

# Delco-server user manual

Lassi Väätämöinen  
[lassi.vaatamoinen@tut.fi](mailto:lassi.vaatamoinen@tut.fi)

## Table of Contents

1. Delco server system architecture.....	3
2. Setting up the server system.....	4
3. Delco server application setup.....	4
Delco-specific setup.....	5
MySQL settings.....	6
4. Publishing a new service.....	7
Creating a Delco file.....	10
Changing service settings.....	11
Removing a service.....	12

# 1. Delco server system architecture

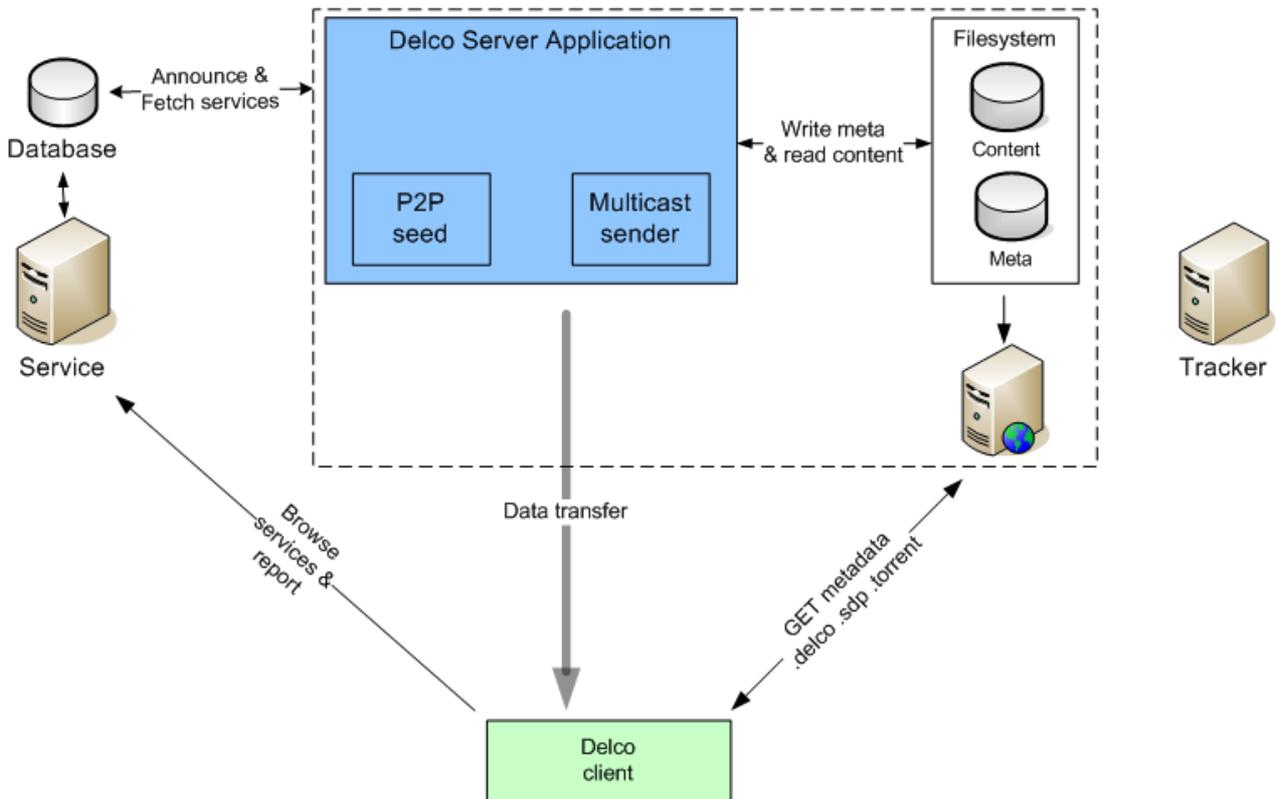


Figure 1. Delco server architecture

General server architecture is presented in figure 1. The server system consists of three logical parts: the Delco server application, the service page and a BitTorrent tracker. The Delco server application is used for creating the service metadata and for starting the actual data transfer (multicast-sessions and peer-to-peer back-end seeds). When a service is created, the service data is put to the database. And after the service is *activated* (service is written into a .delco file), the service data from the database is published on the service web page and a link to the Delco file is available for clients.

For the ease of implementation all actual metadata files (.delco, .torrent., .sdp) are written on the server application local file system. Because the metadata files must be made available for users to fetch, they must be placed in a web server content directory on this local host. This concrete file publishing is presented on the right in figure 1. When a user browsing the service page navigates a link of a given delco file, the actual metadata files related to this delco service are fetched from this local web server.

