

## Conduit and Unison

# SYNCING TOOLS

Find out how the Conduit and Unison utilities can keep your documents and files in sync on multiple computers. **BY DMITRI POPOV**

**K**eeping your documents and files in sync between multiple computers is already a challenge, but other factors make this task even more difficult. Documents and files are stored not only on a desktop machine or laptop, they are often dispersed over different web-based services and applications such as Flickr, del.icio.us, and Gmail. Moreover, with the increasing popularity of portable applications and Live CD Linux distributions, you don't even need a computer – for many users, USB sticks and external hard disks have become the preferred devices for storing documents and files. All this makes a decent synchronization utility a must-have tool, and although rsync [1] remains one of

the most popular synchronization solutions for many users, a few other utilities can be equally or even more useful. In this article, I'll cover two utilities: Conduit and Unison.

### Conduit

Although Conduit [2] is still at an early stage of development, it has a good chance of becoming a killer utility for the Linux desktop for two simple reasons: Synchronization rules can be specified via drag and drop, and Conduit supports data syncing with online applications and services. Conduit is written in Python and PyGTK, and you can install the latest development version of it from source code. Installation instruc-

tions are available on Conduit's website [3]. If you don't want to deal with the source code, you'll be pleased to know that a *.deb* package is available at the GetDeb website [4].

Conduit's main window consists of two panes (Figure 1): The left pane contains a list of supported services (data providers), and the right pane (Canvas) is used to define synchronization rules. Even in its current form, Conduit supports quite a few data providers, including the ever-popular Tomboy, F-Spot, Box.net, Flickr, Gmail, and others. Conduit is capable of syncing individual files and folders. When working with different data providers, it's important to keep in mind that not all of them support two-way synchronization, and a tiny blue arrow next to each data provider icon helps you determine whether the syncing can go both ways or not.

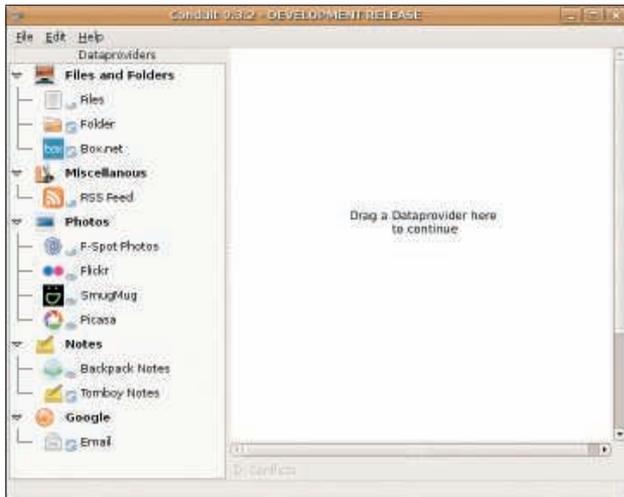


Figure 1: Conduit's dual-pane interface.

To see Conduit in action, I'll create a simple synchronization rule that syncs Tomboy notes to a folder on your USB stick. To do this, click on the folder item in the *Dataproviders* list and drag it onto the Canvas. Right-click on the added *Folder* item and choose *Configure Item*. Select the folder you want to keep your notes in and press *OK*.

Next, drag the *Tomboy Notes* item onto the created group and the rule is ready. The connector arrow between the two items indicates the synchronization direction, and you can switch between one-way and two-way syncing by right-clicking somewhere in the group and choosing the appropriate option. To see whether the rule works properly, right-click on the group and choose *Synchronize Group* (Tomboy must be running during the sync).

Conduit saves the notes as plain text files formatted with Tomboy's markup, and it has a few clever tricks in its arsenal. Because Conduit supports two-way syncing, all the text files you create in the specified folder are added to your *Tomboy Notes*. More importantly, you can sync your notes with Backpack [5], a popular web-based note-taking tool. In fact, you don't even have to create a separate synchronization rule for this. Instead, you can add a new item to the already-defined rule (see Figure 2), thus syncing your notes with the folder on your computer or a USB stick and Backpack at the same time.

To use Backpack with Conduit, you must provide your Backpack API key, which you can find in the *Account* section of Backpack (*Show the API key*).

Conduit provides a way to resolve synchronization conflicts, and you can configure its behavior in the *Edit | Preferences* dialog window under the *Configuration* tab (Figure 3).

Using the bundled data providers, you can do much more than keep your notes and documents in sync. For example, the RSS data provider supports file enclosures, which makes it

a perfect tool for downloading podcasts. Simply create a synchronization rule between an RSS feed and a local folder and you can use Conduit to download the latest podcasts.

Conduit can also come in handy for managing digital photos. If you use F-Spot, you can create a rule that syncs photos with your USB key as well as Picasa and Flickr accounts. The clever part is that Conduit allows you to sync photos on the basis of their tags. For example, you can choose to sync only photos tagged as Favorites or Places – or both.

