

Custom Debian Distributions

Adapting Debian GNU/Linux to special fields

Andreas Tille

Asia Debian Mini-Conf

Beijing, March 1, 2005

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 Techniques
 - Meta-packages
 - User menus
 - Communication
- 3 Prospectus

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 Techniques
 - Meta-packages
 - User menus
 - Communication
- 3 Prospectus

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 Techniques
 - Meta-packages
 - User menus
 - Communication
- 3 Prospectus

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 Techniques
 - Meta-packages
 - User menus
 - Communication
- 3 Prospectus

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
- Company that assembles a distribution = *distributor*
- Business model: selling distribution, support & training
- Examples: Novell, RedHat, Ubuntu, ...
- Debian is just one of them.

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
- Company that assembles a distribution = *distributor*
- Business model: selling distribution, support & training
- Examples: Novell, RedHat, Ubuntu, ...
- Debian is just one of them.

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
 - Company that assembles a distribution = *distributor*
 - Business model: selling distribution, support & training
 - Examples: Novell, RedHat, Ubuntu, ...
 - Debian is just one of them.

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
- Company that assembles a distribution = *distributor*
 - Business model: selling distribution, support & training
 - Examples: Novell, RedHat, Ubuntu, ...
 - Debian is just one of them.

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
- Company that assembles a distribution = *distributor*
- Business model: selling distribution, support & training
- Examples: Novell, RedHat, Ubuntu, ...
- Debian is just one of them.

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
- Company that assembles a distribution = *distributor*
- Business model: selling distribution, support & training
- Examples: Novell, RedHat, Ubuntu, ...
- Debian is just one of them.

- Linux = *core* of operating system
- Needed: *user-specific applications*
- Assemblage of applications around Linux kernel = *distribution*
- Company that assembles a distribution = *distributor*
- Business model: selling distribution, support & training
- Examples: Novell, RedHat, Ubuntu, ...
- Debian is just one of them.

Sorry.

Sorry.

That was wrong.

The Debian Project is an association of individuals who have made common cause to create a free operating system. This operating system that we have created is called

Debian GNU/Linux,

or simply Debian for short.

Moreover, work is in progress to provide Debian for other than Linux kernels, primarily for the Hurd. Other possible kernels are BSD and OpenSolaris.

Custom
Debian
Distributions

Andreas Tille

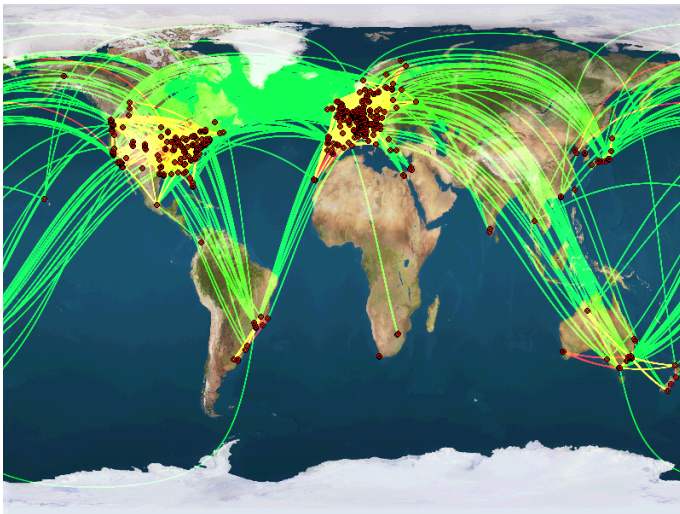
Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus



Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages

User menus

Communication

Prospectus

Commercial distributor

Debian



Commercial distributor

Debian

Company

Structure

Organisation

Commercial distributor

Debian

Company

Structure

Organisation

Employees

Persons

Volunteers

Commercial distributor

Debian

Company

Structure

Organisation

Employees

Persons

Volunteers

CDs, Service

Sells

Nothing

Commercial distributor

Debian

Company

Structure

Organisation

Employees

Persons

Volunteers

CDs, Service

Sells

Nothing

Business plan

Release

If 0 RC-bugs

Commercial distributor

Debian

Company

Structure

Organisation

Employees

Persons

Volunteers

CDs, Service

Sells

Nothing

Business plan

Release

If 0 RC-bugs

Certified

Oracle, SAP, etc.