The tracker is started on its own host entity separately. It was observed during our tests that when starting the tracker on same host as the server application, only one client on the peer-to-peer network at a time was served.

## 2. Setting up the server system

Basic steps for setting up the Delco server system

- Import / setup database
- Web server(s): Database web server & server application local web server
- Start BitTorrent tracker
- Install server application software & needed libraries.

See *Delco-server installation instructions* for details.

## 3. Delco server application setup

When starting the server application for the first time, a server setup dialog is shown (figure 2). User is required to fill in all the fields in order for the service creation and publishing to work properly.

The screenshot shows a Windows-style dialog box titled "DelCo-server configuration". It has a "General" tab selected. The dialog is organized into two main sections: "Delco-specific" and "MySQL settings".

**Delco-specific section:**

- Host web-content URL:
- Web-contentdir:  (dropdown menu shows "src")
- Service content root dir:  (dropdown menu shows "src")
- File repair server URL:
- Report server URL:
- BitTorrent tracker announce URL:
- Channel base URL:

**MySQL settings section:**

- Server address:
- Server port:
- Password:

At the bottom right, there are two buttons: "OK" (with a green checkmark icon) and "Cancel" (with a red X icon).

Figure 2. Delco server initial setup dialog

## **Delco-specific setup**

Meanings for the setup entry fields:

**Host web-content URL** The base URL where clients can fetch service data. To this URL following directories are appended: delco/, torrent/ and sdp/  
E.g. Web content URL for the host is <http://server.host.fi/~delcouser/>

-> URLs for metadata are

<http://server.host.fi/~delcouser/delco/>

<http://server.host.fi/~delcouser/sdp/>

<http://server.host.fi/~delcouser/torrent/>

The .delco, .sdp and .torrent files are placed in these directories respectively.

**Web contentdir** The root directory for service data on local file system. This directory must be mapped to the **host web-content URL** in order to respond for client requests. That is, this local filesystem directory must correspond to the web-content URL of the host. The three directories (delco/, torrent/, sdp/) to this directory path are also appended.

**Service content root dir** The directory on local filesystem that contains actual service content files. All service content must be placed in this directory or in a subdirectory relative to this directory. The root directory must be used in order to parse the fileURI: information for Delco file.

**File repair server URL** The URL for file repair server used for announcing the repair servers to clients in all delco files created.

**Report server URL** Used for announcing report servers to clients. This address must be the address where report script responds on the report server.

**BitTorrent tracker announce URL** is used when creating torrent files. These trackers are announced to clients in the tracker file for connecting to the peer-to-peer network in Delco-system. Multiple tracker entries can be given, separated with spaces.

**Channel base URL** The base URL of channel service used for announcing channels to clients in a delco file containing channel service. A channel ID is appended to this base URL. E.g. Base URL <http://server.host.fi/channel/> -> Complete channel URL <http://server.host.fi/channel/3> (for channel ID = 3).

## MySQL settings

**Server address** Either a host name or IP address of a database server. By default local host is used.

**Server port** Port used in MySQL connections. By default this is 3306.

**Password** Password used for user authentication in MySQL server. Password depends on the database settings, default value is empty.

After all settings have been given and *OK* is clicked, the main window of server application is shown (figure 3).

**NOTE:** If server settings are altered afterwards, all existing services must be deleted, because the servicedata is not altered accordingly when settings are changed. If tracker announce URL is changed, also the .torrent files must be deleted in the local filesystem in order to allow creating .torrent files matching the setup settings.

