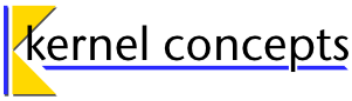
kernel concepts

evolution-kolab plug-in

End-user documentation

Project title	evolution-kolab plug-in
Author	Silvan Marco Fin, Christian Hilberg
Mailing list	-
Created	2011-06-07
Last modified	2011-07-13
Document name	evolution-kolab_user-manual.odt
Version	1.3
License	[CC BY-SA]

Table of contents



License

1 Introduction

evolution-kolab is a project to provide connectivity to *Kolab* groupware servers for *Evolution*. *Evolution* is the *GNOME* PIM[1] and groupware application that provides mail, calendaring and address-book functionality. [Evolution initiative] contains further information about *Evolution*. *Kolab* is a groupware solution for e-mails, appointments, contacts and more. Extensive information on *Kolab* is available from the [Kolab initiative].

1.1 Warning

Currently (Jun 2011), the *evolution-kolab* plug-in is **work in progress** and the plug-in **SHOULD NOT** be deployed in productive environments without carefully chosen safety measures.

1.2 Scope

This document covers installation and configuration of the *evolution-kolab* plug-in. A brief explanation of *evolution-kolab* specific issues is included where applicable. This document is not intended to be a user guide to *Evolution*. Please refer to the *Evolution* documentation, especially the [Evolution user guide] for general operation of *Evolution*. This document applies to *Ubuntu Linux 10.04 (LTS)* on x86 (32bit, 64bit). Furthermore, the updates from Launchpad PPA[2] from Jacob Zimmermann are required. [evo230] describes detailed installation instructions for the PPA.

1 PIM: Personal information management
2 PPA: Personal package archive

2 Installation

This section describes the installation of the *evolution-kolab* plug-in and the optional packages. For information about the different installation methods in *Ubuntu Linux 10.04 (LTS)* refer to [Ubuntu software].

2.1 Installation via package archive

2.1.1 Adding the necessary package repositories

2.1.1.a Using the software center to add the package repositories

Start the *Ubuntu* Software Center: Select *Applications → Ubuntu Software Center* from the *Ubuntu* Desktop. This will bring up the *Ubuntu* Software Center window. Select *Edit → Software Sources* from the applications main menu. A new window “Software Sources” will pop up. Switch to the “Other Software” tab. Press the **Add** button. An unnamed window will open. Into the **APT line** entry field, enter the following line:

```
deb http://evolution-kolab.sourceforge.net/ubuntu lucid main
```

Then confirm the input with the **Add Source** button.

If not already done so, add the Launchpad PPA for [evo230] at this point: Repeat the above step by pressing the **Add** button again and this time enter into the entry field:

```
ppa:jacob/evo230
```

Confirm the input with the **Add Source** button. **Close** the “Software Sources” window.

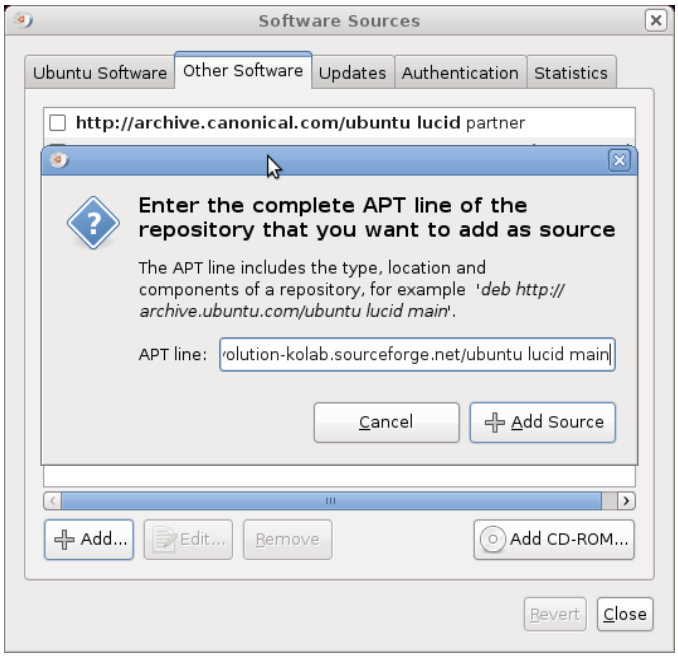


Figure 1: Adding an APT line to “Software Sources”

You will probably be asked to authenticate for privileges during the steps described. Enter your local user's password into the Password entry field and press the **Authenticate** button, if requested to authenticate with the **Authenticate** window. Exit the *Ubuntu* Software Center when done.

2.1.1.b Adding the package repositories via command line

This section describes an alternative way to add the required package sources to the Ubuntu package management. Start a terminal by selecting *Applications* → *Accessories* → *Terminal* from the *Ubuntu* desktop.

To add the package repositories needed for *evolution-kolab,* create a new empty file under `/etc/apt/sources.list.d/` (the file name can be chosen arbitrarily, but must end in `.list`, e.g. `evolution-kolab-evo230.list`) and add the following two lines to it:

```
deb http://evolution-kolab.sourceforge.net/ubuntu lucid main
deb http://ppa.launchpad.net/jacob/evo230/ubuntu lucid main
```

Finally issue the command
```
sudo apt-get update
```

This will update the package information on your system, including the repositories just added. Close the terminal window when done.

2.1.2 Installing the software

2.1.2.a Using the software center to install the software

Start the *Ubuntu* Software Center: Select *Applications* → *Ubuntu Software Center* from the *Ubuntu* Desktop. This will bring up the *Ubuntu* Software Center window.

