## Dasher

# Easy Input

A team of researchers based at Cambridge University is working on developing Dasher [1], an information-efficient text-entry interface. Using natural pointing movements, the user zooms in on characters and thus can completely do without a keyboard. BY HEIKE JURZIK

Dasher is a so-called text-entry interface, that is controlled by natural and continuous pointing movements. The software is particularly useful wherever a "normal" keyboard cannot be used, for example on mobile devices. Dasher also aids text entry when one-handed operating is required, for example when using a joystick, touchscreen, trackball or mouse. In case of zero-handed operations, the software uses devices such as a **head-mouse** or an **eyetracker**. An experienced Dasher user can achieve speeds of up to 25 words per minute with the eyetracker version – that is approximately normal handwriting speed. Trained users can achieve one-handed speeds of up to 39 words per minute using a mouse.

Dasher is being developed by Cambridge University's Inference Group, led by David MacKay, who is a Reader in the Department of Physics. David created the first Dasher version way back in 1997. Since then, the software has been under continuous development and the subject of various research projects, being tested in a number of environments.

Dasher has been Open Source since 2002, and has gained the support of many developers from the Open Source community since then. The current 3.0.* version is available for most Windows systems, as source code and as RPM and Debian packages from the project's homepage.

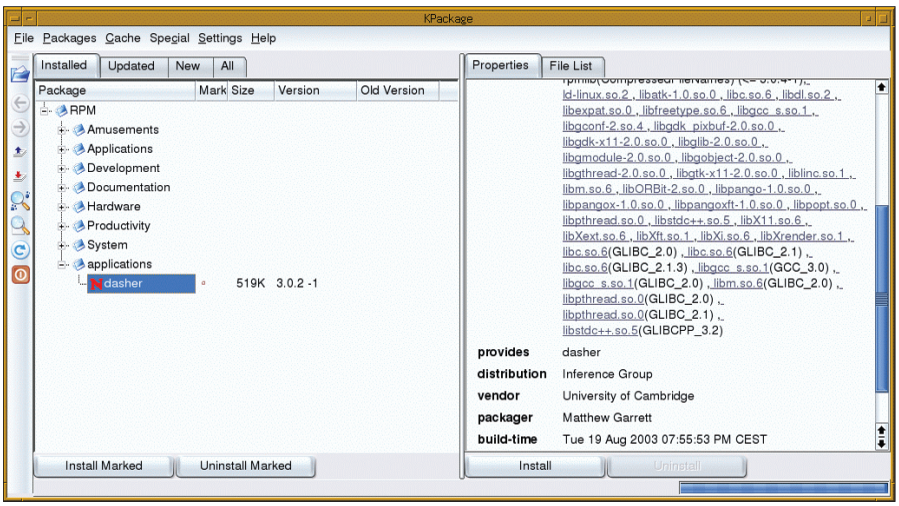


**Figure1: KPackage, showing info and dependencies**

The download page [2] also offers older Dasher versions with Japanese alphabet support, versions for various pocket PCs (Compaq iPAQ H3630, H3650 and H3850, HP Jornada 540, Symbol PPT 2846, Casio Cassiopeia E125 and Toshiba e740, amongst others), a beta release for MacOS X, and many languages are supported, including IPA (**I**nternational **P**honetic **A**lphabet).

## Installation

The subscription CD includes the source code (*dasher-3.0.2.tar.gz*), two RPMs for GTK1 systems (*dasher-3.0.2-1.i386.rpm*, *dasher-3.0.2-1.i586.rpm*), a GTK2 version (*dasher-gtk2-3.0.2-1.i586.rpm*), that can be used on any RPM-based system (with GTK Version 2 or later), a package for Debian "stable" (a.k.a. "woody") (*dasher_3.0.2-1_i386.deb*), and training