Runs in principle

| Commercial distributor | | Debian |
|------------------------|-------------------|-------------------|
| Company | Structure | Organisation |
| Employees | Persons | Volunteers |
| CDs, Service | Sells | Nothing |
| Business plan | Release | If 0 RC-bugs |
| Certified | Oracle, SAP, etc. | Runs in principle |
| Beginners | Preferred by | Administrators |

| Commercial distributor | | Debian |
|------------------------|-------------------|-------------------|
| Company | Structure | Organisation |
| Employees | Persons | Volunteers |
| CDs, Service | Sells | Nothing |
| Business plan | Release | If 0 RC-bugs |
| Certified | Oracle, SAP, etc. | Runs in principle |
| Beginners | Preferred by | Administrators |
| Rpm | Packages | Deb |

| Commercial distributor | | Debian |
|------------------------|-------------------|-------------------|
| Company | Structure | Organisation |
| Employees | Persons | Volunteers |
| CDs, Service | Sells | Nothing |
| Business plan | Release | If 0 RC-bugs |
| Certified | Oracle, SAP, etc. | Runs in principle |
| Beginners | Preferred by | Administrators |
| Rpm | Packages | Deb |
| Market | Customisation | Do-O-Cracy |

Custom
Debian
Distributions

[Andreas Tille](#)

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- **Common goal: best possible operating system**
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - By buying from any other distributor of CD
 - By using the Internet
- Network installation very comfortable

Custom
Debian
Distributions

[Andreas Tille](#)

Debian for
every purpose

What is Debian?
CDD

Techniques
Meta-packages
User menus
Communication

Prospectus

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - Downloading from the Debian FTP site
 - Using the network
- Network installation very comfortable

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
- Network installation very comfortable

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - Buying from any *other distributor* on CD
- Network installation very comfortable

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - Buying from any other distributor on CD
 - Usually installation via internet
- Network installation very comfortable

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - Buying from any other distributor on CD
 - Usually installation via internet
 - Network installation very comfortable

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - Buying from any other distributor on CD
 - Usually installation via internet
- Network installation very comfortable

- Common goal: best possible operating system
- Maintainer: Missing link between developers and users
- Largest collection of ready-to-install Free Software
- Principles of Free Software at distribution level:
Independence of commercial interest of producer
- Two ways to obtain Debian GNU/Linux
 - Buying from any other distributor on CD
 - Usually installation via internet
- Network installation very comfortable

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 Techniques
 - Meta-packages
 - User menus
 - Communication
- 3 Prospectus

- Debian > 10000 packages
- General users interested in a *subset* of those packages
- Care for special groups of target users with different skills and interests
- Easy installation and configuration by grouping specific packages
- **No fork** from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

- Debian > 10000 packages
- General users interested in a *subset* of those packages
- Care for special groups of target users with different skills and interests
- Easy installation and configuration by grouping specific packages
- No fork from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

- Debian > 10000 packages
- General users interested in a *subset* of those packages
- Care for special groups of target users with different skills and interests
- Easy installation and configuration by grouping specific packages
- **No fork** from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

- Debian > 10000 packages
- General users interested in a *subset* of those packages
- Care for special groups of target users with different skills and interests
- Easy installation and configuration by grouping specific packages
- **No fork** from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

- Debian > 10000 packages
- General users interested in a *subset* of those packages
- Care for special groups of target users with different skills and interests
- Easy installation and configuration by grouping specific packages
- **No fork** from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

- Debian > 10000 packages
- General users interested in a *subset* of those packages
- Care for special groups of target users with different skills and interests
- Easy installation and configuration by grouping specific packages
- **No fork** from Debian

Basic idea: Do not make a separate distribution but make Debian fit for special purpose instead

1 Support of target users with common profile:

- Less technical competence
- Unable to install upstream programs with acceptable effort
- No interest in administration
- *Specific subset* of available Free Software
- Need for easy usage
- Defined security profile
- Language barrier

2 Support of administrators with common profile:

- Limited time frame
- Lack of official release cycle
- Lack of specialized knowledge

1 Support of target users with common profile:

- Less technical competence
- Unable to install upstream programs with acceptable effort
- No interest in administration
- *Specific subset* of available Free Software
- Need for easy usage
- Defined security profile
- Language barrier

2 Support of administrators with common profile:

- Limited time budget
- No complete or partial system backup
- Lack of specialized knowledge

