Bangla in GNU/Linux HOWTO

Taneem Ahmed

<taneem@_NOSPAM_bengalinux.org>

Santanu Chatterjee

<santanu@_NOSPAM_Softhome.net>

Progga

<abulfazl@_NOSPAM_juniv.edu>

Sayamindu Dasgupta

<unmadindu@_NOSPAM_bengalinux.org>

2003−04−25

Revision History

Revision 1.1.3	2003−10−23	Revised by: Pro
Some info on Qt and KDE 3.2's ability to render bn text is added at the very end.		
Revision 1.1.2	2003−04−29	Revised by: SDG
Added section on testing the translations. Initial LDP release.		
Revision 1.1.1	2003−04−20	Revised by: Pro
Initial LDP release, section added on Lekho		
Revision 1.1.0	2003−02−20	Revised by: SDG
Added Developers' Guide, made some significant changes in the document structure		
Revision 1.0.0	2003−02−14	Revised by: SDG
Initial Release		

The purpose of this document is to guide you setup and develop support for Bengali (Bangla) in GNU/Linux.

Table of Contents

1. Introduction

1.1. Preface

One of the foremost things on many GNU/Linuxers' wishlist is better support for their vernacular script on their favourite operating system. But Indic scripts are some of the most difficult in the world to add support for, because of inherent complexities like the existence of conjuncts (yuktakshars) and non−standard spellings. However, the Free/Libre/Open Source community is not one that is easily daunted, and this document tries to outline the process of setting up Bangla support in FLOSS applications. Moreover, it also aims to be the "developers' guide" for new developers in the Bangla in GNU/Linux project.

1.2. Copyright Information

1.3. Disclaimer

All copyrights are held by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Naming of particular products or brands should not be seen as endorsements.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals.

1.4. New Versions

The latest version of this HOWTO will always be made available at the www.Bengalinux.org site in a variety of formats.

- HTML .
- Plain text.
- PDF.
- XML.

1.5. Credits

We are extremely grateful to all the developers/users involved in the Bangla in GNU/Linux project, especially the people at Ankur and at the Free Bangla Fonts project .

The XML−stuff has been done with the help of the template written by Stein Gjoen, Gregory Leblanc and Greg Ferguson.

1.6. Feedback

If you have any comments, criticisms, ideas, additions, corrections, questions then please do mail them to `<users@bengalinux.org>` mailing list. We would be happy to help you.

1.7. Conventions used in this document

We have used a number of special formatting to indicate warning messages, commands, filenames, computer outputs etc. They are listed below.

Shell commands
    bash$ ls

Notes

    NOTE

Cautions

    CAUTION

Information

    INFO

Warnings

    WARNING

Filename/Directory
    `/usr/src/linux/`

Applications
    application

Computer Output
    `no such file or directory`

Codes/scripts

```
#!/bin/bash
```

Large Computer Outputs

```
logfile begins
```

1.8. Purpose of this HOWTO

This document explains how to setup and develop support for Bangla (or Bengali) in a GNU/Linux systems. Bangla support in GNU/Linux is by no means complete, but, in modern GNU/Linux distributions, you may be able to write in Bangla, send emails in Bangla, chat in Bangla, save with Bangla filenames (UTF−8....), have some of your graphical applications localized in Bangla, etc. Moreover, we have targeted to have the GNOME core translated into Bangla by August 2003, so that it can be included in the 2.4 release of GNOME. (Update− GNOME 2.4 is now out with partial support for Bangla)

However, note that this documents deals with Bangla support in GNU/Linux in GUI based applications only − we have no immediate plans for supporting Bangla on the console.

Also note that the Bangla support that we speak of here is totally Unicode compliant. We do not support any kind of proprietary/non−standard encoding.

2. Users' Guide

This part is mainly for the user – it does not go into the intricacies of development procedures. However, developers planning to work in this field must also read it – so that they can setup the required develoment environment.

2.1. Setting up the locale for Bengali

The first job that you have to do is to setup the locale for Bangla. The formal description of specific set of cultural habits for some country, together with all associated translations targeted to the same native language, is called the "locale" for this language or country. This sub–section describes how to setup the `bn_BD` locale definition file, which is for Bangladeshi Bengali. However, the Indian Bengali definition file, i.e. `bn_IN`, can be used in the same manner.