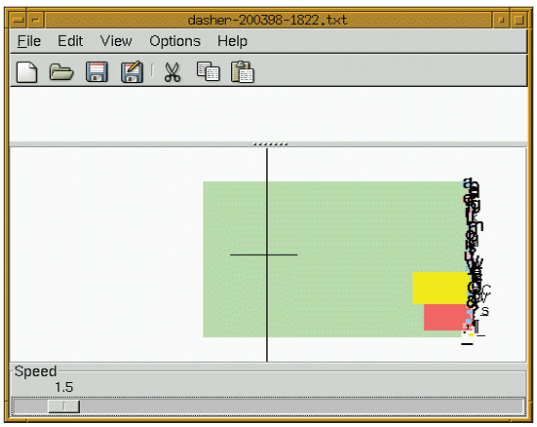


**Figure 2: Alphabet soup – Dasher on initial startup**

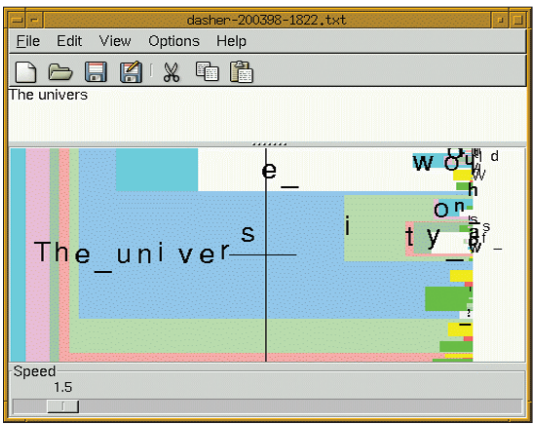


**Figure 3: Dasher in practice – likely words are easy to hit**

texts in various languages. As well as Debian "unstable" and "testing".

Debian users need to become *root* to install the package by typing *dpkg -i dasher_3.0.2-1_i386.deb*. You should make sure that the *libatk1.0-0*, *libc6 (>= 2.2.4-4)* or *libgtk2.0-0 (>= 2.0.2)* libraries are installed first. Type *dpkg --info dasher_3.0.2-1_i386.deb* to discover any dependencies (see Box 1).

Users with RPM-based systems should select the appropriate package from the subscription CD. If you prefer to test whether additional packages will be required, you can type *rpm -i --test package-name.rpm*. To install the package become *root* and type: *rpm -i package-name.rpm*. We've successfully tested the GTK2 RPM version on SuSE 8.2 and 8.1, Red Hat 9.0 and 8.0. If you prefer a GUI-based approach, programs such as YaST, KPackage (see Figure 1) or Gnome-RPM can be used to install the software.

If you wish Dasher to speak any language apart from English, you will need to copy the appropriate training text, as *root*, from the subscription CD to */usr/share/dasher/*.

## Getting Started

Type *dasher &* on the commandline to launch the program. There are two areas below the menus (see Figure 2). The lower one is for selecting the characters and the upper one is where the characters you select appear. If you want to write something in other than the default (English), you'll want to choose your required language before you start. To do so, select *Options/Alphabet…*, click on the entry for the language you need and then click on *Close* to confirm.

Now left-click the input area and drag the mouse to the right of the cross-hairs: letters will swarm towards you. Dasher zooms into the letter aimed at by the mouse – moving the mouse left slows you down or even allows you to go backwards. Click again to halt the program. The software works like a kind of alphabetically sorted bookcase. Imagine that books that start with "a" are top left, and "z" is bottom right. To find a book called "hello world" you first locate the "h". Book titles are arranged alphabetically below the letter, e.g. "hea, heb, hec…". Thus, you can browse the indices until you complete the title. Colors are used for easier orientation. In the current version, uppercase letters are in the yellow box, punctuation marks in the red box near the bottom, and the space character, which is represented by an underscore _, in a white box. This may sound complicated, but all will become clear as soon as you try it.

