Geodetic Observatory Wettzell

e-RemoteCtrl for VLBI

User Manual

Written
by
Alexander Neidhardt (FESG/TUM Wettzell)

and
Martin Ettl (FESG/TUM Wettzell, MPIfR Bonn)

Version: June 17, 2013

Revisions:

June 17, 2013 (A. Neidhardt)	First proof-read version of the user manual (still missing is the station programming part)
June 14, 2013 (A. Neidhardt)	Completed user manual (missing station programming)
April 30, 2013 (M. Ettl)	First basic, incomplete version

Contents

Chapter 1

User Manual

Abstract

This chapter describes the usage of the remote control software for the `Very Long Baseline Interferometry (VLBI)` with radio telescopes and is divided into five sections. The first covers the download possibilities for the software. The second describes the installation procedure of the server and client software. The next section shows, how to start the server and the client part. Section four covers the complete configuration of the server and the client. Finally the fifth section describes the direct working with the graphical user interface of the client. It explains all available displays, features and possibilities for an operator.

1.1   Download of server and client part of the software

The software can be downloaded from the official e-RemoteCtrl Web page http://www.econtrol-software.de
(see fig. 1.1), where the latest stable and the current development version is available. Access data, as
user names and passwords, can be requested via the registration form on the Web page or directly via
email from neidhardt@fs.wettzell.de.
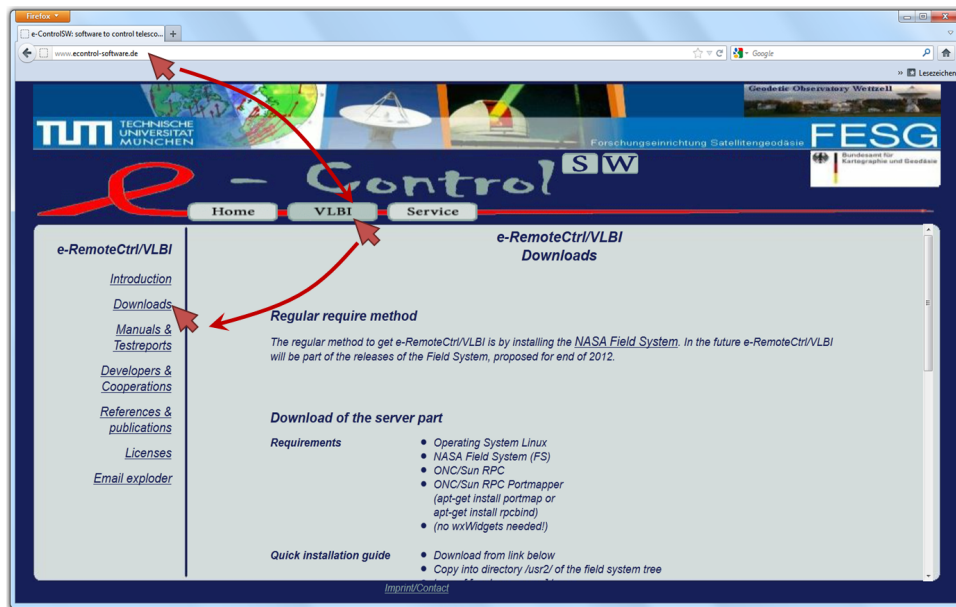
Figure 1.1: The official download page of the e-RemoteCtrl software.

The latest accepted, stable version can also be downloaded from the official `National Aero-
nautics and Space Administration (NASA)` **Field System** `File Transfer Proto-
col (FTP)` server atri.gsfc.nasa.gov. Access data, as user names and passwords, can be directly re-
quested via email from ed.himwich@nasa.gov.

The software can also be sent via email. Please contact neidhardt@fs.wettzell.de.

The software is split into two parts:

1. Server part, which must be installed on the `NASA` **Field System** `Personal Computer (PC)`
   (usually named "`<date>_eremotectrld_stable.tar.gz`" for the stable version and "`<date>_
   eremotectrld_dev.tar.gz`" for the preliminary, development version, where "`<date>`" is re-
   placed by the date of the current version)

2. Client part, which can be installed on each Linux `PC` with a graphical Desktop, to control the `NASA`
   Field System from remote (usually named "`<date>_eremotectrlgui_stable.tar.gz`" for
   the stable version and "`<date>_eremotectrlgui_dev.tar.gz`" for the preliminary, develop-
   ment version, where "`<date>`" is replaced by the date of the current version)

1.2   Installation of server and client part of the software

The installation of the e-RemoteCtrl software consists of two tasks. The first is the installation of the
e-RemoteCtrl server on the `NASA` Field System `PC` . The second consists of the installtion of the `GUI` .

1.2.1 Installation of the e-RemoteCtrl server

The installation does not influence or change the `NASA` Field System installation or configuration. The server is just an add-on to the existing system and copies Field System relevant parts into a local folder. The ideal location for the server directory is `"/usr2/eremotectrl/"` or in the station specific part `"/usr2/st/eremotectrl/"`.

Dependencies and system preparation

The server installation requires a correctly installed and working version of the `NASA` Field System 9.10.x or higher.

The server installation also requires a compiler for C and C++. The server was successfully compiled and tested with the GNU[1] compiler versions 2.95 to 4.6. You can check your current installed C-compiler version with:

```
$ gcc -v
$ Using built-in specs.
$ ...
$ gcc version 4.3.5 (Ubuntu 4.3.5-3ubuntu1)
```

The same can be done for your C++-compiler:

```
$ g++ -v
$ Using built-in specs.
$ ...
$ g++ version 4.3.5 (Ubuntu 4.3.5-3ubuntu1)
```

You should get the same (or a similar) output. If no compiler is installed, please install it with

```
$ sudo apt-get install g++
```

(for Debian like systems) or compile it according to the GNU compiler installation manual.

The building process also requires an installation of the programming language Perl, as it uses a code generator, written in Perl. You can check, if Perl is available on your computer with:

```
$ perl -h
```

If you get a help page, Perl is available. If not, please install it with:

```
$ sudo apt-get install perl
```

Former versions of the e-RemoteCtrl server software required a working installation of a "portmap" (port mapping daemon), which is something as a lookup table for ports of `Remote Procedure Call (RPC)` daemons. As newer Linux versions (e.g. Ubuntu) have some restrictions for the user rights of mapped daemons, the e-RemoteCtrl server does not use the "portmap" daemon anymore. This means, that it requires a fix setup of an Internet address and a port. If the "portmap" daemon should be used for any case, it can be installed with:

```
$ sudo apt-get install portmap
```

Or on newer systems with:

---

[1]http://gcc.gnu.org/

```
$ sudo apt-get install rpcbind
```

To check if a "portmap" daemon is active, run the follwoing command:

```
$ portmap -h
```

If you get a help output, it is already installed and ready on your system. If you get `"command not found"` it is not yet available and you can install it.

In order to allow a secure remote access, a running `SSH` server is required on the Field System `PC`. Usually a `SSH` server and client is installed together on a Linux system. If the server is active, please check the following:

```
$ ps ax | grep sshd
 1927  ?        Ss   0:00  /usr/sbin/sshd
21273 pts/1  S+   0:00  grep sshd
```

If you get the same (or a similar) output, the `SSH` daemon is active on your system. If not, please install it with:

```
$ sudo apt-get install ssh
```

To uncompress the software package, you also need the programs `"tar"` and `"gunzip"`, which are usually part of each Linux installation.

Uncompress and compile the e-RemotCtrl server

Copy the server package into the dedicated directory `"/usr2/eremotectrl/"` or into the station specific part `"/usr2/st/eremotectrl/"`.

The e-RemoteCtrl server software is compressed and packed in a `".tar.gz"` archive. It can be extracted with:

```
$ gunzip <date>\_eremotectrl_stable.tar.gz
$ tar -xvf <date>\_eremotectrl_stable.tar
```

After the extraction, change into the folder `"make"` of the new project directory with:

```
$ cd ./eremotectrl\_stable/make/
```

From here the server software can be compiled with:

```
$ make build
```

After a successful compilation and complete build run for the server code, the `"bin"` folder of the project directory must contain the executables `"ercd"`, which is the server, and `"ercc"`, which is a simple test client (only for specialists).

The server uses a dedicated authentication and authorization mechanism to handle user rights. Therefore it requires a possibility to acquire "root" rights for a very short period, in which the authentication of a user must be checked. Therefore it is necessary to allow this, which is permitted with the "install" call as "root". It can be run with:

```
$ sudo make install
```

1.2.2 Install the e-RemoteCtrl GUI client

The `GUI` client software must be installed on a separate client machine. The client is responsible to request data from the server, which offer these data. It is not required to install the `GUI` client software on the `NASA` Field System `PC` . The `GUI` uses wxWidgets[2] for its graphical output. Therefore the installation of wxWidgets libraries on the dedicated client machine is necessary.

Dependencies and system preparation

On Debian based systems the wxWidgets libraries can be installed with[3]:

```
$ sudo apt-get install libwxgtk2.8-dev
```

In order to make sure, that the correct wxWidgets version is installed on your system, you can use the following check[4]:

```
$ wx-config --version
$ 2.8.12
```

The client installation also requires a compiler for C and C++. It also uses the Perl generator, so that the programming language Perl must be available on the client machine. To uncompress the software package you also need the programs "`tar`" and "`gunzip`", which are usually part of each Linux installation. For the secure connection to the server, the client uses a standard `SSH` client, which is also usually part of each Linux distribution. If one of those programs is not available, see the installation section of the server software, how to install the individual parts.