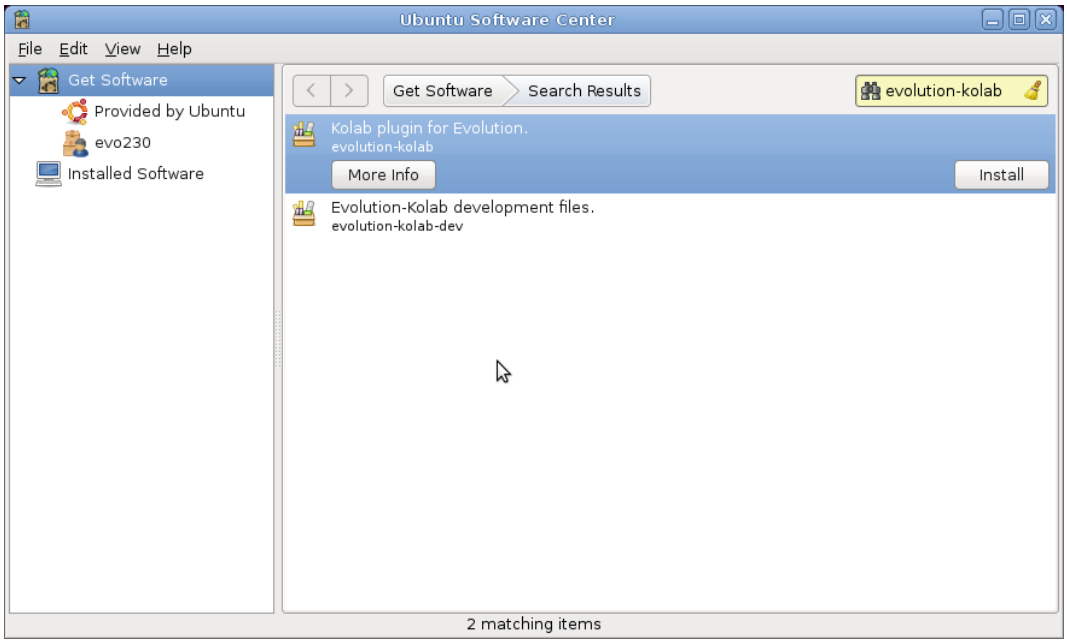


Figure 2: Installing evolution-kolab via the Ubuntu Software Center

Into the search field on the upper left corner, enter `evolution-kolab` as a search term and start the search for packages. From the search results, select the `evolution-kolab` entry. Optionally, select the `evolution-kolab-dev` package as well, if you are interested in installing API documentation files for the plug-in. Start the installation by pressing the **`Install`** button. You will be asked to authenticate; the procedure is the same as when adding the package repositories, described in section 2.1.1.a.

Since the *evolution-kolab* package repository does not yet provide full package validation infrastructure, the packages installed from there cannot be automatically validated. The following dialog will thus open:

Figure 3: Installing packages without signature check

Accept the installation of the untrusted *evolution-kolab* packages after understanding the potential security implications to your system, by pressing the **OK** button. The installation of the selected packages will now take place. Exit the *Ubuntu* Software Center when done.

2.1.2.b Installing the software per command line

Start a terminal by selecting *Applications* → *Accessories* → *Terminal* from the *Ubuntu* desktop.

To install the evolution-kolab packages, type

```
sudo apt-get install evolution-kolab
```

at the command line prompt of the terminal and press **`Enter`**. The terminal should show some output of the **`apt-get`** command, containing among other things:

```
The following extra packages will be installed:
  libcurl3-nss libedata-cal1.2-7 libevolution
The following NEW packages will be installed:
  evolution-kolab libcurl3-nss libedata-cal1.2-7 libevolution
```

and you will be asked to continue as before. Again, press **`Y`** and **`Enter`**. Eventually you will get to see a warning like:

```
WARNING: The following packages cannot be authenticated!
  libcurl3-nss evolution-kolab
Install these packages without verification [y/N]?
```

Press **`Y`** and **`Enter`**. The packages will be downloaded and installed. In case *Evolution* was not installed before, it (and its package dependencies) will be downloaded from their respective repositories and installed alongside *evolution-kolab*. If *Evolution* was installed before, it will be

upgraded to the version from [evo230]. Close the Terminal window when done.

2.2 Alternative installation

If there are problems with the installation via package archive, there is an alternative installation method for the e*volution-kolab* plug-in, which is documented in this section.

2.2.1 Getting the software

Locate the right packages in the e*volution-kolab* package repository:
http://evolution-kolab.sourceforge.net/ubuntu/dists/lucid/main/

The download page lists the following files in the `binary-i386` directory (as of `evolution-kolab_0.2.0`; exact version numbers may differ for newer releases):

Required files (32bit):

```
evolution-kolab_0.2.0_i386.deb
libcurl3-nss_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
libcurl3_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
```

Optional files (32bit):

```
curl_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
evolution-kolab-dev_0.2.0_i386.deb
libcurl3-dbg_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
libcurl3-gnutls_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
libcurl4-gnutls-dev_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
libcurl4-nss-dev_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
libcurl4-openssl-dev_7.19.7-1ubuntu2~6.gbpaa10b9_i386.deb
```

If you are working on a 64bit version of *Ubuntu Linux*, use the files in the `binary-amd64` directory (as of `evolution-kolab_0.2.0`; exact version numbers may differ for newer releases):

Required files (64bit):

```
evolution-kolab_0.2.0_amd64.deb
libcurl3-nss_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
libcurl3_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
```

Optional files (64bit):

```
curl_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
evolution-kolab-dev_0.2.0_amd64.deb
libcurl3-dbg_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
libcurl3-gnutls_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
libcurl4-gnutls-dev_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
libcurl4-nss-dev_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
libcurl4-openssl-dev_7.19.7-1ubuntu2~6.gbpaa10b9_amd64.deb
```

2.2.2 Preparation

Create a new directory (e.g. `/usr/src/evolution-kolab/`) and download the necessary files for your architecture (see section 2.1.1) into this directory.

2.2.3 Installation

Start a terminal by selecting *Applications* → *Accessories* → *Terminal* from the *Ubuntu* desktop. Change to the directory containing the downloaded package files. To install the packages, type:

```
sudo dpkg -i *.deb
```

Eventually you will be prompted for your password.

2.3 Building and installing evolution-kolab binaries manually

Instructions for compiling *evolution-kolab* from its source and installing the binaries manually are provided in the installation notes in the *evolution-kolab* GIT SCM-repository, for *Ubuntu 10.04 "Lucid Lynx"* as well as *Debian 6.0 "Squeeze"*.

3 Configuration

Evolution handles a user's e-mail, contacts and three different types of calendar-related data: calendars, memos and tasks. The e*volution-kolab* plug-in features a special IMAP provider, named “kolab2”, which is a modified version of the “IMAP+” provider (also known as IMAPX). It may be used in *Evolution* to access user's mail folders on the *Kolab* groupware server. The “kolab2” IMAP provider may be used with any IMAP server that complies with the IMAPv4 protocol, with the restriction of not being able to use the *Kolab* specific extensions of the *Kolab* IMAP provider in a non-*Kolab* environment.

For *Kolab* address-book access, e*volution-kolab* provides the ”kolab2” address-book type. Additionally, access to calendars, tasks and memos is provided by the ”kolab2” calendar type.

The following sections describe the steps to setup the different types in *Evolution*.

Important note:
The “kolab2” IMAP provider (this is the Evolution term for a software module that connects to an IMAP server and transports data via an IMAP connection) implements the ANNOTATEMORE draft IMAP extension. These IMAP annotations are used by *Kolab* to maintain folder type information on IMAP folders. For details, see the *Kolab Format Specification*, available from the [Kolab initiative]. The folder types on the *Kolab* server need to be set according to their respective contents (e-mail and PIM data). The *Evolution Data Server* PIM back-ends provided by *evolution-kolab* will ignore any IMAP folder which is not of a *Kolab* PIM data type. You will thus not be able to work with address-book- or calendar-type folders unless they are properly annotated on the IMAP server. Since *evolution-kolab* does presently not support creating IMAP annotations via the *Evolution* front-end (GUI), use another *Kolab* client, e.g. the *Horde* web-client (which is part of a typical *Kolab* installation), in order to create properly annotated PIM IMAP folders on the *Kolab* groupware server. Once a PIM folder is created on the *Kolab* server with the proper IMAP folder type annotation, it can be configured and used in *Evolution*.
Working with *Kolab* e-mail folders from *Evolution* is straight forward, no special steps need to be taken.