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time budget
 - No experience with package management
 - Lack of specialized knowledge

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset of available Free Software*
 - *Need for easy usage*
 - *Defined security profile*
 - *Language barrier*
- ② Support of administrators with common profile:
 - *Small number of users*
 - *Not motivated to contribute upstream*
 - *Lack of specialized knowledge*

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time resources
 - Limited knowledge of system
 - Lack of specialized hardware

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time frame

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time frame
 - Automation of often repeated tasks
 - Lack of specialist knowledge

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time frame
 - Automation of often repeated tasks
 - Lack of specialist knowledge

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time frame
 - Automation of often repeated tasks
 - Lack of specialist knowledge

- ① Support of target users with common profile:
 - Less technical competence
 - Unable to install upstream programs with acceptable effort
 - No interest in administration
 - *Specific subset* of available Free Software
 - Need for easy usage
 - Defined security profile
 - Language barrier
- ② Support of administrators with common profile:
 - Limited time frame
 - Automation of often repeated tasks
 - Lack of specialist knowledge

- [Debian-Junior](#) - Debian for children from 1 to 99
- [Debian-Med](#) - Debian in medicine
- [Debian-Edu \(SkoleLinux\)](#) - Debian in school
- [DeMuDi](#) - Debian Multimedia Distribution
- [Debian-Desktop](#) - Debian for everybody
- [Debian-Lex](#) - Debian for lawyers
- [Debian-NP](#) - Nonprofit Organisations
- [Debian-Enterprise](#) - Enterprise Computing
- [Debian-Accessibility](#) - Blinds and visually impaireds
- [Debian-eGov](#) - E-Government

- *Debian-Junior* - Debian for children from 1 to 99
- *Debian-Med* - Debian in medicine
- *Debian-Edu (SkoleLinux)* - Debian in school
- *DeMuDi* - Debian Multimedia Distribution
- *Debian-Desktop* - Debian for everybody
- *Debian-Lex* - Debian for lawyers
- *Debian-NP* - Nonprofit Organisations
- *Debian-Enterprise* - Enterprise Computing
- *Debian-Accessibility* - Blinds and visually impaireds
- *Debian-eGov* - E-Government

- *Debian-Junior* - Debian for children from 1 to 99
- *Debian-Med* - Debian in medicine
- *Debian-Edu (SkoleLinux)* - Debian in school
- *DeMuDi* - Debian Multimedia Distribution
- *Debian-Desktop* - Debian for everybody
- *Debian-Lex* - Debian for lawyers
- *Debian-NP* - Nonprofit Organisations
- *Debian-Enterprise* - Enterprise Computing
- *Debian-Accessibility* - Blinds and visually impaireds
- *Debian-eGov* - E-Government

- *Debian-Junior* - Debian for children from 1 to 99
- *Debian-Med* - Debian in medicine
- *Debian-Edu (SkoleLinux)* - Debian in school
- *DeMuDi* - Debian Multimedia Distribution
- *Debian-Desktop* - Debian for everybody
- *Debian-Lex* - Debian for lawyers
- *Debian-NP* - Nonprofit Organisations
- *Debian-Enterprise* - Enterprise Computing
- *Debian-Accessibility* - Blinds and visually impaireds
- *Debian-eGov* - E-Government

- Debian-Junior - Debian for children from 1 to 99
- Debian-Med - Debian in medicine
- Debian-Edu (SkoleLinux) - Debian in school
- DeMuDi - Debian Multimedia Distribution
- Debian-Desktop - Debian for everybody
- Debian-Lex - Debian for lawyers
- Debian-NP - Nonprofit Organisations
- Debian-Enterprise - Enterprise Computing
- Debian-Accessibility - Blinds and visually impaireds
- Debian-eGov - E-Government

- [Debian-Junior](#) - Debian for children from 1 to 99
- [Debian-Med](#) - Debian in medicine
- [Debian-Edu \(SkoleLinux\)](#) - Debian in school
- [DeMuDi](#) - Debian Multimedia Distribution
- [Debian-Desktop](#) - Debian for everybody
- [Debian-Lex](#) - Debian for lawyers
- [Debian-NP](#) - Nonprofit Organisations
- [Debian-Enterprise](#) - Enterprise Computing
- [Debian-Accessibility](#) - Blinds and visually impaireds
- [Debian-eGov](#) - E-Government

