Thomas Fischer ⟨fischer@unix-ag.uni-kl.de⟩
Desktop Summit 2011 – 7 August 2011 – Humboldt University of Berlin

KParts Browser Plugin

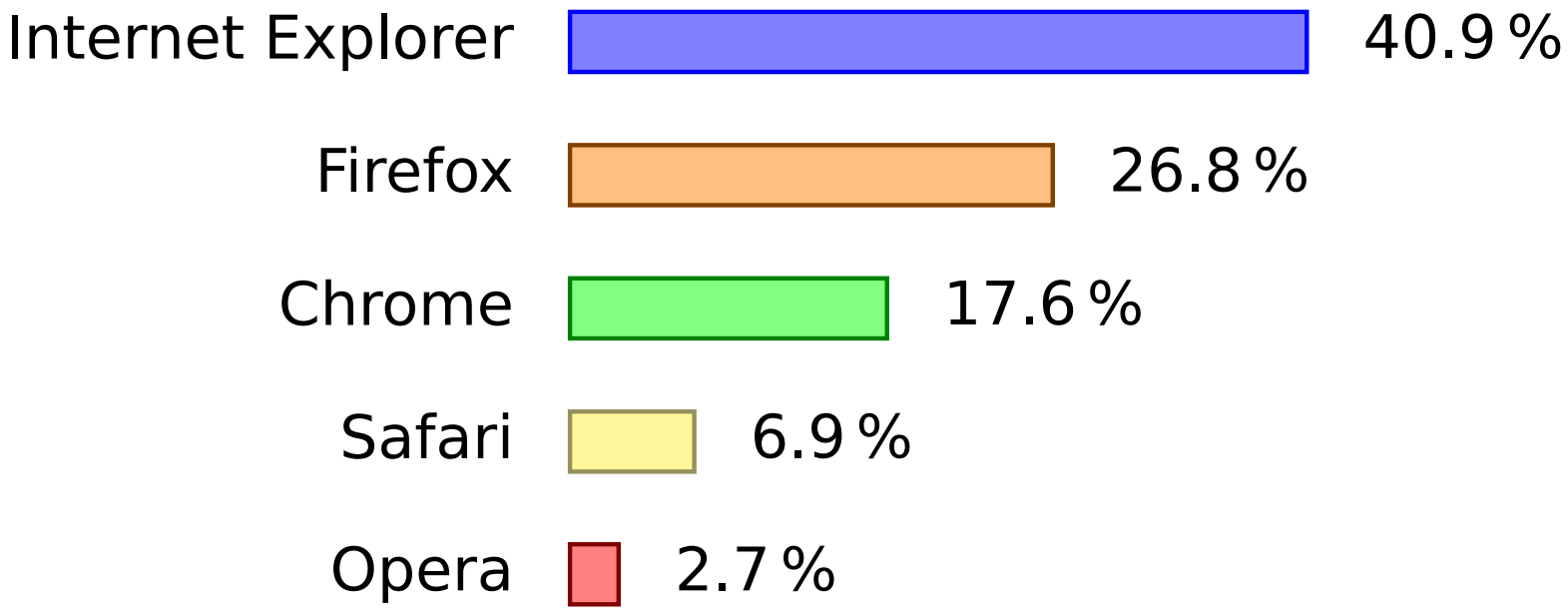
Bringing KDE Components to All Browsers

About Myself

- **Thomas Fischer**
- Background in **computer science**
  - **MSc** in Darmstadt, Germany
  - **PhD** in Kaiserslautern, Germany
- **Lecturer** at the **University of Skövde**, Sweden
- Using **KDE** since **late 2001**
- **Translator** English → German ~2002 to ~2005
- Started **KBibTeX** in 2005
- Started **KParts Plugin** in 2009

What is Your Favorite Browser?

What is Your Favorite Browser?



