Nine projects in 90 MINUTES

Graham Morrison proves just how amazingly adaptable Linux can be by delving into nine projects that should take no longer than the kettle does to boil.

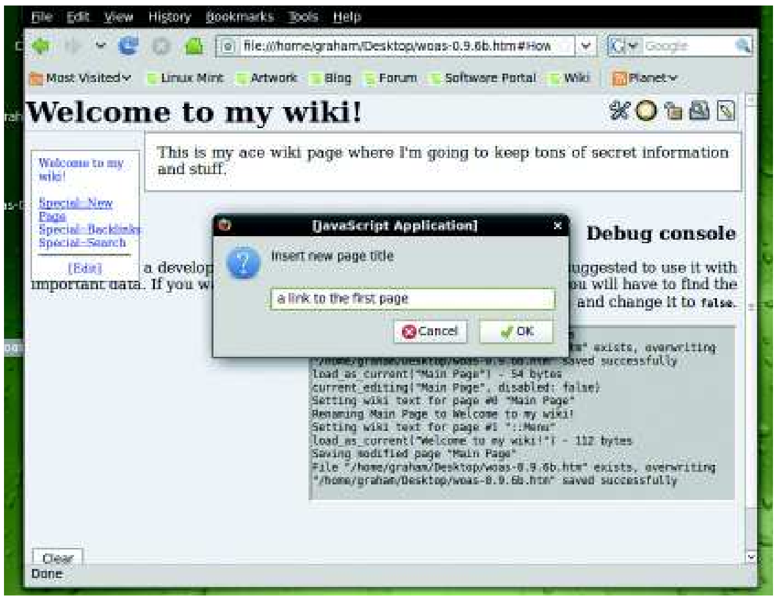
PROJECT 1 Wiki on a Stick

If you fancy taking your own customised pile of documentation and notes wherever you go, there's a perfect solution called *Wiki on a Stick*. As its name implies, this is a project that offers a go-anywhere web-based wiki that's entirely self-contained, and can be used for all the same things a full fat wiki can. The package is provided as a Zip archive, and after you've unzipped this, usually with a double click, you'll find a single **.htm** file remains.

This single file is all there is to it. The entire package is a cleverly constructed XHTML file, which uses both JavaScript and CSS to build a complete wiki that works online and offline. Just double-click to load it into your default browser. You're now running the wiki, and everything you need is right in front of you, including the documentation and a guide to getting started with the software.

"If you've contributed to a wiki before you shouldn't have any problems."

If you've contributed to a wiki before, you shouldn't have any problems with *Wiki on a Stick*. And if you haven't, then there really isn't all that much to the process. It's works just like a limited word processor. Before you get started, you need to first erase all the current data in the wiki, so you can't start your own with a blank template. Look for the Erase All Pages link at the bottom of the first page. Clicking on this and the following two 'are you sure?' messages will remove everything you can see. When the page re-appears, *Firefox* will display a security warning, and you need to click on Allow to enable *Wiki on a Stick* to do its magic. You should also click on Remember This Option so that you're not bothered by this window again.



Welcome to my wiki!

This is my ace wiki page where I'm going to keep tons of secret information and stuff.

a develop... important data. If you w...

Debug console

...uggested to use it with ...ou will have to find the ... and change it to false.

a link to the first page

```
...htm" exists, overwriting
"/home/graham/Desktop/woas-0.9.6b.htm" saved successfully
load_as_current("Main Page") - 54 bytes
current_editing("Main Page", disabled: false)
Setting wiki text for page #0 "Main Page"
Renaming Main Page to Welcome to my wiki!
Setting wiki text for page #1 "::Menu"
load_as_current("Welcome to my wiki!") - 112 bytes
Saving modified page "Main Page"
File "/home/graham/Desktop/woas-0.9.6b.htm" exists, overwriting
"/home/graham/Desktop/woas-0.9.6b.htm" saved successfully
```

› Rather than restrict yourself to the limitations of a flat text file, use *Wiki on a Stick* to transport an entire wiki environment as a single file

Wiki wiki wah-wah

You'll then be sitting in front of a completely blank wiki, and you need to click on the pencil icon in the top-right of the window to create your first page. You can then create a title and add the content according to your requirement. You can also mark up your text using the toolbar in edit mode, and when you're finished, click on the floppy disk icon to save your changes. Further pages can be created by using the menu on the left of the home page, and you can use a simple markup language that you can find listed in the documentation. When you've finished, you don't need to do anything. You've been dynamically changing the contents of the file as you make each change, which means you can simply close your browser and keep your HTML file safe.

