

# Custom Debian Distributions - Making Debian the distribution of choice for specific work fields

Andreas Tille

DebConf 8, Mar del Plata, August 2008

## Abstract

The idea of Custom Debian Distributions was born at DebConf 3 in Oslo and has turned now into a solid tool set that can be used to organise packages targeting at a specific work field inside Debian in a quite efficient way. After five years it is time for a report about status and success as well as continuing to spread the idea amongst people to enable them to spend a minimum effort for the adoption of the tools to get a maximum effect in maintaining a CDD.

One goal of Custom Debian Distributions is to form a group of Debian developers who care for a specific set of packages that are used in the day to day work of a certain user group. The fact that Debian has grown to the largest pool of ready to install packages on the net has led to the side effect that it is hard to maintain for beginners. A Custom Debian Distribution adds some substructure to the currently flat pool of 15000+X packages without a user oriented structure. These substructures are intended to put a focus on special user interest. These substructures are not based on technical matters like Debian installer team, porting teams or teams that are focussing to implement programming language policies.

There are some similarities to Debian-i18n which also has the pure goal to serve the needs of certain end user groups with the difference that the users are grouped not according to their field of work but according to their language. In fact it makes even sense to create CDDs for languages that require certain technical means to optimally support the language regarding direction of writing, special fonts etc. It is known that some countries in Asia build Debian derivatives for this purpose but in principle it is not necessary to derive - the better solution is to make Debian more flexible by starting a CDD effort inside Debian.

The talks will give some examples from the success of CDDs like Debian Edu and Debian Med. One very important outcome of the CDD effort is the ongoing reunification of Linux - the Debian derived distribution that is used in all schools in Extremadura - with Debian Edu. This step means that Debian gets a very large implementation in all schools of Extremadura while on the other hand the effort of development for the people who invented Linux will be reduced. Debian featuring Debian Edu now has a very good chance to become a really good international school distribution because it has roots in five countries (Norway, Spain, France, Germany and Japan) and might become attractive for many more.

The success stories of CDDs would not have been possible outside Debian and thus leaving the path to build Zillions of Debian derivatives that reach a very small user base and working together inside Debian is the main idea of the talk. To

make this idea more attractive in the second part of the talk a description of tools that were developed in the CDD effort will be presented. Especially the newly developed web tools that give a good overview about the packages that are useful for a certain field of work and the QA tools that enable the CDD team members to easily get an overview about packages that need some action. So if people are not yet convinced that a CDD for their purpose makes sense we will catch them by the tools they might get for free if they follow the proposed strategy.

## 1 Symbiosis between experts and developers

As it was explained in last years CDD talk the basic goal of Custom Debian Distributions is to enable the user to focus on the packages that are really needed for his day to day work leading him friendly through the jungle of Debian's > 15000 packages. A user that is working in a certain field is only interested in a defined *subset* of packages and the CDD that is concerned about this field tries to prepare the computer optimally to install this subject with adapted configuration and easily accessible applications. So CDDs are taking care of groups of specialised users turning Debian into a useful tool adapted to their requirements for day to day work and making it to the distribution of choice for their use cases. It should enable and easy installation and automatically configuration whenever possible to make the needed work to fit the intended purpose as small as possible.

The tricky part in developing a CDD is now to tie a solid network of Debian developers, upstream developers ("developing experts") and users (experts in a defined field). It has turned out that gathering upstream developers into a CDD team is quite often not very hard. There are several upstream developers who try to become Debian Maintainer status - a concept which turned out to be quite successfully. The rationale behind this is if it comes to field specific software it is often written by experts in this field to solve the tasks of their daily work. Observations have shown that this software while showing great features regarding the task which should be solved there are often weak parts in the build system or in the general handling of using libraries or wide spread tools. This is exactly the point where Debian developers have good experiences and are able to provide technical help.

So it happens quite often that upstream developers of field specific applications are quite happy if Debian developers want to build Debian packages of the software because they anticipate enhancements of their build system and a security audit. Last but not least they expect a wide distribution of their work to reach a large user base easily.

## 2 Attracting people by providing interesting technical base

The acceptance of new methods is drastically higher if the techniques provided are convincing enough and provide interesting features for the target audience. Considering this we tried to develop simple ways to categorise packages that are useful for certain

tasks. This is done in so called tasks files which are processed using the `cdd-dev` package to build metapackages. The other application of these tasks files is building internationalised web pages which display the packages that are relevant for a certain CDD task with the descriptions of the packages. The translation for the descriptions are drawn from the [Debian Description Translation Project](#) and the more complete the DDTP translation of packages that are relevant for a CDD are the better is the translation of the web pages featuring the CDD tasks. Thus by adding another use case of DDTP translations the effort might become additional participants and the quality of translations - especially those of specific packages which need some expert knowledge for proper translation - might increase.