Remember that this sub–section is not intended to explain how to create a locale definition file, we will just explain how to *use* the locale defition file for Bangladeshi Bengali.

The `bn_BD` file should come with the latest glibc package. in case you have an older glibc, you can download this file from the download section of Bengalinux.org.

Make sure you have some Unicode encoded Bengali fonts installed on your system. If you don't have one, check out fonts section.

Now issue the following command as root:

bash$ **localedef –f UTF–8 –i bn_BD bn_BD.UTF–8**

This makes your system ready to be used with bn_BD locale definition.

Now make sure your X server is ready for Bengali locale. Check the following file
`/usr/X11R6/lib/X11/locale/locale.dir`

It should have the following lines:

```
    en_US.UTF-8/XLC_LOCALE        bn_BD.UTF-8
    en_US.UTF-8/XLC_LOCALE:       bn_BD.UTF-8
```

Now your system is ready to use bn_BD.UTF–8 locale. For example, I use the following command to open up an **xterm** with my own crappy font:

bash$ **LC_ALL=bn_BD.UTF-8 xterm**
-fn"-misc-Probhat-medium-r-normal--0-0-0-0-p-0-iso10646-1" -u8

Remember, if you want to use Bengali in any application, you will have to start that application under an environment where the locale was set to bn_BD.UTF–8, otherwise it won't work.

2.2. Installing the Bengali Fonts

2.2.1. Installing using the rpm package

If you are a rpm [read: RedHat %#$!@] user, then the easiest way to install the fonts from *FBFP* is by using the rpm package. The rpm package can be downloaded from http://www.Bengalinux.org/downloads/freebanglafont−0.2−1.noarch.rpm or http://savannah.nongnu.org/files/?group=freebangfont. Just install the rpm package as you would do for any other rpm package with the command

```
bash$  rpm −ivh freebanglafont−0.2−1.noarch.rpm
```

2.2.2. Installing using individual font file

If you cannot or do not want to use the rpm package, there is always the good old tgz packages. Remember that you should have root privilege when you execute the commands mentioned here.

Download the files you want to install from http://savannah.nongnu.org/files/?group=freebangfont.

Remember that you will find several different tgz packages in *FBFP*. Each of them contain different font faces, i.e. they all look different from one another.

Once you have downloaded all the packages you want, extract all the `*.ttf` file(s). Basically run the following command on each `*.tgz` or `*.tar.gz` you have downloaded.

```
bash$ tar −zxvf your_font_package.tar.gz
```

This will create a new directory that will contain different files that comes with the font package. We are only interested about the `*.ttf` files.

Create a new directory where you want to install these fonts. For the rest of this document, we will assume we are installing the fonts in `/usr/X11R6/lib/X11/fonts/bengali/`

Copy the `*.ttf` files we just extracted in to this directory, and execute the following commands:

```
bash$ cd /usr/X11R6/lib/X11/fonts/bengali
```

```
bash$ ttmkfdir
```

```
bash$ mkfontdir
```

```
bash$ chkfontpath −q −a /usr/X11R6/lib/X11/fonts/bengali
```

Now we will add the new directory to fontconfig configuration. Usually this configuration file is `/etc/fonts/fonts.conf`. Open this file with you choice of text editor and look for the part that looks somewhat like shown below

```
<dir>/usr/X11R6/lib/X11/fonts/Type1</dir>
<dir>/usr/share/fonts</dir>
<dir>~/.fonts</dir>
```

Add the line

```
<dir>/usr/X11R6/lib/X11/fonts/bengali</dir>
```

so that now it looks something like this:

```
 <dir>/usr/X11R6/lib/X11/fonts/Type1</dir>
 <dir>/usr/share/fonts</dir>
 <dir>/usr/X11R6/lib/X11/fonts/bengali</dir>
 <dir>~/.fonts</dir>
```

After you have saved the file, execute the following command:

bash$ **fc−cache**

If you have GNOME print library (libgnomeprint) installed, then execute the following commands:

bash$ **cd /usr/X11R6/lib/X11/fonts/bengali**

bash$ **libgnomeprint−2.0−font−install −−dynamic**

That should be it! If you have any problem, please let us know at <users@bengalinux.org>

2.3. Setting up the Bengali keyboard

The intention of this HOWTO is *not* to explain what is xkb symbol file, or how to create a new layout, etc. There are tons of places on the web where one can find a better and more in−depth HOWTO about XFree86 keyboard[layout/xkb/etc].