On newer Linux distributions (e.g. Ubuntu) the sound libraries are anymore not activated for "`libwxgtk2.8-dev`", so that no sounds (e.g. to acoustically notify Field System errors) can be heard. If you detect this, it is necessary to install the sound libraries explicitly and to compile wxWidgets from scratch. Download the latest stable version of wxWidgets e.g. "`wxGTK-2.8.12.tar.gz`"[5]. Then follow the following steps as "root":

```
$ sudo su
$ apt-get remove libwxgtk2.8-dev
$ apt-get install libsdl-sound1.2-dev
$ apt-get install libgtk2.0-dev
Unpack the wxWidgets sources with "unzip" and "tar" and
change into the newly created folder of the wxWidgets sources.
$ mkdir gtk-build
$ cd gtk-build
$ ../configure --enable-unicode --with-sdl
$ make
$ make install
```

Add the following line to your "`.bashrc`" file in your home directory ("`/home/administrator/.bashrc`"):

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/lib/
```

---

[2]http://www.wxwidgets.org

[3]On RedHat based system, please use the yum-commands "`yum install wxGTK-devel`" and "`yum install wxBase`".

[4]The version number can differ, depending on your Linux-distribution.

[5]http://www.wxwidgets.org/downloads/#latest_stable or with the `FTP` "`ftp ftp.wxwidgets.org`" (User: anonymous; Password: anonymous) from the directory "`./pub/2.8.12`".

Uncompress and compile the `GUI` client

Copy the client package into a dedicated directory.

The e-RemoteCtrl client software is compressed and packed in a "`.tar.gz`" archive. It can be extracted with:

```
$ gunzip <date>\_eremotectrlgui\_stable.tar.gz
$ tar -xvf <date>\_eremotectrlgui\_stable.tar
```

After the extraction, change into the folder "`make`" of the new project directory with:

```
$ cd ./eremotectrlgui\_stable/make/
```

From here the client software can be compiled with:

```
$ make build
```

After a successful compilation and complete build run for the client code, the "`bin`" folder of the project directory must contain the executables "`erc_gui`"

1.3   Starting of server and client part of the software

To use the software, it is necessary to run the server part on the `NASA` Field System `PC` . Then different `GUI` clients can connect to it in parallel. The `GUI` clients must be started on the client machines.

1.3.1   Start the e-RemotCtrl server

The server requires a configuration file for its startup. A first, general version of such a configuration file is located in the folder "`config`". It is named "`eremotectrl.conf`". With this file the server can be started with:

```
$ ../bin/ercd ../config/eremotectrl.conf
```

If the server starts without any error output, it is connected to the `NASA` Field System and ready for remote requests from clients. The server can deal with parallel clients and handles requests sequentially according to the arrival time of the individual request.

For each station it is necessary to customize the configuration file, according to the individual needs and setups. How to do this is explained in section 1.4.

To run the server as a background process and to avoid debugging outputs to the console, start the server in a "`bash`" (shell) with:

```
$ ../bin/ercd ../config/eremotectrl.conf > /dev/null 2> /dev/null &
```

1.3.2   Start the graphical user interface

The `GUI` client requires a configuration file for its startup. A first, general version of such a configuration file is located in the folder "`config`". It is named "`RPCClient.conf`". With this file the client can be started with:

```
$ ../bin/erc_gui ../config/
```

For each station it is necessary to customize the configuration file, according to the individual needs

and setups. How to do this is explained in section 1.4.

If the software starts correctly you should see a similar window as in fig. 1.2.
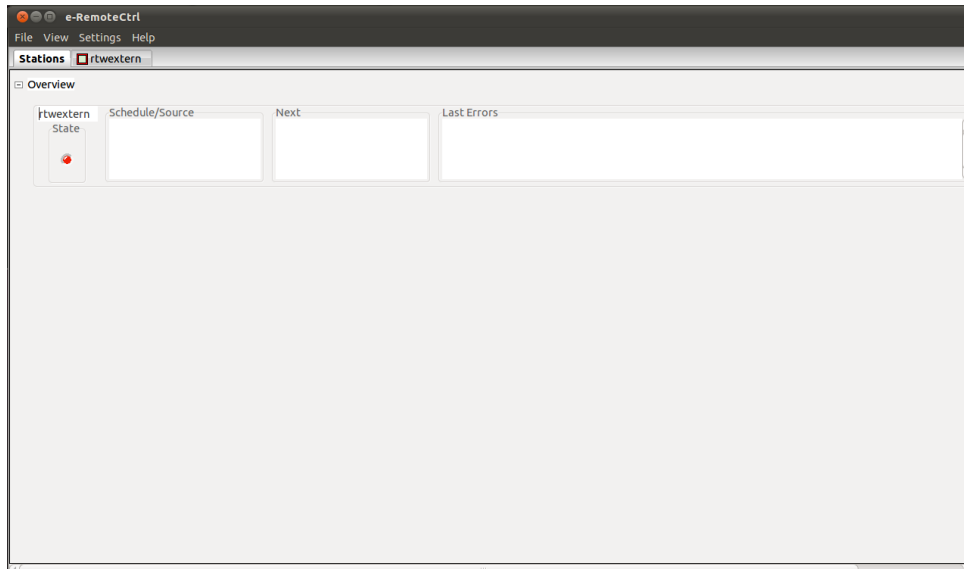

Figure 1.2: The e-RemoteCtrl graphical user interface after startup.

1.4 Configuration of server and client part of the software

This section explains how to configure the e-RemoteCtrl server and client software. Furthermore, it covers the configuration file formats and shows how to customize them.

1.4.1 Create and delete remote control users on the NASA Field System PC

In general it is possible to create standard Linux users with "`adduser`" on the NASA Field System PC , which are then used for the remote access. In any case it is strongly recommended, that each operator gets its own user name and password, to access the system from remote. The standard users of the NASA Field System "`oper`" and "`prog`" should not be used for remote logins. As the e-RemoteCtrl server logs all remote activities into the log file together with the user name of the operator, who commanded them, so that each remote action can be reproduced again, it is essential, that a real person can be assigned by her name as a responsible operator.

Linux allows standard users many possibilities, e.g. they can read foreign directories of other users. This is not always wished by the local person in charge. Therefore it is much better to keep the remote user in his home directory and to restrict the executable programs to a minimum. This can be realized with a "rbash" shell.

To simplify the creation of an "rbash", remote users can be created with the script "`addercduser`", which is available from the download page of the e-RemoteCtrl software. To create a user run the script as superuser ("root") with

```
$ ./addercduser <username>
```

where "`<username>`" is replaced by the real user name, which should be created:

The script offers three different security levels for the user, which should be created.

1. Standard Linux/Unix user with "bash" as shell

2. Partly restricted remote user with "rbash" as shell and the permission to use a small set of executables as programs within the home directory (e.g. "ls", "cd", etc.).

3. Completely restricted remote user with "rbash" as shell and no permission at all to run any programs

A successful creation of a user looks like this (the password must be defined during the creation run, similar to the standard process under Linux):

```
#########################################
# Add a new e-RemoteCtrl user to system #
#########################################
0) Check superuser rights
 => [OK]
1) Enter security level
[0] = Standard Linux/Unix user (bash)
[1] = Partly restricted remote-user (rbash;
      SSH-access with basic programs)
[2] = Completely restricted remote-user (rbash;
      SSH-access without programs)
==> 1
2) Check rbash on system and create link if necessary
 => [OK] rbash exists
3) Check ercd user groups and create it, if necessary
 => [OK] Group ercd exists
4) Create new user
Lege Benutzer >>mytester<< an ...
Lege neuen Benutzer >>mytester<< (1001) mit Gruppe żercdń an ...
Erstelle Home-Verzeichnis >>/home/mytester<< ...
Kopiere Dateien aus >>/etc/skel<< ...
Geben Sie ein neues UNIX-Passwort ein:
Geben Sie das neue UNIX-Passwort erneut ein:
passwd: Passwort erfolgreich geändert
Benutzerinformationen für mytester werden geändert.
Geben Sie einen neuen Wert an oder drücken Sie ENTER für den Standardwert
 Vollständiger Name []:
 Raumnummer []:
 Telefon geschäftlich []:
 Telefon privat []:
 Sonstiges []:
Sind diese Informationen korrekt? [J/n] J
 => [OK] User created
5) Change access rights for .bashrc
 => [OK]
6) Define allowed programs for restricted user in .bashrc
 => [OK]
7) Set program PATH for restricted user in .bashrc
 => [OK]
```

An already created user can be deleted with the following script, which is also available from the download page of the e-RemoteCtrl software:

```
$ ./delercduser <username>
```

A successful deletion of a user looks like this:

```
############################################
# Delete an e-RemoteCtrl user from system #
############################################
0) Check superuser rights
 => [OK]
3) Delete existing user
userdel: warning: can't remove /var/mail/mytester:
Datei oder Verzeichnis nicht gefunden
 => [OK]
```

The created user can then be used for the user roles in the serverconfiguration.

1.4.2 Configuration of the e-RemoteCtrl server

The configuration file for the server is located in the "config" folder of the project directory. The standard name is "eremotectrl.conf". It contains a tagged structure, which separates the single sections with dedicated markup blocks, which are:

```
<eremotectrld>
    <Telescope>
        [...]
    </Telescope>
    <Server>
        [...]
    </Server>
    <FS>
        [...]
    </FS>
    <eQuickStatus>
        [...]
    </eQuickStatus>
    <Logging>
        [...]
    </Logging>
    <StationSpecifics>
        [...]
    </StationSpecifics>
</eremotectrld>
```

The "Telescope" section contains telescope or site specific information. Currently this is just the IVS Station Code, which cannot be requested from the NASA Field System. This entry must be customized to the individual code of the individual site. For the Wettzell 20m radio telescope this section looks like:

```
<Telescope>
    IVSStationCode = Wz
</Telescope>
```

The "Server" section contains all relevant information for the e-RemoteCtrl server connection and access, e.g. ports, timeouts, verbosity levels and user roles. This entry must be customized to the individual network and access settings of the individual site. An example section looks like this:

```
<Server>
    TCPPort = 60225  #d TCP port of server changed during init-phase
    UDPPort = 60226  #d UDP port of server changed during init-phase
    InactiveUsersAutoLogOutTime = 60 #d Auto log out time for inactive
                                     # users in seconds
    VerboseActivated = yes # Activates verbose behaviour with text
                           # outputs per function
```

```
    <RoleManager>
      <User>
          Name       = ettl        # The user name, which was created
                                    # before in the operating system
                                    # with 'addercduser'
          StaticRole = operator     # The user role: observer, notifier,
                                    # scheduler, agent, operator,
                                    # supervisor or superuser
          Blocked    = no           # Tag which defines if a user is
                                    # currently blocked or not (yes/no)
      </User>
      <User>
          Name       = neidh        # The user name, which was created
                                    # before in the operating system
                                    # with 'addercduser'
          StaticRole = supervisor   # The user role: observer, notifier,
                                    # scheduler, agent, operator,
                                    # supervisor or superuser
          Blocked    = no           # Tag which defines if a user is
                                    # currently blocked or not (yes/no)
      </User>
    </RoleManager>
</Server>
```

Each server opens two ports, one for the `User Datagram Protocol (UDP)` and one for the `Transmission Control Protocol (TCP)`, while currently only the `TCP` port is used for connections. The ports for both protocols are defined with `"TCPPort"` or respectively with `"UDPPort"`. Then the server opens a connection channel on these ports. `"InactiveUsersAutoLogOutTime"` defines the time in seconds, after which an inactive user is automatically logged out again, if he does not anymore send frequent requests. `"VerboseActivated"` can be set to `"yes"` or `"no"` and switches the debugging output to the console on or off.

Additionally the `"Server"` section contains the user roles. It is strongly recommended, that each operator gets its own user name and password, to access the system. The standard users of the NASA Field System `"oper"` and `"prog"` should not be used for remote logins. As the e-RemoteCtrl server logs all remote activities into the log file together with the user name of the operator, who commanded them, so that each remote action can be reproduced again, it is essential, that a real person can be assigned by her name as a responsible operator. Remote users can be created with the script `"addercduser"`, which is available from the download page of the e-RemoteCtrl software. Each user can be assigned with a dedictaed static role, which is one of those from fig. 1.27 on page 31.

Each user role definition contains three parts:

1. the `"Name"`, which defines the Linux user name of the individual remote operator,

2. the `"StaticRole"`, which defines the static user role of the remote operator and

3. the `"Blocked"` flag, which allows to block a user temporarily.

The `"FS"` section contains relevant information from the NASA Field System. Currently these are the paths of the requested files. It looks like the following, where usually only the `"HTMLHelpPath"` must be adapted to the real installation directory on the station specific machine:

```
<FS>
    LogPath = /usr2/log/      # Path to the NASA field system logs
    BinPath = /usr2/fs/bin/   # Path to the NASA field system binaries
    FSHelpPath = /usr2/fs/help/  # Path to the NASA field system help
```

```
    STHelpPath = /usr2/st/help/  # Path to the NASA field system station help
    HTMLHelpPath = /usr2/st/eremotectrl_deliverable/doc/htmlhelp/  # Path to
                                            # econtrol help pages
</FS>
```

The default paths are:

- "LogPath": the path of the logging directory of the Field System, which contains all log files.

- "BinPath": the directory with the binaries (executables) of the Field System.

- "FSHelpPath": the directory with all help and manual pages of the Field System.

- "STHelpPath": the directory with all station specific help and manual pages.

- "HTMLHelpPath": the directory with all new Hypertext Markup Language (HTML) help and manual pages, especially for the e-RemoteCtrl software[6].

The "eQuickStatus" section contains relevant settings for the real-time information transfer, to send status information of the telescope to a centralized Web presenter. As this information is defined by the operator of the Web publishing and presenting server, the user just has to change the location of the SSH key file in "SSHKeyfilePath". He must define, where the key file for the connection to the Web presenting server is located on the local Field System PC [7]. The section usually looks like this:

```
<eQuickStatus>
    SendStatus          = yes                    # Send status (yes) or not (no)
    SSHCopyClient       = /usr/bin/scp           # Used secure copy client
    SSHServer           = econtrol.iapg.bv.tum.de # Server, to which the status
                                                 # should be sent to
    SSHUsername         = equickstatus           # User on the e-QuickStatus server
    #ATTENTION: The user rights of the following key file must fit to the user,
    # who starts the ercd (e-RemoteCtrl server), e.g. "-rw------- oper rtx",
    # if oper should start the remote control server.
    SSHKeyfilePath      = ${HOME}/.ssh/iapg_erc_web_rsa # Key file, used for log in
    LocalInfoPath       = /tmp/                  # Temporary path, from where is
                                                 # copied from
    RemoteInfoPath      = ./statusincoming/.     # Remote location, to which should
                                                 # be sent
    TimeOutSec          = 5                      #d SSH control timeout in seconds
</eQuickStatus>
```

The "Logging" section contains all settings for the additional logging functionalities of the e-RemoteCtrl software. In detail these are checklists for operator inputs before, during and after the observation (see section 1.5.9 on page 36) and error classifications to define alarm levels. The section looks like this:

```
<Logging>
    MaxNumberOfBufferedLogLines = 1000 #d The maximum number of buffered log lines
                                    # per user
    <Checklists>
        # The <Before> questions are asked at the beginning of a schedule
        <Before>
            Activated = yes              # Ask these questions (yes/no)
            RequestIfLogTag = exper_init  # The questions are asked when this tag
                                          # is found in the log file
            <Question>
                QuestionText = System: ACU on remote mode # The question
                RequestValueByOperator = no   # The question requires a value from
                                              # the operator (yes/no)
```

---

[6]All classic help and manual pages are automatically converted into HTML pages, according to the found tags.
[7]Key files can be requested from the operators of the Web presenter or "IVS Live" Web page.

