kte-collaborative: Collaborative text editing in KTextEditor and kde-telepathy

Sven Brauch

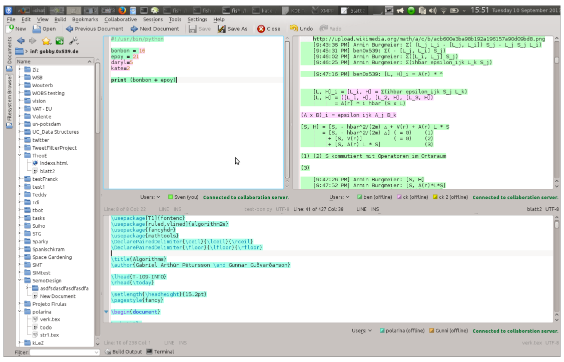
September 11, 2013

kte-collaborative is a project which provides support for collaborative text editing in KTextEditor and enables you to edit documents collaboratively with your Instant Messenger contacts without the need for a centralized server or manual connection setup. It is based on the KTextEditor interfaces for editing text, on the libinfinity library for the protocol and the collaborative editing part, and on kde-telepathy for widgets and functionality related to sharing documents with instant messenger contacts.

1 Purpose of this document

This document is aims to be an advanced user's guide to the program. The main part of the document is meant for pure users of the software, although some sections might be too technical to be of interest for some users.

Sections which are probably mostly relevant for technically interested users are marked with ※.



```
#!/usr/bin/python

bonbon = 16
epoy = 21
daryl=5
kate=2

print (bonbon + epoy)
```

Users: Sven (you) Connected to collaboration server.
Line: 8 of 8 Col: 22 LINE INS test-bon.py UTF-8

```
http://upload.wikimedia.org/math/a/c/b/acb600e3ba98b192a196157a90d09bd8.png
[9:43:36 PM] Armin Burgmeier: Σ( (L_j L_i - [L_j, L_i]) S_j - L_j S_j L_i)
[9:45:31 PM] ben0x539: Σ( - [L_j, L_i] S_j)
[9:46:02 PM] Armin Burgmeier: Σ([L_i, L_j] S_j)
[9:46:25 PM] Armin Burgmeier: Σ(ihbar epsilon_ijk L_k S_j)

[9:47:16 PM] ben0x539: [L, H]_i = A(r) * ^


[L, H]_i = [L_i, H] = Σ(ihbar epsilon_ijk S_j L_k)
[L, H] = ([L_1, H], [L_2, H], [L_3, H])
       = A(r) * i hbar (S x L)

(A x B)_i = epsilon ijk A_j B_k

[S, H] = [S, - hbar^2/(2m) △ + V(r) + A(r) L * S
       = [S, - hbar^2/(2m) △] ( = 0)    (1)
         + [S, V(r)]         ( = 0)    (2)
         + [S, A(r) L * S]             (3)

(1) (2) S kommutiert mit Operatoren im Ortsraum

(3)

[9:47:26 PM] Armin Burgmeier: [S, H]
[9:47:52 PM] Armin Burgmeier: [S, A(r)*L*S]
```

Users: ben (offline) ck (offline) ck 2 (offline) Connected to collaboration server.
Line: 41 of 427 Col: 38 LINE INS blatt2 UTF-8

```
\usepackage[T1]{fontenc}
\usepackage[ruled,vlined]{algorithm2e}
\usepackage{fancyhdr}
\usepackage{mathtools}
\DeclarePairedDelimiter{\ceil}{\lceil}{\rceil}
\DeclarePairedDelimiter{\floor}{\lfloor}{\rfloor}

\title{Algorithms}
\author{Gabríel Arthúr Pétursson \and Gunnar Guðvarðarson}

\lhead{T-109-INTO}
\rhead{\today}

\setlength{\headheight}{15.2pt}
\pagestyle{fancy}

\begin{document}
```

Users: polarina (offline) Gunni (offline) Connected to collaboration server.
Line: 10 of 238 Col: 1 LINE INS verk.tex UTF-8

Figure 1: Example screenshot of kte-collaborative running in kate, displaying three documents from the gobby.0x539.org server.

Contents

2 Purpose of the software

2.1 Things we want to do

kte-collaborative was designed with the following aims in mind:

- Enable the average "non-technical" KDE user who already uses kde-telepathy as his instant messenger to collaboratively edit documents with his Jabber/XMPP contacts, without requiring any network-related setup. A typical example use case would be planning a vacation, and writing down what needs to be done in a collaborative document.

- Allow the same thing also for "technical" use cases, i. e. provide a text editor which is suitable for editing program- or markup code. Typical example use cases include explaining the syntax of a programming language to someone, or writing a university homework in LaTeX together.

- Provide a tool to KDE developers for code review or related purposes. A typical example for this would be a new contributor working on a cpp file in KDevelop for a while, but then running into a problem, and needing help from a more experienced developer. Instead of uploading his changed copy of the document to a pastebin service, and other people downloading that pastebin, changing it and uploading it again, it should be possible for the new contributor to easily share his document for collaborative editing right from KDevelop, and other people should with the same ease be able to look at the document from KDevelop.

- Finally, the project aims to provide a full-featured KDE-based client for the infinote protocol (which is thus compatible with e. g. gobby). This implies that we aim to support all *essential* features the protocol, but not that we aim to provide a tool suitable for every obscure use case someone can come up with.

2.2 Things we do not care about

The project at the current state does explicitly *not* care about the following things, although that might change in the future:

- Performance with more than 25 active users.

- Performance with documents larger than 20.000 lines or 500.000 characters.

- Being bandwidth-efficient.

- Having a small set of dependencies.

3 Installation

Since installation procedure is so different between different systems, it will not be described in detail here; just short hints are given.

3.1 Using a package manager

Usually, you should be able to install this software by searching your package manager for a package named `kte-collaborative` or similar. Install the package, and log out and back in again. Then, launch e. g. Kate to start using the program.