As already mentioned, Conduit is not ready for prime time, and some features

can be a bit shaky or are not implemented. The most glaring omission at the moment is the lack of scheduling capabilities, which means that Conduit can't perform synchronization at pre-defined intervals. In my tests, Conduit proved to be pretty reliable, but you should back up everything before you take Conduit for a spin. Conduit's developer is also seeking help with the project, so if you want to contribute to a project, Conduit is worth considering.

## Unison

Conduit is for users who fancy bleeding-edge software and graphical interfaces. Other users might prefer Unison [6]. Most distributions provide Unison in their repositories, so installing it is not particularly difficult. On Ubuntu, for example, it is a matter of running the `sudo apt-get install unison-gtk` command. Although you can use Unison from the command line, it does feature a graphical front end (Figure 4).

Despite its bare-bones appearance, the front end provides access to most of Unison's features. When you run it in GUI mode for the first time, you'll be prompted to create a new synchronization profile. Simply specify the paths to the directories you want to keep in sync and you are done. Unison then checks both directories and displays all the files

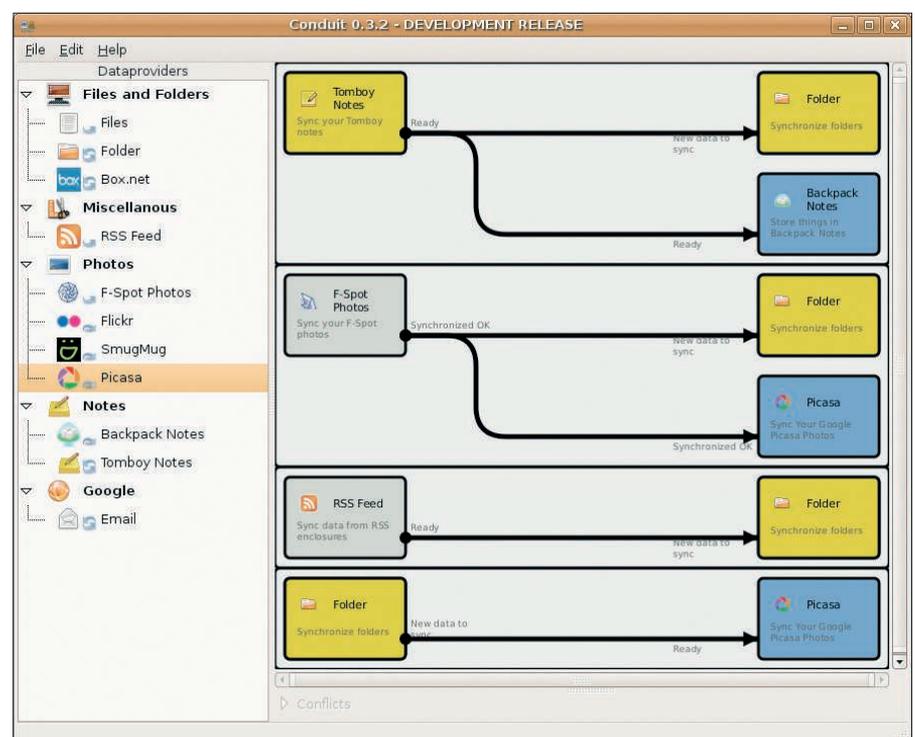


Figure 2: Using Conduit, you can create advanced rules.