3.1 E-mail handling

3.1.1 Initial setup

If *Evolution* is started for the first time, the “Evolution Setup Assistant” dialog window will open and guide you through the setup for an e-mail account. Pressing **`Cancel`** at any time will terminate *Evolution* during the initial setup and the entered data will be lost. *Evolution* denies operation if no e-mail account is configured and will present the “Evolution Setup Assistant” upon startup for as long as no e-mail account is set up.

3.1.2 Setting up a *Kolab* e-mail account

3.1.2.a Configuration pages for receiving e-mail

Step **`Forward`** through the “Welcome” screen and the “Restore from backup” screen. Do not select **`[ ] Restore Evolution from the backup file`** on the latter. The next screen is the “Identity” setup page. Enter your personal data as instructed by the setup screen. Required information are your full name and your *Kolab* e-mail address. Optionally you can specify a reply

address and an organization name. The setting **[ ] Make this my default account** has no impact on the functionality of your *Kolab* e-mail account and may be chosen as desired. Continue by pressing the **Forward** button.

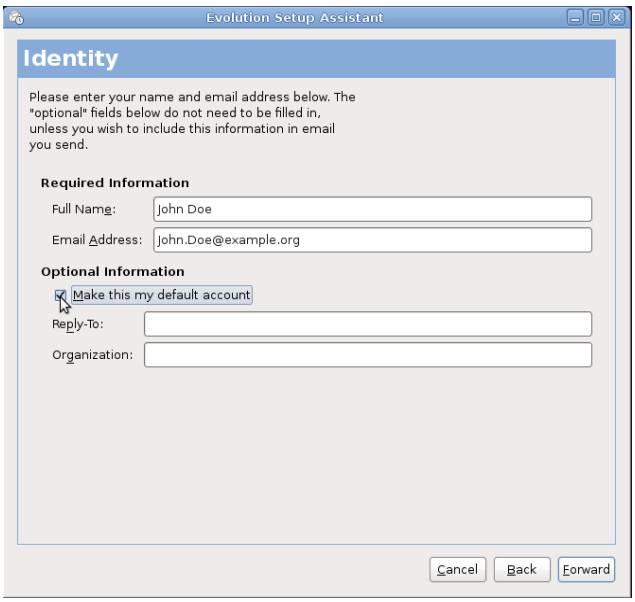


Figure 4: Evolution Setup Assistant: "Identity"

Use the "Receiving Email" page to specify basic account settings:

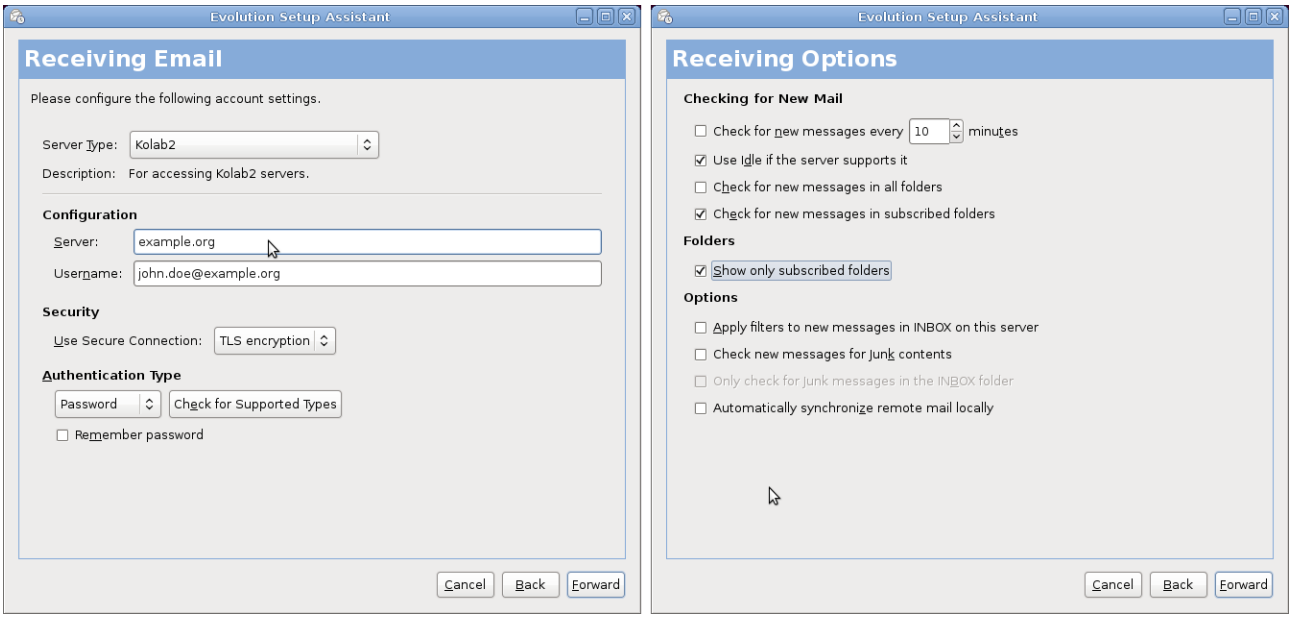


Figure 5: Evolution Setup Assistant: "Receiving Email" and "Receiving Options"

Select "kolab2" from the **Server Type** drop down field. Enter your *Kolab* groupware server's host-name into the **Server** entry field. Enter your *Kolab* user-name into the **Username** entry field. The *Kolab* user-name usually resembles the *Kolab* e-mail address. Select the connection security setting for your connection type from the **Use secure connection** drop down field:

- No encryption: Do not encrypt the connection, data and credentials are transmitted in plain
- TLS encryption: Use the **STARTTLS** protocol after the initial connection to encrypt the connection. This method uses the TLS version 1.0.
- SSL encryption: Connect to the server on a special port, indicating that a secure connection is desired. This method uses SSL version 3.0. Both, TLS encryption and SSL encryption, require a properly configured set of certificates. For information on client certificate environments refer to section 3.4.1. Select **Password** from the drop down field in the "Authentication Type" paragraph. If the *Kolab* groupware server uses a different authentication type, select the method denoted by your administrator.

The button **Check for supported types** will try to look up the supported types from the server. The unsupported methods will be marked as struck out in the authentication type drop down list. If you wish the password to be remembered, check **[ ] Remember password**.

Press **Forward** to continue to the "Receiving options" page.

"Checking for New Mail"

- **[ ] Check for new messages every X minutes**: If checked, *Evolution* will check for new messages after the time interval specified in the spin button.
- **[ ] Use idle if the server supports it**: If checked, *Evolution* will receive status updates from the server without having to poll for them.
- **[ ] Check for new messages in all folders**: Look for new mail in all folders. Especially useful if you decide to set up automated filters or if the server filters incoming mail into different folders.
- **[ ] Check for new messages in subscribed folders**: Look for new mail in the subscribed folders. Relevant only if the next option is unchecked.

"Folders"

- **[ ] Show only subscribed folders**: *Evolution* will display only the subscribed folders, if checked. Otherwise all folders on the server will be displayed.