3.2 By building from source code ※

You can get the source code from the KDE servers using `git clone`:

- for libqinfinity (dependency): `git://anongit.kde.org/libqinfinity.git`
- for kte-collaborative: `git://anongit.kde.org/kte-collaborative.git`

Notable dependencies include the `libinfinity` and, if this is packaged separately in your distribution, `infinoted` packages.

In this case, too, a variety of services needs to be restarted and that requires a variety of commands to do properly; thus, it is probably a good idea to restart your computer or at least log out of your session after installing the plugin to ensure everything works as intended.

3.3 Notes for packaging ※

kde-telepathy-meta should include or at least recommend this package. Additionally to the dependencies which are obvious, the infinote server application, `infinoted`, currently is a hard compile- and runtime dependency (CMake should verify it is present at compile time), so the package must depend on the package containing this application.

4 The user interface

kte-collaborative does not contain a dedicated editor application by itself. Instead, it provides a plugin for the KTextEditor interface, which especially means it will work in any application using KatePart. Prominent examples of such applications include Kate itself, but also KWrite, Kile, and KDevelop. In each of these applications, you should have a "Collaborative" menu like in Figure 2 when the software is installed correctly (if not, see 10.1).

4.1 The "Share document" button

This button is only enabled when you are not editing a collaborative document.

This dialog, displayed in Figure 3, is used to publish a document and make it available to remote users. For details on what the buttons do, refer to Section 5.

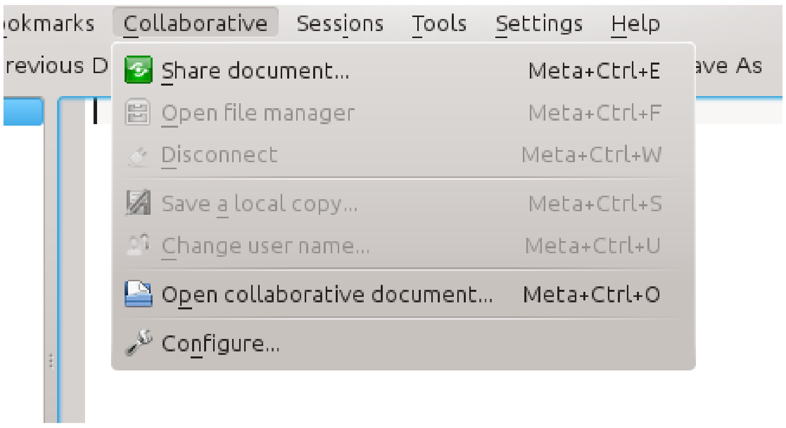
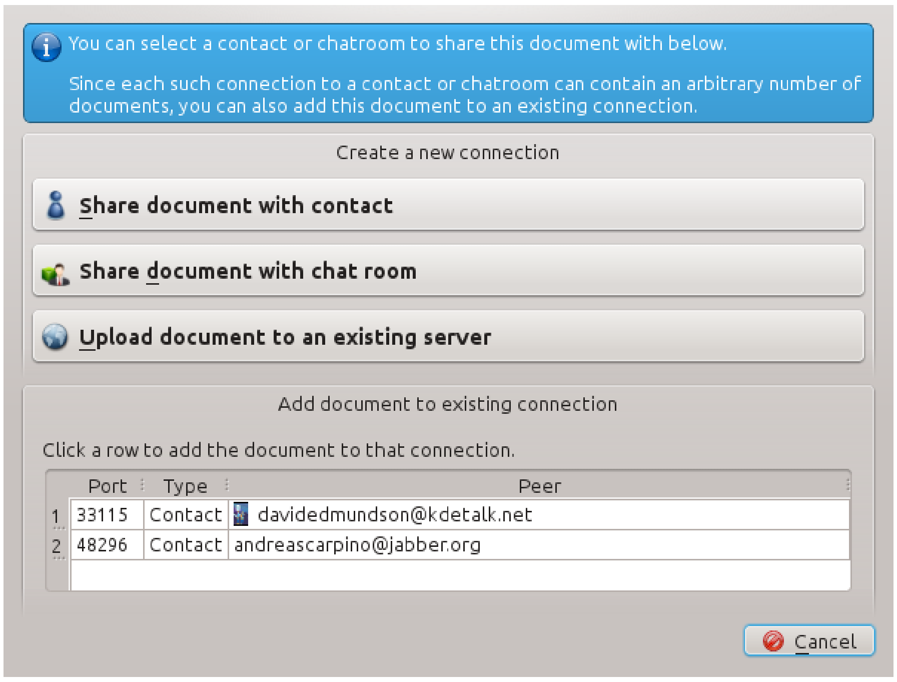

okmarks
Collaborative
Sessions
Tools
Settings
Help
revious D
ave As
Share document... Meta+Ctrl+E
Open file manager Meta+Ctrl+F
Disconnect Meta+Ctrl+W
Save a local copy... Meta+Ctrl+S
Change user name... Meta+Ctrl+U
Open collaborative document... Meta+Ctrl+O
Configure...

Figure 2: The user interface the plugin adds to a host application when it is loaded.



You can select a contact or chatroom to share this document with below.

Since each such connection to a contact or chatroom can contain an arbitrary number of documents, you can also add this document to an existing connection.

Create a new connection

Share document with contact
Share document with chat room
Upload document to an existing server

Add document to existing connection

Click a row to add the document to that connection.

	Port	Type	Peer
1	33115	Contact	davidedmundson@kdetalk.net
2	48296	Contact	andreascarpino@jabber.org

Figure 3: The “Share document” dialog.

4.2 The “Open file manager” button

This button is only enabled when you are editing a collaborative document.

Opens dolphin at the place where the current collaborative document is saved, and thus enables you to add more documents there.

4.3 The “Disconnect” button

This button is only enabled when you are editing a collaborative document.

Disconnects from the current session, and saves the current buffer contents somewhere in `/tmp`. Changes you do will no longer be synchronized to the server, and you will not receive any changes from other users.