```
                    </Question>
                    [...]
            </Before>
            # The <During> questions are asked regularly each full hour in the defined
            # interval times during a schedule
            <During>
                Activated = yes # Ask these questions (yes/no)
                RequestIfLogTag = midob  # The questions are asked when this tag is
                                         # found in the log file
                RequestIntervalTimeHours = 2 #d The questions are asked each
                                             # <IntervalTimeHours>
                <Question>
                    QuestionText = System: ACU ok, no interlocks active # The question
                    RequestValueByOperator = no   # The question requires a value from
                                                  # the operator (yes/no)
                </Question>
                [...]
            </During>
            # The <After> questions are asked at the end of a schedule
            <After>
                Activated = yes # Ask these questions (yes/no)
                RequestIfLogTag = sched_end  # The questions are asked when this tag is
                                             # found in the log file
                <Question>
                    QuestionText = Antenna back in stow position # The question
                    RequestValueByOperator = no   # The question requires a value from
                                                  # the operator (yes/no)
                </Question>
                [...]
            </After>
        </Checklists>
        <LogErrorClassification>
            # Each error identifies the device, the error number and
            # the action started on error with IGNORE, BEEP, ALARM and
            # the alarm level with INFO, WARNING, ERROR
            <LogErrorRule>
                DeviceIndentifier = bo
                ERRORNumber       = -105
                Action            = ALARM
                AlarmLevel        = ERROR
            </LogErrorRule>
            [...]
        </LogErrorClassification>
</Logging>
```

This section can be customized for individual station needs, where the following parameters can be set:

- "`MaxNumberOfBufferedLogLines` " defines the number of lines from the current log file, which are buffered per logged-in user (each user must frequently request, to avoid a loss of log data)

- "`Checklist`" sub-section defines list settings for the "before", "during" and "after" session checklists

    - "`Before`" defines the settings for the checklist, which is released, when the observation starts. The usage can be switched on and off with the "`Activated`" parameter (with "yes" or "no"). A search pattern, for which the log text is scanned, defines the log situation, when the list is released. If the pattern is found the checklist is released. The following part is a sequence of "`Question`" definitions with:

* "`QuestionText`", which is the question text, which must be answered by the operator

* "`RequestValueByOperator`", which can be set to "yes", to request an answer input for the dedicated question from the operator

– "`During`" defines the settings for the checklist, which is released, when the observation is on progress. The usage can be switched on and off with the "`Activated`" parameter (with "yes" or "no"). A search pattern, for which the log text is scanned, defines the log situation, when the list is released. If the pattern is found the checklist is released. Different to the "before" checklist, a time interval can be specified. It allows to pop-up of the list only each specified time period (in hours), so that a found pattern releases the list only, when also the time expires. The following part is a sequence of "`Question`" blocks in the same style as before.

– "`After`" defines the settings for the checklist, which is released, when the observation finishes. The usage can be switched on and off with the "`Activated`" parameter (with "yes" or "no"). A search pattern, for which the log text is scanned, defines the log situation, when the list is released. If the pattern is found the checklist is released. The following part is a sequence of "`Question`" blocks in the same style as before.

* The "`LogErrorClassification`" sub-section consists of a sequence of "`LogErrorRule`" blocks. Each error definition is used to define an action for the situation, when the error statement occurs in the log file, and to define an alarm level for each error. Therefore each block consits of:

– "`DeviceIdentifier`", which identifies the NASA Field System device with a two-character classifier

– "`ERRORNumber`", which defines the error number for the individual device

– "`Action`", which defines the action, which is operated, if the error is found in the log text (possible actions are: "IGNORE" to do nothing, "BEEP" to release a beep sound or "ALARM" to release an alarm sound on the client machine)

– "`AlarmLevel`", which defines the alarm level, to colorize the output (possible classifiers are "INFO" to handle it, like a usual text message, "WARNING" to color the line with the warning color "orange", or "ERROR" to colorize the line with the error alarm color "red" in the client output display)

The last section is "`StationSpecifics`". It contains station specific parts, which are handed over to station code modules of the e-RemoteCtrl software. One essential part here is the "`SysMon`" section, which is used to configure the system monitoring handling in the e-RemoteCtrl server. In general the individual developer of station-specific system monitoring tools can define own structures, as this section is only used in the station specific part (see chapter 2 on page 45).

It is necessary to restart the e-RemoteCtrl server after each change of the configuration file.

1.4.3   Configuration of the e-RemoteCtrl client

The e-RemoteCtrl client can be configured with a configuration file located in the "`config`" folder of the client project directory. The structure is similar to the already described of the server. But the content is completely different. It is possible to manually change values directly in the configuration file. A much more sophisticated way is to change the values directly in the graphical user interface in the "Configure" display. Therefore the configuration of the client is only described for this graphical handling.