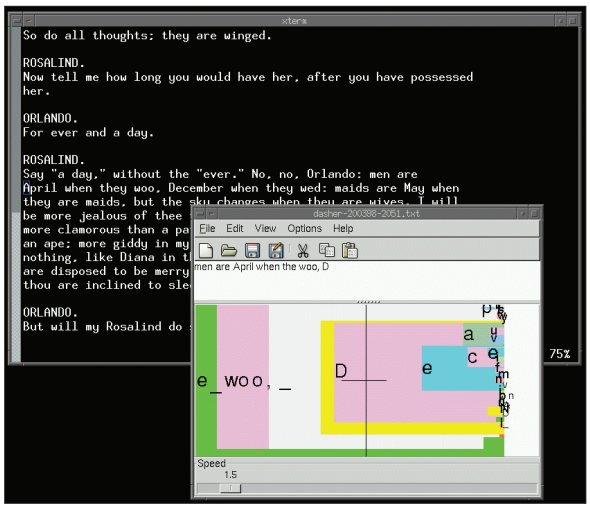


**Figure 4: Poetry in Motion**

The developers describe Dasher as "zooming in on an alphabetical library, steering as you go". The amount of "shelf-space" dedicated to a letter depends on the probability of that letter occurring. Thus, a lot less space is dedicated to the letter "x" than to "e", and letters that occur more often are easier to find. Dasher knows a lot of words from its dictionary files, and this makes it easier and quicker to find probable entries. Figure 3 shows the alternative paths that open up after zooming in on the letters "univers" – the letter "i" (followed by the letters "ty", which would lead to "universtity"), and the letter "e" followed by an underscore (representing a space character), which would lead to "universe".

While you are getting used to the program, it is advisable to select a slow speed by dragging the slide control at the bottom of the window to the left (1.0 is the slowest setting). The speed at which Dasher zooms in on at letters depends on how far to the right you move the mouse pointer.

## Under the Hood

When you launch Dasher for the first time, it creates configuration files in your own home directory. Settings, such as the last language selection, the zoom speed etc., are stored in *~/.gconf/apps/*

### GLOSSARY

**Head-mouse:** *This is a mouse replacement usable by people who cannot use their hands. It uses a reflection point on the forehead or on spectacles, and an infra-red sensor at the computer to follow head movements and thus simulate mouse movements.*

**Eyetracker:** *An eyetracker is an unobtrusive, remote human-computer interface that tracks where the user is looking, normally using a combination of laser sensors and cameras. This allows a user to control a computer by simply moving their eyes.*

### Box 1: Dasher for Debian

```
huhnix:~ # dpkg --info dasher_3.0.2-1_i386.deb
 new debian package, version 2.0.
 size 224702 bytes: control archive= 1467 bytes.
[...]
 Package: dasher
 Version: 3.0.2-1
 Section: x11
 Priority: optional
 Architecture: i386
 Depends: libatk1.0-0 (>= 1.0.1), libc6 (>= 2.2.4-4), libexpat1 (>=
1.95.2-6), libgconf2-4 (>= 1.1.9), libglib2.0-0 (>= 2.0.1), libgtk2.0-0
(>= 2.0.2), liblinc1 (>= 0.1.21), liborbit2 (>= 2.3.107), libpango1.0-0
(>= 1.0.1), libstdc++2.10-glibc2.2 (>= 1:2.95.4-0.010810), xlibs (>>
4.1.0)
 Installed-Size: 668
```

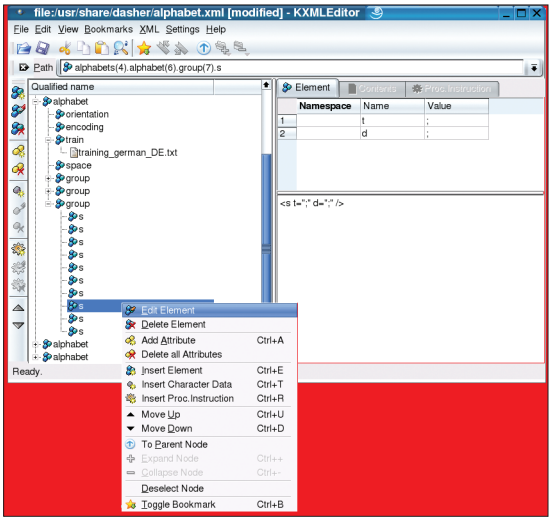


Figure 5: Modifying alphabet.xml


Figure 6: The dawn of Dasher

*dasher/%gconf.xml* so that they persist between sessions. Dasher "remembers" the new words that you use during a writing session, adding them to your personal training text in the hidden *~/.dasher* directory. This feature means that you can create your own training texts before launching into longer writing projects. Let's imagine you need to write an essay on Shakespeare's "As you like it"; it would be a good idea to create a new file, *~/.dasher/training_as_you_like_it.txt*. You could, for example, feed the file with quotes found online. Web sites with electronic books or texts, such as at Project Gutenberg [3], are a good source. Simply load the new training text (*File/Import Training Text...*) and start writing – Dasher already knows what you're talking about (see Figure 4). The Dasher homepage provides a detailed guide on creating training texts for new languages [4].