PROJECT 2 Encrypt your files

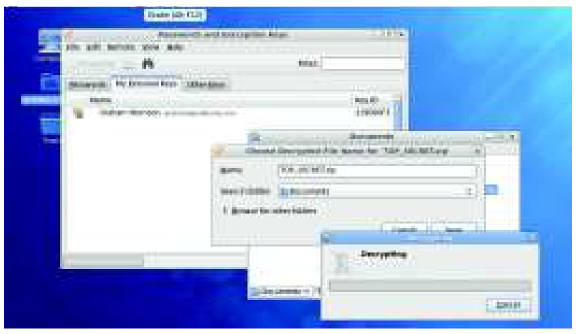
While many people look upon encryption as being a shifty technology for people who must have something to hide, the reality is that sometimes it can be absolutely essential. Bank details, credit card numbers, online receipts and invoices are commonly found on the average laptop, and this data can be extremely dangerous in the wrong hands. Encrypting this kind of data makes a great deal of sense. It will stop anyone getting hold of it should you lose your laptop, for example, and it also means you can send files through email or over an open access point without worrying about it being intercepted.

Fortunately, standard encryption no longer involves messing around with userspace filesystems or encrypting your home directory. It needs just a single right-click on the Gnome desktop, from where you can select Encrypt. If there's no such entry, you will need to install the **seahorse-plugins** package, then log out and back into your Gnome desktop.

You should then navigate to the file or folder you want to encrypt, right-click on the item, and select Encrypt. If it's the first time you've done this you'll be informed that no encryption keys have been found and that you need to build your own security credentials. Click on Okay, and the main *Seahorse* application will launch.

This is the tool behind Gnome's mysterious Passwords and Encryption Keys entry in the Accessories menu. Click on File > New and PGP Key from the list that appears. You will then need to enter your name and email address, click on Create and think of a passphrase. *Seahorse* will then generate the key, which can take a while depending on your CPU.

When this has finished, you should see the new key listed in the Personal Keys page of *Seahorse*, and you should go back to the file or folder and select Encrypt again. This time a Recipients window will open, and you should choose yourself from the list. If someone else sends you their public key, you can send files to them knowing that only they will be able to decrypt the file. You should then find a file or Zip archive with the **.pgp** extension, and your data is now secure. To decrypt the file, just double-click on it and enter your passphrase.

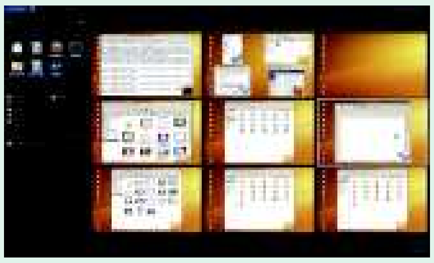

› **Encrypting your files on the Gnome desktop is so easy, there's no excuse for not using it.**

PROJECT 3 Advanced window management

Gnome Shell has the potential to completely change the way you use your desktop, and in many ways, it's what virtual desktops should always have been. Fedora 12 and Ubuntu 9.10 users can open their package managers and install the **gnome-shell** package without any further difficulty. With the package installed, it needs to be started on the command line by typing **gnome-shell --replace**. This is because it uses its own window manager and closes whatever happens to be running. The *Gnome Shell* replacement is much more austere, but it still features its own eye candy. When it's running, just move your mouse up to the top-left corner of the screen. You see the current desktop zoom away into a small window within a black background.