The configuration display can be opened, after selecting the dedicated telescope tab, by clicking on the "Configure" section (see fig. 1.3). For the first configuration, no password is needed. After the configuration with `SSH` a password is required (see the next section).
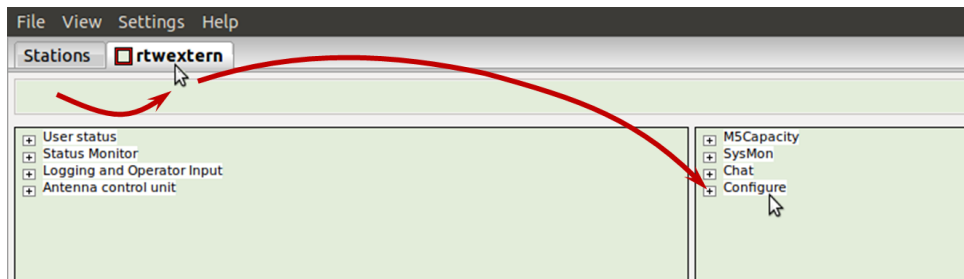
Figure 1.3: Opening of the configuration display.

It opens the configuration display (see fig. 1.4).

This display is separated into several tabs, where different configurations can be made for:

- the connection from a Sun `RPC` client

- the connection to a Sun `RPC` server

- the `SSH` communication with tunnels

- the management of shortcuts (hot keys)

- the appearance of the `GUI`
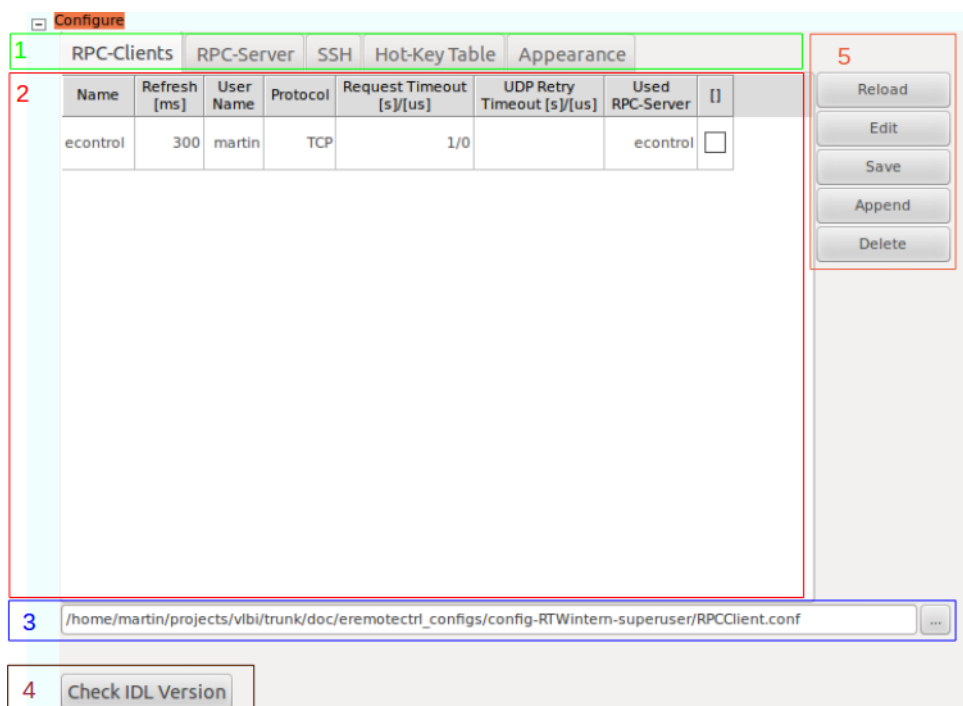
Figure 1.4: The e-RemoteCtrl configuration display

Also the management of the configuration file (like saving, reloading or showing of the file) can be handled in this display (see fig. 1.4, section 5). The path to the currently selected configuration file and a possibility to switch to another file is available here (3). Additionally it is possible to compare the versions of the e-RemoteCtrl client and server, after a connection was established (4).

In detail this means:

- Section 1 in fig. 1.4
  This section contains the configuration tabs, where the user can change between several configuration menus.

- Section 2 in fig. 1.4
  After the selection of a specific tab, the configuration part for the individual topic is loaded into this area.

- Section 3 in fig. 1.4
  This section shows the path to the configuration file. By pushing on the button at the right, a dialog box appears, in which a new configuration file can be selected. After the selection of a new file, the new parameters of that file are loaded automatically after checking the user credentials with user name and password (see section 1.5.4 on page 27).

- Section 4 in fig. 1.4
  By pushing on this button, the versions of the idl2rpc-generator for the communication modules and of the `Interface Definition Language (IDL)` file with the communication interface definition is checked, if a valid connection could have been established. Both version numbers give information about the available functions and valid communication levels. The appearing dialog (see fig. 1.5) compares the server versions (remote) with the client versions (local). If there is any mismatch between the version numbers, it is highly recommended to equalize the versions before usage. Otherwise the connection between the client and the server might not be stable, especially if the `IDL` file version differs.
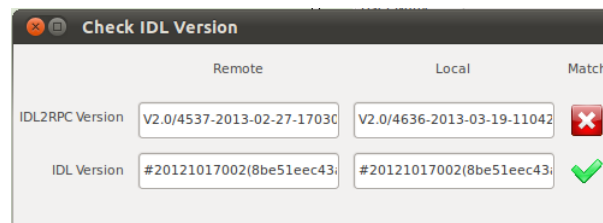
Figure 1.5: The dialog window to compare the version numbers of the server with those of the client, showing a mismatch of the idl2rpc-generator versions in the example here.

- Section 5 in fig. 1.4
  This section offers five buttons for:

  – Reload
    By pushing on this button, the content of the current configuration file is loaded again. This can be used to undo local changes or to restore a previous setting.

  – Edit
    By pushing on this button, a simple editor window appears, in which the configuration file can be edited and saved manually. This view is only for the experts!!!

  – Save
    By pushing on this button, the current settings will be saved into the current configuration file.

– Append
By pushing on this button, an empty row will be added to the setting tables of the tabs:

  * "RPC-Clients" configuration
  * "RPC-server" configuration
  * "Hot-Key table" management

– Delete
By pushing on this button, a selected row will be deleted in the configuration tabs, which are described for the "Append" button.

The following sections describe the configuration tabs in detail.

First tab: the connection settings for the RPC client

The first tab of the configuration display is for the settings of the client part of the communication (see fig. 1.6).

Figure 1.6: The configuration tab for the settings of the client part of the communication.

The following entries (columns) can be changed here and are in detail:

• Name:
"Name" defines the internal name of the client. This is only relevant for other software, where more parallel clients are used to connect with different servers. For e-RemoteCtrl it must be "econtrol".

• Refresh [ms]:
This column defines the refresh rate of the GUI in milliseconds. A good value is a quarter of the reaction time of a human being, which is about 250 to 300 ms. A value of 300 ms means, the client tries to request data from the server each 300 milliseconds after a previously finished communication, in order to update the displays.

• User name (is obsolete):
The user name is not necessary anymore. Only for older versions before the authentication and authorization mechanisms it was necessary to assign a user to chat messages.

• Protocol:
Here it would be possible to select TCP or UDP as network protocol. For e-RemoteCtrl connections, it is fixed to TCP .

• Request timeout [s]/[us]:
The request client timeout[8] defines, after which time a communication request must reply. The value is defined in seconds and microseconds. The values must be greater than 0. If a request

---

[8]Also take a look into the idl2rpc.pl manual[1]

could not be finished within this time, the communication request is interrupted and returns with an error. The value is usually a heuristic experience. For very well connections 1 second is enough. For world-wide connections it is better to set a much higher value. In best case the value is three times smaller than the definition for the "SSH" timeout settings (see the configuration description there). The request timeout is valid for both protocol types.

- UDP Retry Timeout [s]/[us]:
  The client retry timeout[9] is only valid for `UDP` . It defines the time after which a request is repeated again. For e-RemoteCtrl this setting is not required.

- Used RPC-Server:
  This column defines the internal name of the interface. This is only relevant for other software, where more parallel clients are used in the same `GUI` to connect to different servers. It must be one of the names, which are defined in the "RPC-Server" tab. For e-RemoteCtrl it must be `"econtrol"`.

Second tab: the connection settings for the RPC server

The second tab is used to configure the communication settings to connect to the server (see fig. 1.7).
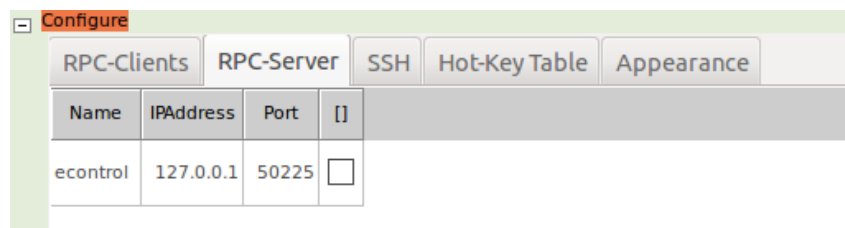
Figure 1.7: The configuration tab for the settings of the server part of the communication.

The following entries (columns) can be changed here and are in detail:

- Name
  "Name" defines the internal name of the server. This is only relevant for other software, where more parallel servers are used. For e-RemoteCtrl it must be `"econtrol"`, which corresponds to the definition in the "RPC-Clients" tab in the column "Used RPC-Server"

- IPAddress
  The `Internet Protocol (IP)` -Address of the server can be defined here. This must correspond to the settings in the "SSH" tab. If no `SSH` connection is used (direct connection), then the real address of the server must be used here. If a `SSH` connection tunnel is defined in the "SSH" tab, the address of the local host `"127.0.0.1"` must be used here (see the next configuration section).

- Port
  The server uses a dedicated network port number, which must be defined here. This must correspond to the settings in the "SSH" tab. If no `SSH` connection is used (direct connection), then the real port of the server must be used here. If a `SSH` connection tunnel is defined in the "SSH" tab, the forwarded port of the "System Access Settings" from the "SSH" tab must be used here (see the next configuration section).

---

[9]Also take a look into the idl2rpc.pl manual[1]

Third tab: the configuration of `SSH` tunnels for the communication

The `GUI` client can use automatically administrated `SSH` tunnels to communicate through firewalls. The third tab is used to setup these tunnels (see fig. 1.8).
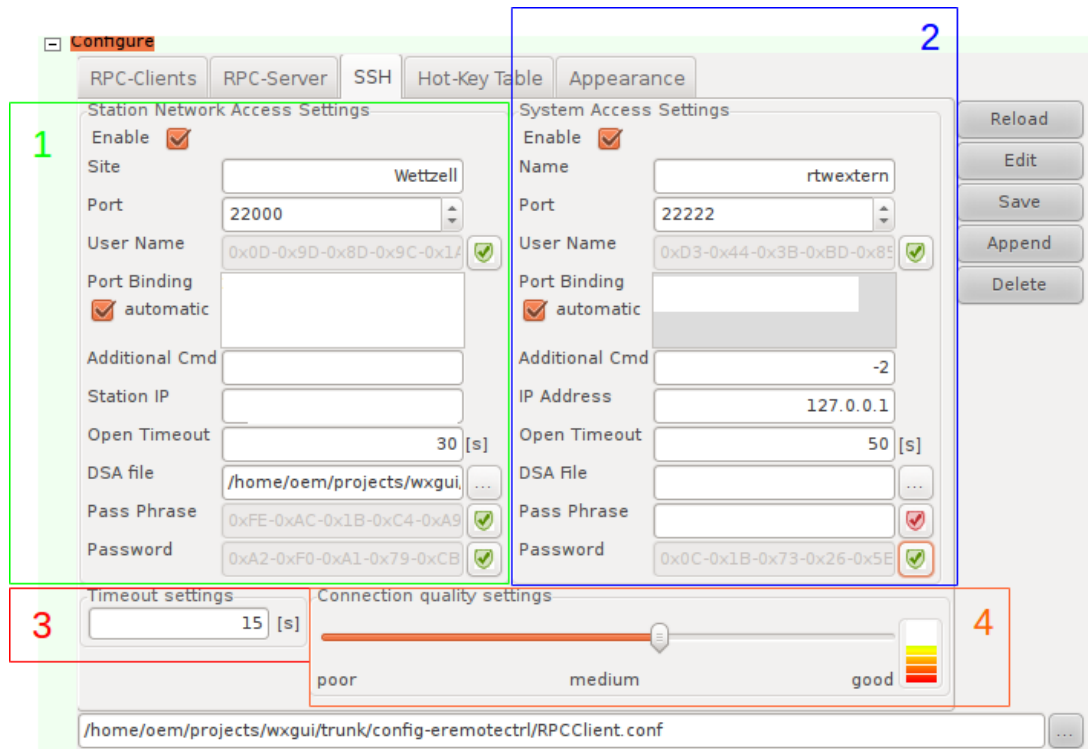
Figure 1.8: The configuration tab for the settings of the SSH tunnels.

The `SSH` configuration consists of four sub-sections.

- Section for the "Station Network Access Settings" to pass firewalls of an observatory with a tunneled `SSH` connection (see the green section in fig. 1.8). If this is used, also the "System Access Settings" must be defined. Each `SSH` setting section has the following elements:

  - Enable checkbox
    If this checkbox is enabled, all other controls for the individual `SSH` access are activated and are used for the connection to the e-RemoteCtrl server.

  - Site (Name)
    This text box contains the name of the computer or system, to which should be connected with the individual `SSH` -tunnel.

  - Port
    This text box contains the port of the `SSH` server on the remote system or firewall.

  - User Name
    This text box contains the user name on the computer with the `SSH` server (compare `ssh -l <username>`). In the e-RemoteCtrl versions with authentication the text is encrypted. It can be de- and encrypted with the button on the right of the text box (green is encrypted, red is clear text). The user name and password of the "System Access Settings" are furthermore the access and decryption data to decrypt the configuration file after saving it with new user and password settings. These data are also required for the first access to the remote control tab after starting the software.

– Port Binding
The port binding is a definition to set, which local port is tunneled to which computer (`IP` address and port) on the remote side (compare `ssh -L<localport>:<ip>:<remoteport>`). All requests to the local port are then forwarded to the computer and port, which is defined here.

– Port Binding (automatic)
If the automatic mode is active, an algorithm tries to determine the current port binding settings from internal definitions. It is currently recommended not to use this automatic setup.

– Additional command
Additional flags, options and parameters can be entered in this text box and are used as calling arguments for the `SSH` client.

– IP Address/Station IP
This textbox contains the IP-Address of the `SSH` server, to which a connection should be established with `SSH` .

– Open Timeout
The open timeout defines an additional waiting time, while starting up the connection to a `SSH` server. Within this time (in seconds) the server must respond and complete the login procedure. If this time expires, the connection attempt is interrupted and starts again.

– DSA file
If a `SSH` connection uses a key file (DSA or RSA), the path to this file must be entered here.

– Pass Phrase
If the key file uses a pass phrase, it must be entered into this text box. In the e-RemoteCtrl versions with authentication the text is encrypted. It can be de- and encrypted with the button on the right of the text box (green is encrypted, red is clear text).

– Password
This text box contains the password for the user on the computer with the `SSH` server. In the e-RemoteCtrl versions with authentication the text is encrypted. It can be de- and encrypted with the button on the right of the text box (green is encrypted, red is clear text). The password and the user name of the "System Access Settings" are furthermore the access and decryption data to decrypt the configuration file after saving it with new user and password settings. These data are also required for the first access to the remote control tab after starting the software.

• Section for the "System Access Settings" to access the `NASA` Field System `PC` via a tunneled `SSH` connection (see the blue section in fig. 1.8). For the elements see the explanations for the "Station Network Access Settings" section.

• Section to define the timeout for the `SSH` connection (see the red section in fig. 1.8). After the defined time (in seconds) the `SSH` connection is closed and re-established again automatically. The timeout settings should correspond to the settings of the "Request Timeout" in the "RPC-Clients" tab. In best case it is a value, which is three times of the "Request Timeout" in the "RPC-Clients" tab. The `SSH` timeout avoids a blocked `SSH` tunnel over long distance communications on the world-wide network.

