



Akonadi – Pimpin' your PIM

Till Adam / Volker Krause



Topics

- Why do we need Akonadi?
- Design Overview
- Release: Here it is :)
- Contribute



Why?

Limitations of KDE3

- KResource framework limitations:
 - Data is not shared
 - Not designed for asynchronous access
 - Hard to extend to other data types
 - Basically no shared common code
 - KMail limitations:
 - Only limited backend abstraction
- **Designed for small amounts of local data**



Why?

Goals

- As much as possible shared, type independent functionality
- Easy to extend to new data types
- Unified API to access PIM data, independent of the actual data source
- Scalability
- **Useable for the whole free desktop**



Design Overview

- Client/Server approach
- [add picture]



Server

- Fully type independent
- Cache for remote data
- Change notification
- Conflict detection



Basic Objects

- Filesystem-like structure:
 - Collections
 - Items
- Items can consist of multiple parts so clients can access only the actually needed data



Resource Agents

- Connect Akonadi to external data sources
 - local files (maildir, iCal, vCard, ...)
 - mail- or groupware servers
 - web services
- Translate data formats



Client/Server Communication

- Two communication channels:
 - D-Bus for control data
 - IMAP-like protocol for content data
- Standard formats for content data (MIME, iCal, vCard, etc.)
- **Toolkit and language independent interface**



Clients Libraries

- Currently only one: libakonadi, C++/KDE
- Consists of type-independent part and type specific plugins



Release

Here it is :)

- [TODO]



Requirements

- Server:
 - D-Bus
 - Qt 4.3
 - MySQL Server binary, does not need to be configured and running
- Clients:
 - recent kdelibs + kdepimlibs



How do I use it?

- Starting/stopping Akonadi:
`akonadi_ctl start/stop`
- Akonadi Console:
 - Manage resource agents
 - Browse content
 - Watch client/server communication



Release



[screenshot pending]



What can I do?

- Contributing to the Akonadi server
- Additional Client library implementations
- Language bindings
- Add resource agents for new backends
- Extend Akonadi to support new types
- Using Akonadi in applications



Akonadi Server

- Fulltext/Metadata indexing & search:
 - Virtual collections
 - Search in backends (IMAP, LDAP, etc.)
- Additional features:
 - Local subscriptions / Subscription profiles
 - Filtering
- Performance/Database optimization



Client Libraries

- Currently only one: KDE/C++
- Possible approaches:
 - Native implementations:
 - Native data types, easy integration
 - Language bindings:
 - Scripting languages, RAD



Extending Akonadi

- Support additional backends: groupware servers, web services, ...
- Support for additional data types: Notes, IM messages, RSS feed entries, ...
- See “Developing PIM applications with Akonadi” [TODO: fill in date/time]



Using Akonadi in Applications

- Port existing applications
- New possibilities:
 - Integrate PIM data wherever useful:
 - Every mail address can be linked to your addressbook
 - Every date can be linked to your calendar
 - Plasma applets / Desktop widgets



Resources

- [#kontakt](#) on [irc.freenode.org](#)
- [pim.kde.org/akonadi](#)
- kde-pim@kde.org
- Talk: “Developing PIM Applications with Akonadi” later today [### add correct time/location]