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.
Debian for beginners

- 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.
Debian for beginners

- 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](https://www.novell.com), [RedHat](https://www.redhat.com), [Ubuntu](https://www.ubuntu.com), …
- [Debian](https://www.debian.org) 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.
Debian for beginners

- 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.
Differences to commercial distributions

Commercial distributor

Debian
Differences to commercial distributions

- Commercial distributor
- Company
- Structure
- Organisation

Debian
Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td></td>
<td>Organisation</td>
</tr>
<tr>
<td></td>
<td>Volunteers</td>
</tr>
</tbody>
</table>
# Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td>CDs, Service</td>
<td>Sells</td>
</tr>
</tbody>
</table>

- **Company**: Structure
- **Employees**: Persons
- **CDs, Service**: Sells

- **Debian**:
  - Organisation
  - Volunteers
  - Nothing
Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td>CDs, Service</td>
<td>Sells</td>
</tr>
<tr>
<td>Business plan</td>
<td>Release</td>
</tr>
</tbody>
</table>
## Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td>CDs, Service</td>
<td>Sells</td>
</tr>
<tr>
<td>Business plan</td>
<td>Release</td>
</tr>
<tr>
<td>Certified</td>
<td>Oracle, SAP, etc.</td>
</tr>
<tr>
<td></td>
<td>Runs in principle</td>
</tr>
</tbody>
</table>

- Company: Structure
- Employees: Persons
- CDs, Service: Sells
- Business plan: Release
- Certified: Oracle, SAP, etc.

- Volunteers
- Organisation
- Nothing
- If 0 RC-bugs

Beginners Preferred by Administrators
Rpm Packages Deb
Market Customisation Do-O-Cracy
### Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td>CDs, Service</td>
<td>Sells</td>
</tr>
<tr>
<td>Business plan</td>
<td>Release</td>
</tr>
<tr>
<td>Certified</td>
<td>Oracle, SAP, etc.</td>
</tr>
<tr>
<td>Beginners</td>
<td>Preferred by</td>
</tr>
</tbody>
</table>
## Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td>CDs, Service</td>
<td>Sells</td>
</tr>
<tr>
<td>Business plan</td>
<td>Release</td>
</tr>
<tr>
<td>Certified</td>
<td>Oracle, SAP, etc.</td>
</tr>
<tr>
<td>Beginners</td>
<td>Preferred by</td>
</tr>
<tr>
<td>Rpm</td>
<td>Packages</td>
</tr>
</tbody>
</table>
# Differences to commercial distributions

<table>
<thead>
<tr>
<th>Commercial distributor</th>
<th>Debian</th>
</tr>
</thead>
<tbody>
<tr>
<td>Company</td>
<td>Structure</td>
</tr>
<tr>
<td>Employees</td>
<td>Persons</td>
</tr>
<tr>
<td>CDs, Service</td>
<td>Sells</td>
</tr>
<tr>
<td>Business plan</td>
<td>Release</td>
</tr>
<tr>
<td>Certified</td>
<td>Oracle, SAP, etc.</td>
</tr>
<tr>
<td>Beginners</td>
<td>Preferred by</td>
</tr>
<tr>
<td>Rpm</td>
<td>Packages</td>
</tr>
<tr>
<td>Market</td>
<td>Customisation</td>
</tr>
</tbody>
</table>
Particular features

- **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
**Particular features**

- **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**
Particular features

- 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
Particular features

- 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
Particular features

- 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
Particular features

- 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
Particular features

- 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
**Particular features**

- 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*
Customising Debian

- 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*
Customising Debian

- 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*
Customising Debian

- 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*
Customising Debian

- 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
Customising Debian

- 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*
Motivation

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
Motivation

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
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

Support of administrators with common profile:
- Limited time frame
- Automation of often repeated tasks
- Lack of specialist knowledge
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

Support of administrators with common profile:
- Limited time frame
- Automation of often repeated tasks
- Lack of specialist knowledge
Motivation

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
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

Support of administrators with common profile:

- Limited time frame
- Automation of often repeated tasks
- Lack of specialist knowledge
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

Support of administrators with common profile:
- Limited time frame
- Automation of often repeated tasks
- Lack of specialist knowledge
Motivation

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
Motivation

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
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

Support of administrators with common profile:
- Limited time frame
  - Automation of often repeated tasks
  - Lack of specialist knowledge
Motivation

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
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

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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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
Existing Custom Debian Distributions

- **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 - adaptable for any purpose?

- 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"
Debian - adaptable for any purpose?

- 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"_
Debian - adaptable for any purpose?

- 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"*
Debian - adaptable for any purpose?

- 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"*
Debian - adaptable for any purpose?

- 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"_
Debian - adaptable for any purpose?

- 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"
Debian - adaptable for any purpose?

- 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"*
Debian - adaptable for any purpose?