- *Debian-Junior* - Debian for children from 1 to 99
- *Debian-Med* - Debian in medicine
- *Debian-Edu (SkoleLinux)* - Debian in school
- *DeMuDi* - Debian Multimedia Distribution
- *Debian-Desktop* - Debian for everybody
- *Debian-Lex* - Debian for lawyers
- *Debian-NP* - Nonprofit Organisations
- *Debian-Enterprise* - Enterprise Computing
- *Debian-Accessibility* - Blinds and visually impaireds
- *Debian-eGov* - E-Government

- Debian-Junior - Debian for children from 1 to 99
- Debian-Med - Debian in medicine
- Debian-Edu (SkoleLinux) - Debian in school
- DeMuDi - Debian Multimedia Distribution
- Debian-Desktop - Debian for everybody
- Debian-Lex - Debian for lawyers
- Debian-NP - Nonprofit Organisations
- Debian-Enterprise - Enterprise Computing
- Debian-Accessibility - Blinds and visually impaireds
- Debian-eGov - E-Government

- Debian-Junior - Debian for children from 1 to 99
- Debian-Med - Debian in medicine
- Debian-Edu (SkoleLinux) - Debian in school
- DeMuDi - Debian Multimedia Distribution
- Debian-Desktop - Debian for everybody
- Debian-Lex - Debian for lawyers
- Debian-NP - Nonprofit Organisations
- Debian-Enterprise - Enterprise Computing
- Debian-Accessibility - Blinds and visually impaireds
- Debian-eGov - E-Government

- [Debian-Junior](#) - Debian for children from 1 to 99
- [Debian-Med](#) - Debian in medicine
- [Debian-Edu \(SkoleLinux\)](#) - Debian in school
- [DeMuDi](#) - Debian Multimedia Distribution
- [Debian-Desktop](#) - Debian for everybody
- [Debian-Lex](#) - Debian for lawyers
- [Debian-NP](#) - Nonprofit Organisations
- [Debian-Enterprise](#) - Enterprise Computing
- [Debian-Accessibility](#) - Blinds and visually impaireds
- [Debian-eGov](#) - E-Government

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field of medicine etc.
- In contrast to employees of companies every single Debian developer has the freedom and ability to realize his vision
- Every developer is able to influence the development of Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- Developed by about 1000 volunteers
- Flexible, not bound on commercial interest
- Strict rules (policy) glue all things together
- Common interest of each individual developer:
Get the best operating system for himself.
- Developers have children in real life or work in the field
of medicine etc.
- In contrast to employees of companies every single
Debian developer has the freedom and ability to realize
his vision
- Every developer is able to influence the development of
Debian - he just has to *do* it.

Do-O-Cracy = "The doer decides"

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 **Techniques**
 - **Meta-packages**
 - User menus
 - Communication
- 3 Prospectus

- Main part of the work of a distributor

- Contains

- Executable programs

- Configuration files

- Scripts (e.g. `install`, `postinst`, `prerm`, `preinst`, `rm`, `unpack`, `unpack-alternative`)

- Instructions to other packages

- Files that are part of archives of files

- Enables easy handling

- Easy installation of all dependencies

- Clean upgrades

- Easy handling of package removal

- Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages

→ More than just an archive of files
- Enables easy handling
 - Easy installation of all dependencies
 - Clean upgrades
 - Easy handling of package archives
 - Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
- More than just an archive of files
- Enables easy handling
 - Easy installation of all dependencies
 - Easy upgrades
 - Easy handling of package changes
 - Basis of modern Linux-like distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
- More than just an archive of files
- Enables easy handling
 - Easy installation of all dependencies
 - Easy removal of packages
 - Easy handling of package versions
 - Basis of modern Linux/Unix distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` **scripts**
 - Relations to other packages
- More than just an archive of files
- Enables easy handling

→ *Example: `dpkg-query -f='${Package} ${Version} ${Architecture}\n'`*

→ *Example: `dpkg-query -f='${Package} ${Version} ${Architecture}\n'`*

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
- More than just an archive of files
- Enables easy handling

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
- More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages

→ More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies
 - Clean upgrades
 - Easy handling of security updates

→ Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
 → More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies
 - Clean upgrades
 - Easy handling of security updates
 → Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
 → More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies
 - Clean upgrades
 - Easy handling of security updates
 → Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
 → More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies
 - Clean upgrades
 - Easy handling of security updates
 → Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
 → More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies
 - Clean upgrades
 - Easy handling of security updates
 → Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Main part of the work of a distributor
- Contains
 - Executable programs
 - configuration
 - `{pre/post}install + {pre/post}remove` scripts
 - Relations to other packages
 - More than just an archive of files
- Enables easy handling
 - Ensures compliance of all dependencies
 - Clean upgrades
 - Easy handling of security updates
 - Basis of modern GNU/Linux distributions

Problem: Increasing number of packages

- Special packages with following content:

- 1 Relations on other Debian packages (essential)

- "Depends" for necessary packages
- "Recommends" for recommended packages
- "Provides" for metapackages or non-DEB packages

- 2 Menu entries (recommended)

- Customization of menu entries provided by `xdg-menus`
- (Optional: `xdg-user-dirs` & `xdg-desktop-portal`)

- 3 Adapted configuration (optional)

- `debconf` / `debconf-templates` / `debconf-templates`
- `debconf` / `debconf-templates` / `debconf-templates`

- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:
 - ➊ Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ➋ Menu entries (recommended)
 - Customization of default menu for particular package
 - Customization of /usr/local
 - ➌ Adapted configuration (optional)
 - Customized options or pre-installing
 - Customized scripts
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:

- 1 Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
- 2 Menu entries (recommended)
- 3 User menu (optional)
- 4 Desktop environment (optional)
- 5 Adapted configuration (optional)

- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:

- 1 Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
- 2 Menu entries (recommended)
- 3 User menu (optional)
- 4 Adapted configuration (optional)

- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:

- 1 Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages

- 2 Menu entries (recommended)

- Customisation of default entries possible

- 3 Adapted configuration (optional)

- Customisation of default configuration possible

- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:

- 1 Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
- 2 Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
- 3 Adapted configuration (optional)

- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- *Special packages* with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:
 - 1 Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - 2 Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - 3 Adapted configuration (optional)
 - `debconf` questions or pre-seeding
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using `cdd-dev` package

- Special packages with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
 - *debconf* questions or pre-seeding
 - *cfengine* scripts
- Special meta package: *<cdd>-common*
- Meta-packages → kind of task packages
- Easily built by using *cdd-dev* package

- Special packages with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
 - *debconf* questions or pre-seeding
 - *cfengine* scripts
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using *cdd-dev* package

- Special packages with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
 - *debconf* questions or pre-seeding
 - *cfengine* scripts
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using *cdd-dev* package

- Special packages with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
 - *debconf* questions or pre-seeding
 - *cfengine* scripts
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using *cdd-dev* package

- Special packages with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
 - *debconf* questions or pre-seeding
 - *cfengine* scripts
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using *cdd-dev* package

- Special packages with following content:
 - ① Relations on other Debian packages (essential)
 - "Depends" for necessary packages
 - "Recommends" for interesting packages
 - "Suggests" for not so important or non-free packages
 - ② Menu entries (recommended)
 - Customisation of default entries possible
 - Documentation entry if useful
 - ③ Adapted configuration (optional)
 - *debconf* questions or pre-seeding
 - *cfengine* scripts
- Special meta package: `<cdd>-common`
- Meta-packages → kind of task packages
- Easily built by using *cdd-dev* package

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1
Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1
Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1
Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name
Description: short description
             long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name

Description: short description
             long description as in any debian/control file

Depends: dependency1, dependency2, ...

Suggests: suggests1, suggests2, ... (optional)

$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name

Description: short description
             long description as in any debian/control file

Depends: dependency1, dependency2, ...

Suggests: suggests1, suggests2, ... (optional)

$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name
Description: short description
             long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1

Task: task name
Description: short description
             long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1
Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see /usr/share/doc/cdd-dev/examples/tasks

Control file

```
$ cp -a /usr/share/doc/cdd-dev/examples/tasks .
$ cat tasks/README
$ edit tasks/task1
Task: task name
Description: short description
               long description as in any debian/control file
Depends: dependency1, dependency2, ...
Suggests: suggests1, suggests2, ... (optional)
$ cp tasks/task1 tasks/<meta-package-name>
```

