

(cn)Chapter 13

(ct)Upstart, Landscape, and the Desktop Tracker

(1)Certification Objectives

- Understand Upstart
- Manage with Landscape
- Configure the Desktop Tracker
- Two-Minute Drill
- Self Test

Upstart is the replacement for init, the first process. As Upstart is intended to fulfill all of the functions of init, it includes configuration files and commands which correspond to the functionality of init and /etc/inittab.

Updates and other maintenance commands are also possible with Landscape, Canonical's system management service. Be aware that while trial subscriptions are available, Landscape requires a support subscription through Canonical.

Finally, tracker is more than just a database which facilitates searches. In the words of the Ubuntu Wiki, "Tracker is the glue that helps developers connect GNOME together". However, the UCP curriculum specifies only basic knowledge of the tracker system.

Start ItE

(s)Inside the Exam

(3)Manage events and services using upstart (123.6)

Upstart is relatively new, and is now integral in the Linux boot process. As suggested by the description in the UCP curriculum, you need to understand the configuration files in the /etc/event.d/ directory. You'll also need to know how to interact with the Upstart service with the **initctl** command.

(3)Manage systems within the enterprise using Landscape (123.7)

The Landscape service supports Web-based administration of multiple systems. You need to know how to configure Ubuntu clients, register them with a Landscape account. From the Landscape interface, you also need to know how to review a hardware and software inventory, monitor system resources, and remotely install packages.

(3)Perform Desktop searches using tracker (125.5)

The tracker system supports faster searches within files and applications. Based on the command line and graphical tracker utilities, you need to know how to search with keywords. the UCP curriculum specifies the **tracker-search-tool** and **tracker-preferences** commands as well as the tracker-utils package.

End ItE**(5)Understand Upstart**

If you haven't installed a new version of Linux lately, you may be in for a shock. There's no `/etc/inittab` configuration file in today's Ubuntu releases. Upstart, the replacement for the System V init program, is designed to meet the demands of the latest plug and play hotplug environments. During the boot process, this is especially helpful with filesystems mounted on portable and network devices. The Upstart system is also responsible for starting services, configuring terminal, mounting partitions, and more. There are five packages associated with Upstart, as described in Table 13-1.

Upstart Package	Description
upstart	The basic Upstart package, which includes the init , initctl , stop , start , and status commands, as well as control-alt-delete and sulogin scripts
upstart-logd	The Upstart logging package, which includes the console logging service
upstart-compat-sysv	The Upstart package designed for comaptibility with the original SysV init package; it includes runlevel scripts for the <code>/etc/event.d/</code>

Upstart Package	Description
	directory, as well as commands such as halt , poweroff , reboot , shutdown , runlevel , and init
startup-tasks	For essential system tasks; as of the Hardy Heron release, it adds the <code>/etc/event.d/</code> directory
system-services	For terminal scripts in the <code>/etc/event.d/</code> directory

Table 1: Upstart Packages

These packages are automatically installed even in that most minimal of installations, based on the Ubuntu JeOS (Just enough Operating System) CD.

(2)Upstart Service Scripts

Pay special attention to the scripts in the `/etc/event.d` directory. Future Linux releases may see service scripts currently in the `/etc/init.d` directory move there to become more of a part of the Upstart system. One service script was already moved there for the Hardy Heron release, the startup logging daemon, **logd**. The configuration file, `/etc/event.d/logd`, is straightforward. First, logging is stopped in runlevels 0, 1, and 6 with commands like:

```
stop on runlevel 0
```

Next, log output, unless otherwise configured, is sent to the console:

```
console output
```

The `/sbin/logd` daemon is run by default. If it is stopped for any reason, it is restarted, as depicted by the **respawn** directive.

```
exec /sbin/logd
respawn
```

(2)Upstart Scripts Replace /etc/inittab

If you remember the directives in the `/etc/inittab` configuration file, you'll recognize the contents of the other files in the `/etc/event.d` directory. As the hardware advantages of Upstart are

essentially transparent to most users, this section focuses on how Upstart provides equivalent functionality to `/etc/inittab`. For reference, I include most of the default active directives from the `/etc/inittab` configuration file installed on my system with the Dapper Drake release of Ubuntu Linux. All of these directives are incorporated in various configuration files in the `/etc/event.d/` directory.

The first line in a standard `/etc/inittab` file specifies the default runlevel. To review, for Debian-based distributions including Ubuntu, the default runlevel as signified by the following directive is 2:

```
|id:2:initdefault
```

The default runlevel is now configured with Upstart's `rc-default` configuration file in the `/etc/event.d` directory. In essence, the script takes the default runlevel from `/etc/inittab` if the file exists and a directive like that above is included; otherwise, it's configured to set a runlevel of 2.

Next, there are the scripts which are run independent of runlevel, as configured by

```
|si::sysinit:/etc/init.d/rcS
```

which effectively runs all scripts in the `/etc/rcS.d` directory. These are all "start" scripts, as they start with a capital S. In other words, they're started whenever Linux is booted. The same scripts are run through Upstart, courtesy of the `/etc/event.d/rcS` script.

The next directive from my old `/etc/inittab` specifies what needs to be done when Linux is booted into single user mode, as specified by the "recovery mode" option described earlier in the GRUB configuration file. The **S** runlevel is synonymous with the **single** shown in the GRUB configuration file, which starts "single user mode". The **wait** directive stays in the noted runlevel. The **sulogin** command, when invoked by the "recovery mode" option, boots and logs into the root account.

```
|~~:S:wait:/sbin/sulogin
```

The equivalent Upstart functionality is available from the `/etc/event.d/rcS-sulogin` file, which invokes the **sulogin** command unless `/etc/inittab` exists on the system.

These directives from `/etc/inittab` provide options; only one is run, based on the specified runlevel. For example, if the default runlevel is 2, the associated command runs the scripts in the `/etc/rc2.d` directory.

```
l0:0:wait:/etc/init.d/rc 0
l1:1:wait:/etc/init.d/rc 1
l2:2:wait:/etc/init.d/rc 2
l3:3:wait:/etc/init.d/rc 3
l4:4:wait:/etc/init.d/rc 4
l5:5:wait:/etc/init.d/rc 5
l6:6:wait:/etc/init.d/rc 6
```

The equivalent Upstart script for runlevel 2 is `/etc/event.d/rc2`, which confirms the desired runlevel and then runs the following command:

```
exec /etc/init.d/rc 2
```

Linux systems are configured to accept the Ctrl-Alt-Delete key combination often associated with Microsoft systems. When the associated signal is received, the `/etc/inittab` command specifies the following command, which starts the **shutdown** command in one second (**-t1**), limited by any existing rules specified in an `/etc/shutdown.allow` file (**-a**), if it exists, and reboots now (**-r now**).

