



The Technology Behind KDE 2.0

Richard Moore, rich@kde.org

This Presentation Will Cover

- User interface improvements
- The improvements in interoperability
- The facilities of the KDE libraries
- Some of the design decisions we took
- On-going work for KDE 2.1

Better Interoperability

- NET Compliant window manager
- XDnD drag and drop protocol (and Motif)
- Standardised desktop files
- XSMP based session management
- Support many different platforms
- Better integration with non-KDE apps
- Built-in SMB support

More Flexibility

- New window manager ‘KWin’
 - A reusable window management library
 - The user interface is provided by plugins
- New panel ‘Kicker’
 - Supports plugin applets
- Icon themes
 - New sets of icons
 - Dynamic effects

KDE Theme Engine

- As simple as a configuration file, or pure C++ or a mixture of both
- Drawn dynamically, not just pixmaps
- Behaviour as well as appearance
- Supports GTK pixmap themes
- Built on the Qt style facilities

Qt Improvements

- Support for themes
- More widgets and other basic tools
- Unicode strings
- Graphical UI Builder
- XML Parser
- Qt Object Model
- QActions

The KDE Libraries

- Used by all KDE applications
- Wide range of facilities
- Well documented platform for developers
- Ensures consistency between applications
- Divided up into several smaller libraries to minimise unnecessary overhead

KIO

- Network transparent I/O library
- Non-blocking
- Built in HTTP, HTTPS, FTP, NFS, local files, gopher, SMB...
- Filter protocols such as tar and gzip
- New protocols can be plugged in
- LDAP, Diamond Rio, WebDAV under development

KSycoca

- System Configuration Cache
- Read-only binary cache of the text configuration files
- Design for rapid access
- Automatically updated when a configuration file is modified
- One copy shared by all processes

KNotify

- Built on DCOP
- Standard way for apps to notify the user
- Messages can be logged, show dialogs, play a sound
- Supported by KWin, Kit and various standard components (such as KCompleter)

MCOP and aRts

- Multimedia Communication Protocol
- High performance binary protocol designed for streaming audio, video etc.
- aRts real-time synthesiser
- Supports wav, mp3, mpeg2, MIDI...
- Plugin architecture
- No dependency on KDE

DCOP

- Desktop Communication Protocol
- Fast and lightweight
- Built on the X11 ICE protocol
- Very easy to use
- Includes an XML based IDL
- Many bindings exist
 - C++, C, Java, Python, Perl, XMLRPC...
- Command line and GUI user tools

Why not CORBA?

- We tried CORBA for nearly a year before we decided to ditch it.
- Failed to deliver many of its promised benefits
- Steep learning curve meant less people working on KDE
- CORBA can only deliver one part of the component system
- Poor C++ IDL

KParts

- Component framework for KDE
- Dynamically loaded libraries
- User interface is defined as XML and merged with the host application
- Supports plugins
- Used throughout KDE
- Very easy to develop for
- Automatic style guide compliance

Where do we go from here?

- Much faster release cycle
- Version 2.1
 - New extension API for the panel
 - UI improvements
 - Maybe a new language bindings module
 - Updated applications
 - Evolutionary rather than revolutionary changes
- ViaVoice integration