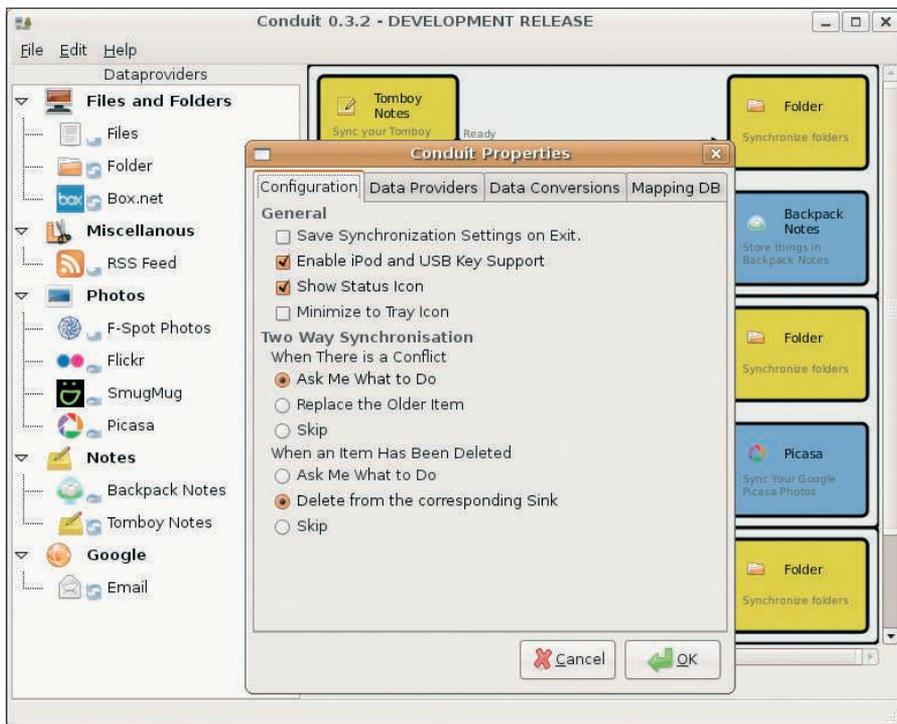


Figure 3: In the Preferences window, you can specify how Conduit should resolve synchronization conflicts.

to be synchronized. Pressing the Go button performs the sync.

Although the basic operation of Unison is pretty straightforward, it offers a few advanced features, including a powerful and flexible conflict-resolution mechanism that provides a comprehensive set of options.

Once Unison has scanned the directories, you can specify the way it should handle conflicts with the options available under the *Actions* menu. For example, you can choose to solve conflicts in favor of the desired directory by choosing either the *Resolve all conflicts in favor of first root* or *Resolve all conflicts in favor of second root* command. Also, you can force all changes from one directory to another and replace older files with newer ones.

Specifying all the synchronization settings manually each time you run Unison is hardly ideal. Fortunately, Unison provides an elegant solution – multiple profiles: For each profile, you can specify settings in a *.prf* file and save it in the *~/unison* directory. Say you want to keep your Firefox profiles in sync on two different machines. Create a new *firefox.prf* text file and start by specifying paths to the *.mozilla* directories on both the local and remote machines:

```
root = /home/user/.mozilla
root = ssh://remotehost/home/ .mozilla
```

As you can see, Unison supports synchronization via SSH, which is a handy feature for syncing remote machines.

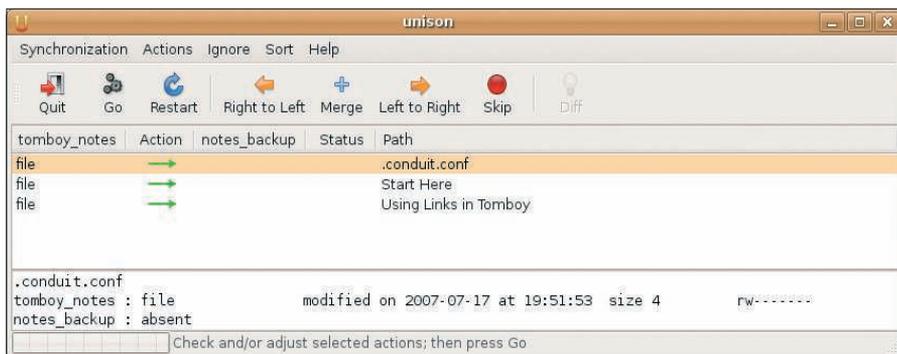


Figure 4: Unison's GUI.

Unison also supports a wide range of synchronization options, which you can specify in the profile file. For example, if you want the synchronization operation to run without user interaction, you can use the batch option *batch = true*. To force Unison to resolve conflicts by keeping the newest files is equally simple: *prefer = newer*. If you want Unison to preserve the files' modification times, add the *times = true* option. The final result will look something like this:

```
root = /home/user/.mozilla
root = ssh://remotehost/home/ .mozilla
batch=true
prefer=newer
times=true
```

If you place the file into the *~/unison* directory, you can then sync Firefox profiles using the *unison firefox* command.

This simple example is just a sample of what you can do with Unison, and to get the most out of this excellent synchronization tool, be sure to read the manual [7].

### Final Word

Unison is a great option for users who need a reliable and mature synchronization tool. Using options available in Unison, you can create rules that sync your data exactly the way you want. If you are looking for a utility that can help you easily sync data stored on different web services, Conduit is the ticket. Even in its current state, Conduit looks like a killer application for users who need to keep tabs on their data spread over different locations and services. ■

### INFO

- [1] Rsync: <http://samba.anu.edu.au/rsync/>
- [2] Conduit: <http://www.conduit-project.org/>
- [3] Conduit installation instructions: <http://www.conduit-project.org/wiki/Build>
- [4] GetDeb: <http://www.getdeb.net/app.php?name=Conduit>
- [5] Backpack: <http://backpackit.com/>
- [6] Unison: <http://www.cis.upenn.edu/~bcpierce/unison/>
- [7] Unison manual: <http://www.cis.upenn.edu/~bcpierce/unison/download/releases/stable/unison-manual.html>