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

In contrast, Upstart's `/etc/event.d/control-alt-delete` file executes the following command, with fewer limitations. Of course, the command can be modified to better suit your needs.

```
exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

Many server systems are protected by Uninterruptable Power Supplies (UPS). When connected to a Linux system, they may send one of three signals to the system, which was interpreted by `/etc/inittab` with the following commands:

```
pf::powerwait:/etc/init.d/powerfail start
pn::powerfailnow:/etc/init.d/powerfail now
po::powerokwait:/etc/init.d/powerfail stop
```

These options require the `powstatd` package, also known as the "Configurable UPS Monitoring Daemon". It depends on the noted `/etc/init.d/powerfail` script as well as the `/etc/powstatd.conf` configuration file. But as `/sbin/powstatd` is now a regular daemon controlled by the complementary `/etc/init.d/powstatd` script, it's technically not part of Upstart.

Linux systems typically include 6 consoles, also known as gettys. In `/etc/inittab`, they're started with the following directives, where all six gettys are made available in runlevels 2 and 3.

NOTE Command line console terminals are known as gettys, not getties, based on a somewhat odd acronym, as a getty get a TTY, short for teletypewriter. The TTY is the predecessor for the current command line console, which "gets" the TTY.

```
1:2345:respawn:/sbin/getty 38400 tty1
2:23:respawn:/sbin/getty 38400 tty2
3:23:respawn:/sbin/getty 38400 tty3
4:23:respawn:/sbin/getty 38400 tty4
5:23:respawn:/sbin/getty 38400 tty5
6:23:respawn:/sbin/getty 38400 tty6
```

In the Upstart system, each of these gettys are started with `tty x` files in the `/etc/event.d` directory, where x represents the terminal number. While these files are more descriptive, they contain the same information. All six terminals are started in runlevels 2 and 3. When a user logs out of a terminal, the **respawn** directive regenerates that terminal with a new login screen. Communication in a text screen is plenty fast at 38400 bits per second.

(2)The Role of `initctl`

The jobs associated with the boot process are a subset of the standard processes which run on any Linux system. Each startup job has a process identifier, and the **initctl** command controls these jobs. There are several other commands linked to **initctl**, including **start**, **stop**, and **status**.

But first, to review current jobs related to the Upstart system, run the following command:

```
$ sudo initctl list
```

The output lists jobs available from the scripts configured in the `/etc/event.d/` directory. Generally, these jobs are either waiting to be used:

```
| control-alt-delete (stop) waiting
```

or they're running:

```
| tty1 (start) running, process 5889
```

Jobs in the `/etc/event.d/` directory can be started with the **sudo initctl start *filename*** command. You could also substitute *stop* or *status* to stop or review the status of a job. The *emit* option starts and then stops a job. Thus, there are two ways to execute the command in the `/etc/event.d/control-alt-delete` script:

```
| $ sudo initctl start control-alt-delete  
| $ sudo initctl emit control-alt-delete
```

Technically, the **initctl emit** command can be used to add switches or other data to a script configured in the `/etc/event.d/` directory. But once the reboot cycle is started, the **initctl emit** command doesn't get a chance to stop the shutdown process.

For more information on the capabilities of **initctl**, run the **initctl help** command.

(5)Manage With Landscape

Landscape is Canonical's Web-based system management service. It allows administrators to manage the Ubuntu systems on their networks from a single Web-based interface. While Landscape is not freely available, it is a convenient administrative front-end, which makes it possible to monitor and administer individual systems without always having to connect to each system. For those of you familiar with other system management tools, Landscape is functionally similar to the Red Hat Network, Microsoft's System Management Server (or System Center Configuration Manager), and Novell's Zenworks Suite.

More information, including the conditions for a free trial, is available from www.canonical.com/projects/landscape. When I asked for a free trial, I was sent an email invitation for an account, as depicted in this chapter. The account is limited to five systems. Full access is available with any Canonical support subscription.

(2)Landscape Features

Landscape supports the system management of Ubuntu computers. It allows you to administer users, security updates, and even custom repositories. Several systems with common requirements can be managed as a group. Packages can be managed individually by system or group.

The Landscape system monitoring tools allow you to review resource usage, manage processes, and audit actions by other administrators. It also supports a hardware inventory of each system. It includes direct feedback to the Canonical support team. As they have read-only access to the information in your Landscape account, it helps them diagnose any problems you may have.

(2)Create a Landscape Account

If you do not have a Canonical support subscription, you can get a 60-day trial of Landscape. As of this writing, it's based on registration at <http://forms.canonical.com/dashboard/survey/response.jsp>, which is also currently available from www.canonical.com/landscape/register. If you're interested in more permanent access to Landscape, the registration form allows you to say if you're "interested in buying Landscape now".

Once you receive an email confirmation from Landscape, make a note of the registration account, such as *name-first*. You'll need this account name in a moment.

Navigate to (or click on) the embedded link, and follow the instructions. Remember your passphrase, as you'll need it when logging into Landscape, as shown in Figure 13-1. It can and probably should be different from the registration password. You'll also need the registration password when registering Landscape systems.

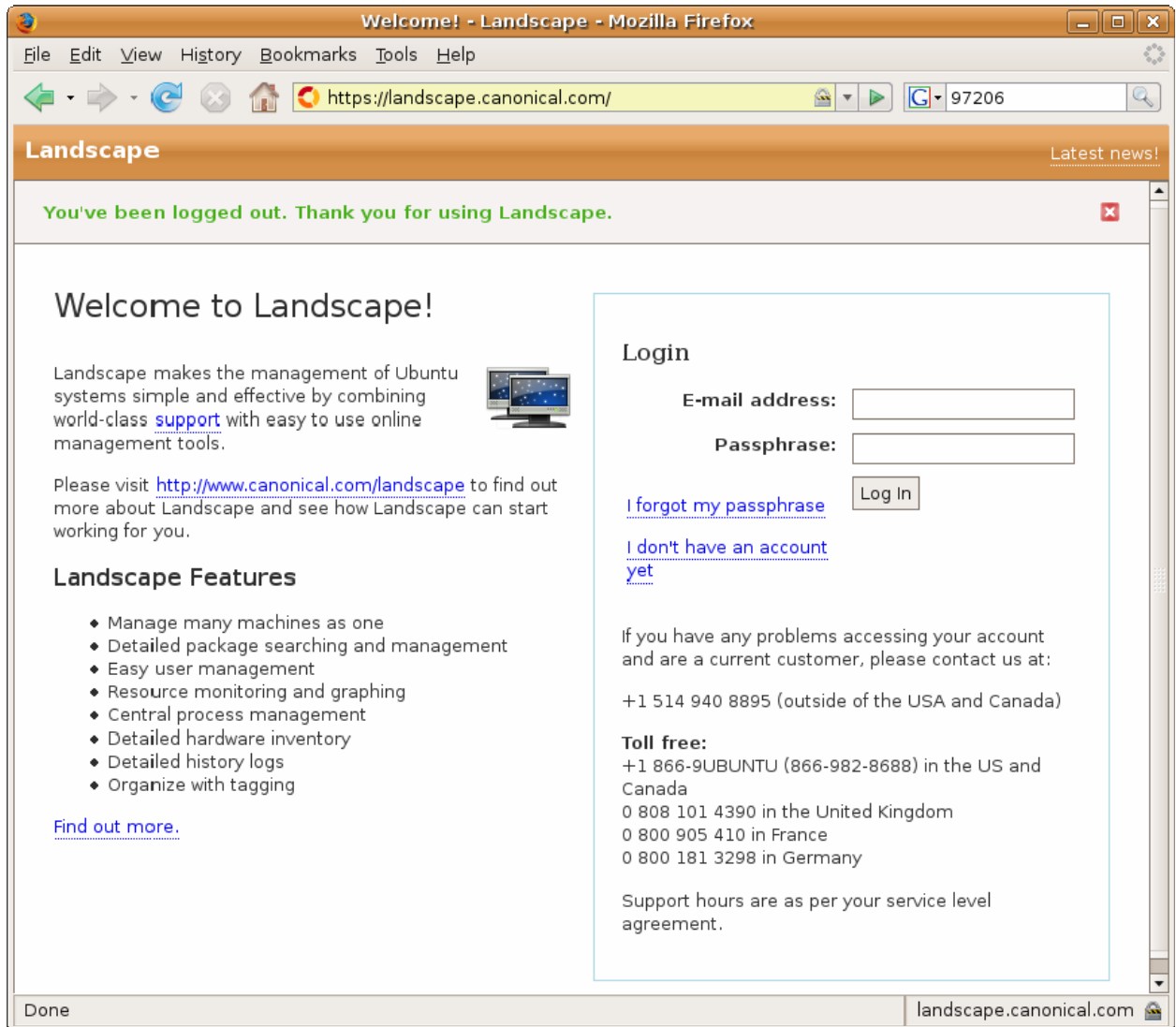


Figure 13-1. Landscape Login Screen

(2)The Landscape System Registration Process

To register an Ubuntu client with Landscape, you need to update the associated `/etc/apt/sources.list` configuration file with the appropriate repository. If you're registering a system based on the Hardy Heron release, add the following line to that file:

```
deb http://landscape.canonical.com/packages/hardy . /
```

If you're working with a different release, substitute for *hardy* accordingly.

While not absolutely required, you should download the appropriate GPG (GNU Privacy Guard) key with the following command:

```
$ gpg --keyserver-options no-http-proxy --keyserver keyserver.ubuntu.com \  
--recv-key C605E80D
```

If there is a proxy server between the local system and the Internet connection, replace *no-http-proxy* with *http-proxy*, and you'll be prompted for such as needed. If you don't acquire the GPG key, there will be "GPG error" and "Warning" messages to that effect.

If successful, you'll see a message about a public key associated with the "Landscape Development Team", along with several keys in the current user's home directory, in the `.gnupg/` subdirectory. You can then make the key available to the **apt-*** commands as follows:

```
$ gpg --armor --export C605E80D | sudo apt-key add -
```

The message I see when this command is successful is:

```
OK
```

Now you can update the local repository cache, to include information from the newly added Landscape-related repository:

```
$ sudo apt-get update
```

And install the Landscape client package:

```
$ sudo apt-get install landscape-client
```

The command installs a number of dependent packages. I tried it on three different Hardy Heron systems; between 7 and 41 were installed from my configuration to help enable remote administration. In the first case, most of the dependent packages were already installed. In either case, expect this command to take some time to download and install associated packages.

Finally, the following command starts the process of registering the client. You'll be prompted for required information. The computer title can be a descriptive human readable name; my entries are in bold:

```
$ sudo landscape-config
```

```
The Landscape client must be started on boot to operate correctly.
```

```
Start Landscape client on boot? (Y/n): Y
```

```
Starting Landscape client: landscape-client
```

```
.
```

```
This script will interactively set up the Landscape client. It will ask you a few questions about this computer and your Landscape account, and will submit that information to the Landscape server. After this computer is registered it will need to be approved by a account administrator on the pending computers page.
```

```
Please see https://landscape.canonical.com for more information.
```

```
The computer title you provide will be used to represent this computer in the Landscape user interface. It's important to use a title that will allow the system to be easily recognized when it appears on the pending computers page.
```

```
This computer's title: Hardy Heron Test One System
```

You'll also need the account name from the email, the registration password (which is the passphrase described earlier), any applicable proxy server information. While Landscape allows script management of clients from the Web-based interface, it doesn't normally allow it from the client. Substitute accordingly for *name-first* and *password*. A registration password isn't necessarily required; after logging into your Launchpad account, review <https://landscape.canonical.com/account/name-first> for more information.

```
Account name: name-first
```

```
A registration password may be associated with your Landscape account to prevent unauthorized registration attempts. This is not your personal login password. It is optional, and unless explicitly set on the server, it may be skipped here.
```

```
If you don't remember the registration password you can find it at https://landscape.canonical.com/account/name-first
```

```
Account registration password: password
```

```
Please confirm: password
```

```
The Landscape client communicates with the server over HTTP and HTTPS. If your network requires you to use a proxy to access HTTP and/or HTTPS web sites, please provide the address of these
```

```
proxies now. If you don't use a proxy, leave these fields empty.
```

```
HTTP proxy URL:
```

```
HTTPS proxy URL:
```

Be aware that this information is saved in clear text in the `/etc/landscape/client.conf` configuration file – in clear text.

```
Enable script execution? [y/N] y
```

If you Enable Script Execution, the **landscape-config** utility prompts for users with appropriate permissions:

```
Script users: michael,donna
```

Finally, the **landscape-config** utility prompts you to request a new registration for this computer:

```
Request a new registration for this computer now? (Y/n): Y
```

```
Please wait... System successfully registered.
```

Now confirm the connection from your Landscape account. Log in to the account at <http://landscape.canonical.com>, as shown in Figure 13-1. Use the e-mail address and passphrase used during the registration process. When the account accepts the login, it takes you to <https://landscape.canonical.com/dashbord>, with a link to pending computers. Click the link, and you'll be able to accept (or reject) that computer as one to be administered from this Landscape account.

(2)Work With Landscape

In this section, I'll show you the basic tools associated with Landscape, with an emphasis on the monitoring and administration of remote systems. I assume you're already logged into your Landscape account, and have registered at least a couple of computers. If you haven't already done so, this is an excellent opportunity to master the techniques associated with registering a Landscape system, as described so far in this section.

The administrative power of Landscape comes from the way computers can be configured. The standard Landscape screen is shown in Figure 13-2, and the different monitoring options are shown at the bottom of the screen.

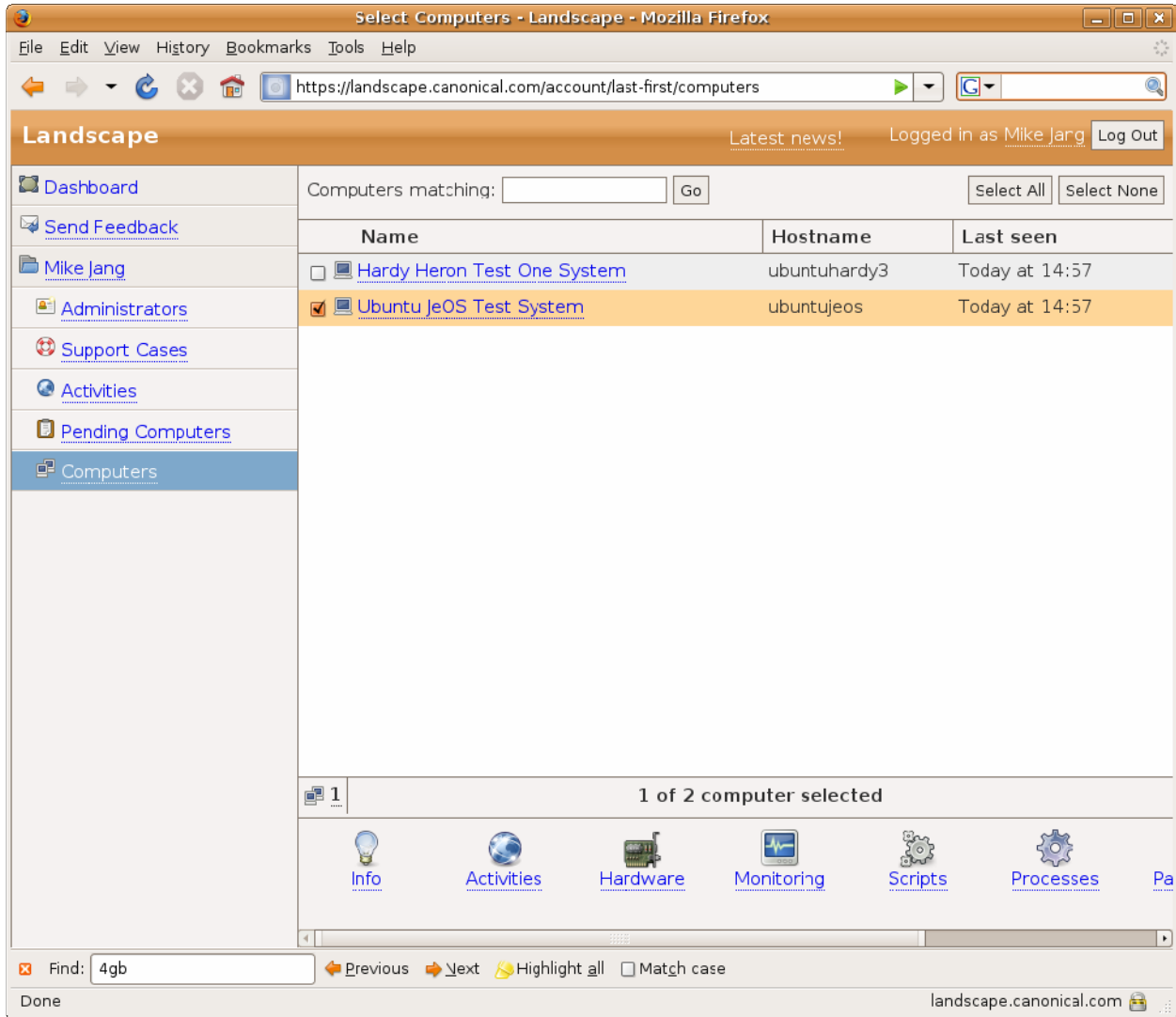


Figure 13-2. Landscape Control Screen

(3)Info

Basic information for each configured system is included in the Info screen, including the release, CPU, and amount of RAM.

(3)Activities

Any currently running or pending activities are listed on this screen.

(3)Hardware

Detailed hardware information for the selected system is available on this screen, similar to the information available from the **lshw** and **hal-device** commands. Further information based on how the hardware was detected is available by hyperlink. This is the "hardware inventory" described in the UCP curriculum.

(3)Monitoring

The Monitoring screen includes graphical information on system load, RAM and swap space use, system temperature, and disk usage. Similar information is available from the client command line with the **top** and **df** commands. This is the "monitor system resource usage" described in the UCP curriculum.

(3)Scripts

The Scripts screen allows you to configure and run a script, based on the shell interpreter of your choice, run as the user of your choice. As Landscape provides an administrative interface, scripts can even be run as the root user.

(3)Processes

The Processes screen displays currently all running processes. Similar information is available from the client command line with variations on the **ps x** command.

(3)Packages

The Packages screen supports lists of currently installed and upgradeable packages. This is the information available in the output to the **dpkg -l** and **apt-get upgrade** commands. There is also a Request Upgrades button which can download and install upgradable packages on selected systems. This is the "remotely install package updates to multiple machines" described in the UCP curriculum.

(3)Users

The Users screen supports management of users and groups. The options available from this screen are effective front-ends to user management commands such as **useradd**, **usermod**, **groupadd**, and **groupmod**.

(5)Work the Desktop Tracker

The Desktop Tracker is designed as much more than just a search database. When fully configured, it automatically links files to appropriate applications. But it's not that configurable or capable at least of this writing. I believe that's why the UCP curriculum is limited to saying "Candidates should be able to search for files based on keyword matches within file content using graphical and command line tools."

Not counting dependencies, there are four relevant Tracker packages, as described in Table 13-2. As described throughout the book, you'll want to install these packages with the **apt-get install *package1 package2*** or similar command.

Tracker Package	Description
tracker	The main Tracker package, which provides the object framework, daemon, metadata extractor, and thumbnail creator
tracker-search-tool	The Tracker GUI search tool and preference configurator
tracker-utils	Tracker text commands which searches by or for metafile type, folder, database query, statistics, status, and file tags
libdesklar-tracker	Tracker GUI applet

Table 2: Tracker Packages**(3)How to Disable the Tracker**

It may not be appropriate to enable Tracker on all Ubuntu systems. As a processor and disk intensive service, it could easily drain a laptop battery, or reduce the resources available to other applications. Be aware that the default Tracker installation is configured to start automatically in the GNOME Desktop Environment. I've disabled it on my laptop system in the

Sessions Preferences tool, which can be started in a GUI with the **gnome-session-properties** command. It can be disabled in the Session Preferences window, under the Startup Programs tab.