"Options"

- **[ ] Apply filters to new messages in INBOX on this server**: Select this option if you want to apply filters to incoming messages.
- **[ ] Check new messages for junk contents**: If checked, *Evolution* will check new messages for junk content. For additional information refer to [Evolution-Spam].
- **[ ] Only check for Junk messages in the INBOX folder**: If checked, *Evolution* will restrict the Spam checks to the INBOX folder.
- **[ ] Automatically synchronize remote mail locally**: From the *Evolution* (2.30) user guide:

  Evolution, by default, downloads only the header information such as From, Subject, and Date. The body of the message and the attachments are downloaded only when the message is clicked on, saving time and network usage. This is useful if you do not read all of the messages you receive.

  If this option is checked, *Evolution* fetches the headers as well as the body of the

message simultaneously. In this case, the time taken to open a message is comparatively less. In addition, all mail is available for offline reading, if this option is checked.

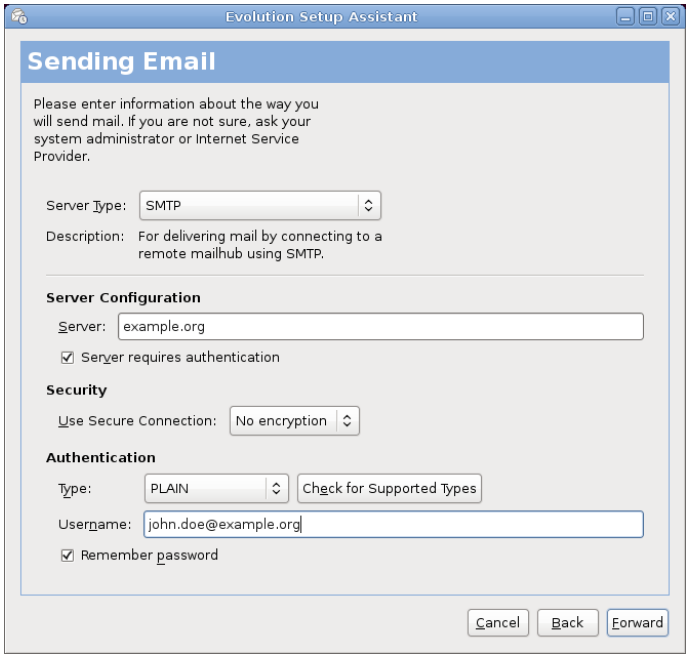
3.1.2.b Configuration pages for sending e-mail

Proceed to the next page of the assistant with the **Forward** button. Next, configure the options for sending e-mail in *Evolution*. Select **SMTP** from the **Server Type** drop down field. Specify your *Kolab* groupware server's host-name in the **Server** entry field. Check the **[X] Server requires authentication** check box.

The *Kolab* user-name usually resembles the *Kolab* e-mail address. Select the connection security setting for your connection type from the **Use secure connection** drop down field. The details for configuring security options from **Use secure connections** in the section 3.1.2.a apply.

Select **Plain** from the **Type** drop down list, if not announced otherwise for the *Kolab* groupware server. The button **Check for supported types** will try to look up the supported types and strike through the unsupported ones in the **Type** drop down list. Use your *Kolab* user-name (*Kolab* account e-mail address) as user-name. If desired, check the **[ ] Remember password** check box.

Proceed to the “Account Management” page by pressing the **Forward** button. On this page, enter an arbitrary identifier into the **Name** for the e-mail account just entered. Continue with the **Forward** button to the “Done” page. Click **Apply** to save the settings.



3.1.2.c Account Editor

If you want to modify details of an existing e-mail account or delete it, select *Edit → Preferences* from the main menu. In the “Evolution Preferences”, select the “Mail Accounts” icon on the left side

of the dialog. This shows the e-mail account view in the main viewing area. Select the *Kolab* account you want to deal with.

- If the selected account is to be deleted, press the **Delete** button. This will bring up a dialog window, asking for confirmation of the removal. Pressing **Delete** will remove the account settings from *Evolution*. Pressing **Do not delete** will abort the removal.
- If instead the settings are to be modified, press the **Edit** button. This will bring up the “Account Editor” window for the selected account. The “Account Editor” lists six different tabs. The tabs *Identify*, *Receiving Email*, *Receiving options* and *Sending Email* correspond to the pages of the same name as described in section 3.1.2. The tabs *Defaults* and *Security* contain general options regarding e-mail accounts. For more information on these option tabs refer to [Evolution-Accounts].
  Press the **Cancel** button at the bottom if you want to close the “Account editor” window without saving any changes made to the settings. If you want to accept the changes just made, press the **OK** button.

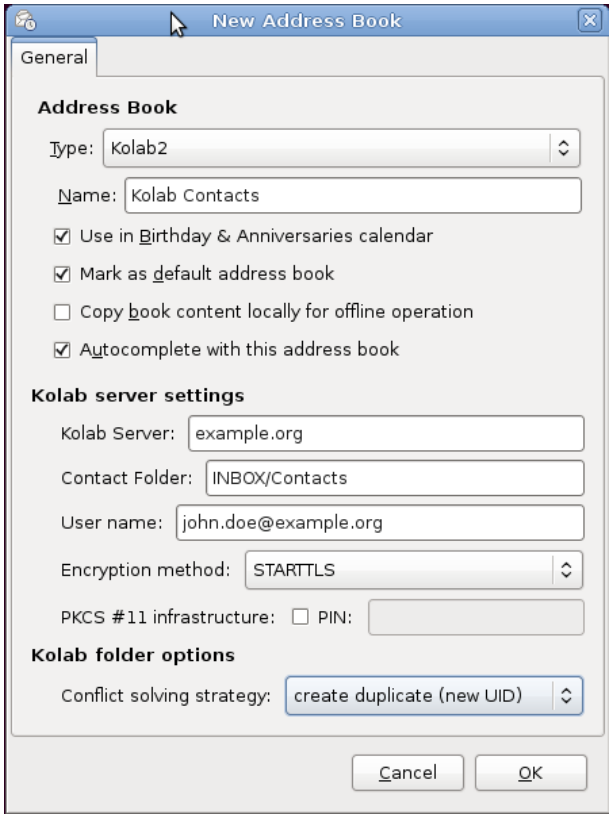
Close the “Evolution Preferences” window via the **Close** button when done in this dialog.

3.2 *Address-book type*

All steps in this section are performed in the address-book view. To switch to the address-book view of *Evolution,* press the **`Contacts`** button.

3.2.1 Adding a *Kolab* address-book

Select *File* → *New* → *Addressbook* from the main menu. A dialog window “New Address Book” will open. From the **`Type`** drop down field, select “kolab2”. Enter an arbitrary identifier for the *Kolab* address-book into the **`Name`** entry field.



3.2.1.a Generic Options

- **`[ ] Use in Birthday & Anniversaries calendar`**: Available birthday and anniversaries events from this address-book's contact entries are made available to the calendar view for display.
- **`[ ] Mark as default addressbook`**: Self explanatory.
- **`[ ] Copy book content locally for offline operation`**: This setting has no effect at all for the “kolab2” provider. Furthermore, *Kolab* address-books provide for this automatically.
- **`[ ] Autocomplete with this addressbook`**: While entering e.g. e-mail addresses into the recipients field or in the attendee list of a meeting event, this address-book will be

searched for matching entries and matches will be offered for completion.

