** | **Description** |
| Software Activation | Many proprietary software vendors use online software activation. Before the user can run a newly installed application, it must be activated online with the software vendor. The software vendor tracks how many times each license code is activated and will block further activations after the license limit has been met. For personal software licenses, the limit is usually one to three activations. Enterprise license limits are governed by the enterprise license agreement. For example, an organization may purchase a 100-user license from the software vendor.  Two different forms of online software activation are used by software vendors:   * *One-time activation*. With one-time activation, the license is activated once, usually right after the product is installed. Once activated, the product remains activated. * *Persistent activation*. With persistent activation, the license is continuously re-activated online at a preconfigured interval. This allows the software vendor to deactivate installations if the conditions of the license agreement have been violated.   Using online activation helps software vendors ensure their products are used in compliance with the license agreement. However, it also has several drawbacks:   * Online activation mechanisms can fail if an internet connection isn't available. * A small system change can deactivate the software. For example, adding RAM to the system or upgrading the CPU could cause the activation system to think the application has been illegally copied to a new computer system. The customer typically has to contact the vendor to reactive the software. * Operating system upgrades or migrations can deactivate the software. |
| Media DRM | DRM has commonly been used with digital media files, including books, music, and videos. In the late 2000's most online digital media vendors implemented DRM to encrypt media files. Without the appropriate key, the files couldn't be decrypted and played by the customer.  In recent years, however, there has been a movement away from DRM on the part of many large digital media vendors. This has occurred because of several key shortcomings of DRM, including:   * A lack of portability. For example, DRM-protected digital files from a given vendor would play only using software or hardware from that same vendor. * A lack of backup support. DRM protection frequently made it such that protected digital files could not be backed up. * Weak encryption. Many applications are available on the internet that can strip DRM protection from digital files. * Obsolescence could cause older digital files to no longer be playable on newer hardware. * Excessive management overhead. Trying to manage DRM for legally purchased digital files required an excessive amount of management overhead on the part of the digital media vendor. Small system errors on the vendor's network could cause customers to lose access to digital files that were legally purchased.   To address these shortcomings, many vendors have adopted DRM alternatives such as:   * Encouraging customers to not illegally share digital files * Making digital files so inexpensive and easy to access that it doesn't make sense to make illegal copies |