4.4 The “Save a local copy” button

This button is only enabled when you are editing a collaborative document.

Saves the document to a local file, but stays connected to the collaborative session. Asks for a location the first time you trigger it, future calls will overwrite that document.

4.5 The “Change user name” button

This button is only enabled when you are editing a collaborative document.

Change your user name, i. e. the name you appear with for remote users to a different one. For technical reasons, this will disconnect you from the session and re-join it with the new user name. The way colors are picked also implies that you will have a different color.

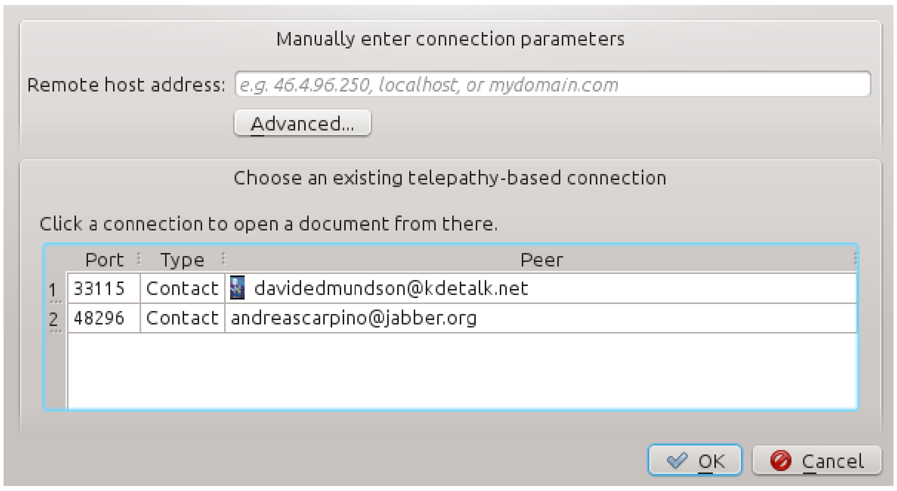
4.6 The “Open collaborative document” button

This dialog, pictured in Figure 4, is used to open a collaborative document, either from an existing Telepathy-initiated connection, or from a classic server. The “Advanced” button offers some advanced options, such as using a different port. For more information, refer to Section 7.1.

An example public server is `gobby.0x539.de`.

4.7 The “Configure” button

This button opens the configuration module, as seen in Figure 5. For more information on the configuration options it provides, see Section 8.



Manually enter connection parameters

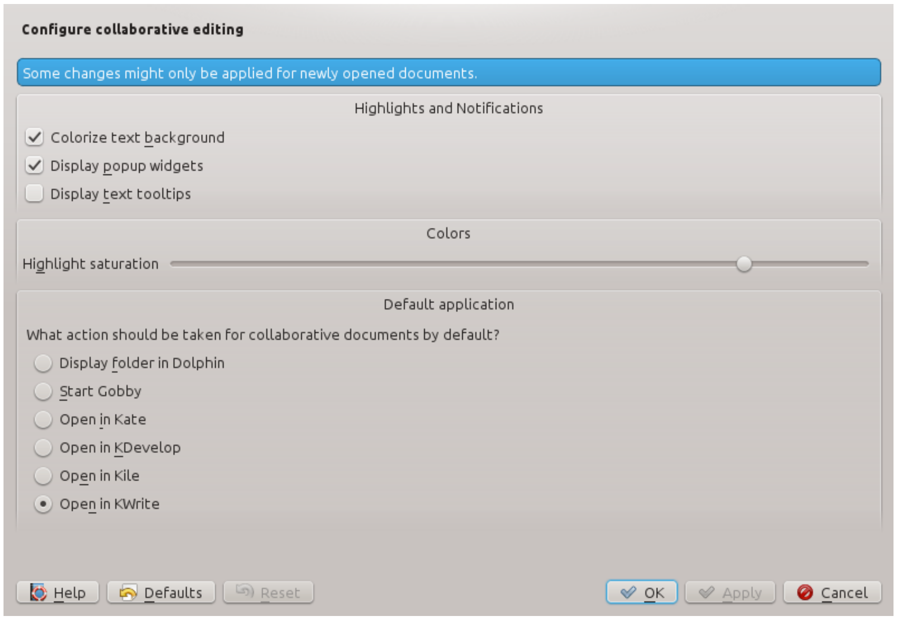
Remote host address: *e.g. 46.4.96.250, localhost, or mydomain.com*

Choose an existing telepathy-based connection

Click a connection to open a document from there.

	Port	Type	Peer
1	33115	Contact	davidedmundson@kdetalk.net
2	48296	Contact	andreascarpino@jabber.org

Figure 4: The “Open document” dialog.



Configure collaborative editing

Some changes might only be applied for newly opened documents.

Highlights and Notifications

- ✓ Colorize text background
- ✓ Display popup widgets
- ☐ Display text tooltips

Colors

Highlight saturation

Default application

What action should be taken for collaborative documents by default?

- ○ Display folder in Dolphin
- ○ Start Gobby
- ○ Open in Kate
- ○ Open in KDevelop
- ○ Open in Kile
- ● Open in KWrite

Figure 5: The Settings dialog.

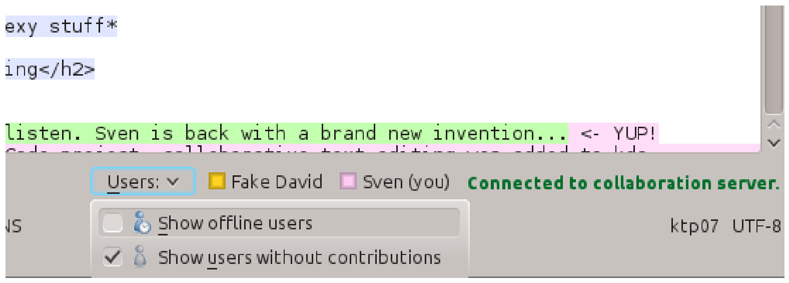

exy stuff*
ing</h2>
listen. Sven is back with a brand new invention... <- YUP!
Users:
Fake David
Sven (you)
Connected to collaboration server.
Show offline users
Show users without contributions
ktp07 UTF-8

Figure 6: The status bar below the document.