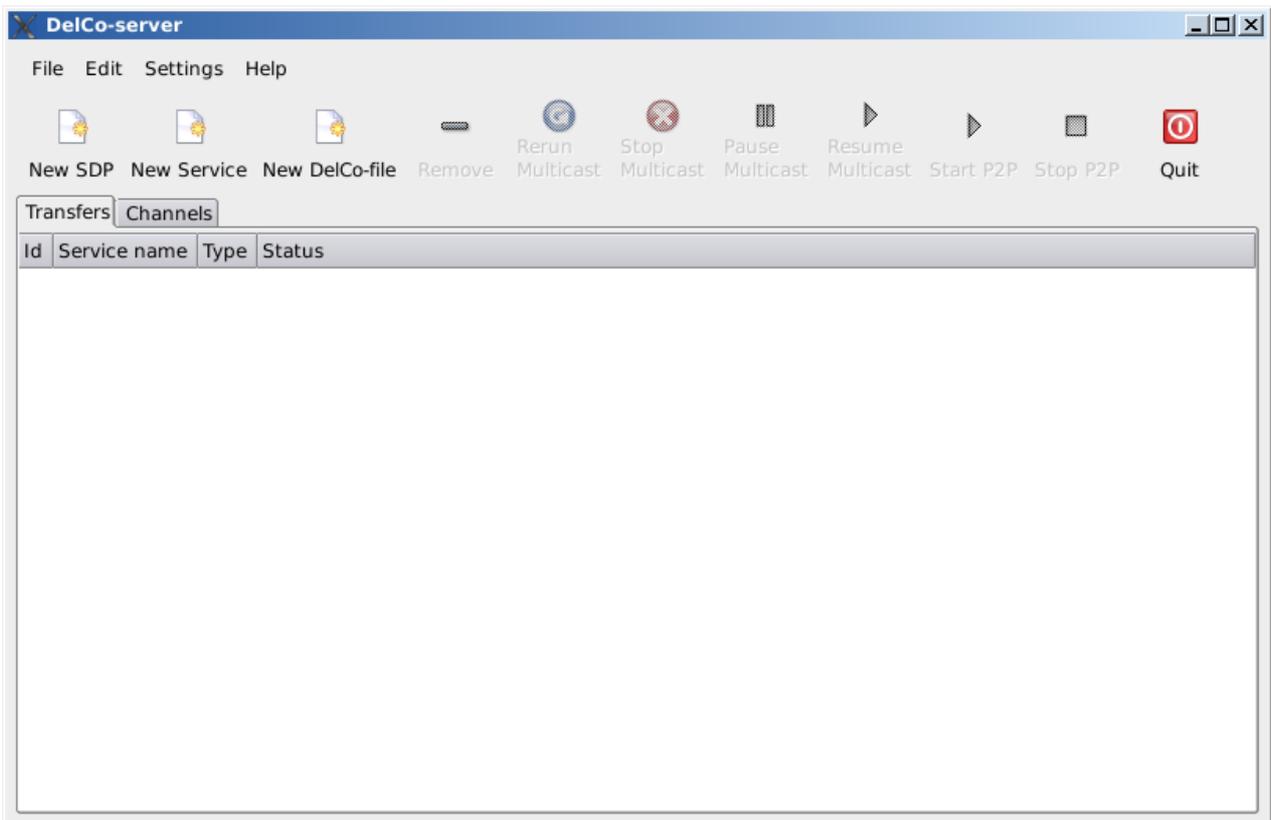


Figure 3. Delco server main window

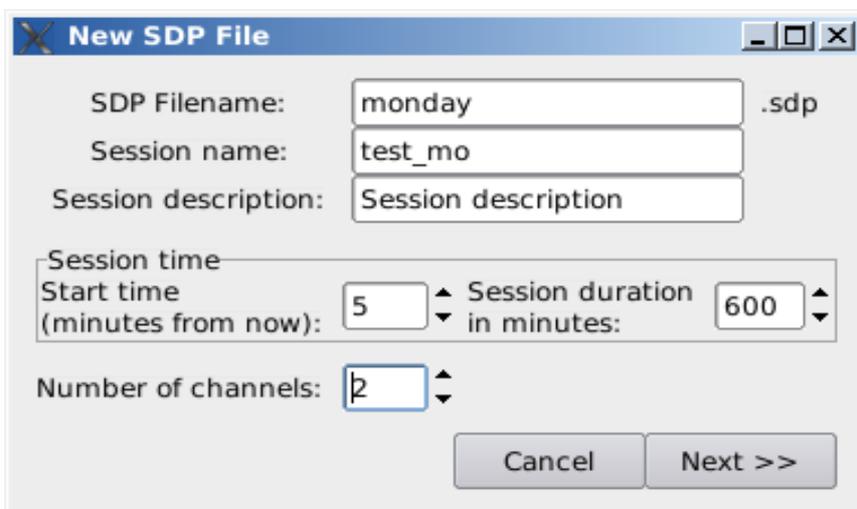
## 4.Publishing a new service

Starting from scratch, creating a new service must be started by creating a new SDP-file (a multicast session description). Pre-existing files that are accepted and can be parsed by FLUTE application may be used, but also the server application SDP creating dialog can be used.

The SDP dialog is started with window presented in Figure 4. First a filename must be given. It is recommended that no special characters (spaces, commas etc.) are used to ensure proper operation. Next, the session name must be given. Also an optional session description may be entered.