While it's possible to disable Tracker solely when a system is running on batteries, you'll want to choose whether the Tracker service should be allowed to load a system, especially right after the boot process. However, Tracker can make it easier for applications such as those associated with media players to find usable files.

(2)Configure the Tracker

The easiest way to configure the Tracker is with the GUI Tracker Preferences tool, available with the **tracker-preferences** command. As shown in Figure 13-3, it includes five tabs of detailed configuration information. Changes are made to each user's tracker.cfg file, in the `~/.config/tracker/` subdirectory.

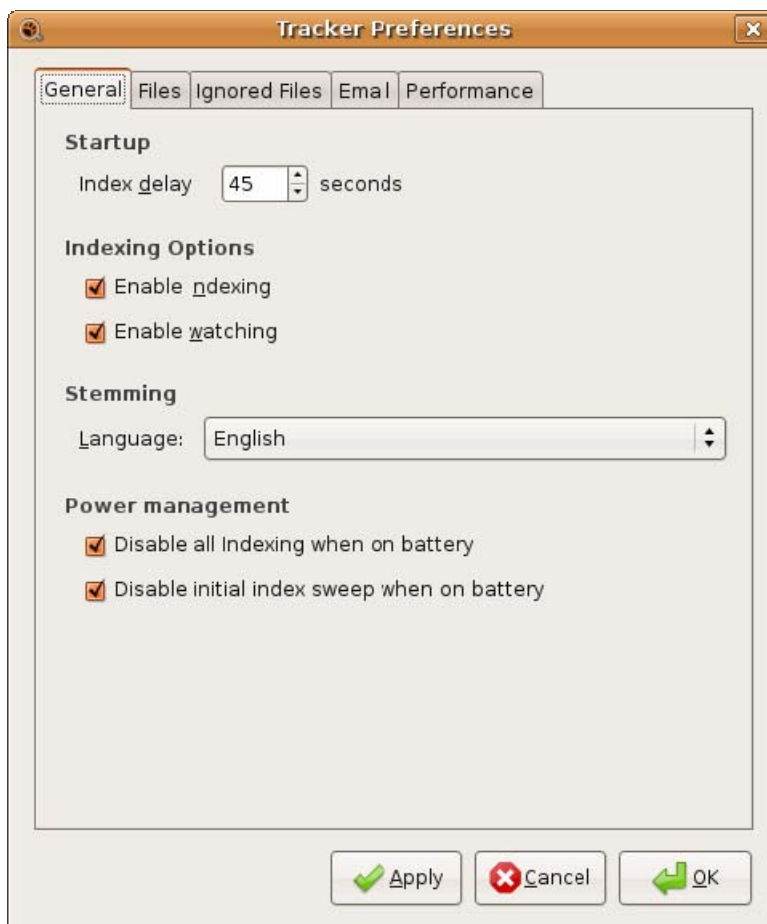


Figure 13-3. Tracker Preferences**(3)General Tab**

There are five options associated with the Tracker Preferences window, General tab. I describe them in Table 13-3, with the associated directives in each user's tracker.cfg configuration file.

Option	Directive	Description
Enable Indexing	EnableIndexing	Indexing provides the search database.
Enable Watching	EnableWatching	Watching is often associated with specific directories.
Language	Language	Searches can be configured in several languages.
Disable All Indexing When On Battery	BatteryIndex	Disables Tracker updates when AC power is disconnected.
Disable Initial Index Sweep When On Battery	BatteryIndexInitial	Disables the initial Tracker update when AC power is disconnected.

Table 3: Tracker General Tab Configuration**(3)Files Tab**

There are six options associated with the Tracker Preferences window, Files tab. I describe them in Table 13-4, with the associated directives in each user's tracker.cfg configuration file.

Option	Directive	Description
Index File Contents	EnableFileContentIndexing	Index the text contents of files.
Generate Thumbnails	EnableThumbnails	Activate the generation of

Option	Directive	Description
		thumbnails.
Index Mounted Directories	SkipMountPoints	Include mounted directories in the index.
Index And Watch My Home Directory	WatchDirectoryRoots	Include the user's home directory in the index.
Watch Directories	WatchDirectoryRoots	Include configured directories in the index.
Crawled Directories	CrawlDirectory	Index configured directories after login.

Table 4: Tracker Files Tab Configuration**(3)Ignored Files Tab**

There are two options associated with the Tracker Preferences window, Ignored Files tab. I describe them in Table 13-5, with the associated directives in each user's tracker.cfg configuration file.

Option	Directive	Description
Ignored Paths	NoWatchDirectory	Specify a directory to not watch.
Ignored File Patterns	NoIndexFileTypes	Specify a file pattern to not track.

Table 5: Tracker Files Tab Configuration**(3)Email Tab**

There are four options associated with the Tracker Preferences window, Email tab. I describe them in Table 13-6, with the associated directives in each user's tracker.cfg configuration file. The only option which can be activated is related to the Evolution personal

information manager. Thunderbird and KMail are alternative GUI email formats available for Linux. As implied by the last entry, other email applications are available.

Option	Directive	Description
Enable Evolution Email Indexing	IndexEvolutionEmails	Includes Evolution files in the Tracker index.
Enable Thunderbird Email Indexing	IndexThunderbirdEmails	Includes Thunderbird files in the Tracker index.
Enable KMail Email Indexing	IndexKmailEmails	Includes KMail files in the Tracker index.
Additional Mbox Style Mailboxes to Index	AdditionalMboxesToIndex	Includes other specified email systems in the Tracker index.

Table 6: Tracker Email Tab Configuration

(3)Performance Tab

There are six options associated with the Tracker Preferences window, Performance tab. I describe them in Table 13-7, with the associated directives in each user's tracker.cfg configuration file.

Option	Directive	Description
Throttling: Indexing Speed	Throttle	Higher values decrease indexing speed
Minimize Memory Usage (Slower Indexing)	LowMemoryMode=true	Reduces the amount of memory for the Tracker
Use Additional Memory For Faster Indexing	LowMemoryMode=false	Increases the amount of memory for the Tracker
Perform Fast Index Merges (May Affect System	FastMerges	Index merge speed

Option	Directive	Description
Performance)		
Maximum Amount Of Text To Index	MaxTextToIndex	Maximum size of text to index, in bytes
Maximum Number Of Unique Words To Index	MaxWordsToIndex	Maximum number of words to index

Table 7: Tracker Performance Tab Configuration

(2)Command Line Tracker Management Tools

There are a number of command line tools associated with the Tracker service. They're installed with the tracker-utils package. These tools include the following commands: **tracker-files**, **tracker-meta-folder**, **tracker-query**, **tracker-search**, **tracker-stats**, and **tracker-tag**. These commands by and large return errors if the Tracker daemon, **trackerd**, is not running.

To review the current status of the Tracker service, the **tracker-status** command can help. Typically, it returns a message like the following, which suggests that Tracker is at work. It could also be *Initializing* or *Idle*.