This is the world of *Gnome Shell*. Recent documents and applications are listed in the left-hand side, while each virtual desktop is previewed on the right. You can add more virtual desktops by clicking on the large plus symbol, and browse your applications by clicking on the More button.

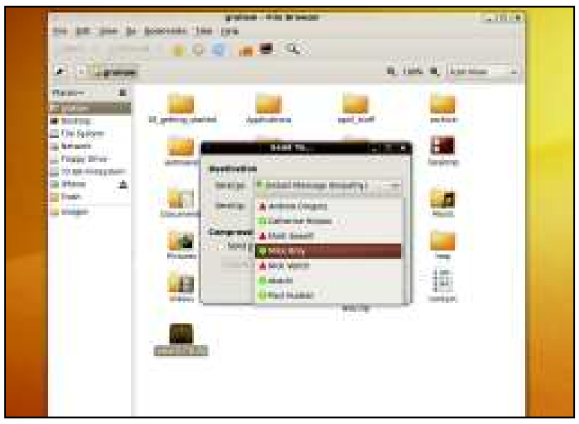
But the best thing is that all of this layout is interactive. Drag an app from the left panel on to a virtual desktop, for example, and it's launched automatically. Drag a file, and it will be opened automatically. Drag a folder, and its contents will be shown on the virtual desktop of your choice. From this *Gnome Shell* view you can move windows around or click on a desktop to zoom back into normal operation.


› **Click on your username to enable a quick-launch panel on the left border of the screen.**

PROJECT 4 Send files faster

Another neat feature of the latest Gnome desktops found on Ubuntu and Fedora is their ability to quickly send files to any of your contacts, either through email, instant messaging, a network drive or an optical disc. From the desktop, right-click on a file or folder and select the Send To option. For best results, you should be using the *Empathy* instant messenger application and *Evolution* for email, both of which are well-integrated into the Gnome desktop environment. If *Empathy* is running, for example, you can select Instant Message from the Send As drop-down menu, and the Send To list will be populated with contacts currently online. Your recipient will receive an incoming file message just as they would had you sent the file directly from your instant messenger client.

If you're trying to send a folder, this will be archived and sent as a Zip file by default. This is probably best for general users, but if you're sending your collection to another Linux user, you might want to choose **.tar.bz2** from the drop-down list, as these are generally smaller and better supported. The only problem is that Gnome performs no sanity check on what you're trying to send. This means you could inadvertently send a virus to a Windows user, for instance, or package too much data for the transmission protocol.


› **Modern messaging frameworks, such as *Empathy*, are becoming increasingly integrated into the desktop environment.**

PROJECT 5 Blog from home

Blogging things might have lost a little momentum in 2009, but it's still a great communication medium that can help you keep in touch with friends and family and any number of other people interested in your soldering hobbies and automatons. Home broadband connections are also fast enough to cope with this limited readership, which means it makes more sense to run your blog from a home server than it does one of the online commercial services. They mostly use *WordPress* anyway, which has become something of a standard in blog-based content management, and many people are using it for far more than simple communication.

If you've got an always-on Linux machine at home, installing *WordPress* is relatively easy. Ubuntu, OpenSUSE and Fedora include prebuilt packages, for instance, and after downloading the necessary files, you need only mess around with configuration briefly. Just grab **apache**, **mysql-server**, **wordpress** and **php-mysql** and let the package manager sort out the dependencies. When this has finished, Ubuntu users need to enter a password for the *MySQL* root user through the package manager's GUI, and you should be able to open a browser on the same machine and point it at **http://localhost** to make sure the *Apache* web server is running.