Session start and stop times are entered as minutes from current time. If non-zero session start time is given, this time will be waited before starting multicast session. Finally, number of channels used for the sessions are entered. Maximum channel amount is five.

Next part of the SDP dialog is to enter the channel information (figure 5). There are three fields per channel: Channel IP address, port number and time-to-live (TTL) field. Currently only IPv4 addresses are supported. Default values are entered in the boxes by the application, with FLUTE default channel IP address, but the port numbers differ (are incremented by one per channel). However, when defining multiple SDP files, the user must take care for entering proper channel IP and port addresses, because the values are same by default for every SDP file created using the dialog.

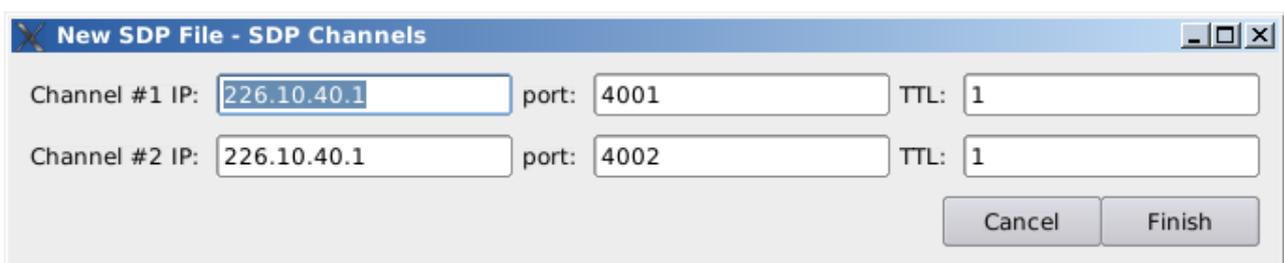


The screenshot shows a dialog box titled "New SDP File". It contains the following fields and values:

- SDP Filename: monday .sdp
- Session name: test\_mo
- Session description: Session description
- Session time: Start time (minutes from now): 5, Session duration in minutes: 600
- Number of channels: 2

Buttons: Cancel, Next >>

Figure 4. SDP dialog part 1



The screenshot shows a dialog box titled "New SDP File - SDP Channels". It contains the following fields and values:

- Channel #1 IP: 226.10.40.1 port: 4001 TTL: 1
- Channel #2 IP: 226.10.40.1 port: 4002 TTL: 1

Buttons: Cancel, Finish

Figure 5. SDP dialog for channel definition (part 2)

After the SDP file is created, a single service must be created (figure 6 and 7). First a service name is given. Then service type must be chosen and other parameters entered accordingly. Optional service description will be used for advertising BASIC services in a channel.

In the service content dialog the files contained in a service are entered. These files must be chosen from service content root (default) defined in server setup or from one of its subdirectories. After content has been selected .torrent-files for service are created, if P2P is chosen to be used within the service and the torrent files do not already exist in the web content directory.

If *CHANNEL* service is to be created user must check the channel radio- button in the service dialog. Next a channel name is requested. After the channel is created a popup containing created channel URL is shown. This channel is activated only after the channel service is *activated* by publishing it in a .delco file. After the channel is published in a .delco file, the channel service can be seen in the "Channels"-tab in the main window. After activating the channel all new .delco files created are put into all channels shown in the tab. (this is for demoing channel functionality).

A *STREAM* is basically an ordinary transfer server, but additional info for client control is inserted into .delco file. A stream service **MUST** be created only for streamable content.

A created service is saved in the serverdatabase, and will be stored there until being deleted via "New delco" dialog. Stored services can be reused when server is restarted.

**NOTE:** If a service in the database is wanted to be reused after restart, the pointer files (.torrent, .sdp) and content files related to the service **MUST** be available on local filesystem in original path.

**NOTE:** The server software does not currently offer a way to edit the service data afterwards. However, the multicast session descriptions can be "edited" to some extent. That is, the an SDP file can be redefined using the "New SDP" dialog (figures 4 and 5) by naming the "new" file as a file already existing in a service. This way expired sessions can be "extended" by defining new start and stop times. The filename which to write can be seen in serviceinfo view (figure 11) accessible on the context menu when right-clicking on a service on the main window.