There is nothing special about setting up a Bengali Keyboard, just follow the steps described below. Remember everything you need in the following steps can be downloaded from the download section of Bengalinux.org, except the machine and the OS :−).

Make sure you have some Unicode encoded Bengali fonts installed on your system. If you don't have one, check out fonts section.

Download the xkb symbol file `ben` and copy it to the directory where the other xkb symbol files are. Usually this would be `/usr/X11R6/lib/X11/xkb/symbols`

BTW, usually only root can copy files into this directory.

You are now one command away from starting the keyboard. On my system I just type the following command:

bash$ **setxkbmap −symbols "us(pc101)+ben+group(ctrl_shift_toggle)"**

On your system you are most likely to use a different value for `−symbols`option. Let me explain what they mean, then you can decide what should be the value for your system. My default keyboard layout for X is `us(pc101)`. If you don't know what is the default you use, check the `XF86Config` file As the Bengali keyboard layout is defined in the second group we can simply add everything from `ben` to our keyboard layout. Now you have setup your keyboard with `us(pc101)` as group one and `ben` as your second group. What does that mean? Well in simple words: with X you can have different layouts in different groups and you can switch between these different layouts. And that is exactly why we added "group(ctrl_shift_toggle)". This means we can switch between the two groups by pressing **ctrl−shift** buttons together. Matter of fact there are other options for the toggle switch. If your XFree86 is new enough, you will find a file named `group` in the same directory you copied `ben`. If your system does not have this file do not worry. You can use an option like "grp:shift_toggle" which will toggle by pressing both shift buttons. In which case your command becomes:

```
bash$ setxkbmap −symbols "us(pc101)+ben" −option "grp:shift_toggle"
```

2.4. Some "special" Bengali only applications

2.4.1. Lekho

Lekho is a text editor for creating Bangla Unicode files without using any type of Unicode font.

This section must not be taken as "The Complete Reference of Lekho". At best it can be regarded as a short introduction to the editor. The main purpose of this section is to make you aware of the presence of this wonderful editor. In any case, the primary source of information on Lekho is it's website or the documentation provided with the package distribution.

You may find Lekho helpful if you are

1. Too lazy/busy to install the necessary Locale, Opentype fonts and Keyboard.
2. Currently using a Gnome version prior to 2.0 .
3. Don't have gtk+ in your system at all.
4. Want to export as Latex for use with Bangtex macro package.

Some of the major features of Lekho are as follows

- Keyboard Layout− Phonetic (i.e. k=ka, kh=kha etc)
- Minimum Requirements− Qt 2.3
- Font Usage− Adarshalipi
- Save As− UTF−8, UTF−16, ISO−8859−1
- Export As− HTML, Bangtex
- Development Status− Beta
- Support− Through Mailing list

1. Download the latest tarball from Lekho's website at http://lekho.sourceforge.net (Rumours are that an RPM is on the pipeline. By the time you read this HOWTO, it may be already out. So please check it if you prefer RPMs). Now untar the tarball:

   ```
   bash$ tar xfz Lekho.tar.gz
   ```
2. 'CD' to the Lekho source directory and then go through the traditional way of compilation+installation:

   ```
   bash$ ./configure
   ```

```
bash$ make

bash$ make install
```

3. At the very end of 'make install', the Adarshalipi fonts are copied to a seperate directory and the name of this directory is also printed. Now add that directory to your fontpath. There are several ways to do this depending on the XFree86 version you are using. Please consult Lekho's documentation for details.

Although Lekho is still in development stage, it has enough features to meet the requirements of most people IMHO.

---

3. Developers' Guide

This part of the HOWTO is mainly for developers – though the user who is interested in the internals can also go through it ;–). However, the developer should not start from here – he/she should at first read the preceding section to setup the system for Bengali support.