You now need to open up the command line and switch to your administrator's account, using either **su** on Fedora or **sudo bash** on Ubuntu. If you're not using Ubuntu, you may need to create a root password for the *MySQL* server. Just type the following, replacing **password**:

```
mysqladmin -u root password 'password'
```

The server can now be started by typing either **service mysqld start** (Fedora) or **service mysql** (Ubuntu) and we now need to create the database that *WordPress* will use for the blog's content. Type the following to connect to *MySQL* and create a database. Once again, swap **password** for something difficult:

```
mysql -u root -p
mysql> CREATE DATABASE wordpress;
mysql> GRANT ALL PRIVILEGES ON wordpress.* TO "wp_
user"@"localhost"
            -> IDENTIFIED BY "password";
mysql> FLUSH PRIVILEGES;
mysql> EXIT
```

Now that the database has been created, we're going to configure the *WordPress* installation. We first need to link the installation directory to your *Apache*'s root HTML location:

```
ln -s /usr/share/wordpress /var/www/html/blog
```

The last step is to open the **wp-config.php** file found within **/var/www/html/blog**' with your favourite text editor, and make sure the following values are set:

```
define('DB_NAME', 'wordpress_db');
define('DB_USER', 'wp_user');
define('DB_PASSWORD', 'password');
define('DB_HOST', 'localhost');
```

If your distribution is using SELinux security (as Fedora does), type **echo 0 >/selinux/enforce** to disable it for the current session. This is bad, but we don't have the space to explain SELinux configuration. Finally, point your local browser at **http://localhost/blog/wp-admin/install.php**. You should see the *WordPress* install screen. Enter a title for the blog and your email address, then click on the Install Blog button. You'll then see a page displaying a username of 'admin' and a randomly generated password. You need to remember these, but they should also be emailed to the address you entered in the previous step. Click on Log In' enter those details, and you're up and running. Don't forget to forward port 80 to your server from the internet.

› **You don't have to use *WordPress* on some impersonal online blogging service – just run it from home.**

PROJECT 6 Create a network wormhole

We all need to copy files to servers, whether it's music to our media streamer, or work documents to the backup folder. But connecting to these servers can sometimes be less than intuitive and often tedious. KDE 4.3 has the perfect solution. You can use the desktop folder feature to access a remote site, and conjure a constant portal from where you can grab files, drop new ones and see what's happening in almost real time.

But before you can enable this feature, you first need to create a saved bookmark to your remote site. The easiest way to do this is through the *Dolphin* file manager. Open the Places panel by pressing F7 and make the location editable by pressing Ctrl+l. You can now type the remote network address of your server. We highly recommend accessing your data using SSH, and KDE uses the Fish protocol to connect to SSH servers, even if they don't include 'sftp'. The format for the location URL is **fish://user@server/path/to/folder**, but you could swap **fish** for **ftp** or **smb** (*Samba*) for other protocols. When you press Enter, *Dolphin* will attempt to connect to the server and ask you for the password. You should then see your remote files in the file list. Now drag the location icon directly to the left of the location URL into the Places pane. This means you only need to click on this icon to connect to the server, but it also means that KDE file requesters and applications can also connect with a single click.

To open the portal on your desktop, click on the Plasmoid cashew on the top-right of the desktop, unlock it if you need to, then click on Add Widget. Drag the Folder View Plasmoid on to your desktop. Click on this, then on the spanner on the window's border. In the configuration window that appears, enable Show A Place and choose your server from the drop-down list. The Plasmoid will update to show the contents of the remote site, and you can interact with this destination exactly as you can a local folder.

› **Forget connecting to file servers – just create a window on the remote filesystem.**

PROJECT 7 Manage content with Drupal

Drupal is the software back-end used by many different websites to present and manage a variety of information. It's one of the most widely used pieces of software of its kind, and it's an incredibly powerful piece of open source programming. But the best thing about it its installation routine is very similar to *WordPress*, and uses many of the same packages. Just add **drupal** to the list of dependencies for *WordPress*, for example, and you've got everything you need.

Drupal is installed into **/usr/share/drupal**. and you need to make a symlink from this directory to where *Apache* holds its HTML.

```
ln -s /usr/share/drupal /var/www/html/drupal
```

You then need to open the *Apache Drupal* config file, **/etc/httpd/conf.d/drupal.conf**, and remove the **#** symbol from the line that contains **Allow from 127.0.0.1**. When *Drupal* is up and running, you'll have to come back to this configuration file and remove the **#** from the **Allow from all** line too, so that people can access your site from the internet. When you're making changes to *Apache* like this, you need to restart the web server, which can be done from Fedora by typing **service httpd restart** or from Ubuntu by typing **service apache2 restart**.