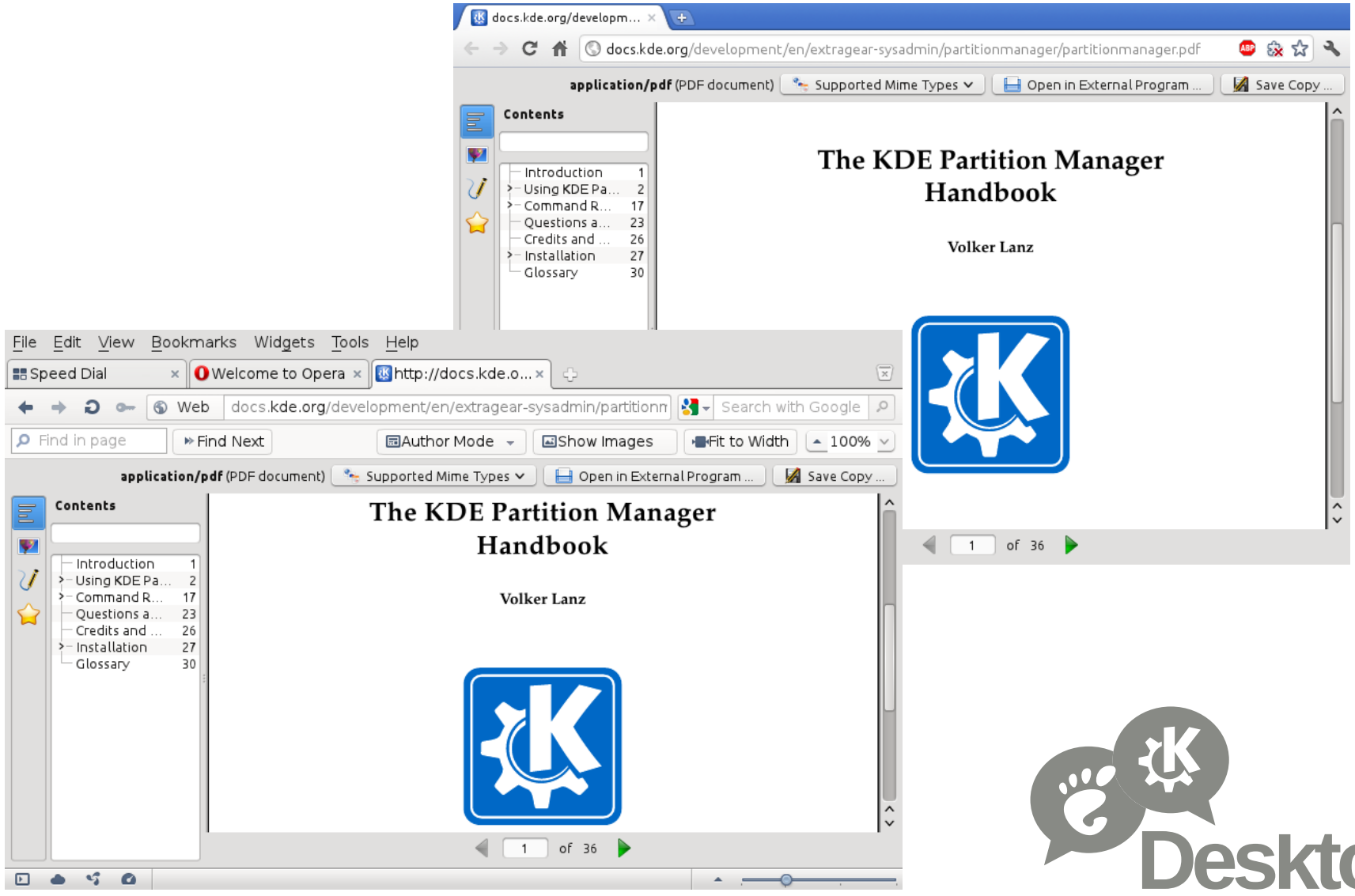
Wikipedia: Median values over several browser usage statistics for June 2011

Desktop summit

What is the KParts Plugin?

- It is a **Netscape-compatible browser plugin** as you know from Flash or Acrobat Reader
- It uses KParts to present **embedded documents**
  - PDF, PostScript, …
  - Office documents
  - Audio and video files
- **330 supported mimetypes** on my laptop