To properly support a new language, you not only need training texts, but an alphabet that defines the character set (**Unicode** as of Dasher version 3). This data is stored in an XML file called *alphabet.xml* in */usr/share/dasher/* and as distributed already contains support for the following languages :

English alphabet – limited punctuation, English alphabet with lots of punctuation, French, German, Polish, Polski and Portuguese (Brazilian)

Each alphabet starts with an *<alphabet>*-tag that is used to name the alphabet. This is followed by a series of mappings from what is displayed to what is output by Dasher. Thus, a newline character shows up as follows in the XML file:

```
<s d="Â¶" t="&#xa;"/>
```

The first argument (*d="Â¶"*) is the paragraph symbol, and this is what you see when you zoom into the characters; the second argument (*t="
"*) is the HTML escape sequence for **LF**. To add missing characters, you can edit this file as superuser root (see Figure 5).

## Linux Magazine at the labs

We took a trip to Cambridge, UK, to meet David MacKay and Matthew Garrett from the Dasher team to find out about Dasher's history and future plans. Project founder David MacKay developed the initial Dasher prototype after deciding keyboards were too big for handhelds. The first version was running on Windows and Linux, in C and **Tcl** (see Figure 6). David Ward continued work on Dasher between 1998 and 2002 as part of his PhD research, and performed a large number of experiments and turned the program into a working software system. An active group of

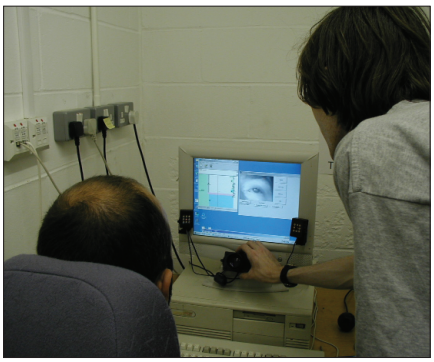

Figure 7: Hands-on adjusting the eyetracker

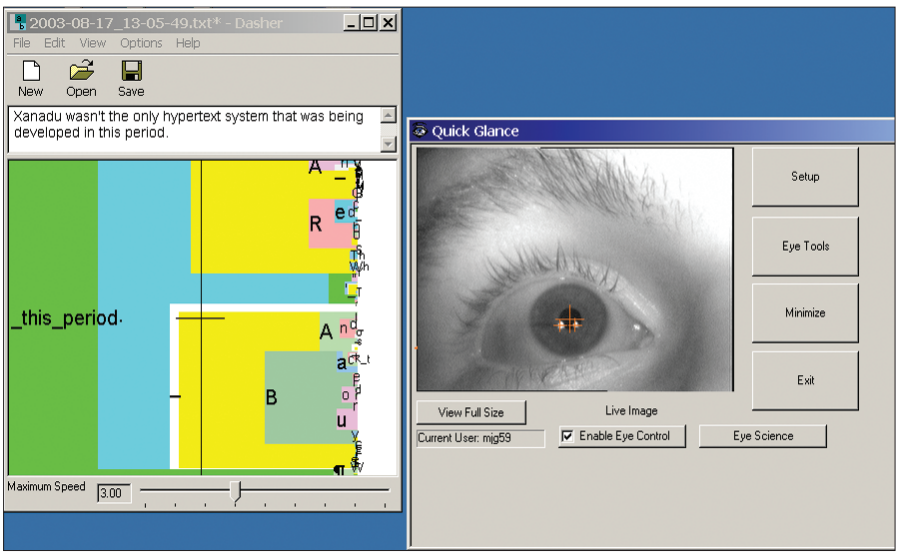


Figure 8: The eyetracker at work – just don't blink!

Open Source developers continues the development of Dasher today.