3.2.1.b Kolab server settings

- **`Kolab Server`**: Enter the host-name of the *Kolab* groupware server into this entry field.
- **`Contact Folder:`** Enter the complete IMAP folder name of the desired contact folder. Ask your administrator for the correct path to the folder if unsure. **Note**: The folder path needs to be specified starting from the IMAP root, with leading slashes ("/") omitted. In many installations, the PIM data folders will be located under a top level INBOX directory, so the path to specify here may look like `INBOX/Contacts` for a contacts folder. The part of the path following `INBOX/` may be language specific, depending on which client initially created the folders.
- **`User name`**: Enter your *Kolab* user-name into this entry field.
- **`Encryption method`**: Select the desired connection security method from this drop down field. The available options are: **`No encryption`**, **`SSL encryption`**, and **`STARTTLS`**. The details for configuring security options from **`Use secure connections`** in section 3.1.2.a apply.
- **`PKCS #11 infrastructure`**: Access private key data for certificates from a PKCS #11 device, like a smart card, a TPM or any compatible device. This is usually used in conjunction with client certificates to authenticate the client to the server. See section 3.4.1 for details on deployment of client certificates and usage in case of *Evolution*.
- **`PIN`**: Heed the warning!

WARNING This is insecure and PINs are not intended to be used like this. This part of the *evolution-kolab* plug-in demonstrates solely the possibility of PKCS #11 deployment and is by not meant to be used in critical environments.

Enter the PIN to access your PKCS #11 device here. Mind that the PIN will be saved in plain text altogether with the rest of the information to access this address-book. Due to limitations in the architecture of *Evolution 2.30*, there is no sane way of getting the PIN from the *Evolution* front-end (GUI) to the back-ends (*EDS*) which handle PIM data and server connections.

3.2.1.c Kolab folder options

- **`Conflict solving strategy`**: Select a strategy to counter the effects of a conflict. See section 3.4.2 for an explanation of conflicts and the options. The available options are **`Take Newer (last modified)`**, **`Take Remote (server-side)`**, **`Take Local (client-side)`** and **`Take Both (resulting in two different, parallel entries)`**. The strategy which is configured for a certain folder will be used to solve any synchronization conflicts which may occur after an *Evolution* offline session when reconnecting to the *Kolab* server. No further user interaction will occur.

3.2.2 Modifying a *Kolab* address-book

From the address-book list, select the *Kolab* address-book you wish to modify. Then select *Actions*

→ *Address Book Properties* from main menu. This will bring up the “Address Book properties” dialog. It resembles the “New Address Book” dialog window from section 3.2.1, except that there is no selector for the address-book type. The server settings are inactive, the options in this section can not be changed on an existing address-book. If you want to access the *Kolab* address-book with one of these options changed, the address-book configuration needs to be deleted from the list of address-books and a new one must be created. This is a limitation of the underlying library and cannot be circumvented at the moment. However, if the newly added address-book uses the same IMAP folder path as the previously deleted one, the PIM data contained in that folder will be found and used by the new address-book entry, so it does not need to be downloaded from the server again.
Changing the **`Conflict solving strategy`** becomes effective only after *Evolution* is restarted.

3.2.3 Removing a *Kolab* address-book from the view

From the address-book list, select the *Kolab* address-book you wish to delete. Then select *Edit* → *Delete Address Book*. An unnamed dialog window will ask for approval. If you press the **`Delete`** button, the address-book will be removed from your configuration. No data will be deleted on the *Kolab* groupware server by this action.

3.3 Calendar/tasks/memos type

Calendar, tasks and memos are handled by the same piece of software. In case of *Kolab*, that also means that the configuration dialog is the same. This documentation describes the configuration steps for calendars. Tasks and memos are configured just in the same way.

The configuration instructions in this section apply to the calendar view (respectively tasks or memos). To switch to the appropriate view, press the button labeled **`Calendar`** (or **`Tasks`** or **`Memos`**, respectively).

3.3.1 Adding a new calendar-, task- or memo-view

Select *File* → *New* → *Calendar* from the main menu. This will bring up the “New Calendar” dialog window. From the **`Type`** drop down field, select “kolab2”. Enter an arbitrary identifier for your *Kolab* calendar into the **`Name`** entry field.

3.3.1.a General settings

By pressing on the colored **`Color`** button, a color which will be used as event background color can be selected.

- **`[ ] Copy calendar contents locally for offline operation`**: This setting has no effect at all for the “kolab2” provider. Furthermore, *Kolab* calendars provide for this automatically.
- **`[ ] Mark as default calendar`**: Self explanatory.

3.3.1.b Kolab server settings

- **`Kolab Server`**: Enter the host-name of your *Kolab* groupware server into this entry field.
- **`Calendar Folder:`** Enter the complete IMAP folder name of the desired calendar folder.

Ask your administrator for the correct path to the folder if unsure. **Note**: The folder path needs to be specified starting from the IMAP root, with leading slashes (“/”) omitted. In many installations, the PIM data folders will be located under a top level INBOX directory, so the path to specify here may look like `INBOX/Calendar` for a calendar folder. The part of the path following `INBOX` may be language specific, depending on which client initially created the folders.

- **`User name`**: Enter your *Kolab* user-name into this entry field.
- **`Encryption method`**: Select the desired connection security method from this drop down field. The available options are: **`No encryption`**, **`SSL encryption`**, and **`STARTTLS`**. The details for configuring security options from **`Use secure connections`** in section 3.1.2.a apply.
- **`PKCS #11 infrastructure`**: Access private key data for certificates from a PKCS #11 device, like a smartcard, a TPM or any compatible device. This is usually used in conjunction with client certificates to authenticate the client to the server. See section 3.4.1 for details on deployment of client certificates and usage in case of *Evolution*.
- **`PIN`**: Heed the warning!

WARNING This is insecure and PINs are not intended to be used like this. This part of the *evolution-kolab* plug-in demonstrates solely the possibility of PKCS #11 deployment and is by not meant to be used in productive environment.

Enter the PIN to access your PKCS #11 device here. Mind that the PIN will be saved in plain text altogether with the rest of the information to access this address-book. Due to limitations in the architecture of *Evolution*, there is no sane way of getting the PIN from *Evolution's* front-end (GUI) to the back-ends which handle the PIM data and server connections.

3.3.1.c Kolab folder options

- **`Conflict solving strategy`**: Select a strategy to counter the effects of a conflict. See section 3.4.2 for an explanation of conflicts and the options. The available options are **`Take Newer (last modified)`**, **`Take Remote (server-side)`**, **`Take Local (client-side)`** and **`Take Both (resulting in two different, parallel entries)`**. The strategy which is configured for a certain folder will be used to solve any synchronization conflicts which may occur after an *Evolution* offline session when reconnecting to the *Kolab* server. No further user interaction will occur.

3.3.2 Modifying a *Kolab* calendar

Click on the calendar you wish to modify, from the list of calendars. Then right click on the calendar to bring up the context menu for the *Kolab* calendar. Select *Properties* from the context menu. The “Calendar properties” dialog window for this calendar will open. It resembles the “New Calendar” dialog window from section 3.3.1, except the **`Type`** drop down field. You may not modify the *Kolab* server settings of an existing calendar. If you need to change data on these fields, you have to delete the calendar from the list of calendars and add it again as a new calendar. This is a limitation of the underlying library and can not be circumvented at the moment. However, if the newly added calendar uses the same IMAP folder path as the previously deleted one, the PIM data contained in that folder will be found and used by the new calendar entry, so it does not need to be downloaded from the server again.