Some Screenshots



Desktop summit

Live Demonstration

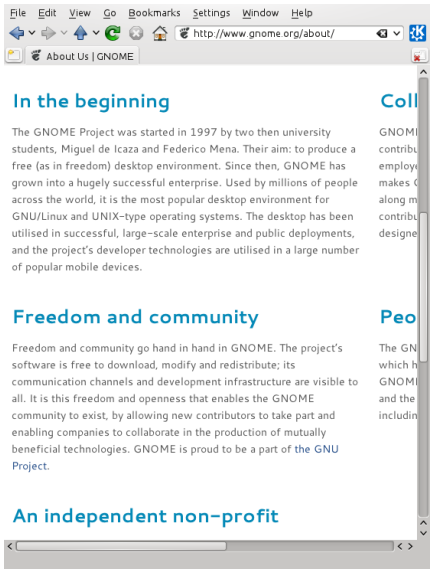
```
1 <html><head><title>Plugin Test</title></head>
2 <body><h1>Plugin Test</h1>
3 <p><object type="application/vnd.oasis.opendocument.
    presentation" data="http://blog.freifunk.net/files/
    freifunk-mitmachen.odp" width="900" height="600">
    Plugin not available?</object><br/>Same file as a <a
     href="http://blog.freifunk.net/files/freifunk-
    mitmachen.odp">link</a>.</p>
4 <p><object type="application/pdf" data="http://www.bvg
    .de/index.php/de/binaries/asset/download/20909/file
    /1-1" width="800" height="700">Plugin not available?
    </object><br/>Same file as a <a href="http://www.bvg
    .de/index.php/de/binaries/asset/download/20909/file
    /1-1">link</a>.</p>
5 </body></html>
```

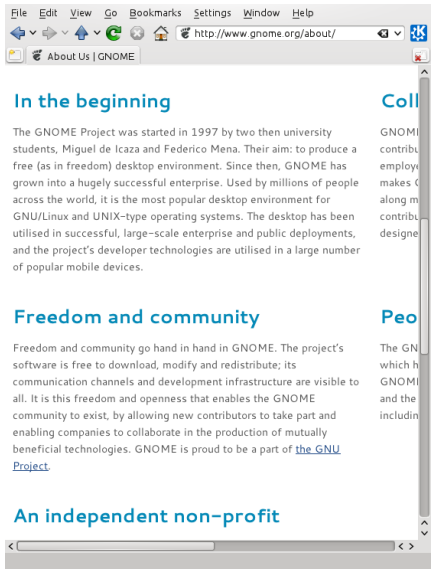
Advantages using the KParts Plugin

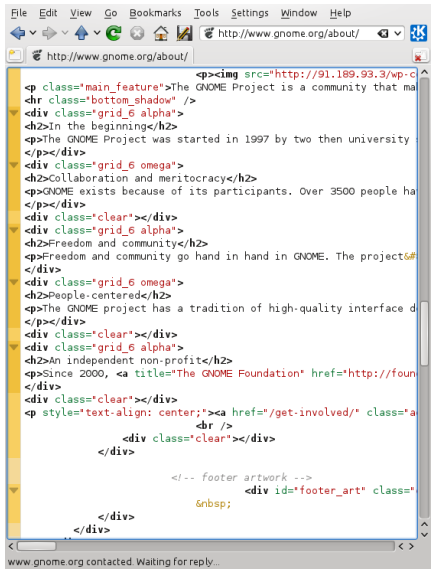
- Brings your **favorite applications** to your **favorite browsers**
- Brings browser plugins to **architectures where not alternatives exist**
- **Open-source browser plugin** instead of binary-only, closed-source browser plugins
- **Save to file** for later use
- **Open in external application** which does not need to be KDE-based

KParts for Gnome Users

- **Component framework** (like Bonobo?)
- **Example** Konqueror uses a **KHTML** part to view web pages or a **Kate** editor component for text