3.1. The general development process

The development center is the www.Bengalinux.org website. The model followed is the generic FLOSS type "bazaar" model (slightly localized version ;–), and you get all the goodies (CVS, shell access, scp, email lists, etc). If you are interested in developing, send a mail to `<core@bengalinux.org>` announcing your intentions.

A developer must be subscribed to the mailing list of the project he or she is work on – please see the Projects section of Bengalinux.org.

Usual netiquettes are applicable to all the lists – be polite/gentle, do not send HTML mails, do not shout, etc.

3.2. Pre–requisites

- Must know how to read and write proper Bangla (at least Grammar should be OK).
- Must have some sort of editor which can be used to edit Unicode encoded Bangla. (Proprietary editors are not advisable – no support will be provided.)
- (This one is optional) – must have basic idea about l10n (if possible, RTFM by going through the GNU gettext info page)
- Must be willing to follow the general FLOSS philosophy and procedures while developing for the project – and in general behave like a saint ;–).

3.3. Translators' Guide

3.3.1. Procedural Guide

Translation is pretty straight forward – you need to edit the `*.po` (Portable Object) files. To get the PO files – check out which files are untranslated from http://developer.gnome.org/projects/gtp/status/gnome–2.2/bn/index.html. Download any one of the untranslated files and open it in an editor that supports Bangla (sorry – vi/emacs won't work here). I personally use Yudit – and other possible options are Lekho and the good old Gedit2.

Before you start working on something, make sure that you announce the file you are working on at `<gnome–translations@bengalinux.org>`. This may save a lot of trouble and frustation ;–).

At the beginning of the file you will see some fields for filling in the translators' name, email, translation time, package name, etc. Fill them up, and proceed to the next part.

You will find some lines like

```
#: atk/atkobject.c:1183
msgid "invalid"
```

```
msgstr ""
```

The first line specifies the filename and the position in which the string in the second line occurs. The second line contains the actual string that is to be translated and the third line is the placeholder for the translated string in Bangla. Thus, in a translated file this set would look like

```
#: atk/atkobject.c:1183
msgid "invalid"
msgstr "bhul"
```

Here, I have written the word *bhul* in English due the limitations of present day technology. However, when you do the actual translation, you will need to write the translated strings in Bangla – and not English.

You will have to translate all the sets in a po file before it is considered to be complete. When it is complete, mail it to the translation list `<gnome-translation@bengalinux.org>`, so that the other developers can also go through it.

Before submitting the translation as final – please check the file with the command:

```
bash$ msgfmt -c filename_bn.po -o /dev/null
```

where `filename_bn.po` is the file you are working on.

3.3.2. Style Guide

While translating, remember not to *kill the language* ;–). Do not overdo the translation – do not use *Sadhu Basha* . Also, when the computer is addressing the user, it is *samman–shuchak (karun, balun, etc)* while the computer is addressed by *karo, balo, etc*. If a Bangla equivalent sounds too out of the ordinary – keep the English word (in Bangla script) – it can be always changed/improved later. *DO NOT* overdo anything.

Also, keep proper nouns and trademarks/brandnames as they are. (Bourne Again Shell (BASH) should not be called *Aabaar Bourne Shell*).

Use the more common word (*jaanlaa* and not *gabaakhya*). Also always remember that the words need not be very much applicable to the actual action – an Englishman from the nineteenth century could not even have guessed what "Save" can do in a computer ;–).

If you have comments regarding a certain translated string, please write that (commented out with a "#"), so that we can go through it later. If you have more than one equivalent string in your mind, write all of them, and comment all of them except the one you think is the best.

However, while commenting, remember that you *cannot* have any comments between the *msgid* and *msgstr* lines.

That means

```
#: atk/atkobject.c:1183
msgid "invalid"
# yadayadayada
msgstr "bhul"
```

is invalid. Instead you should write−

```
#: atk/atkobject.c:1183
msgid "invalid"
msgstr "bhul"
# yadayadayada
```

Also ...last but in no way the least, spellcheck, spellcheck, spellcheck. During translation session − always sit with your dictionary, or even better, the *Bangla Baanaan Abhidhan* from the Bangladesh Bangla Academy. The bspeller from Bengalinux may be useful here.

Another recommended book is the *Samartha−ShabdaKosh* (Bangla Thesaraus) by Ashok Mukherjee from Sahitya Samsad.