Changing the **`Conflict solving strategy`** becomes effective only after *Evolution* is restarted.

3.3.3 Removing a calendar from the view

Select a calendar and right click it in the list. This will bring up the context menu for the *Kolab* calendar. From the context menu select **`Delete`**. An unnamed confirmation dialog window will open and ask for approval. If you press **`Delete`**, the *Kolab* calendar will be removed from *Evolution's* calendar view. No data will be deleted on the *Kolab* groupware server by this action.

3.4 Additional information

3.4.1 Client certificates

Utilizing client certificates to authenticate the *Evolution* client against the *Kolab* server services is possible to a large extent. However, there are some issues involved. The document [Evo-Kolab-PKCS11] is available from the project download directory along with this document and the project sources. It describes the configuration details for the *Kolab* groupware server. Additionally, the steps to set up a PKCS #11 device and to configure *Evolution* to be able to use such a PKCS #11 device are explained in chapter “3 Accessing *Kolab* services using client certificates” of that document. Please refer to [Evo-Kolab-PKCS11] for information about getting client certificates to work with *Evolution* and the *evolution-kolab* plug-in.

The PIM part of *Evolution* (contacts, calendar, memos, tasks) rely on the same infrastructure as the e-mail part, thus, if everything is set up according to [Evo-Kolab-PKCS11] (especially chapter 3), you should be able to use the PKCS #11 infrastructure as described in sections 3.2.1.b and 3.3.1.b for the PIM part. **Heed the warnings for the PIN entry fields!**

Currently there is no way of requesting user input from the back-ends – refer to [EDS-Arch] for additional information on back-end architecture. It is a limitation of *Evolution* and EDS for the time being. Therefore, the current implementation of e*volution-kolab* plug-in offers the possibility to enter the PIN at configuration dialog level, which leads to the PIN being saved to a configuration entry on-disk. As of *Evolution 2.30*, the location is the *gconf* database, and the PIN will be stored in plain text (i.e., unencrypted).

3.4.2 Conflict solving strategies

When a user changes existing PIM data of a *Kolab* groupware folder, the e*volution-kolab* plug-in needs to write these changes to the *Kolab* server. A conflict is the state when two data sets derived from the same origin exist. This condition usually occurs if multiple write accesses to a *Kolab* PIM folder happen and at least one of the clients was in offline state when modifying the PIM data. (In online mode, the e*volution-kolab* plug-in checks for server side changes right before opening a PIM object for modification, minimizing the risk of running into a conflict situation.)

Example: Two employees, Abraham and Bebraham, use a shared calendar for special company events. Abraham utilizes the *Horde* web front-end to modify the data, whereas Bebraham uses *Evolution* and the e*volution-kolab* plug-in. Bebraham starts *Evolution*. The e*volution-kolab* plug-in retrieves all data from the *Kolab* groupware server and after that, Bebraham switches *Evolution* to offline operational mode. Abraham starts his browser and modifies today's existing event “fire drill”. Abraham changes start time from 10 am to 1 pm. He submits his changes. Meanwhile, Bebraham starts working on his appointments in offline mode. He opens the “fire

drill” event which is shown to him with the outdated start time 10 am. Since he is working offline, he does not see the current changes from Abraham. Bebraham now moves the “fire drill” from today to tomorrow and saves the changes. Then he switches to online state and the e*volution-kolab* plug-in tries to synchronize the changes to “fire drill” event with the data on the *Kolab* groupware server. At this point, a conflict occurs, since the original “fire drill” event was modified on two independent locations (independent, since Bebraham was working in offline mode).

The e*volution-kolab* plug-in is able to detect these conflicts. It is not possible to request user interaction during the synchronization of data. Due to this limitation, the desired action is selected at configuration time so that the e*volution-kolab* plug-in can act accordingly:

- **`Take Newer (last modified)`**: The e*volution-kolab* plug-in reads in the server's data and the client's data and decides according to the “last modification” time-stamp of the changes, which one will be used. If the server version is newer, the client's changes will be dropped. If the client's version is newer, the server's version will be overwritten. Keep in mind that this only works as expected if the system clocks of all clients involved in the changes are synchronized. Unexpected behavior will occur if this synchronization strategy is selected and the system clocks of the clients concurrently modifying PIM data are out-of-sync.
- **`Take Remote (server-side)`**: Drops the local (client) changes and uses the data from the server in all cases where a synchronization conflict occurs. If a PIM object has been changed on the client but not on the server, the modified client data is written to the server.
- **`Take Local (client-side)`**: Uses the PIM data the user entered in offline mode to overwrite the data on the server, regardless of any synchronization conflict situations. With this option, the own client data is considered authoritative.
- **`Take Both (resulting in two different, parallel entries)`**: A duplicate entry for conflicting data is created on the *Kolab* server in case of a synchronization conflict situation. The duplicate entry will carry an identity (UID) different from the original entry. No data is dropped. Regarding PIM object duplication, however, should a *Kolab* client mistakenly create a duplicate of an existing object and store it under the same UID as the original, only the first object seen by the plug-in is shown in *Evolution*. The user needs to interact manually via IMAP with the *Kolab* objects on the *Kolab* groupware server to recover from this error situation, e.g. by creating an “IMAP+” e-mail account in *Evolution* to connect to the *Kolab* server.

4 Using *evolution-kolab* with *Evolution*

For general information about using *Evolution* (2.30) please consult [Evolution user guide].

Special considerations about using *Evolution* and the e*volution-kolab* plug-in are explained in chapter 5 “Issues“.

Please be sure to understand the **“Important Note”** about *Kolab* folder types and IMAP annotations mentioned at the start of chapter 3 “Configuration”, page 10, for proper operation of *evolution-kolab.*

Note on synchronizing PIM data with the *Kolab* server: The synchronization button in the *Evolution* user interface toolbar, labelled **`Send/Receive`**[3], synchronizes e-mails *only*. It does so regardless of the *Evolution* view (e-mail, contacts, calendar, tasks, memos) currently active. No PIM data will be synchronized by pressing this button, only e-mails will be. In order to synchronize PIM data changes between *Evolution* and the *Kolab* server, please follow the instructions given in the usage sections for contacts and the calendar-type PIM objects.

4.1 Working with e-mails

For the “kolab2” IMAP account type to work well, you need to connect to a *Kolab* IMAP service. This service supports the IMAP folder annotations which do carry information about the different folder types *Kolab* supports and which are needed for proper operation of the “kolab2” IMAP account type. To connect to any other IMAP server, use the “IMAP+” or “IMAP” account type instead.