4.8 The status bar

The status bar below the document, as seen in Figure 6, currently indicates

- the connection status of the session (connected, loading, disconnected);
- and the users which are part of the session.

If the status bar is not present, then the document you are editing is not a collaborative document.

The users list shows a little colored box indicating the color the user will appear with in the document, and the user's name. Your own name and color is marked with the suffix "(you)" behind your name. Offline users are marked with the "(offline)" suffix and by the right lower half of their color box being grayed out.

If more users are online than fit into your status bar, the users list will collapse and only show users' color boxes. Moving the mouse over one of the boxes will show the user name in a tooltip. If even the collapsed form would not fit into the status bar, only the amount of users is displayed instead.

Clicking the "Users" button gives two options for customizing who is displayed in the users list:

- Checking *Show offline users* will also show all users which have been in the session at some point in the past, but currently are not. This setting is disabled by default.
- Checking *Show users without contributions* will show all users which currently have text belonging to them in the document, or had text belonging to them since you joined the session. Disabling this checkbox is useful to reduce the list to the participating users in case e. g. three people are writing a protocol and 20 are watching. This setting is enabled by default.

Thus, by default, the toolbar will simply display everyone who is online; but you can for example make it display everyone who wrote a part of the document in the past instead by setting "Show offline users" to true and "Show users without contributions" to false.

5 Different ways to share a document

There are two fundamtentally different ways to share a document with someone:

- By adding it to an existing server.
- By using Telepathy to build a connection to a Jabber contact or chatroom.

In either case, a connection will be built and you can share an arbitrary number of documents and folders over this same connection. For this purpose, you can for example use the “Open file manager” button (see 4.2) and just copy documents into the window which opens.

5.1 Technical details of Telepathy-based connections ※

Even in the peer-to-peer connections built by Telepathy, there currently always is a server running. The server is run by the person initiating the request. The server is called *infinoted* and is started by kte-collaborative in the background in this case. Telepathy then establishes the connection and binds the port the server is listening on on your computer to some port on the remote computer, with black magic to actually transfer data between the two. This black magic can work in three ways:

- If you are in the same local network as the target or have a publically reachable IP address, the connection is established directly.
- Otherwise, an attempt is made to use a proxy server for the connection.
- If both fails, an attempt is made to build an *in-band bytestream* connection, which transfers data base64-encoded inside XMPP stanzas, and thus basically uses the jabber server as a proxy.

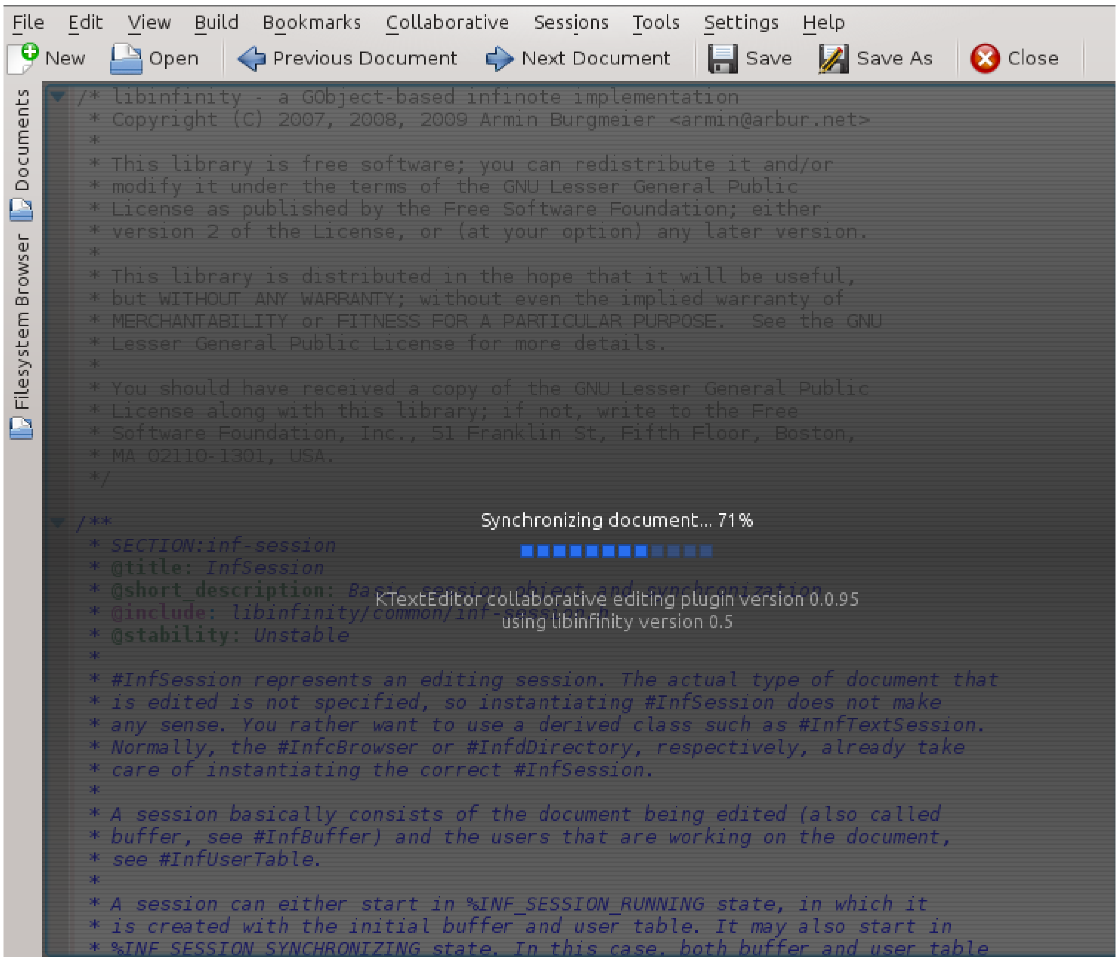
In any of the above cases, all parties will end up having a local port on their computer (e. g. 127.0.0.1:41531) with all data flowing through that port being transferred to the others by Telepathy. Thus, all documents from Telepathy-built connection always end up having 127.0.0.1 as their host address.