Figure 6. Service dialog

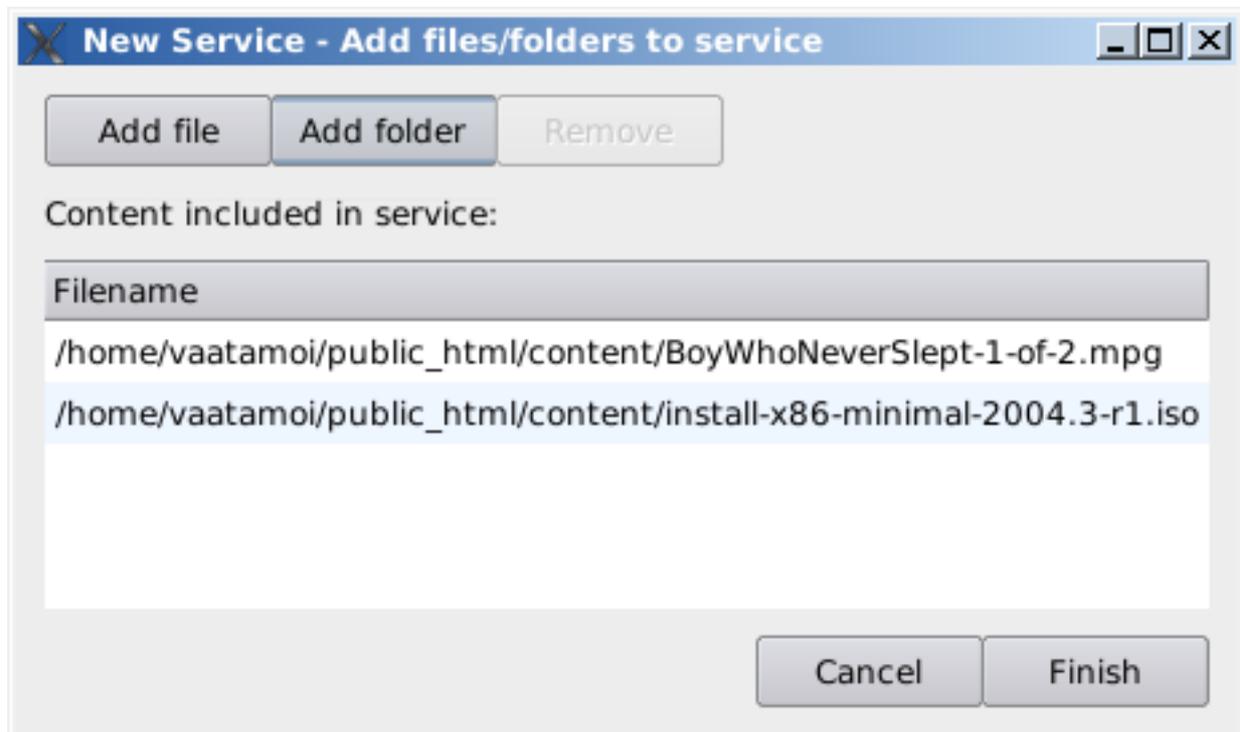


Figure 7. Service content dialog

## Creating a Delco file

All services created are listed in the window (figure 8). A new delco file is created by choosing one or more services from the list. Selected services will be contained in the .delco file being created.

It can also be chosen whether to start the multicast session immediately (if not checked, session must be started manually) and whether to start service provider P2P seed immediately, or only to publish the pointer file for clients to fetch. Also the P2P seed can be started manually afterwards.

After "Create" is clicked, the .delco file is written to local filesystem under the web content root and links are published in the service page. Activated services (i.e. those contained in a .delco file) are then shown in the server application main window (figure 9).