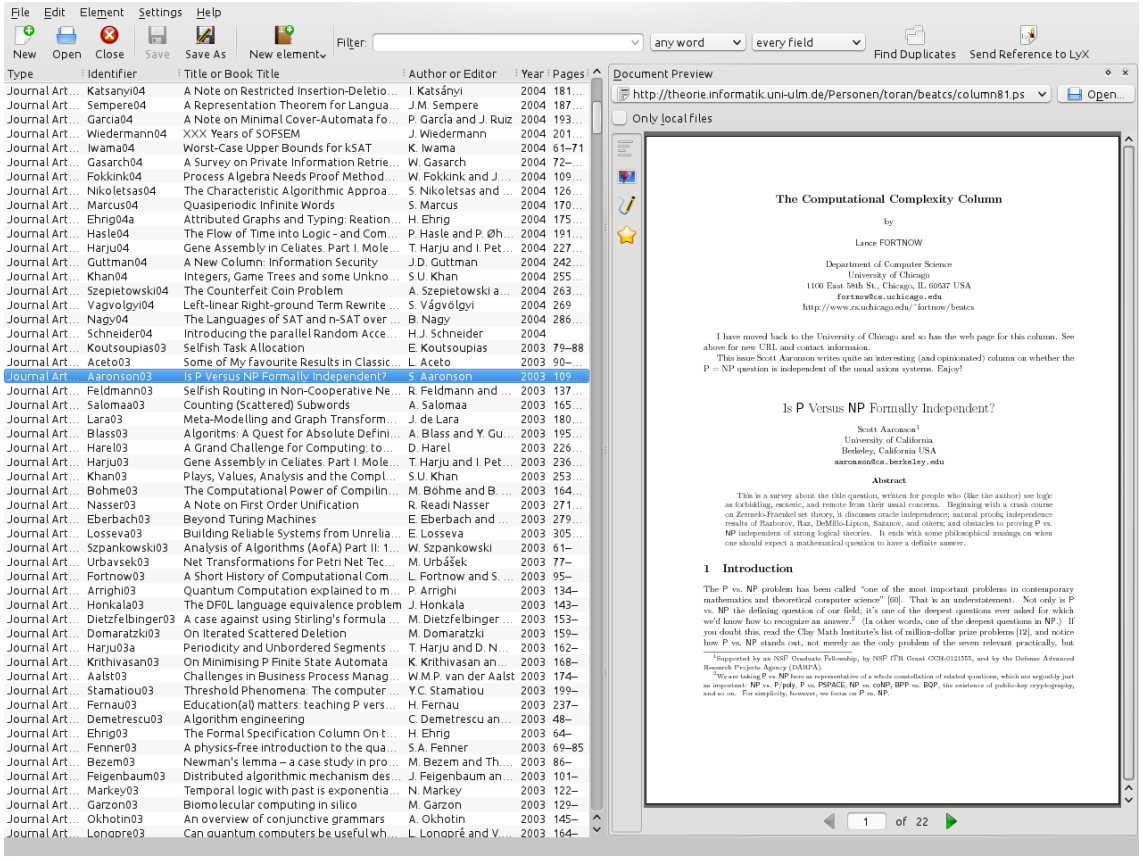




Desktop summit

KParts – Everywhere in KDE

- **Okular** is only a shell around the Okular plugin
- Same holds for **Kate**, **Dolphin**, **KOffice**, …



Desktop summit

Netscape Plugin API

- **Plugin architecture** supported by most browsers
- Small API between browser and plugin:
  ~ **15 functions**

Source: Wikipedia

1. At start, **browser scans for plugins**
2. Plugins notify browser about supported **mime types**
3. Browser encounters request for plugin while surfing, locates **plugin matching mime type**
4. Browser initializes plugin, sends **mime type**, **raw data**, and **space to show data in**
5. Plugin uses **X11 protocol magic** to embed itself

Writing Plugins with Qt

- **Wrapper code** between NPAPI and a QWidget
  BSD-licensed code by Nokia, not part of Qt tar ball
- Support for X11/Netscape browser plugins dates back to **Qt 3.1** (~ 2003)

1. Let your object **inherit** QWidget and QtNPBindable
2. **Implement virtual functions** of QtNPBindable
   e. g. to read raw data from `QIODevice`
3. Create a **helper object** for plugin initialization
4. Implement your QWidget **as you like**
   (e. g. load KParts)

Source: Qt/Nokia

Code: KParts Plugin's Public Interface

```
1 class KPartsPlugin : public QWidget, public
    QtNPBindable
2 {
3     Q_OBJECT

5 public:
6     KPartsPlugin(QWidget *parent = NULL);

8     bool readData(QIODevice *source, const QString &
        format);
9 }
```

Code: Plugin Initialization

```
1  class QtNPClassList : public QtNPFactory {
2  public:
3    QtNPClassList() {
4      initAllMimeTypes();
5    }
6    QObject *createObject(const QString &) const {
7      return new KPartsPlugin();
8    }
9    QStringList mimeTypes() const {
10     return allMimeTypes;
11   }
12 };

14 QtNPFactory *qtns_instantiate() {
15     return new QtNPClassList;
16 }
```