For the same reason, the peer-to-peer connections have exactly the same properties as the classical client-server connections, especially regarding having multiple documents and folders in one connection.

6 Sharing a document using Jabber/XMPP

6.1 Sharing a document with one contact

To share a document with a contact, first open the document you want to share. Then open the “Share document” dialog, and click “Share document with contact”. You will be prompted to select a contact from your contact list. The capability to collaboratively edit a document is advertised on XMPP; thus only contacts which actually have the



```
/* libinfinity - a GObject-based infinote implementation
 * Copyright (C) 2007, 2008, 2009 Armin Burgmeier <armin@arbur.net>
 *
 * This library is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Lesser General Public
 * License as published by the Free Software Foundation; either
 * version 2 of the License, or (at your option) any later version.
 *
 * This library is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
 * Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public
 * License along with this library; if not, write to the Free
 * Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston,
 * MA 02110-1301, USA.
 */

/**
 * SECTION:inf-session
 * @title: InfSession
 * @short_description: Basic session object and synchronization
 * @include: libinfinity/common/inf-session.h
 * @stability: Unstable
 *
 * #InfSession represents an editing session. The actual type of document that
 * is edited is not specified, so instantiating #InfSession does not make
 * any sense. You rather want to use a derived class such as #InfTextSession.
 * Normally, the #InfcBrowser or #InfdDirectory, respectively, already take
 * care of instantiating the correct #InfSession.
 *
 * A session basically consists of the document being edited (also called
 * buffer, see #InfBuffer) and the users that are working on the document,
 * see #InfUserTable.
 *
 * A session can either start in %INF_SESSION_RUNNING state, in which it
 * is created with the initial buffer and user table. It may also start in
 * %INF_SESSION_SYNCHRONIZING state. In this case, both buffer and user table
```

```
Synchronizing document... 71 %

KTextEditor collaborative editing plugin version 0.0.95
using libinfinity version 0.5
```

Figure 7: The “splash screen” of the plugin while loading a collaborative document

program installed and configured correctly will be selectable.

The other person will receive a notification about the request, and can choose to deny or accept it. Your editor should immediately load the collaborative document (see Figure 7). The other person should join shortly after.

6.2 Sharing a document with a chatroom

This works the same as sharing a document with a contact; just use the “Share document with chat room” button. Contacts joining the chatroom later on will be offered to join the collaborative session, too.

Note that due to the effects described in Section 5.1, when the user who initiated the session disconnects, all users will be disconnected.

6.3 Opening an existing document from a Telepathy-initiated connection

When you accidentially or purposefully close a document on a connection initiated as described in Sections 6.1 and 6.2, the connection is not closed; instead, it stays alive in the background and you can re-open the document using the "Open document" dialog (Figure 4): the widget in the lower part of the dialog should list the connection to the contact, and you can click it to get an ordinary "Open file" dialog listing all files in the connection.

6.4 Adding a document to an existing connection

Similar to what was described in the previous section, you can add a document to a connection initiated as described in Sections 6.1 and 6.2 by using the "Share document" dialog (Figure 3). Other users in the connection will be notified of this and are given opportunity to join the session immediately, or at any later time using the technique described in the previous section.

6.5 Technical details of establishing a connection using Telepathy ※

Telepathy uses a technology called *Tubes* to establish bytestream connections between two arbitrary instant messenger contacts. This is currently only implemented for Jabber/XMPP, though. Section 5.1 lists the possibilities of how such a connection can work internally.

For the user of the API, in this case kte-collaborative, Telepathy offers a DBus interface to request such *Channels* (a Tube is a special Channel). The details of Telepathy Channel dispatching are quite complicated, but a short outline of how it works in kte-collaborative shall be given here:

- Some application, for example Kate or also ktp-text-ui (kde-telepathy's chat window application), requests a new Tube channel with type "infinote". In the parameters of the channel requests, it specifies which documents should be opened initially and whether an editor needs to be launched on the initiating side. The latter is set to true for requesting channels from ktp-text-ui, and to false for requesting channels from e. g. Kate.
- First, this request is handled by an application called `infinoteservertubehandler` *on the same computer*. If not running, the application is launched via DBus activation. It takes care of starting the infinote server (see also Section 7.5), then tells Telepathy to proceed sending the request to the other person.
- The channel request is received by the other person; an application called `infinoteclienttubehandler` (without the dash) is started, and takes care of launching the editor on the client's computer as soon as Telepathy indicates that the connection was built successfully.
- Closing an editor window has no effect whatsoever on the connection. A con-

nection is closed when your Jabber account disconnects, or when one of the tube handler applications is terminated for some reason.

For debugging purposes, you can start the client- and server tube handler applications (they are located in your `libexec/` directory) by hand.

6.6 Technical details of the notifications ※

There's a separate daemon called `infinotenotifier` which is launched and watches all connections for documents being added (the infinote protocol provides push-notifications for this purpose, so just watching a connection consumes no bandwidth). When it detects that a document was added to an existing connection, it displays a notification to the user. You can disable those – as always – in KDE's systemsettings program, under Notifications.

7 Using a classical server to share a document

A *classical server* is defined here as a server (as in physical machine) standing somewhere running an infinote server, whilst having a public IP address and probably a domain name. An example for such a classical server is `gobby.0x539.de`.