*For each meta package this skeleton of a
debian/control entry is needed*

see */usr/share/doc/cdd-dev/examples/tasks*

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
    change the variables of the example
$ edit debian/rules
    change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

```
$ cd ..
$ mv debian ..
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
    change the variables of the example
$ edit debian/rules
    change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .  
$ cat debian/README  
$ edit debian/control.stub  
change the variables of the example  
$ edit debian/rules  
change the variable _CDD_ to your CDD  
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .  
$ cat debian/README  
$ edit debian/control.stub  
  change the variables of the example  
$ edit debian/rules  
  change the variable _CDD_ to your CDD  
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Directory debian

```
$ cp -a /usr/share/doc/cdd-dev/examples/debian .
$ cat debian/README
$ edit debian/control.stub
  change the variables of the example
$ edit debian/rules
  change the variable _CDD_ to your CDD
$ make -f debian/rules dist
```

Source - tar.gz is ready

Build package

```
$ debuild
```

In principle you are ready.

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .  
$ cat common/README  
$ edit common/{conf,control,common.1}  
change variable _CDD_
```

Initialisation of role based menu system

- Meta-package will depend from common packages
- will support 1 user menu
- strongly suggest to use

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Install support packages
- Strongly suggest to use

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggest to use *auto-apt*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggested

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
  change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggested

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
  change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggested

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
  change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggested

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
  change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggested

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/common .
$ cat common/README
$ edit common/{conf,control,common.1}
  change variable _CDD_
```

Initialisation of role based menu system

- Meta packages will depend from common package
- Installs *auto-apt* helper
- Strongly suggested

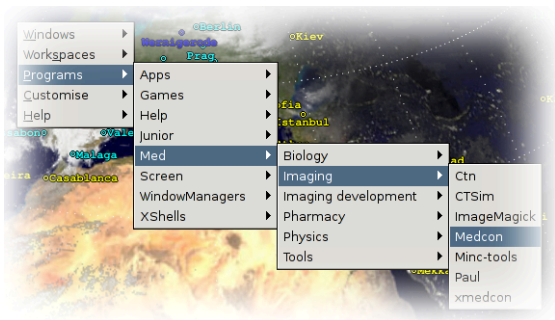
- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions
- 2 Techniques
 - Meta-packages
 - **User menus**
 - Communication
- 3 Prospectus

- User with role `<cdd>` gets extra user menu



- User with role `<cdd>` is provided with additional menu
- Do not bother other users with those additional menus
- Implemented in current Debian-Med packages
- Role system is implemented by flexible plugins

- User with role `<cdd>` gets extra user menu



- User with role `<cdd>` is provided with additional menu
- Do not bother other users with those additional menus
- Implemented in current Debian-Med packages
- Role system is implemented by flexible plugins

- User with role `<cdd>` gets extra user menu



- User with role `<cdd>` is provided with additional menu
- Do not bother other users with those additional menus
- Implemented in current Debian-Med packages
- Role system is implemented by flexible plugins

- User with role `<cdd>` gets extra user menu



- User with role `<cdd>` is provided with additional menu
- Do not bother other users with those additional menus
- Implemented in current Debian-Med packages
- Role system is implemented by flexible plugins

- User with role `<cdd>` gets extra user menu



- User with role `<cdd>` is provided with additional menu
- Do not bother other users with those additional menus
- Implemented in current Debian-Med packages
- Role system is implemented by flexible plugins

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

```
$ cp menu/task1/menu/dep1.menu /usr/share/doc/cdd-dev
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

Optional

```
$ cp -a /usr/share/doc/cdd-dev/examples/menu .
$ cat menu/README
$ cat menu/task1/README
$ edit menu/task1/dep1.menu
$ edit menu/task1/dep2.txt
$ edit menu/task1/dep3.html
```

Create valid menu entries with sensible descriptions

Find task names

```
$ cp menu/task1 menu/<meta package name>
```

*User menu entries for every dependencies - at least
pointing to documentation*

- 1 Debian for every purpose
 - What is Debian?
 - Custom Debian Distributions

- 2 Techniques
 - Meta-packages
 - User menus
 - **Communication**

- 3 Prospectus

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- *Project at Alioth*
- *Subversion Repository*
- *Common developer mailing list*
Use this list often for CDD issues!
- User mailing lists for most CDDs
<http://lists.debian.org/debian-<cdd>>
- Web pages for most CDDs
<http://www.debian.org/devel/debian-<cdd>>

- *Project at Alioth*
- *Subversion Repository*
- *Common developer mailing list*
Use this list often for CDD issues!
- User mailing lists for most CDDs
<http://lists.debian.org/debian-<cdd>>
- Web pages for most CDDs
<http://www.debian.org/devel/debian-<cdd>>