The internationalised web pages which are generated automatically out of the information inside the tasks files is a key documentation feature which is a really helpful tool for developers of the CDD as well as very informative for users because they immediately get an overview about all ready to install software that might be helpful for their day to day work. Thus we try to promote these pages as the main entry point for information for our users what we have done, which work is in progress and what's on our TODO list. This information might give them a good motivation to join the project. The first step might be to provide better translations for the package descriptions which is certainly a task which is better done by experts using the package themselves instead by Joey Randomtranslator who tries his best in a word by word translation but if he does not know the real usage of the package it is hard to provide a really useful translation.

Once we were able to rise users interest they might be interested to do the next step to install and try the packages in question. This is the point where the upstream developer of the software becomes a new user which might report bugs or give hints for enhancements and might become a coworker finally. This way Debian, or more specifically the CDD that supports the specific field, has helped to increase the user base of a software and thus the potential developer base.

The process to establish a certain piece of Free Software described above seems to be quite straightforward but without a linking instance like a CDD in between upstream developer and users the propagation of specialised Free Software is often everything else than straightforward. If the idea of Free Software reaches a specialist who is working on a specific solution he is happy to release the code on his private web page – and that it is just there. It is not very common to use well known source code repositories like [Savannah](#) or [SourceForge](#) or even implementing a version control system to promote group development. In contrary if an other specialist is seeking for a solution for the same problem he has to invent extended Google queries to find the project in question – if he has the idea to seeking and before he is starting simply from scratch.

Some volunteers have realised this situation and provide extensive link lists either on static HTML pages or Wikis to enable others to join the catalogue effort. The problem here is that these link lists are often incomplete and what matters even more are not directly connected to immediately installable and executable binaries.

There are some similar efforts like CDD in other distributions for instance there is a comparable effort to package biological Free Software done by [Gentoo](#) or [FreeBSD](#) because also other distributors realised the problem described above. The difference between such kind of installable software collections and a CDD is that a Custom Debian Distribution tries to do more than just packaging specific software. It is rather

about forming a team of maintainers who try to build a consistent system around several tasks in a specific field, care for easily installation using metapackages, making sure that everything works together smoothly and working actively as missing link between upstream developers and users.

### 3 CDD is more than packaging specific software

The main work of a distributor is providing precompiled binaries of Free Software, caring for smooth installation and upgrades as well as security fixes for the distributed packages. In the case of Debian which maintains the largest pool of ready to install binary software packages this is quite a large amount of work. The huge amount of software includes larger subset of software for very specific use and this is the playground of CDD packaging teams who closely work together to bundle their competence on packages with a specific user base.

It turns out that there is a good chance of cooperation between CDDs on a technical level because several jobs to do are at least similar. This idea is the basis of the whole CDD effort: Making sure that the wheel that drives a certain CDD is only invented once and adopted for all others. There is a similar situation in the internationalisation teams: There is a well defined group of users (speakers of a certain language) who need special support (translations at various places) and it make sense if language teams just work together and use common tools like DDTP server and others.

While this translation work is one part of the internationalisation team it can not stop here. It is also about proving that Debian is flexible enough to incorporate this kind of changes instead of forcing users to make language based forks of Debian. Unfortunately there are many people out there who feature a wrong concept of using Free Software. They read the license as they are allowed to modify the software as modifying their local copy and tweak it until it fits their needs. They treat this constant forking as the normal way to customise their distribution. The internationalisation team has done a good job in propagating the idea that it is better to include translations into Debian than adding translations over and over for every new release of Debian.

In principle the maintainers of a CDD have the same job: Attracting derivivers who have not yet understood the power of internal customisation inside a CDD to reach their goal – optimal support of their users – more efficiently. The most convincing example how a CDD managed to merge a derivative back to Debian is that the educational branch of [LinEx](#) is working on unification with Debian Edu since end of year 2007.

### 4 Techniques

The techniques used are based on sorting certain packages of the Debian pool into certain remits or in the terminology we have chosen in CDDs “tasks”. So the tasks files are listing dependencies from Debian packages and different tools are using these as common source of information.

## 4.1 Building metapackages and taskset descriptions

The `cdd-dev` can be used to build a set of metapackages and taskset descriptions. Because the technique is described in very detail in the [article about CDD](#) there is no need to repeat the content here. I would rather ask you to follow the link and especially read section *6.1 Metapackages*.

The build process using `cdd-dev` also included the creation of a `CDD-tasks` package which contains information for taskset to enable the tasks of a CDD via taskset.

## 4.2 Web pages based on common scripts

A quite new feature are the web pages that are build based on the very same information that is used to build the metapackages. This enables users to get a very quick overview because they see all packages included in the task together with the description. Because the target audience does not necessarily is comfortable with English language the descriptions are even translated in case there is such a translation provided by the [Debian Description Translation Project](#). Such pages are available for the following projects:

- [Debian Edu](#)
- [Debian GIS](#)
- [Debian Junior](#)
- [Debian Med](#)
- [Debian Science](#)

In addition to the packages existing inside Debian there is an easy way to specify prospective packages that should be included into Debian in the future. These packages are listed on the web pages as well. To read more about this feature just have a look into the [article about CDD](#) especially section *8.1 Existing and prospective packages*.

There is no need to copy the information of the just existing article about CDD which is continuously updated and thus this document ends here with the strong recommendation to read the technical details there.