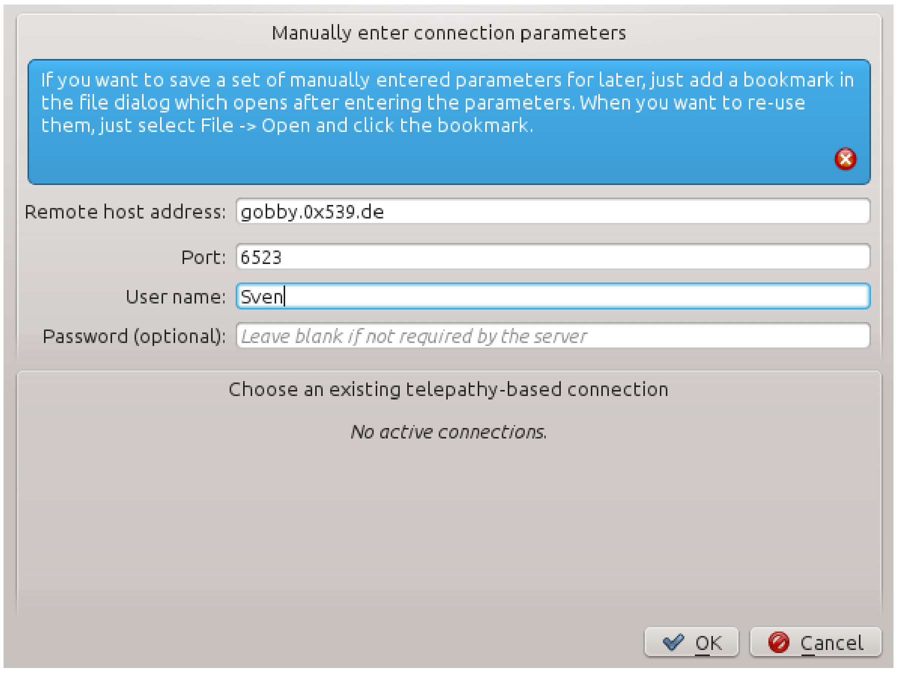
The obvious disadvantage of such a server is that you need a centralized place where this server is located; thus you either need to run it yourself, or all your documents will most likely be public and editable by everyone. The former most likely requires you to rent a server somewhere, which isn't free, and also requires some setup effort. Still, there are cases where centralized servers are useful; especially when you want a set of documents to be available over a longer period of time.

7.1 Opening an existing document from a classical server

The "Open collaborative document" dialog, as described in Section 4.6, can be used to conveniently connect to a classical server (see Figure 8). In the normal case, just enter the server host name in the "Remote host address" field and click Ok. You will be presented a list with documents on the server; click one to join it.

7.2 Adding a document to a classical server

To upload a document to an existing server, you can use the "Share document" dialog in much the same way (Figure 9): Click the "Upload document to an existing server" button, enter the server parameters, and select a file name. Your editor will automatically join the newly created document.


Manually enter connection parameters

If you want to save a set of manually entered parameters for later, just add a bookmark in the file dialog which opens after entering the parameters. When you want to re-use them, just select File -> Open and click the bookmark.

Remote host address: gobby.0x539.de

Port: 6523

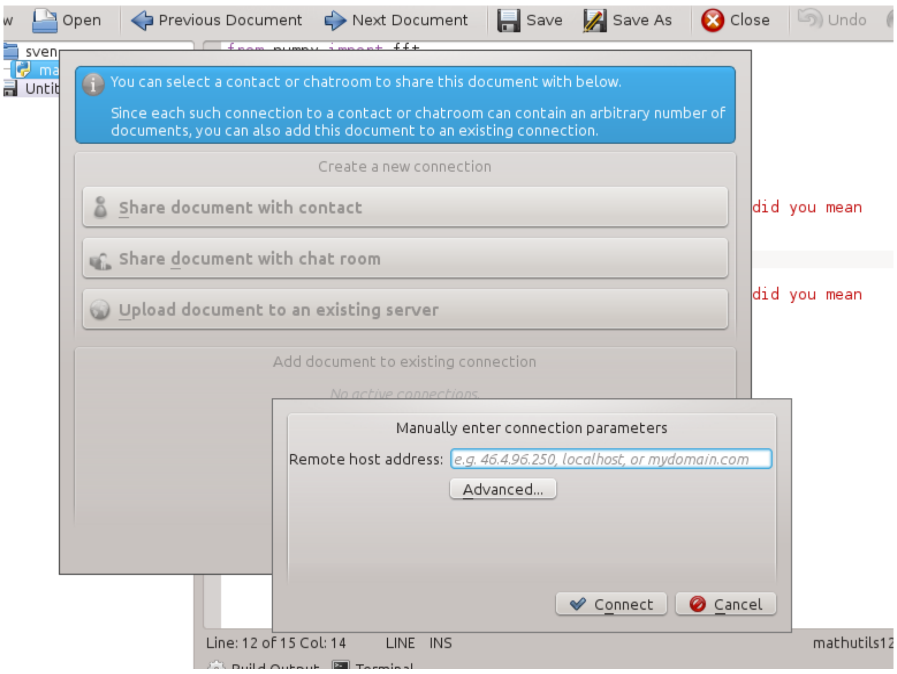
User name: Sven

Password (optional): *Leave blank if not required by the server*

Choose an existing telepathy-based connection

No active connections.

Figure 8: The "Open collaborative document" dialog with manually filled out connection parameters


You can select a contact or chatroom to share this document with below.

Since each such connection to a contact or chatroom can contain an arbitrary number of documents, you can also add this document to an existing connection.