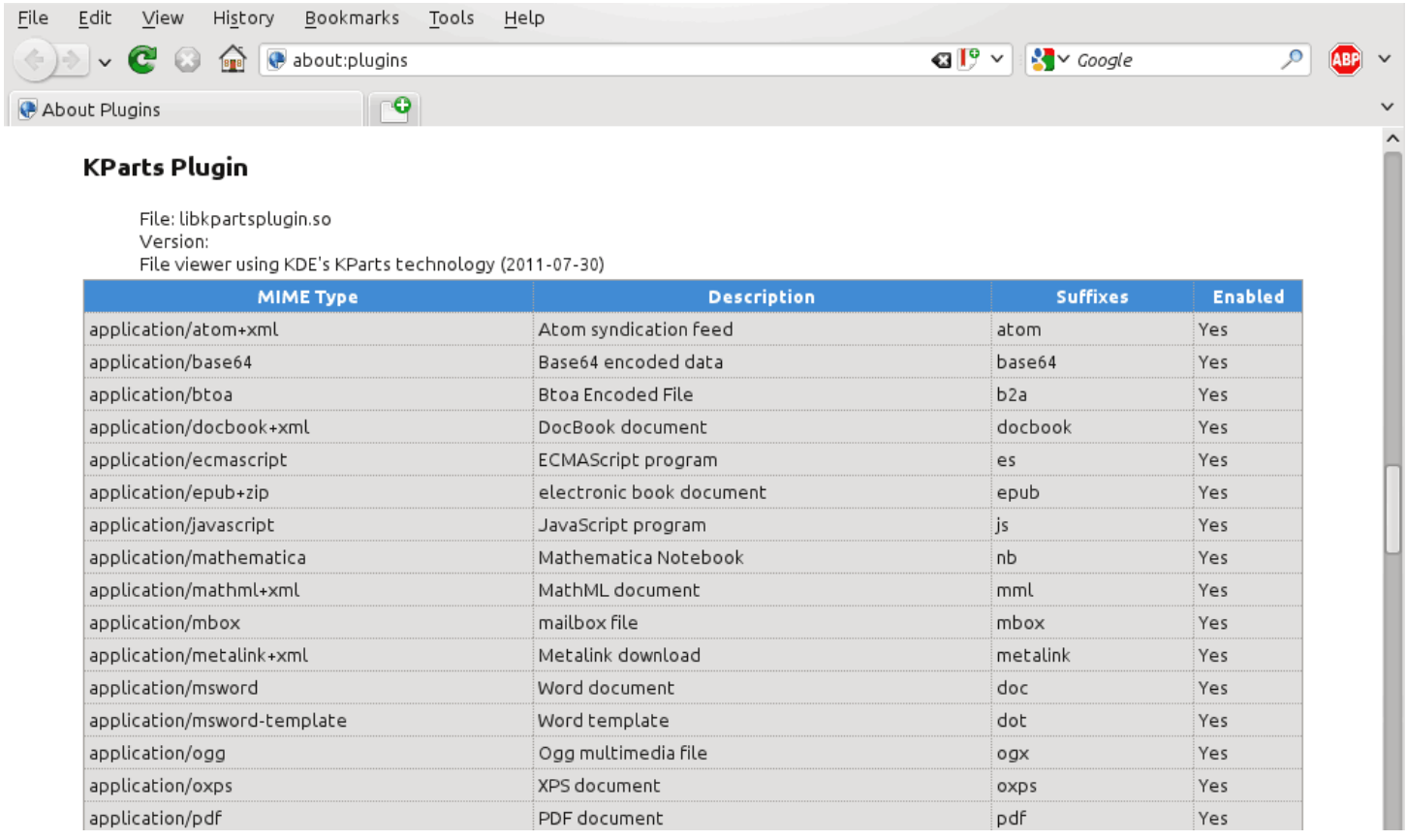
Code: Supported Mime Types

```
1  static void initAllMimeTypes() {
2    KMimeType::List mtL = KMimeType::allMimeTypes();

4    foreach (const KSharedPtr<KMimeType> &mtP, mtL) {
5      QString mimetype = mtP->name();
6      QString ext = mtP->mainExtension().mid(1);
7      QString descr = mtP->comment();

9      KService::List list = KMimeTypeTrader::self()->
         query(mimetype, "KParts/ReadOnlyPart");
10     if (!list.isEmpty())
11       allMimeTypes.append(QString("%1:%2:%3").arg(
           mimetype).arg(ext).arg(descr));
12   }
13 }
```