We now need to make some drastic changes to *Drupal*'s permissions to enable the installation routine to work. You can do this by typing the following commands as system administrator, and you should probably change them back to something more sensible after you've got *Drupal* working:

```
cd /etc/drupal
chmod 777 default
cp default/default.settings.php default/
settings.php.
chmod 666 default/settings.php default/
default.settings.php
chmod -R 777 /var/www/html/drupal
```

We now need to create a *MySQL* database for *Drupal* to store all its data, like this:

```
mysql -u root -p
mysql> CREATE DATABASE drupal;
mysql> GRANT ALL PRIVILEGES ON
drupal.* TO "dp_user"@"localhost"
mysql> IDENTIFIED BY "password";
mysql> FLUSH PRIVILEGES;
mysql> EXIT
```

Now you should be able to point a browser on the same machine at **http://localhost/drupal/install.php** to jump into the *Drupal* installation routine. You'll be asked for the name of the database you just created, along with the username and password to access it, and the configuration will be generated automatically. It's then a case of entering a name for your site and a contact email address, followed by an admin account. Save and continue, and your login details will be emailed to you. As a last step, go back to **/etc/drupal/** and type **chmod 755 default**, and inside the *Drupal* directory, type **chmod 644 settings.php default.settings.php**. You can log in and start playing with your fresh *Drupal* configuration. If you have SELinux trouble, see our *WordPress* project.

PROJECT 8 Get Conky

When you first look at the popular and rapidly evolving *Conky*, you might be forgiven for thinking it's yet another system monitor tool, the kind that quietly sits on your desktop telling you how hot your CPU is running and the weather outside your window. It is these things, but it's also so much more, capable of transforming your desktop from a panel-based launch menu into a streamlined experiment in minimalism.

Most distributions will bundle a **conky** package, and if you're lucky it will be added to your application menu for quick access. If not, open a command line and type conky or press Alt+F2 and enter **conky** as the command to execute. With no further configuration, you'll see a pretty standard resource manager appear on your desktop. The great thing about *Conky* is that you can change almost anything about the tool through its configuration file. But rather than create your own, it's much easier to grab a copy from someone who's already done all the hard work for you.

The internet is full of *Conky* configuration files, and many will include a screenshot to illustrate the results of all the author's hard work. You can find many examples on the Ubuntu forum, at **http://ubuntuforums.org/showthread.php?t=281865**, or on the *Conky* blog, **http://blog.conky.be** for example. Just grab the config file you're interested in, paste it into a text file and save. You may also need to install further packages, such as **lm-sensors** for CPU temperature information, for instance. You can then relaunch *Conky* by typing **conky -c**, followed by the name of the configuration file, and you'll see the new configuration appear on your desktop background. If you're happy with the results, add it to your desktop's startup routine.

› There's so much you can do with *Conky*. Here's a Christmas theme by BigRZ.

PROJECT 9 Free your music

We all have a music collection, and for most of us, that also means we have a library of MP3 files. But MP3 files aren't as free as Ogg Vorbis and Flac, and this can create problems when you switch to a distribution like Fedora that doesn't include MP3 playback by default. And switching between formats isn't easy either. You will lose a little quality, as there are very few tools that are up to the task of converting a library without humbling your system and your brain.

But there is one, and it's so good that you no longer need to suffer with MP3 incompatibility if you don't want to. It's the *Perl Audio Converter*, or *PACPL* as it's better known. But what most people don't realise is that after installing the *PACPL* package for your distribution, the tool integrates into your KDE desktop environment. This means you can right-click on a single folder, or even your entire collection, and choose Actions > PACPL Convert from the pop-up menu. You will then be presented with a colossal list of possible end formats, and when you select one, you can choose any further options such as quality and a destination folder for the converted files. It's then just a case of leaving PACPL alone. On KDE it will display a notification of each file it converts, and before long your music collection will be set free. LXF

› You don't have to stop the music just because it's the wrong format.