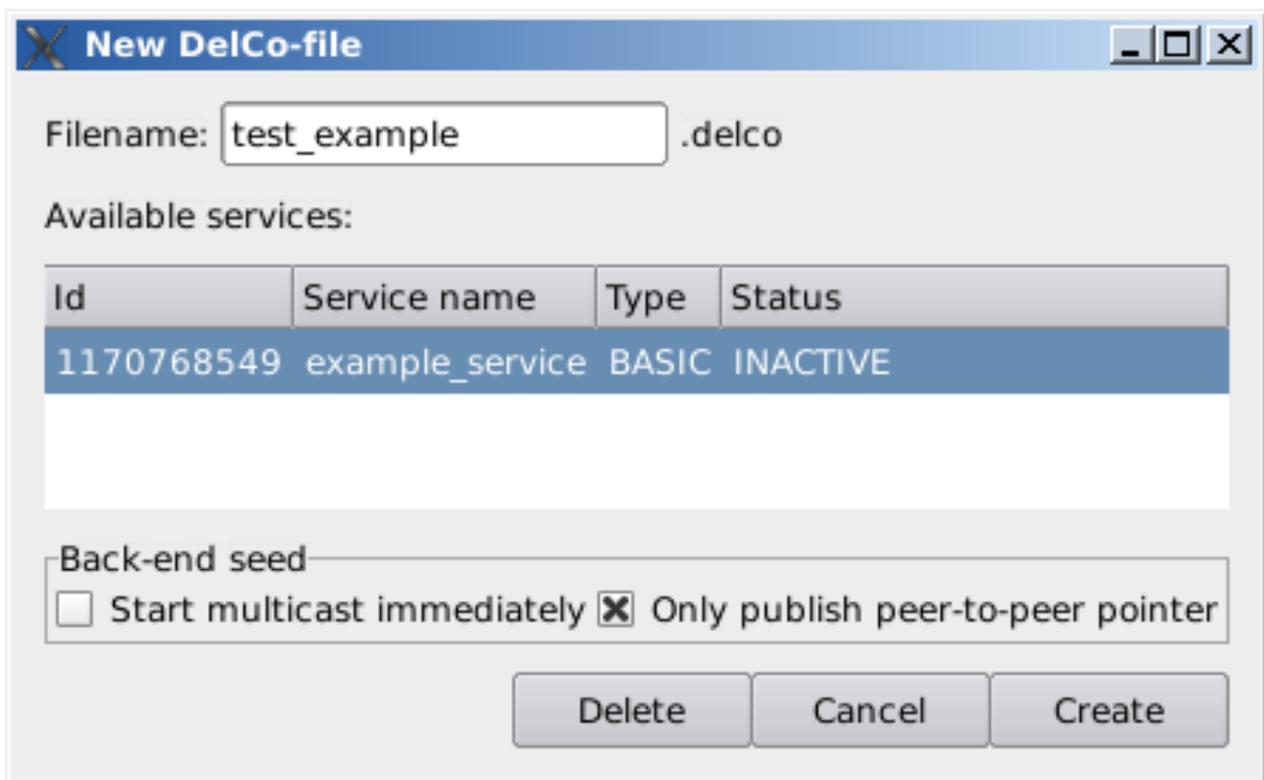


Figure 8. .delco file dialog

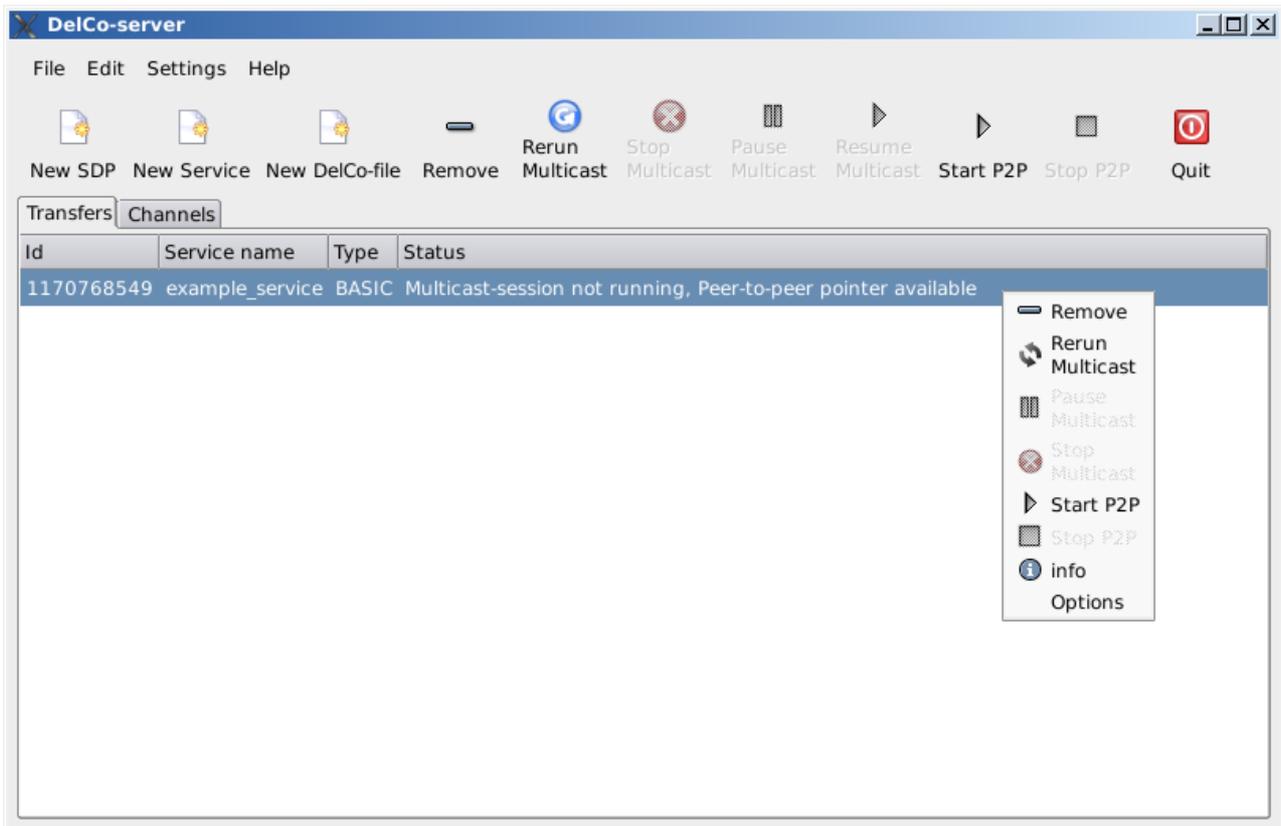


Figure 9. Main window with context menu after creating a service

## Changing service settings

Service settings can be modified in two different manners. First, the initial service settings can be globally defined on the second tab of server configuration. These settings are applied by default to all new services created.

The *service specific* settings (figure 10) can be changed by choosing a running service in the main window and choosing "Options" on the context menu (or "Server configuration" on Settings-menu). When changing service specific settings, the service name is shown on the settings-dialog. Three parameters may be changed here: upload rates for both multicast and peer-to-peer delivery and amount of multicast rounds for the session. (**NOTE:** Setting multicast rounds take effect only on newly started session, so if a multicast session is already running amount of rounds are NOT set on the running session). If the amount of rounds is set to zero, the session is continuous, i.e. multicast rounds are run as long as session time specified in SDP allows.

The transfer rate settings take effect immediately. On multicast the transfer rate setting given is the actual data rate of multicast sender. On peer-to-peer the transfer rate given is the amount of upload rate allowed for every item within the service. E.g four items in a session, transfer rate set to 150 kB/s => actual allowed upload rate 600kB for the peer-to-peer part of the service.

(**NOTE:** if wishing to de-select a service, ctrl + right-click is required, this is a GNOME -specific feature)