In Cambridge, we were able to experiment with the head-mouse and eyetracker in their lab. Although the eyetracker (Eyetech's *Quickglance Eyetracker* [7], which costs about 3000 US dollars) only supports the Windows platform, the head-mouse apparently has drivers for Linux. One thing that became apparent while we were attempting to train the eyetracker was that the settings and adjustments required help by another person (see Figure 7). People who wear spectacles are faced with another problem, as the frame will tend to reflect too strongly. Unfortunately, even extended calibration is no guarantee for useful results – just wink, or move your head slightly, and the focus of the eyetracker will shift. However, assuming you have accustomed yourself to the rigid posture and the camera, composition speed is quite acceptable (see Figure 8). Other eyetrackers use a second camera to track head movements. It's been reported to the Dasher team that a tracker by Metrovision [8] supports this feature.

The head-mouse proved far more flexible. The developers in Cambridge use a "Smart-Nav Headmouse" by NaturalPoint [9]. The reflection point is attached to a spectacle frame by means of a flexible cable (see Figure 9). The small camera is attached to a computer. The purchase price is a lot lower (approx. 300 US Dollars), and the head-mouse is more accurate. The main thing is to ensure that the user's head is supported to provide more stability. It is even conceivable that one could attach the reflection point to the user's foot to create a "toe-mouse".

**Figure 9: Quick and flexible – the head-mouse**

**Figure 10: David MacKay with the breath mouse**


**Figure 11: Dasher on the iPAQ**

The so-called "breath mouse" [10] is completely new and was developed in Cambridge. Using only an optical USB mouse, an elastic band and a belt, David MacKay created a system that users can control simply by breathing (see Figure 10). The stomach muscles move the mouse up and down. Dasher's one-dimensional mode (*Options/One Dimensional*) provides support for this sort of device.

## Future Directions

The Dasher team has bold ambitions regarding future developments. The Palmtop variant already works well (see Figure 11), and the current developer version of Dasher can interact with programs such as AbiWord, although you will need to check this Dasher variant out of CVS [11] and compile it yourself. So far, Dasher can only interact with GTK2-based programs, but work is in progress on a Qt-based version.

The developers are also looking to integrate Dasher with speech recognition software. Dasher could help users by allowing them to zoom into alternative selections for words not recognized correctly by their system. By combining the probabilities provided by the voice system and language modeling, Dasher could emphasize the most likely words, massively improving the usability of current systems. ■

## GLOSSARY

**Unicode**: *Unicode is a standardized character set defined by the International Organization for Standardization (ISO). Unicode is interesting because it tries to provide a unified encoding for all known characters. This not only includes the letters of the Latin alphabet, but also Greek, Cyrillic, Arabic, Hebrew, Chinese, Korean, and many other alphabets [5].*

**LF**: *LF is the ASCII Line Feed character. On Unix like systems such as Linux it's used as the "new line" character.*

**Tcl**: *The Tool Command Language [6] (pronounced "tickle") is a scripting language that while simple, provides all the major elements of programming languages. With the addition of Tk (Toolkit) (which provides Widgets, e.g. menus or buttons) this speeds up the process of programming software applications with graphical interfaces.*

## INFO


[1] Project homepage *http://www.inference.phy.cam.ac.uk/dasher/*

[2] Download: *http://www.inference.phy.cam.ac.uk/dasher/Download.html*

[3] Shakespeare's "As you like it" at Project Gutenberg: *http://www.ibiblio.org/gutenberg/etext98/2ws2510.txt*

[4] Creating Dasher training texts: *http://www.inference.phy.cam.ac.uk/dasher/Training.html*

[5] Unicode info: *http://unicode.e-workers.de/*

[6] Tcl homepage: *http://www.tcl.tk/*

[7] Eyetech's *Quickglance Eyetracker*: *http://www.eyetechds.com/*

[8] Metrovision homepage: *http://www.metrovision.fr/*

[9] NaturalPoint: *http://www.naturalpoint.com/*

[10] Breath Dasher: *http://www.inference.phy.cam.ac.uk/dasher/development/breath/*

[11] CVS version: *http://www.inference.phy.cam.ac.uk/dasher/Develop.html*