After successful configuration of an e-mail account of type “kolab2”, all *Kolab* IMAP folders which are not annotated to contain PIM data (this is, e-mail folders and such of unknown type) will be displayed in the e-mail view of *Evolution*. After initial setup and first connection to the *Kolab* groupware server, *Evolution* needs to be restarted once. The “kolab2” e-mail account can then be used just like an account of type “IMAP+” (this is the implementation on which “kolab2” is based) and behaves just like this account type, apart from not showing folders which are known to contain *Kolab* PIM data.

All IMAP folders created under the “kolab2” account type in the *Evolution* e-mail view will be annotated as e-mail folders on the *Kolab* server. Creation of nested folders is possible. It is currently not possible to create or manage IMAP folder annotations, , IMAP folder permissions or IMAP quotas in the *Evolution* front-end (GUI). If you need such functionality, you can resort to a different *Kolab* client (e.g. the *Kolab* web front-end) for these tasks, or contact the *evolution-kolab* project in order to support an implementation effort.

If a PIM data IMAP folder on the *Kolab* groupware server for some reason contains an e-mail folder or a folder of unknown type, the PIM data folder will be shown, but it will appear as though it did not contain any data. Its subfolder, however, which is of e-mail type or unknown type, will be selectable and can be used just like any other *Evolution* e-mail folder.

4.2 Working with contacts

To operate properly, currently the e*volution-kolab* plug-in must be run at least once in online state, connecting to a *Kolab* server with already set up (i.e. annotated) PIM folders.

3 The same action can be triggered by pressing the **F9** key or by selecting *File* → *Send/Receive* from the *Evolution* main menu.

Working with contacts on *Kolab* groupware servers in case of single user and client access to the contacts folders corresponds to normal operation as described in the [Evolution user guide], except that an Internet connection is required. The e*volution-kolab* plug-in synchronizes data in case of online state as soon as the data is submitted to the *Evolution* user interface. In case of offline state, data is saved in a local database, which is used to synchronize the local data with the server as soon as online state is activated again.

An environment where different users or clients access a shared *Kolab* groupware server folder works very similar to the single user environment described above. Special attention has to be paid to changes to server objects through other connected clients. Due to limitations in *Evolution 2.30*, the back-end can not be instructed to refresh the data by simple button or menu interaction. There are currently two reliable ways only of updating contact data in *Evolution*:

- restart *Evolution*
- enter a search expression in the contact search field or delete / modify the field, if it is not empty (the search may be cleared thereafter, as all contacts will now be updated, even those which were not results of the previously started search)

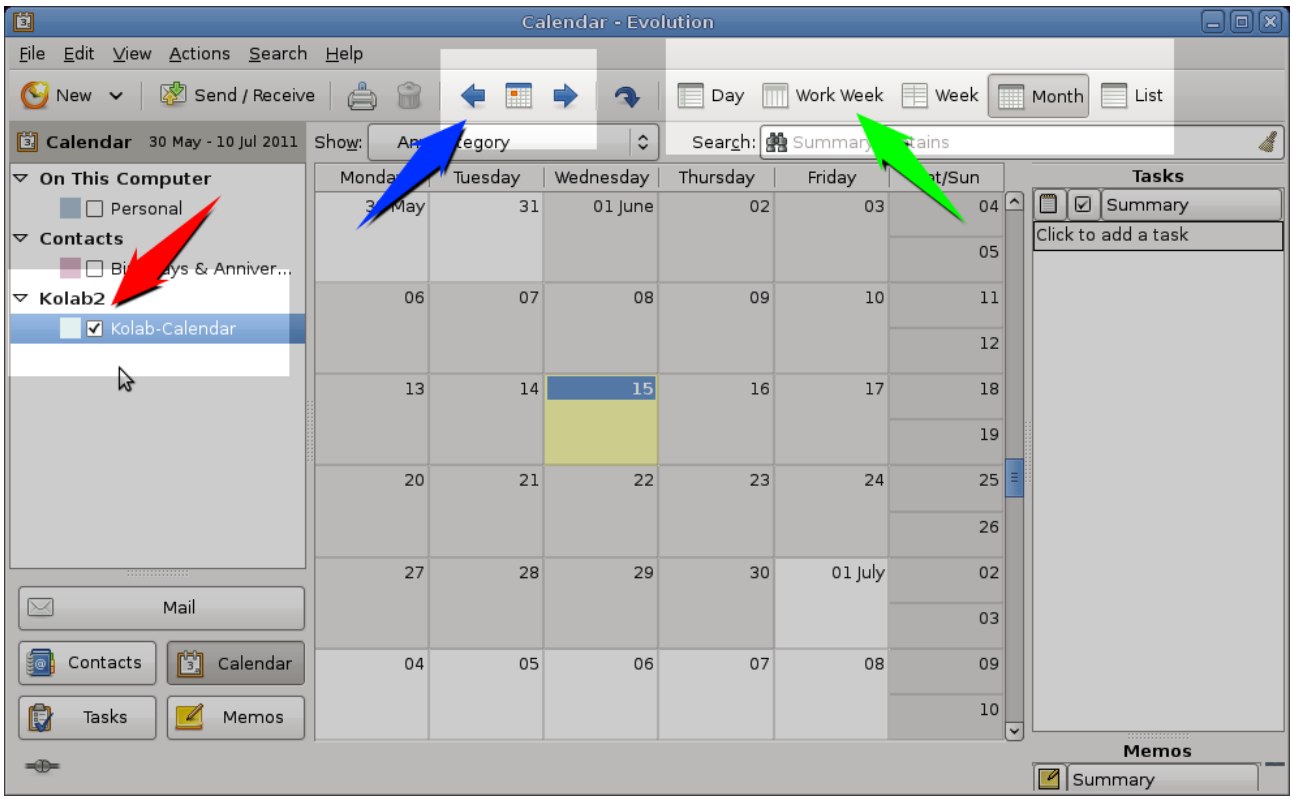
4.3 Working with calendars, memos and tasks

To operate properly, currently the e*volution-kolab* plug-in must be run at least once in online state, connecting to a *Kolab* server with already set up (i.e. annotated) PIM folders.

Calendars, memos and tasks may be used in the same way as described in the [Evolution user guide]. There is a restriction with calendars: the *Kolab* groupware server does not support detached reoccurring events. If a single occasion of a reoccurring event is modified, it will be stored as a standalone event and the link to the related (parent) event is lost.

Working with PIM types on shared *Kolab* groupware server folders (or if accessing a private folder with multiple *Kolab* clients concurrently) has an update problem. If PIM data on the *Kolab* groupware server was changed, the changes will not be propagated to the *Evolution* front-end, unless one of the following steps is performed:

- Deactivate and activate again the *Kolab* calendar in the calendar list (red arrow  in figure 8).
- Switch the calendar to another view (**`Day`**, **`Work week`**, **`Week`**, **`Month`** or **`List`**) (green arrow  in figure 8)
- Use the **`Go back`** and **`Go forward`** buttons from the toolbar to switch back and forth (blue arrow  in figure 8)


Calendar - Evolution
File Edit View Actions Search Help
New
Send / Receive
Day
Work Week
Week
Month
List
Calendar 30 May - 10 Jul 2011
Show:
Search:
On This Computer
Personal
Contacts
Kolab2
Kolab-Calendar
Tuesday
Wednesday
Thursday
Friday
Tasks
Summary
Click to add a task
Mail
Contacts
Calendar
Tasks
Memos
Memos
Summary

5 Issues

5.1 Limitations

Several limitations currently reduce the usability of *Evolution* in context of a *Kolab* groupware server environment.