Create a new connection

Add document to existing connection

No active connections

Manually enter connection parameters

Remote host address: *e.g. 46.4.96.250, localhost, or mydomain.com*

Line: 12 of 15 Col: 14 LINE INS

Figure 9: The "Share document" dialog after clicking the "Upload document to an existing server" button

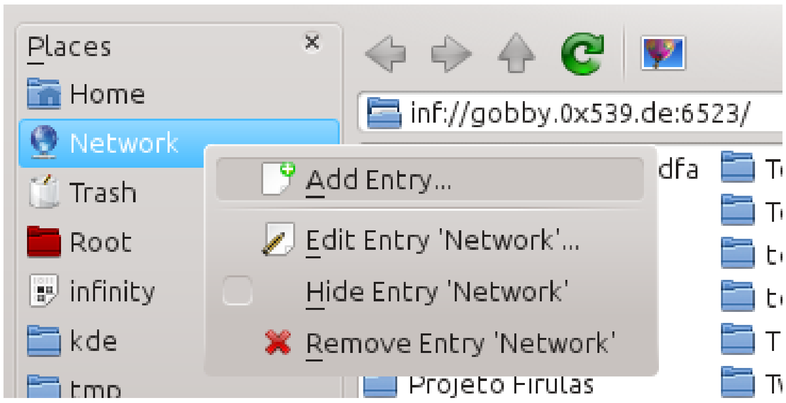


Figure 10: To save a set of parameters, just open a dialog which shows the server's files as described in sections 7.1 and 7.2, and add a bookmark.

7.3 Re-using parameters for classical servers

In case you frequently use the same server, you do not have to enter the server address each time you want to load or create a document there. The dialog described in sections 7.1 and 7.2 are basically just a user interface to create an URL of the form

```
inf://user:password@servername:port/
```

or, in a simpler case

```
inf://servername/
```

which then opens a file selection dialog for that base URL. Since those are just perfectly normal URLs, you can create a bookmark for them as usual, and use that bookmark to refer to the server later. For this purpose, just use the "Add bookmark" function as usual, by right-clicking the bookmark bar in the file dialog (see Figure 10)

You can now just use *the normal Open and Save As actions* you'd use to open or save local files to open or create collaborative documents, by just clicking e. g. File → Open and then your bookmark. You do not need to use the (more verbose) Collaborative → … dialogs any more.

Note that this does currently not work for Telepathy-initiated connections. Those will have a different URL format every time you connect.

7.4 Technical details on `inf://` URLs ※

As you might have guessed, there's a KIO slave which handles the `inf://` protocol. Knowing this, you can do more fun things. For example, you can open collaborative documents from the command line:

```
kate inf://myusername@gobby.0x539.de/main.cpp
```

It gets even more powerful when you additionally use kioclient:

```
kioclient cp myfile.cpp inf://gobby.0x539.de/myfile.cpp
kioclient ls inf://gobby.0x539.de/test1/
```

The KTextEditor plugin will treat any document with an URL which starts with `inf://` as a collaborative document and will attempt to synchronize it. It will use the user name and password provided in the URL as your user name and password in the session. It does not care how you open that URL.

7.5 Starting an infinote server ※

The server executable is called *infinoted* or sometimes *infinoted-0.5* or *infinoted-0.6*. You can just run it to get an infinote server running on localhost, like this for example:

```
mkdir /tmp/document_root
infinoted --security-policy=no-tls -r /tmp/document_root
```

You can then use it to collaboratively edit documents with yourself:

```
kioclient cp test.cpp inf://localhost/test.cpp
kate inf://localhost/test.cpp
```

If you have a public IP or can control your router well enough, you can edit documents with people in this way.

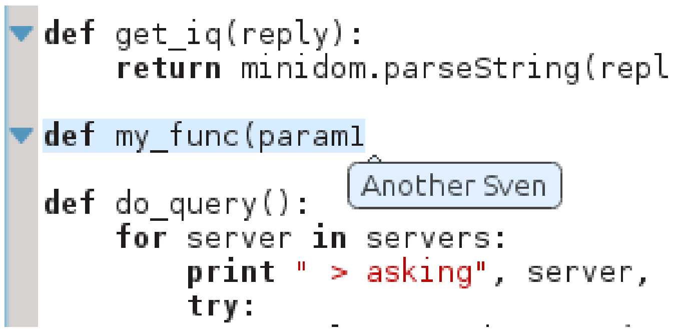
8 Configuration options

kte-collaborative offers a few configuration options to adjust the program to your needs. Figure 5 shows the settings dialog.

8.1 Changing the text colors

Changing the “Saturation” slider will make the colors used to highlight which user wrote what more or less intense (dragging it to the right means more intense colors). You need to re-join the session for this setting to take effect. If you don’t like the colored text background, you can also turn it off completely here.

A useful feature related to this can be found in the context menu of the view: right-click anywhere in the document and click “Clear user highlights” to clear all the background highlighting. That way, when new changes are made you can easily spot them.

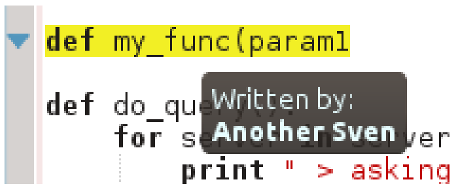


```
def get_iq(reply):
    return minidom.parseString(repl

def my_func(param1

def do_query():
    for server in servers:
        print " > asking", server,
        try:
```

Figure 11: Popup widgets telling which user is writing stuff



```
def my_func(param1

def do_qu
    for s          ver
        print " > asking
```

Figure 12: Tooltips tell which user wrote the text under the mouse

8.2 Changing the display of the popup widgets

By default, the plugin will draw small popup widgets when a user types text to indicate who is typing (see Figure 11). You can disable them by unticking the box.

8.3 Changing the display of the tooltips

There's an option to display tooltips which tell what user wrote what text when you move the mouse over that text (see Figure 12). Since it can easily be annoying while editing, you can turn it off by unticking the box. For this, too, you need to close and re-open the document (better restart the editor, since this setting is bound to the view which is not guaranteed to be recreated on re-opening the document).

8.4 Changing the default application for Telepathy-initated connections

You can also select the application you want to be started when initiating a connection from Telepathy here (this matters when a contact shares a document with you via Jabber or when you share a document via the button in the KTp chat window). There's various editors listed, including Gobby; you can also just open the folder in Dolphin and then proceed from there.