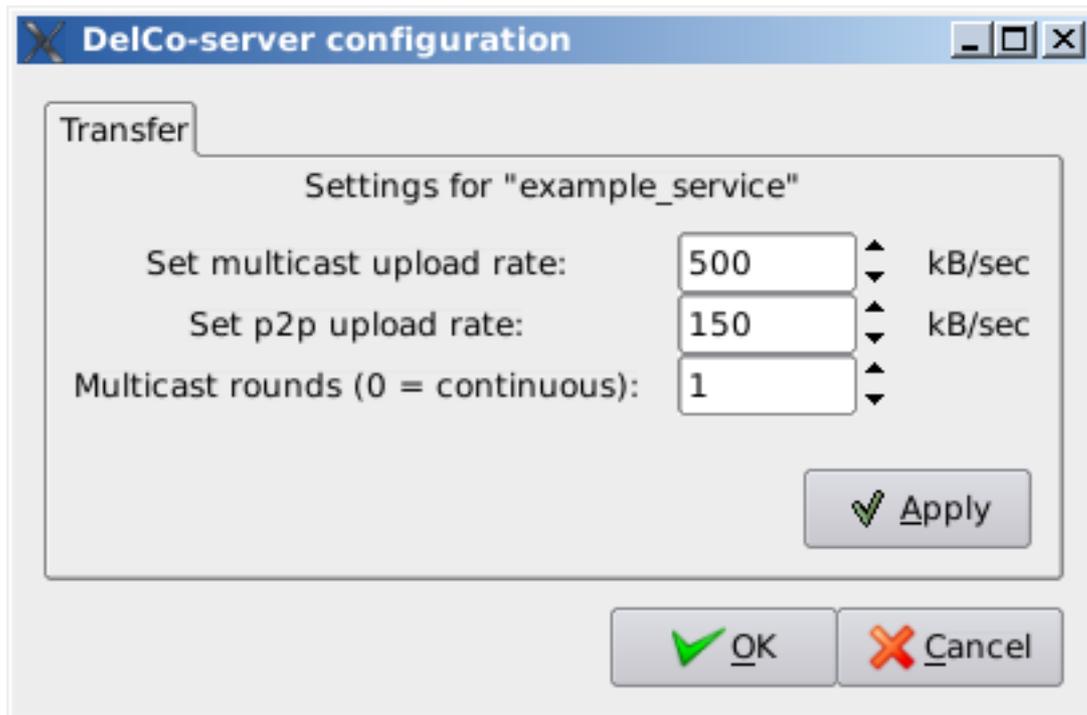


Figure 10. Service-specific settings dialog

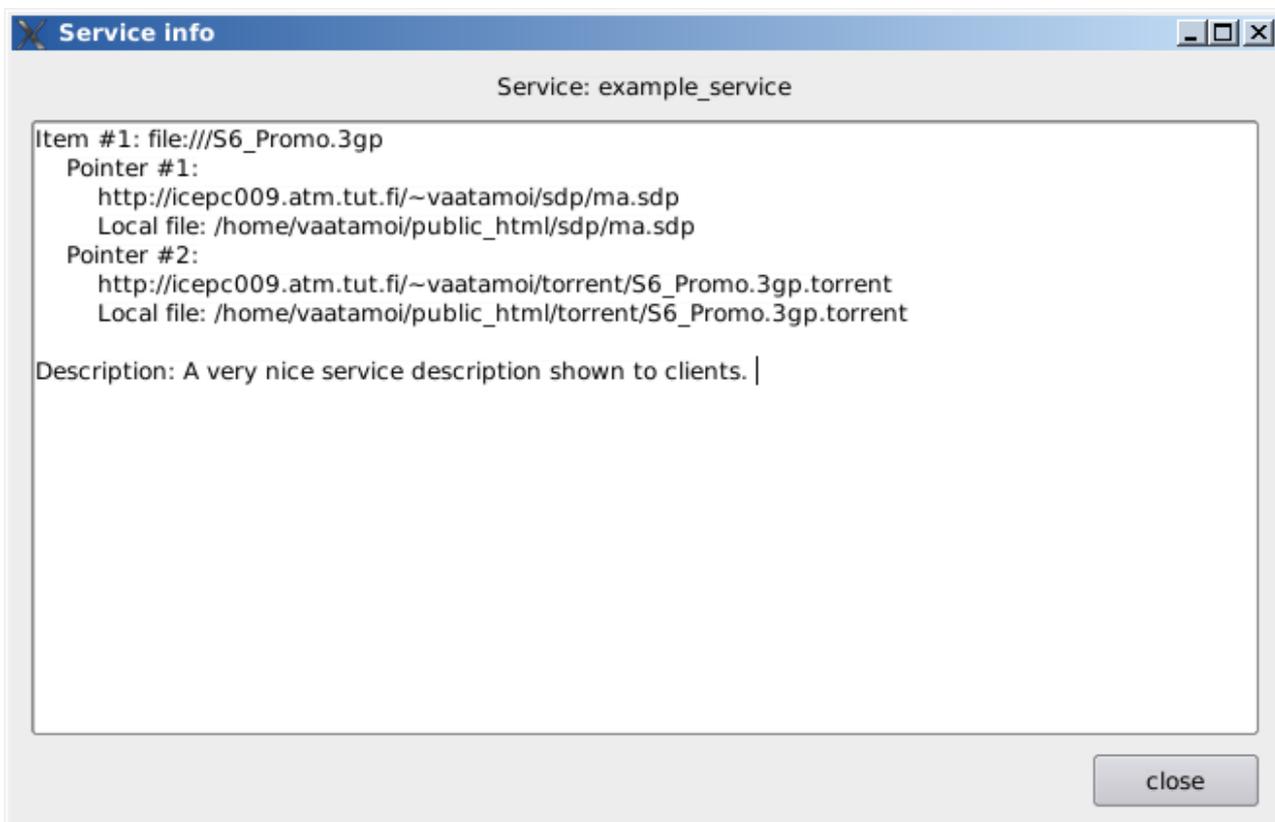


Figure 11. Service information view

## ***Removing a service***

A service can be “deactivated” (i.e. it is removed from a .delco file) by choosing appropriate service from the main window and using “Remove”. When this is done, all transfers within the service are stopped. When service is being deactivated, it is also removed from the .delco-file context. When all services originally contained in the .delco file are removed, the .delco file is deleted.

A deactivated service is still available for delco file-creating in the “New Delco-file”-dialog (figure 8), so the deactivated service can be reused in another (new) .delco file. If service is deleted also here by choosing the service and clicking “Delete”, the servicedata is removed completely (also from database) and may not be used anymore.