• Section to adjust internal parameters to deal with different network and connection qualities (see the orange section in fig. 1.8). With the slider it is possible to select different steps of communication qualities from "poor" to "good". A well adapted quality avoids unnecessary reconnects.

The following sections describe, how to setup different connections with the different, available `SSH` tunneling combinations.

1. **Connection to the server without any `SSH` tunnel:** (direct connections are not recommended for e-RemoteCtrl, as authentication, opening of remote terminals, etc. are not possible here)
   For direct connections, it is necessary to disable all `SSH` configurations. This activates only the
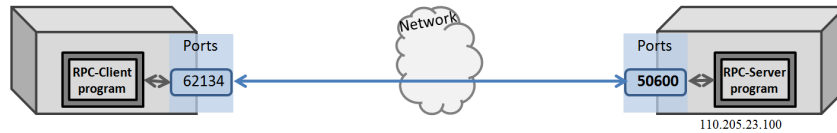
Figure 1.9: Schematic diagram to illustrate a direct connection between a client and server.
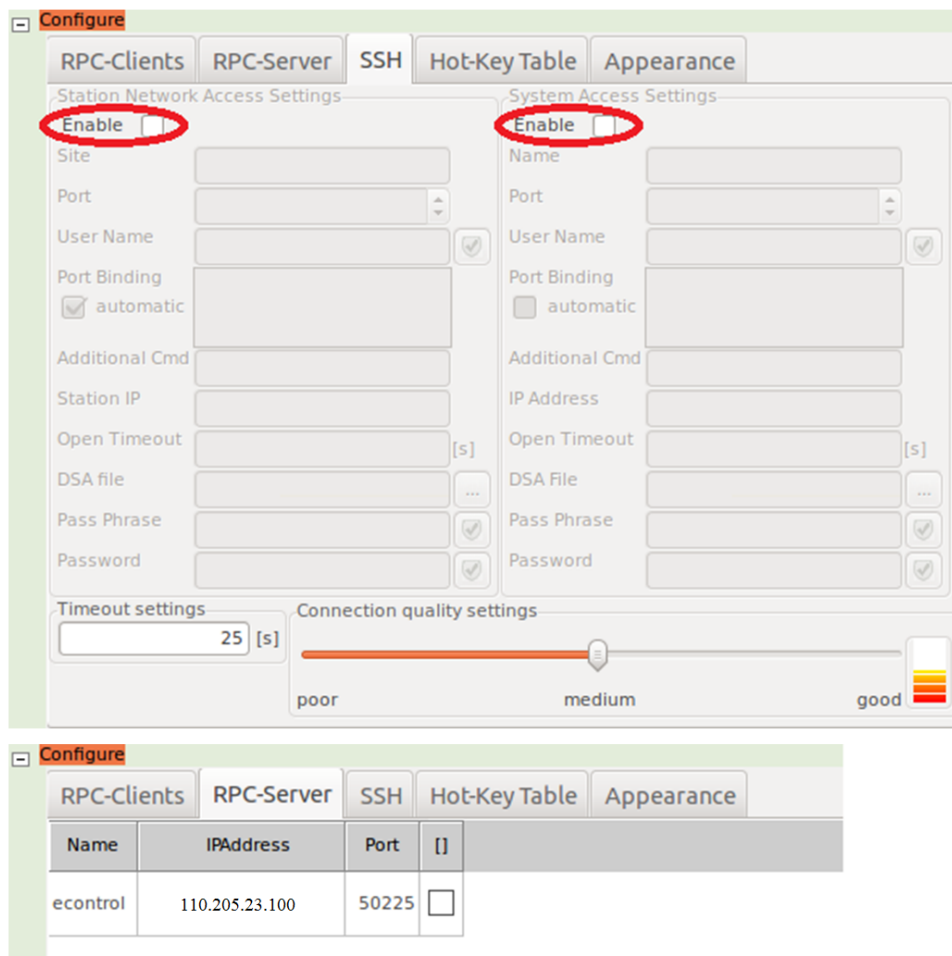
Figure 1.10: An example for a configuration of the "SSH" and the "RPC-Server" tab for a communication without any SSH.

    settings in the "RPC-Server" tab. There the real `IP` address of the e-RemoteCtrl server and the real port must be defined.

2. **Connection to the server with a `SSH` tunnel to access the `NASA` Field System `PC` :** This is the standard way for the most telescopes. It can be used, if the `NASA` Field System `PC` is directly accessible without a firewall, or if a firewall automatically forwards an external port without any intermediate components to the `NASA` Field System `PC` .

    For a `SSH` connection to the `NASA` Field System `PC` it is necessary to enable the "System Access
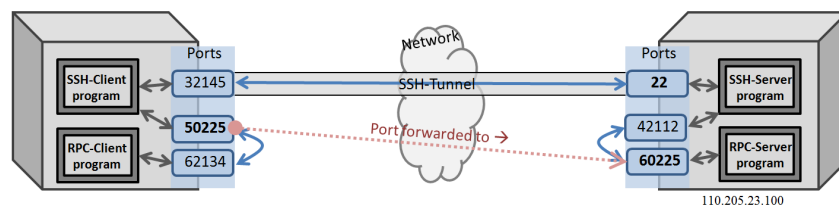
Figure 1.11: Schematic diagram to illustrate a connection between a client and server, using a `SSH` tunnel to access the `NASA` Field System `PC` .
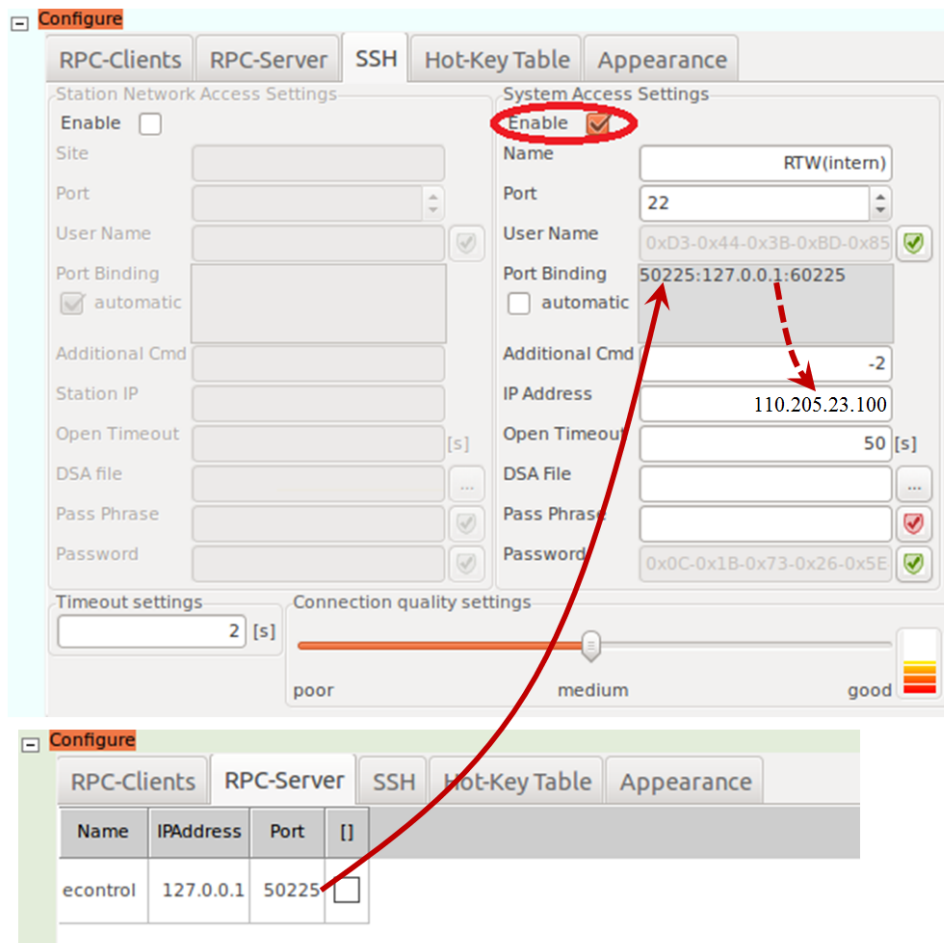
Figure 1.12: An example for a configuration of the "SSH" and the "RPC-Server" tab for a communication with a SSH-tunnel to the NASA Field System PC.

Settings". It is necessary to define a name for the connection, the port to the `SSH` server (standard is port "22"), an user name and a password for the computer with the `SSH` server, the `IP` address of the `SSH` server and the port binding. The port binding defines the forwarding of local ports (in fig. 1.12 port "50225") to the remote machine, which can be the computer with the `SSH` server (as in fig. 1.12 with address "127.0.0.1") or another computer in the same network ("Off-host forwarding"). Added to those `IP` address is the remote port, to which the local port is forwarded (in fig. 1.12 port "60225"). If necessary a key file with or without a pass phrase can be used and additional arguments for the `SSH` client can be defined.

For connections with a `SSH` tunnel it is necessary to change the "RPC-Server" tab, as the client

must now connect to the local port (in fig. 1.12 port "50225") at address "127.0.0.1", which is automatically forwarded with the defined `SSH` tunnel.

3. **Connection to the server with `SSH` tunnels to pass a firewall of an observatory and to access the `NASA` Field System `PC` :**
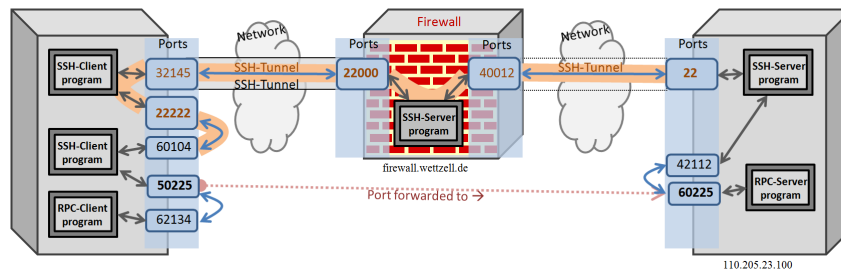
Figure 1.13: Schematic diagram to illustrate a connection between a client and server, using a first `SSH` tunnel to access an observatory firewall and a second `SSH` tunnel to access the `NASA` Field System `PC` .
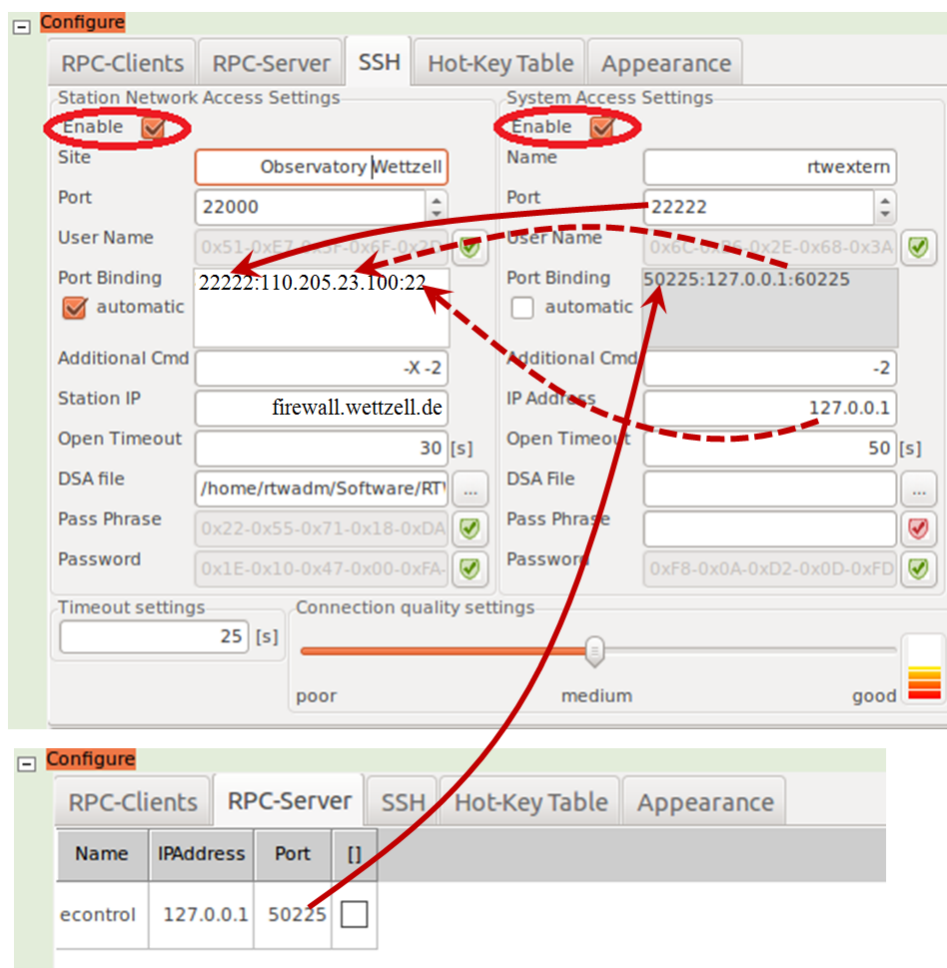
Figure 1.14: An example for a configuration of the "SSH" and the "RPC-Server" tab for a communication with a first SSH-tunnel to a firewall and a second, internal tunnel to the NASA Field System PC.

For a `SSH` connection to the `NASA` Field System `PC` via a second `SSH` connection, passing a firewall, it is necessary to enable both, the "System Access Settings" and the "Station Network Access Settings".

The "Station Network Acess Settings" creates a connection to a firewall of an observatory, which is for-warded to the `SSH` port of the `NASA` Field System `PC`. The "Station Network Acess Settings" defines the main tunnel through the firewall and requires to define a name for the connection, the port to the `SSH` server on the firewall (usually different from the standard port, e.g. here "22000"), a user name and password for the firewall (maybe in combination with the usage of a key file and a possible pass phrase), the `IP` address of the firewall and the port binding for the firewall. The port binding of the "Station Network Access Settings" defines the forwarding of local `SSH` ports (in fig. 1.14 port "22222") to the `SSH` server port on the `NASA` Field System `PC` (here at port "22" on the machine "110.205.23.100"). If this first tunnel to the firewall is established, another `SSH` client can connect to the new, local port and will be automatically forwarded to the port of the `SSH` server on the `NASA` Field System `PC`.

This forwarded `SSH` connection is then used for a second `SSH` tunnel, which forwards the real com-munication between client and server to the port of the e-RemoteCtrl server on the `NASA` Field System `PC`. This is defined in the "System Access Settings". The configuration of this is quite similar to the previously described setup with only one `SSH` connection. The only main differences are, that the port of the `SSH` server must now be the newly generated, forwarded, local port of the `SSH` tunnel through the firewall (in fig. 1.14 port "22222") and that the `IP` address of the used `SSH` server is now the local host "127.0.0.1", as it should use the newly generated and forwarded port on the local machine to connect to the `SSH` server.

With these two possibilities it should be possible to solve most of the combinations for an access to a system behind observatory firewalls.

Fourth tab: the configuration of hot keys

The fourth tab of the configuration display offers the possibility to define own hot-keys (see fig. 1.15), which start commands or programs on the remote `NASA` Field System `PC`. Each hot-key is a combination of the "CONTROL" key plus the "SHIFT" key plus a dedicated key letter (`"CONTROL"+"SHIFT"+<key_letter>`, e.g. `"CONTROL+SHIFT+p"`).