- 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"*
Debian for every purpose

- What is Debian?
- Custom Debian Distributions

Techniques

- Meta-packages
- User menus
- Communication

Prospectus
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
What are 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
What are 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
What are 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
What are 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
What are 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
What are 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
What are packages

- Main part of the work of a distributor
- Contains
  - Executable programs
  - configuration
  - \{pre/post\}install + \{pre/post\}remove \texttt{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

\textit{Problem: Increasing number of packages}
What are 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
What are 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
What are 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
What are 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
Meta-Packages

**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
Meta-Packages

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
Meta-Packages

**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
**Meta-Packages**

- **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
Meta-Packages

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
Meta-Packages

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
Meta-Packages

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
Meta-Packages

- **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
Meta-Packages

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
Meta-Packages

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

```bash
$ 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
Short introduction to build meta-packages

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
Short introduction to build meta-packages

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
Short introduction to build meta-packages

Control file

```bash
$ 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

```bash
$ 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
Short introduction to build meta-packages

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
Short introduction to build meta-packages

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
Short introduction to build meta-packages

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
Short introduction to build meta-packages

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
## Short introduction to build meta-packages

### Control file

```bash
$ 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
<table>
<thead>
<tr>
<th>Directory <code>debian</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>$ cp -a /usr/share/doc/cdd-dev/examples/debian .</td>
</tr>
<tr>
<td>$ cat debian/README</td>
</tr>
<tr>
<td>$ edit debian/control.stub</td>
</tr>
<tr>
<td>change the variables of the example</td>
</tr>
<tr>
<td>$ edit debian/rules</td>
</tr>
<tr>
<td>change the variable <code>_CDD_</code> to your CDD</td>
</tr>
<tr>
<td>$ make -f debian/rules dist</td>
</tr>
</tbody>
</table>

Source - `tar.gz` is ready

Build package

$ debuild

*In principle you are ready.*
Short introduction to build meta-packages (2)

Directory `debian`

```bash
$ 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

```bash
$ 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.
Short introduction to build meta-packages (2)

Directory `debian`

```sh
$ 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.*
Short introduction to build meta-packages (2)

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.
Short introduction to build meta-packages (2)

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.
Short introduction to build meta-packages (2)

Directory `debian`

```bash
$ 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

```bash
$ 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.
Short introduction to build meta-packages (2)

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.
$ 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
$ 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
$ 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**

```bash
$ 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

```bash
$ 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
Debian for every purpose

- What is Debian?
- Custom Debian Distributions

Techniques

- Meta-packages
- User menus
- Communication

Prospectus
User with role \(<\texttt{cdd}\)> gets extra user menu

- User with role \(<\texttt{cdd}\)> 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
Role based user menus

- 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
Role based user menus

- 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
Role based user menus

- 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
Role based user menus

- 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
Changing user menus

Optional

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

Create valid menu entries with sensible descriptions

User menu entries for every dependencies - at least pointing to documentation
Changing user menus

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
Changing user menus

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
Changing user menus

Optional

```bash
$ 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

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

User menu entries for every dependencies - at least pointing to documentation
Changing user menus

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
Changing user menus

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\ \text{package\ \name}\>

User menu entries for every dependencies - at least pointing to documentation
Changing user menus

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/\text{<meta package name>}

User menu entries for every dependencies - at least pointing to documentation
Changing user menus

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
Changing user menus

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
### Changing user menus

#### Optional

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$ cp -a /usr/share/doc/cdd-dev/examples/menu .</code></td>
<td>Copy file to current directory</td>
</tr>
<tr>
<td><code>$ cat menu/README</code></td>
<td>Display README file content</td>
</tr>
<tr>
<td><code>$ cat menu/task1/README</code></td>
<td>Display README file content in task1</td>
</tr>
<tr>
<td><code>$ edit menu/task1/dep1.menu</code></td>
<td>Edit menu entry for dep1</td>
</tr>
<tr>
<td><code>$ edit menu/task1/dep2.txt</code></td>
<td>Edit menu entry for dep2</td>
</tr>
<tr>
<td><code>$ edit menu/task1/dep3.html</code></td>
<td>Edit menu entry for dep3</td>
</tr>
</tbody>
</table>

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

Techniques
- Meta-packages
- User menus
- Communication

Prospectus
Developer platforms

- **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>
Developer platforms

- **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>](http://lists.debian.org/debian-<cdd>)
  - Web pages for most CDDs
    - [http://www.debian.org/devel/debian-<cdd>](http://www.debian.org/devel/debian-<cdd>)
**Developer platforms**

- **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>](http://lists.debian.org/debian-<cdd>)
- Web pages for most CDDs  
Developer platforms

- 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>
Developer platforms

- **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
Current state

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
Current state

- 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
Current state

- 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
Current state

- 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
Current state

- 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: \textit{Debian-Edu (SkoleLinux)}
- Back integration: \textit{DeMuDi}
- \textit{How to start a Custom Debian Distribution}
- Basic problem: reinventing the wheel frequently
- Debian-Fork possible $\rightarrow$ but CDD makes more sense
- Fork: requires adaptations continuously
- CDD: Work has to be done \textit{only once}
- Debian = Give and Take
Customisation regarding languages
- Dependencies of certain font packages
- Adapted input / output methods
- Special configuration
CDD in Asia

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

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

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
Future

- 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*
Future

- 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
Future

- 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
Future

- 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
Future

- 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>