- The architecture of *Evolution* and the *Evolution-Data-Server (EDS)* does not provide for a generic mechanism to request input from the user from the back-ends. Back-ends are the two processes that handle PIM data for *Evolution* and are referred to as *Evolution Data Server*. The user interface has to know which data is required for the next back-end action to take place and requests data from the user, before that action is performed. The required data is then transmitted via the existing API. The *Evolution* Data Server API has no infrastructure for e.g. PKCS #11 PIN input or conflict solving strategies. So this information has to be made available to the back-end by other means.
  In case of PKCS #11 deployment, this breaks the concept of the trusted environment in which a user supplies his secret PIN. In case of conflict resolution, the back-end can not ask for user interaction, if it detects a conflict. Instead, the strategy has to be selected at configuration time and the back-end handles any conflict accordingly.
- The implementation of *Libcamel* was modified for the e*volution-kolab* plug-in, but only to the extent to support *Kolab* specific extensions to the IMAP protocol. *Libcamel* itself was developed for the purpose of providing access to different e-mail storing entities, IMAP servers being one of these. Its intended utilization was in the *Evolution* program and it was never meant to be used elsewhere. The design of the *Evolution Data Server* made it necessary to use *Libcamel* not only outside of *Evolution*, but also multiple times within a distinct process in the back-end. To account for these limitations, some restrictions had to be made in the back-end implementation which utilizes *Libcamel*. Connections to the same IMAP service of a *Kolab* groupware server share some of their data structures within Libcamel. It is not possible to change the details of one of the connections, without disturbing the overall operation of the whole *Libcamel* in the back-end process. Therefore, modifications to credentials and the server details are prohibited at configuration level. Changing these details requires the existing connection to be removed from the configuration and set up anew.
- *Evolution 2.30* does not supply generic means to refresh a view. If data changes on the *Kolab* groupware server through access from a third party, *Evolution* does never recognize these changes, unless the user triggers an operation that will lead to a new request to the back-end.
  Because the user interface of *Evolution* does not provide any menu item or button to trigger a generic refresh event, the user needs to work around this and trigger a refresh by requesting the user interface to change the view of a calendar or enter search strings into contact search fields.

5.2 Evolution (2.30) Bugs

Table 1 lists known bugs which interfere with the operation of *Evolution* generally and with the operation of the e*volution-kolab* plug-in in particular.

No.	Description	Type	Related bugs	Related external bugs
E1	Address-book view not updated (workaround: in online mode, do an address-book search (you can clear the search after it finished to see all contacts again), or restart *Evolution*)	contact	[3285204]	
E2	Inline attachments for iCalendar are dismissed by *Evolution* if the PIM object is edited	event, task, note	[3301561]	
E3	*Evolution* has the year 2038 problem for the start date	event, task	[3286426]	
E4	Calendar view not updated (workaround: switch month view or toggle off/on the calendar)	event	[3306337]	
E5	*Evolution* due datetime for tasks is displayed in UTC timezone but other times are displayed in the local timezone	task	[3285847]	
E6	*Evolution* directory traversal does not stop when deleting mail folders (workaround: none, may cause severe data loss)	mail		[636855]
E7	*Evolution* does display all day events with recurrence as not all day events (check-box "All Day Event" is not activated; time is shown for start and end time ("00:00"))	event	[3307866]	
E8	LibCamel crashes if using certificates infrastructure which is not properly set up (SSL/StartTLS, the back-ends crash in ssl_bad_cert(), camel-tcp-stream-ssl.c:885)	libcamel		[12475]
E9	When running *Evolution* under KVM or VMware virtualization, the sync conflict resolution is not working. No conflicts are detected, thus the client side object is always written onto the server, overwriting server side changes regardless of the configured sync strategy. Whether or not this is an *Evolution* bug or an e*volution-kolab* back-end implementation issue has not yet been verified.	contact, event, task, note	[3317200]	

Table 1: Known Evolution bugs, impacting the evolution-kolab plug-in

6 References

[CC BY-SA] -, Attribution-ShareAlike 3.0 Unported (CC BY-SA 3.0), -, http://creativecommons.org/licenses/by-sa/3.0/
[Evolution initiative] Matthew Barnes et al., Evolution - GNOME live!, 2011, http://live.gnome.org/Evolution/
[Kolab initiative] -, Kolab Groupware:: Home, 2010, http://www.kolab.org/
[Evolution user guide] Akhil Laddha et al., Evolution (2.30) user guide, 2009, http://library.gnome.org/users/evolution/stable/
[evo230] Jacob Zimmermann, Evolution 2.30 built for Ubuntu 10.04, 2010, https://launchpad.net/~jacob/+archive/evo230
[Ubuntu software] Ubuntu Documentation Team, Adding, Removing and Updating Applications, 2008, https://help.ubuntu.com/10.04/add-applications/C/index.html
[Evolution-Spam] Akhil Laddha et al., Stopping Junk Mail (Spam), 2009, http://library.gnome.org/users/evolution/stable/usage-mail-organize-spam.html
[Evolution-Accounts] Akhil Laddha et al., Working with Mail Accounts, 2009, http://library.gnome.org/users/evolution/stable/config-prefs-mail-identity.html
[Evo-Kolab-PKCS11] Silvan Marco Fin, Securing Kolab network connections and using security devices (PKCS #11) for client authentication with Evolution, 2010, http://sourceforge.net/projects/evolution-kolab/files/
[EDS-Arch] Milan Crha et al., Evolution / EDS_Architecture, 2010, http://live.gnome.org/Evolution/EDS_Architecture
[3285204] -, Delete contact does not work correctly, 2011, https://sourceforge.net/tracker/?func=detail&aid=3285204&group_id=343941&atid=1439212
[3301561] -, Attachment in Contacts is lost after editing in Evolution, 2011, https://sourceforge.net/tracker/?func=detail&aid=3301561&group_id=343941&atid=1439212
[3286426] -, Date does not work correct, 2011, https://sourceforge.net/tracker/?func=detail&aid=3286426&group_id=343941&atid=1439212
[3306337] Christian Hilberg, Calendar view not updated, 2011, https://sourceforge.net/tracker/?func=detail&aid=3306337&group_id=343941&atid=1439212
[3285847] -, The time is not correct, 2011, https://sourceforge.net/tracker/?func=detail&aid=3285847&group_id=343941&atid=1439212
[636855] Christian Hilberg, IMAPX folder deletion traversal does not stop, 2010, https://bugzilla.gnome.org/show_bug.cgi?id=636855
[3307866] -, "all day event"-option is not synchronized, 2011, https://sourceforge.net/tracker/?func=detail&atid=1439212&aid=3307866&group_id=343941
[12475] -, MeeGo crash stats - 12475, -, http://crashdb.meego.com/backtrace.php?number=12475
[3317200] Christian Hilberg, Sync conflict detection not working under KVM, 2011, https://sourceforge.net/tracker/?func=detail&aid=3317200&group_id=343941&atid=1439212