The first time you edit a collaborative document using a Telepathy-initiated connection, a dialog will ask you about your preferred editor application. The setting you pick there is the same as the one in the settings dialog.

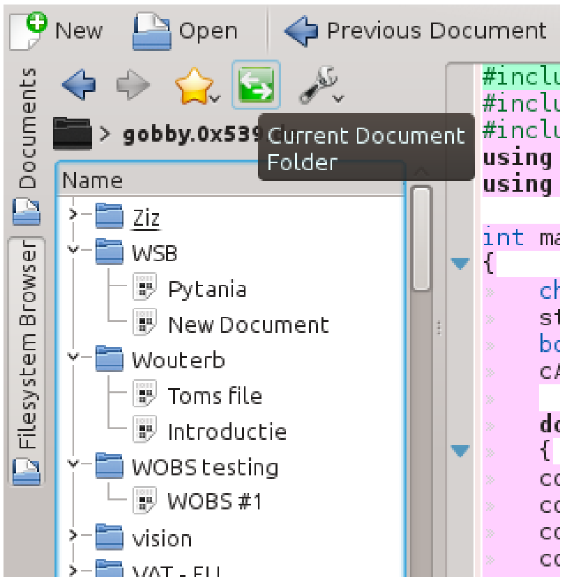


Figure 13: The “File system browser” sidebar, which is available in some applications such as Kate an KDevelop, can easily be used as a browser for collaborative documents.

If you have an application you’d like to be added to the list, let me know.

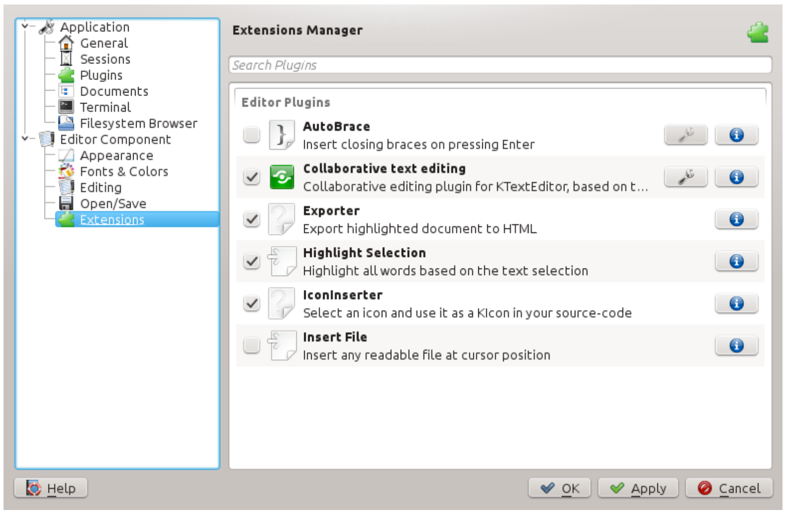
9 Useful tips

9.1 Document sidebar

One thing you probably want if you have more than one document on a server (especially when you’re working with a classic server) is a sidebar which displays all available documents. In Kate, to get this, first open a document from the server you want to work on. Then, go to Settings → Configure Kate → Plugins and enable “File system browser”. That should give a tab in the left sidebar which opens a file system browser. Click the “sync” icon (shown in Figure 13) to make the sidebar display the items from the server. You can also use the “Favorites” feature to make a bookmark for this server, to easily open it next time.

9.2 Sharing multiple files

This isn’t exactly new if you read the rest of the document, but not very obvious, so it might be worth stating: If you have one document on a server, you can always click Collaborative → Open file manager, and then just copy some files or even whole folders there. For example, you could copy the whole kte-collaborative repository tree there, to have it available for editing with others.



Editor Plugins

- ☐ **AutoBrace** Insert closing braces on pressing Enter
- ✓ **Collaborative text editing** Collaborative editing plugin for KTextEditor, based on t…
- ✓ **Exporter** Export highlighted document to HTML
- ✓ **Highlight Selection** Highlight all words based on the text selection
- ✓ **IconInserter** Select an icon and use it as a KIcon in your source-code
- ☐ **Insert File** Insert any readable file at cursor position

Figure 14: KTextEditor plugin settings module as it appears in Kate.

Be aware though that copying files to such a server is computionally expensive and will take some time. You'd better not try copying images or build files there. It'll probably work, but it will be slow and useless.

Sharing multiple files without opening a document ※ Knowing how the `inf://` URLs work, you can of course just open e. g. `inf://gobby.0x539.de/` in dolphin, create a folder there, and copy some files over.

10 Troubleshooting

Whatever happens, if something goes wrong after you changed anything related to your Telepathy or kte-collaborative installation (especially when you just installed the program), please restart your computer and try if it works after that. Services should automatically reload when their configuration changes, but many just don't do it correctly.

10.1 The "Collaborative" Menu isn't there

The collaborative editing plugin is not loaded. Look for the "Editor → Extensions" configuration module in your application (in Kate you can find it in "Settings → Configure Kate") and make sure the "Collaborative text editing" plugin is present and enabled, as shown in Figure 14.

10.2 Creating a Telepathy-based connection fails or breaks after a while ※

Shortly put, it's a bug, please let us know about it. It would be very very useful to have the following information:

- The text in the "Gabble" tab of the "ktp-debugger" program (type the latter in a console to get it) from *both* sides of the connection attempt
- The Jabber server(s) you were using.

Going a bit more into detail, this is a difficult topic, and there's a few known issues right now, especially these:

- Jabber.org seems to be quite broken for in-band bytestreams and sometimes loses packets. We can't do much about this.
- Randomly stream-initation fails with a "initiator does not have tube capabilities" error from gabble; reason is unknown, we're investigating. For the user, this shows as a "Connection refused" error box when sharing a document.
- Few servers provide a working proxy so often the not-too-great in-band bytestream solution is used as a fallback; we're working on getting a proxy list online.

Either way, when you encounter one of these errors, please let us know and include the information mentioned above – it's very helpful to get more data points about these errors.