3.3.3. Testing the translations

One of the most tricky things for a translator is understanding the context. For example, by simply looking at a pofile, a translator will have no clue whether *Open* stands for *Kholo* or *Khola*. To do context sensitive translation, a translator has to check out his/her translations in action, that is, in the application interfaces. This section describes a hack to see the translations without having to recompile the program.

This process works for most systems with GDM as the login manager. However, I cannot guarantee that this will work on your system.

To begin with, setup the locale for Bangla and install some Bangla fonts.

Once that is done, get all the translated pofiles in a single folder in your machine (to avoid mess), and one by one convert them into MO files with the command:

bash$ **msgfmt −o file.mo file.version.gnomeversion_bn.po**

While doing this, remember to strip all the version and language information from the mo file's name, ie, `gnome−games.gnome−2−4.bn.po` should be converted to `gnome−games.mo`.

If you are unsure about a package, do a **rpm −ql packagename**, and you will get the naming scheme for the package. For example, a part of the output of **rpm −ql *gdm*** reads

```
/usr/share/locale/ru/LC_MESSAGES/gdm−2.4.mo
/usr/share/locale/sk/LC_MESSAGES/gdm−2.4.mo
/usr/share/locale/sl/LC_MESSAGES/gdm−2.4.mo
/usr/share/locale/sv/LC_MESSAGES/gdm−2.4.mo
/usr/share/locale/ta/LC_MESSAGES/gdm−2.4.mo
```

From this, it becomes apparent that one has to issue the command **msgfmt -o *gdm-2.4.mo gdm2.gnome-2-2.bn.po*** for the GDM package.

Once you are finished with that, copy the MO files to `/usr/share/locale/bn_IN.UTF-8` or `/usr/share/locale/bn_BD.UTF-8`, depending on how you set up your locale.

Now, open up `/etc/X11/gdm/locale.alias` and add the following lines to it.

```
        Bengali                  bn_IN.UTF-8,bn_IN
```

After this, restart X, select Bengali in the Languages menu and relogin into GNOME. You will be presented with a Bangla system.

For testing a single application, you don't have to go through all this stuff. Just setup the locale, install the font and put the MO file in the proper place. Now, fire up xterm, and issue the command **export LANG=bn_IN.UTF-8** or **export LANG=bn_BD.UTF-8**. After this, from xterm itself, start the program you want to check, and it will start up with a Bangla interface.

A. Appendix

This section contains some miscellaneous information.

A.1. Related Websites

- www.Bengalinux.org or www.AnkurBangla.Org: The mother site of all major projects related to Bangla in GNU/Linux.
- http://www.nongnu.org/freebangfont: The font wing of Ankur.
- www.unicode.org: The official site of the Unicode Consortium.

A.2. Volunteering for the Bangla in GNU/Linux project

If you want to help us in bringing Bangla support in GNU/Linux and in other Free/Libre/Open Source Software (FLOSS) applications, then please let us know at `<users@bengalinux.org>`. We are looking for mentors, developers, beta testers, translators, font designers to volunteer for our project. Even if you are not a conventional technically skilled person, you may help us by letting us know your comments on the work we are doing (especially the translation jobs.)

Also, if you are a font developer, you may consider (re)licensing your work under the GNU GPL so that we can reuse the fonts in our projects (of course, with due credit to you).

A.3. Software which has Bangla rendering support

Rendering of the Bangla script is a complex job, especially when one has to be within the Unicode limits. Hence, the technology behind Bangla rendering is considered to be quite "bleeding edge", and only the latest software support it. For example, Bangla text rendering support is there only in the latest Qt toolkit (Qt 3.2) which is used for KDE and Pango, which is the text rendering/layout engine of GTK 2. Pango 1.1.1 and above has a seperate module for Bangla. In addition to that, in the Free Software world, Yudit can render Bangla text correctly. However, the situation is improving slowly, and in the near future, we will probably see Bangla rendering in Mozilla, OpenOffice, etc. In fact, the beta version of OpenOffice (at the time of writing) has partial support for Bangla – however, it still has some serious problems with the split vowel signs. Starting from version 3.2.0, KDE applications (will) officially support Bangla rendering too.