```
Tracker daemon's status is Indexing
```

If you want more information about statistics associated with the Tracker service, run the **tracker-stats** command. Statistics are associated with the different kinds of files which have been indexed; the first few lines of output from my system are:

```
-----fetching index stats-----
default : 0
Files : 47716
Folders : 24282
Documents : 3305
Images : 4713
Music : 40
Videos : 49
Text : 10061
```

```
Development : 1115
Other : 4151
Emails : 36655
```

These statistics are interesting; you can identify more about each category with the **tracker-files** command. For example, to identify all available videos on the local system, I run the following command:

```
$ tracker-files -s Videos
/home/michael/Photo/101_0941.MOV
/home/michael/Photo/101_0940.MOV
/home/michael/Photo/101_0942.MOV
```

But note the output. These files are all in my home directory. Which makes sense, as the applicable Tracker configuration file is in my home directory, in `.config/tracker/tracker.cfg`.

Now that Tracker files are indexed by their contents, they can be searched with the **tracker-search** command. When I use it to search for UCP, it reveals the following:

```
/home/michael/UbuntuCert/ch01.pdf
/home/michael/Ubuntu Book Overlap.odt
/home/michael/overlapemail.odt
```

But that's just a snapshot. I see that the PDF file for the first chapter is shown, which means the Tracker service can read the contents of PDF files. However, Tracker can't read the contents of binary files saved in Microsoft Word `.doc` format, even if such files were saved from OpenOffice.org Writer.

On the other hand, Tracker does by default read all files in the user's home directory and subdirectories. To review the results in a specific directory, the **tracker-meta-folder** command can help. Just be aware that the **tracker-meta-folder** command requires the complete path to the folder; I can use it to search for main files collected for this book with the following command:

```
$ tracker-meta-folder /home/michael/UbuntuCert
```

However, it does not search subdirectories. Yes, that makes it sort of like the **ls** command.

The **tracker-query** command depends on configuration files in `.rdf` format, which is beyond the scope of the UCP curriculum.

One popular option for Tracker files are tags, which have similarities to the meta tags associated with blog posts. The Tracker service doesn't automatically add a tag. However, if I wanted to add a UCP tag to the PDF files for this book, I'd run the following command:

```
$ tracker-tag -a UCP UbuntuCert/ch??.pdf
```

I can then review the tags associated with a specific file with the following command:

```
$ tracker-tag UbuntuCert/ch11.pdf
```

(2)Graphical Tracker Management Tools

As Tracker is configured to work in the GNOME Desktop Environment, you can take advantage of the Tracker service with several graphical tools, including the Deskbar Applet and the Tracker Search Tool. In the future, Tracker searches should be integrated with the Nautilus File Browser, but it has not yet happened for Ubuntu Linux as of the Hardy Heron release.

When enabled, the Deskbar Applet enables easy searches via an icon on the top taskbar in the GNOME Desktop Environment. It requires the installation of the deskbar-applet package. Click on the applet to open a text search window. When I use it to search for "UCP", I see PDF formatted chapters as well as emails with UCP included in the text.



Illustration 13-1.

The Tracker Search Tool is another graphical method for searching through the Tracker database. It goes one step further than the Deskbar Applet. As shown in Figure 13-4, I've used this tool to search for UCP from the Tracker database, and the output includes excerpts from associated PDF files and more.

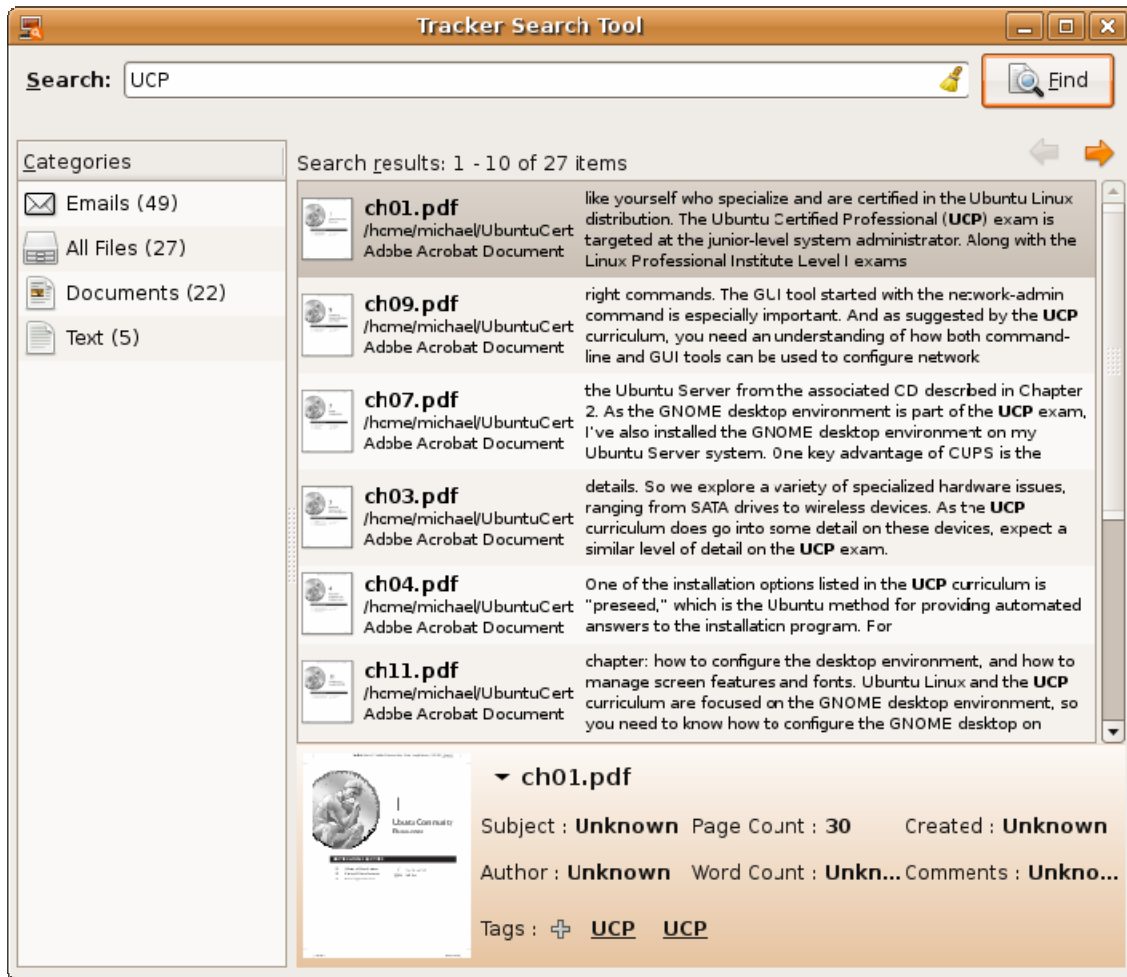


Figure 13-4. The Tracker Search Tool

(6) Certification Summary

This chapter addresses three topics included with the latest revision to the UCP curriculum: Upstart, Landscape, and the Desktop Tracker. The Upstart service is the replacement for the init process and associated configuration files. Upstart service scripts with the functionality of key directives in the older /etc/inittab configuration file can be found in the /etc/event.d/ directory.

As Canonical's web-based management service, Landscape is one way for Ubuntu administrators to manage groups of computers from a single screen. It supports the monitoring and administration of one or more connected systems. It requires registration of each client from a special repository.

The Desktop Tracker is more than just a search database. As shown in the Tracker configuration tools, it can index the contents of a variety of files in specified directories. It even can index the contents of Evolution emails. Once configured, there are a variety of graphical and command line tools available which can take advantage of the configured search database.

(6)Two-Minute Drill

Here are some of the key points from the certification objectives in Chapter 13.

(3)Understand Upstart

- Upstart replaces the init system, including `/etc/inittab`
- Upstart job scripts are stored in the `/etc/event.d/` directory.
- The **initctl** command can be used to control Upstart job scripts.

(3)Manage With Landscape

- Landscape is Canonical's Web-based system management service.
- The configuration of a Landscape client requires access to a special repository.
- Clients can be registered with the **sudo landscape-config** command.

(3)Configure the Desktop Tracker

- The Desktop Tracker is more than just a search database for file contents.
- The **tracker-preferences** command starts the GUI Tracker Preferences configuration tool.
- The **tracker-search** command can search the Tracker database.
- The Deskbar Applet and Tracker Search tool can search the Tracker database.

(6)Self Test

The following questions will help you measure your understanding of the material presented in this chapter. Read all the questions carefully, as there may be more than one correct answer. Some questions are “fill in the blank” and normally require an exact answer. Choose all correct answers for each question.

(3) Understand Upstart

1. Which of the following configuration files have been replaced by the Upstart system?

- A. /etc/init
- B. /etc/inittab
- C. /sbin/init
- D. /etc/event.d/control-alt-delete

2. Specify the full path to the Upstart configuration script associated with the first terminal getty.

3. Which of the following scripts in the /etc/event.d/ directory specifies the default runlevel for Ubuntu Linux?

- A. rc2
- B. rc5
- C. rc-default
- D. rcS

4. Which of the following commands does not restart a system?

- A. sudo restart**
- B. sudo init 6**
- C. sudo initctl control-alt-delete**
- D. sudo shutdown -h**

5. If you have a script in the /etc/event.d/ directory with additional parameters to be added at the command line, which of the following commands would you start with?

- A. sudo initctl start**
- B. sudo initctl emit**
- C. sudo initctl jobs**

D. sudo initctl status**(3)Manage With Landscape**

6. Based on the Hardy Heron release, which of the following repositories include the packages required for Landscape clients?
 - A. deb http://security.ubuntu.com/ubuntu hardy ./
 - B. deb http://us.archive.ubuntu.com/ubuntu hardy ./
 - C. deb http://landscape.canonical.com/packages/hardy ./
 - D. deb http://archive.canonical.com/packages/hardy ./
7. What is required for official continuous access to Landscape? [Two answers are correct]
 - A. Nothing. It's an open source tool.
 - B. Any Ubuntu support contract from Canonical
 - C. A separate subscription
 - D. A trial subscription
8. Specify the command which starts the steps required to register a client with a Landscape account. Assume the appropriate packages are installed. Do not use the full path to the command.

9. Which of the following packages must be installed on a Landscape client?
 - A. landscape
 - B. landscape-server
 - C. landscape-config
 - D. landscape-client
10. Which of the following tasks can not be performed from the Landscape Web-based administration tool, unless you create a specialized script?
 - A. User management

- B. Security upgrades
- C. Format new disks
- D. Load and memory monitoring

(3)Configure the Desktop Tracker

11. In what directory does the Desktop Tracker start indexing files by default?
- A. The top level root directory (/)
 - B. The /etc/ directory
 - C. The /home/ directory
 - D. The user's home directory
12. What is the default Tracker configuration file?
- A. /etc/tracker/tracker.cfg
 - B. ~/tracker.cfg
 - C. ~/tracker/tracker.cfg
 - D. ~/.config/tracker/tracker.cfg
13. Which of the following packages (not commands) include command line search tools for the Desktop Tracker?
- A. tracker-preferences
 - B. tracker-utils
 - C. tracker-search
 - D. tracker-status
14. Which of the following commands, with *ServiceType* specified, can list all indexed files of a specific type?
- A. **tracker-files -s *ServiceType***
 - B. **tracker-query -s *ServiceType***

C. **tracker-search -s ServiceType**

D. **tracker-preferences -s ServiceType**

15. If you wanted to add the tag *grits* to the *abc.def* file in a Tracker indexed file, what command would you use? Do not include the path to the command. Assume the file in question is in the local present working directory.

(6)Lab Questions

(3)Lab 1

In this lab, you'll use the **initctl** command to test various processes in the `/etc/event.d/` directory. First, review those jobs currently running with the following command:

```
$ initctl list
```

Does it work? Unless you've created a root user password and have logged in as such, it won't work. In that case, you'll need to preface the command with a **sudo**. Now try stopping the job associated with the sixth virtual console:

```
$ initctl stop tty6
```

Confirm that the job is stopped, using the **sudo initctl list** command noted earlier. Now try to access the console. Press Ctrl-Alt-F6 to access that console. Can you log in? What happened?

Now try to restore the `tty6` job. You may need to return to a different console. One way to do so is by pressing Ctrl-Alt-F1. Then run the following command.