- *Project at Alioth*
- *Subversion Repository*
- *Common developer mailing list*
Use this list often for CDD issues!
- User mailing lists for most CDDs
<http://lists.debian.org/debian-<cdd>>
- Web pages for most CDDs
<http://www.debian.org/devel/debian-<cdd>>

- *Project at Alioth*
- *Subversion Repository*
- *Common developer mailing list*
Use this list often for CDD issues!
- User mailing lists for most CDDs
`http://lists.debian.org/debian-<cdd>`
- Web pages for most CDDs
`http://www.debian.org/devel/debian-<cdd>`

- *Project at Alioth*
- *Subversion Repository*
- *Common developer mailing list*
Use this list often for CDD issues!
- User mailing lists for most CDDs
`http://lists.debian.org/debian-<cdd>`
- Web pages for most CDDs
`http://www.debian.org/devel/debian-<cdd>`

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible → but CDD makes more sense
- Fork: requires adaptations continuously
- CDD: Work has to be done *only once*
- Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible → but CDD makes more sense
- Fork: requires adaptations continuously
- CDD: Work has to be done *only once*
- Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
 - Basic problem: reinventing the wheel frequently
 - Debian-Fork possible → but CDD makes more sense
 - Fork: requires adaptations continuously
 - CDD: Work has to be done *only once*
 - Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
 - Debian-Fork possible → but CDD makes more sense
 - Fork: requires adaptations continuously
 - CDD: Work has to be done *only once*
 - Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible → but CDD makes more sense
 - Fork: requires adaptations continuously
 - CDD: Work has to be done *only once*
 - Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible → but CDD makes more sense
- Fork: requires adaptations continuously
- CDD: Work has to be done *only once*
- Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible → but CDD makes more sense
- Fork: requires adaptations continuously
- CDD: Work has to be done *only once*
- Debian = Give and Take

- Probably most popular CDD: Debian-Edu (SkoleLinux)
- Back integration: DeMuDi
- How to start a Custom Debian Distribution
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible → but CDD makes more sense
- Fork: requires adaptations continuously
- CDD: Work has to be done *only once*
- Debian = Give and Take

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- Customisation regarding languages
- Dependencies of certain font packages
- Adapted input / output methods
- Special configuration

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- Customisation regarding languages
- Dependencies of certain font packages
- Adapted input / output methods
- Special configuration

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- Customisation regarding languages
- Dependencies of certain font packages
- Adapted input / output methods
- Special configuration

Custom
Debian
Distributions

[Andreas Tille](#)

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- Customisation regarding languages
- Dependencies of certain font packages
- Adapted input / output methods
- Special configuration

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- More Custom Debian Distributions
 - Further development of CDD tools
 - Spreading the idea to developers and users
 - *Customised live CDs of CDDs*
 - *New way to distribute Debian*
 - Extra *Packages* file for each CDD

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- More Custom Debian Distributions
- Further development of CDD tools
- Spreading the idea to developers and users
- *Customised live CDs of CDDs*
- *New way to distribute Debian*
- Extra *Packages* file for each CDD

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- More Custom Debian Distributions
- Further development of CDD tools
- Spreading the idea to developers and users
- *Customised live CDs of CDDs*
- *New way to distribute Debian*
- Extra *Packages* file for each CDD

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages

User menus

Communication

Prospectus

- More Custom Debian Distributions
- Further development of CDD tools
- Spreading the idea to developers and users
- *Customised live CDs of CDDs*
- *New way to distribute Debian*
- Extra *Packages* file for each CDD

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages

User menus

Communication

Prospectus

- More Custom Debian Distributions
- Further development of CDD tools
- Spreading the idea to developers and users
- Customised live CDs of CDDs
- New way to distribute Debian
- Extra *Packages* file for each CDD

Custom
Debian
Distributions

Andreas Tille

Debian for
every purpose

What is Debian?
CDD

Techniques

Meta-packages
User menus
Communication

Prospectus

- More Custom Debian Distributions
- Further development of CDD tools
- Spreading the idea to developers and users
- *Customised live CDs of CDDs*
- *New way to distribute Debian*
- Extra *Packages* file for each CDD

This talk can be found at

<http://people.debian.org/~tille/talks/>

Andreas Tille <tille@debian.org>