Firefox Lists Supported Mime Types



KParts Plugin

File: libkpartsplugin.so
Version:
File viewer using KDE's KParts technology (2011-07-30)

MIME Type	Description	Suffixes	Enabled
application/atom+xml	Atom syndication feed	atom	Yes
application/base64	Base64 encoded data	base64	Yes
application/btoa	Btoa Encoded File	b2a	Yes
application/docbook+xml	DocBook document	docbook	Yes
application/ecmascript	ECMAScript program	es	Yes
application/epub+zip	electronic book document	epub	Yes
application/javascript	JavaScript program	js	Yes
application/mathematica	Mathematica Notebook	nb	Yes
application/mathml+xml	MathML document	mml	Yes
application/mbox	mailbox file	mbox	Yes
application/metalink+xml	Metalink download	metalink	Yes
application/msword	Word document	doc	Yes
application/msword-template	Word template	dot	Yes
application/ogg	Ogg multimedia file	ogx	Yes
application/oxps	XPS document	oxps	Yes
application/pdf	PDF document	pdf	Yes

Desktop summit

Code: Creating the KPart

```
1  bool KPartsPlugin::readData(QIODevice *source, const
     QString &format) {
2    KTemporaryFile tempFile
3    copyIODevice(source, &tempFile);

5    KService::Ptr service = KMimeTypeTrader::self()->
       preferredService(format, "KParts/ReadOnlyPart");
6    KParts::ReadOnlyPart *part = service->createInstance
       <KParts::ReadOnlyPart>((QWidget*)this, (QObject*)
       this);

8    QWidget *partWidget = part->widget();
9    m_gridLayout->addWidget(partWidget, 1, 0, 2, 4);

11   part->openUrl(KUrl(tempFile.fileName()));

13   return true;
14 }
```

Source Lines of Code

Total Physical Source Lines of Code (SLOC)	**377**
Development Effort Estimate, **Person-Years** (Basic COCOMO model, Person-Months = $2.4 \times \text{KSLOC}^{1.05}$)	0.07
Schedule Estimate, Years (Basic COCOMO model, Months = $2.5 \times \text{person-months}^{0.38}$)	0.20
Estimated Average Number of Developers (Effort/Schedule)	0.36
Total Estimated **Cost to Develop** (average salary = US$ 56 286/year, overhead = 2.40)	US$ 9701 6850 €

generated using David A. Wheeler's 'SLOCCount'

Code Not Shown Here

- **Error Handling**
- **Save** as File and **Open** in External Program
- **Selection Dialog** if multiple KParts available
- **Black-listing** of mime types
- **Enabling/disabling** of mime types
- Graphical user interface (**GUI**)

Where to get it

- Check your **distribution**
  (Gentoo, Ubuntu, Debian, Fedora, openSUSE, ...)
- Fetch the **sources** at
  `http://www.t-fischer.net/kpartsplugin/`
- **Compile** it

```
1 tar -xvf kpartsplugin-20110806.tar.bz2
2 mkdir b ; cd b
3 cmake ../kpartsplugin-20110806
4 make
5 sudo make install
```

- You may need to install **development packages** for KDE and Qt

… and Gnome?

- **Epiphany** supports KParts Plugin out of the box
  (if you have a minimum KDE installation and parts installed)
- How to make it **Gnome-aware**
  (a) **Rewrite** KParts Plugin in C using Gnome libraries, loading Gnome components
  (b) Write KPart **wrapper** around Gnome components
- **Share** configuration files
- **Reuse** code?
  Unlikely, too much code is Qt/KDE-specific

Future Extensions

- **Internationalization**
  (so far, `KLocale` insists on `en-US`)
- Integration of KPart's **Shortcuts**, **Menu** and **Toolbars**
- More advanced browser plugin features (scripting, printing, ...)
- Port to **Windows and Mac** using the KDE framework
- **ActiveX** support (QtBrowserPlugin supports this)

Thomas Fischer ⟨fischer@unix-ag.uni-kl.de⟩
Desktop Summit 2011 – 7 August 2011 – Humboldt University of Berlin

Thank you for your attention

Questions?