```
$ initctl start tty6
```

Can you access the `tty6` console now?

(3)Lab 2

As I cannot assume that all readers have a Canonical support subscription, this is not a mandatory lab. Yes, I ask you to go through the basic steps to set up a trial subscription in this

lab. However, if you've done so before, Canonical does not have to grant you a second trial subscription.

1. Review the Canonical page associated with Landscape. As of this writing, it can be found at www.canonical.com/projects/landscape.
2. Review the current options for regular and trial subscriptions, as well as Landscape without support.
3. If you don't already have a subscription, and don't want to pay for "Landscape without support," register for a trial. As of this writing, the conditions for Trial subscriptions are listed at www.canonical.com/landscape/register.
4. Whatever method you use, when you receive the email inviting you to a Landscape account, make a note of the account name, and click the included URL link.
5. Specify the login information as prompted. If you're prompted to include a passphrase, be aware that passphrase can include spaces, as well as letters, numbers, and punctuation.
6. Navigate to the login screen at <http://landscape.canonical.com>. Login with the registration email address and passphrase.
7. Click on your name in the left pane. You should see a phrase such as "You can register new clients by following these instructions" with a link. As of this writing, the link leads to <https://landscape.canonical.com/account/jang-m/how-to-register>.
8. Follow the instructions on one of your Hardy Heron test clients to register that system on the now active Landscape account.

(3)Lab 3

In this lab, you'll set up the Desktop Tracker to search through files in a subdirectory. Depending on the system, it may take some time for the Desktop Tracker to actually index the text contents of all files in the current home directory, especially if you're running this lab on your home system (which I do not recommend).

This lab assumes the Desktop Tracker isn't already running. If it is, you can stop it by killing the associated process. I identify the associated process identifier with the **ps aux | grep**

trackerd command (it's in the second column of the output), and then kill the Desktop Tracker process by running the **kill** command on the process identifier number.

If this is a test system, you may want to create some files in the local directory. Use file names different from the contents of each file.

1. Run the **tracker-status** command. If the system has just started indexing, you'll see a message to that effect. An error message similar to the following indicates that the tracker service is not currently running.

```
tracker-status: internal tracker error: The name org.freedesktop.Tracker
was not provided by any .service files
```

2. The Tracker system may be indexing for quite some time. You can monitor the current status with the **tracker-stats** command.
3. After a few minutes or so, the **tracker-status** command returns the following message:

```
Tracker daemon's status is Idle
```

4. Now run the **tracker-search** command with the search term of your choice. If this is a test system, use text based on the contents of files created in the preamble of this lab.
5. Run the **tracker-tag -a IloveLinux testfile** command. Substitute an existing filename for *testfile*.
6. Run the **tracker-tag** command by itself. You'll see a list of tags and how many times that tag matches a file.
7. Run the **tracker-tag -s IloveLinux** command. Do you see the *testfile*?

(6)Self Test Answers

(3)Understand Upstart

1. **B.** The `/etc/inittab` configuration file is no longer used on current Ubuntu Linux releases, as it has been replaced by the files associated with the Upstart system.
 - There is no standard `/etc/init` configuration file. The `/sbin/init` command and

/etc/event.d/control-alt-delete script are files now provided by the Upstart system. Therefore, answers **A**, **C**, and **D** are incorrect.

2. The configuration script for the first terminal getty is **/etc/event.d/tty1**.
3. **C**. The rc-default script, in the /etc/event.d/ directory, specifies the default runlevel for Ubuntu Linux, at least as configured by the Upstart system.
 While rc2 is associated with runlevel 2, which is normally the default runlevel, it is actually specified in the rc-default script. The rc5 script is associated with runlevel 5, the default for some other Linux distributions. The rcS script is configured to run standard start scripts, independent of runlevel. Therefore, answers **A**, **B**, and **D** are incorrect.
4. **D**. The **sudo shutdown -h** command halts a system.
 While rc2 is associated with runlevel 2, which is normally the default runlevel, it is actually specified in the rc-default script. The rc5 script is associated with runlevel 5, the default for some other Linux distributions. The rcS script is configured to run standard start scripts, independent of runlevel. Therefore, answers **A**, **B**, and **D** are incorrect.
5. **B**. The **emit** option to the **initctl** command supports the use of additional parameters.
 The **sudo initctl start** command is not complete, without a specified job to start. The **sudo initctl jobs** command uses the terminal to await for changes in related job status. The **sudo initctl status** command lists the current status of related jobs. Therefore, answers **A**, **C**, and **D** are incorrect.

(3)Manage With Landscape

6. **C**. Before installing a Landscape package, most administrators will add the deb <http://landscape.canonical.com/packages/hardy> ./ line to the /etc/apt/sources.list configuration file.
 The URLs listed in the other answers relate to security, standard, and partner packages. Therefore, answers **A**, **B**, and **D** are incorrect.
7. **A** and **C**. Continuous access to Landscape is available to any user with any Canonical support subscription. A dedicated Landscape account is also available via subscription.

- Landscape is not open source, at least not as of this writing. Trial subscriptions expire after a short time. Therefore, answers **B** and **D** are incorrect.
- 8. The command which starts the steps to register a client with a Landscape account is **landscape-config**.
- 9. **D**. As strange as it sounds, the landscape-client package must be installed on a Landscape client, before it can be configured as such.
 - There are no related packages named landscape, landscape-server, or landscape-config. Therefore, answers **A**, **B**, and **C** are incorrect.
- 10. **C**. You can't format a new disk remotely on a Landscape client.
 - User management, security upgrades, and load and memory monitoring are all fairly easy tasks from the Landscape interface. Therefore, answers **A**, **B**, and **D** are incorrect.

(3)Configure the Desktop Tracker

- 11. **D**. By default, the Desktop Tracker starts indexing files in each user's home directory.
 - The Desktop Tracker starts only with user logins; unless the default configuration is changed, it only has permissions in the user's home directory Therefore, answers **A**, **B**, and **C** are incorrect.
- 12. **D**. The default Desktop Tracker configuration file is tracker.cfg, in each user's home directory, in the .config/tracker/ subdirectory.
 - The other suggested locations for the default Desktop Tracker configuration file do not exist; therefore, answers **A**, **B**, and **C** are incorrect.
- 13. **B**. The tracker-utils package includes a number of command line Desktop Tracker options, including **tracker-query**, **tracker-search**, **tracker-status**, and **tracker-stats**.
 - The other answers are not Desktop Tracker packages. Therefore, answers **A**, **C**, and **D** are incorrect.
- 14. **D**. By default, the Desktop Tracker starts indexing files in each user's home directory.
 - The Desktop Tracker starts only with user logins; unless the default configuration is changed, it only has permissions in the user's home directory Therefore, answers **A**, **B**, and **C** are incorrect.

15. The command which adds the tag grits to the local abc.def file is **tracker-tag -a grits abc.def**.

(6)Lab Answers

(3)Lab 1

This lab provides the parameters to experiment with the **initctl** command. You're welcome to use **initctl** to experiment with other file-based scripts in the `/etc/event.d/` directory. Just be aware that some of these scripts could change the current runlevel or even reboot or halt the local system.

(3)Lab 2

Be aware, this lab has to be optional, as Canonical has the right to refuse an application for a Trial subscription to Landscape.

(3)Lab 3

This lab is designed to give you a feel for some of the capabilities of the Desktop Tracker from the command line interface. I assume that GUI tools are easier for you to handle, and are therefore something you should explore on your own.