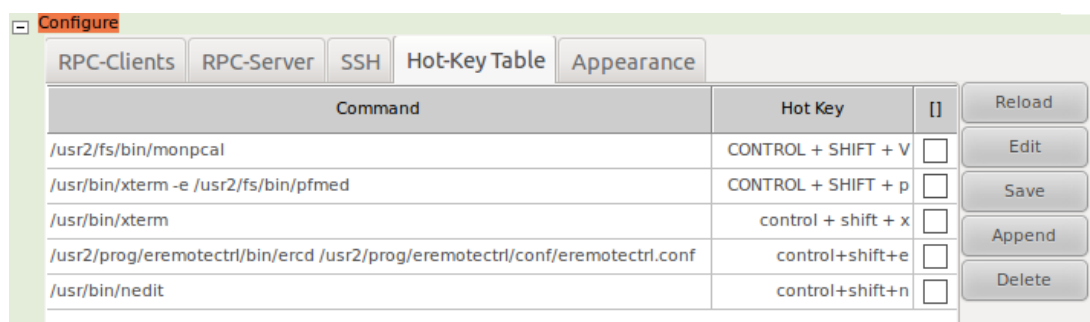
Figure 1.15: The configuration tab to define hot keys for the key combinations "CON-TROL"+"SHIFT"+<key_letter>.

Each hot-key is in an own line, where the first column defines the command (e.g. `"/usr/bin/xterm -e /usr2/fs/bin/pfmed"` to open the Field System editor "pfmed", see fig. 1.15), which should be executed on the remote `NASA` Field System `PC`. If command line programs are started, the us-age of a terminal window (e.g. "xterm") is required. The second column defines the hot-key with `"CONTROL"+"SHIFT"+<key_letter>` (e.g. `"CONTROL+SHIFT+p"` as shortcut for the Field System editor "pfmed", see fig. 1.15). The third column[10] offers a possibility to mark single lines, which are

---

[10]If long commands are used, it is necessary to scroll horizontally.

then changed, when the "Save" button is pushed, or deleted, when the "Delete" button is pushed.

After a hot-key is defined, the configuration is saved and the connection to the server on the NASA Field System PC is re-established again, holding the defined hot keys simultaneously starts the defined command on the remote machine.

The hot-keys use separate SSH connections, to start the remote programs. These connections are also controlled by the GUI , and will also be closed after quiting the GUI or if a timeout requires to re-establish the SSH connections. The status of these SSH connections can be controlled with the SSH status log window (see section 1.5.7 on page 29).

Fifth tab: the configuration of the appearance of the GUI

This tab enables the configuration of the appearance of the complete GUI (see fig. 1.16). This currently means the background color and the headline output.
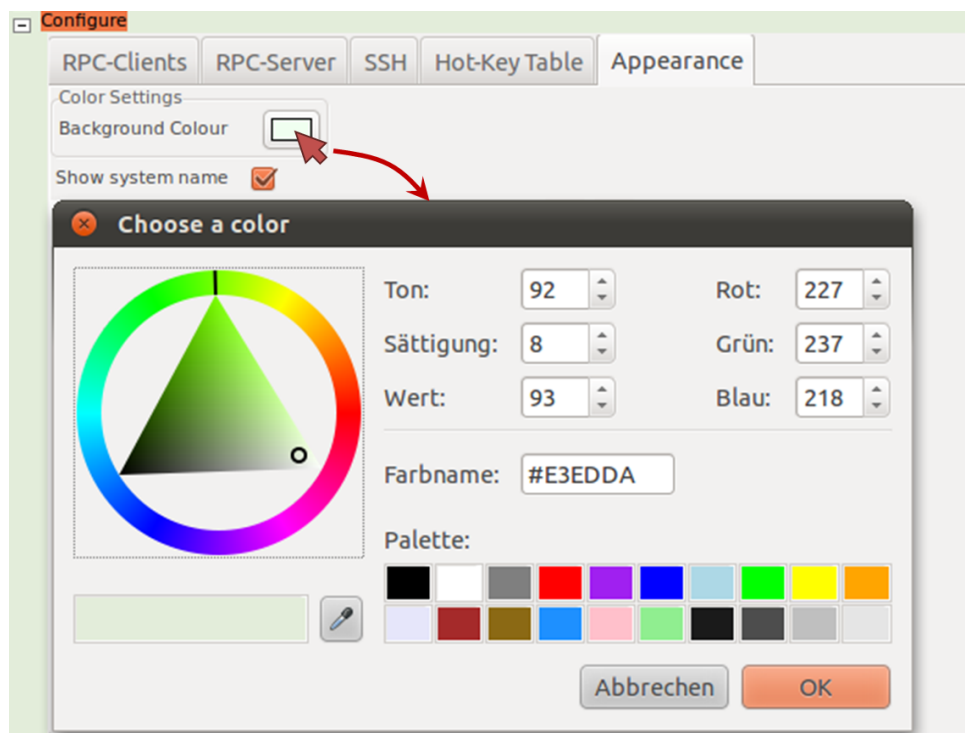
Figure 1.16: The configuration tab to change the appearances of the GUI.

The first configuration parameter is the background color of the GUI . Different background colors are quite useful, if different telescopes should be controlled from one operator room. Different colors for each telescopes separate them optically and simplify the identification. A click on the button with the currently used color opens a further dialog (see fig. 1.16), in which a predefined color can be selected from a palette or a completely individual color can be mixed using a "color wheel". After the selection of a new color, a push on the "OK" button adopts the new color. The selection then becomes activated after saving of the new configuration.

The next parameter of this tab allows to show the system/telescope name in the headline of each remote control tab. The name is taken from the "Status Monitor" of the e-RemoteCtrl GUI , which shows

the system name from the `NASA` Field System `PC` . Enabling the "Show system name" parameter, activates the output, after saving of the changed configuration.

1.5 Work with the graphical user interface (client)

This section describes the features of the `GUI` of e-RemoteCtrl. Start the `GUI` client as described in section 1.3 on page 6.

1.5.1 The stations tab

The startup screen shows the tab with the stations overview (always on the most left side of the tabs, see fig. 1.17), where a table with the most important information about all controlled stations is printed. In the current version of e-RemoteCtrl it is just a design demo without a functionality. Therefore it is currently not relevant for the control operations!
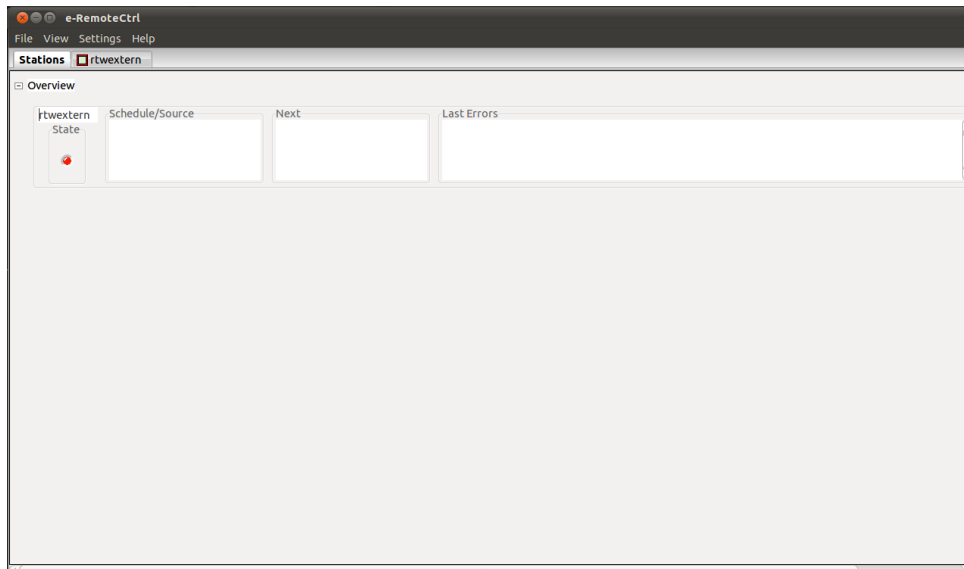
Figure 1.17: The tab with the overview about all controlled stations (currently not functional).

1.5.2 Login

If a `SSH` tunnel is defined with user name and password, those access data are also used to encrypt the sensible data (e.g. user data) in the configuration file. After starting of the e-RemoteCtrl `GUI` with such an encrypted configuration file, an user name and a password must be entered, if a remote control tab is selected (see fig. 1.18). Newer versions of e-RemoteCtrl obligatory require encrypted configuration files, to handle the authentication and authorization of user roles.

After the entering of an user name and a password, a push on the "OK" button checks the entered user credentials. If a valid user name and password was entered, the operator gets access to the remote control tab of the telescope. If wrong data are entered, an error dialog pops up, to inform the operator about the wrong user information (see fig. 1.19).
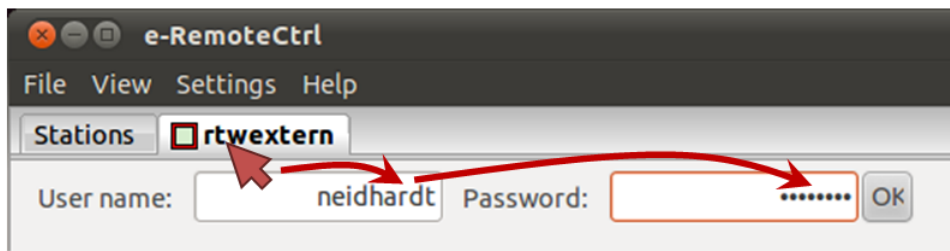
Figure 1.18: The login screen of the e-RemoteCtrl GUI, which requests a user namen and password for encrypted configuration files.
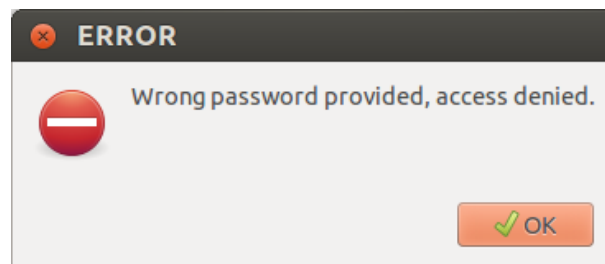
Figure 1.19: The dialog with the note, that the entered user account data are wrong.

1.5.3 The e-RemoteCtrl tab for one individual station

After the operator has entered a valid user account information, the tab opens, which shows the individual e-RemoteCtrl windows for one individual, activated, but not yet connected station. (see tab "rtwextern" in fig. 1.20). Each controlled station is represented with such an individual control tab with all the control elements, to run remote operations. The tab names are those names, defined in the SSH configuration of the "System Access Settings" for the individual control connection. Ahead of the tab name is a small rectangle. The border symbolizes the error status of the connection (red is not connected, green is successfully connected). The filling color of the rectangle is the same color as the background of the individual control tab. This can be defined in the "Appearance" configuration (see section 1.4.3 on page 24).

The main part of the control window consists of three areas:

- The headline area, where the station name and the currently operated schedule is shown, if it is activated in the "Appearance" tab of the configuration and if the client is successfully connected to the server on the NASA Field System PC .

- The left control area with the main control and monitor elements.

- The right control area with the control and monitor elements of the peripheral devices.

The seperator between the left and the right control area can be dragged from left to right, to point the main focus onto one of both sides (ideal on small laptop displays).
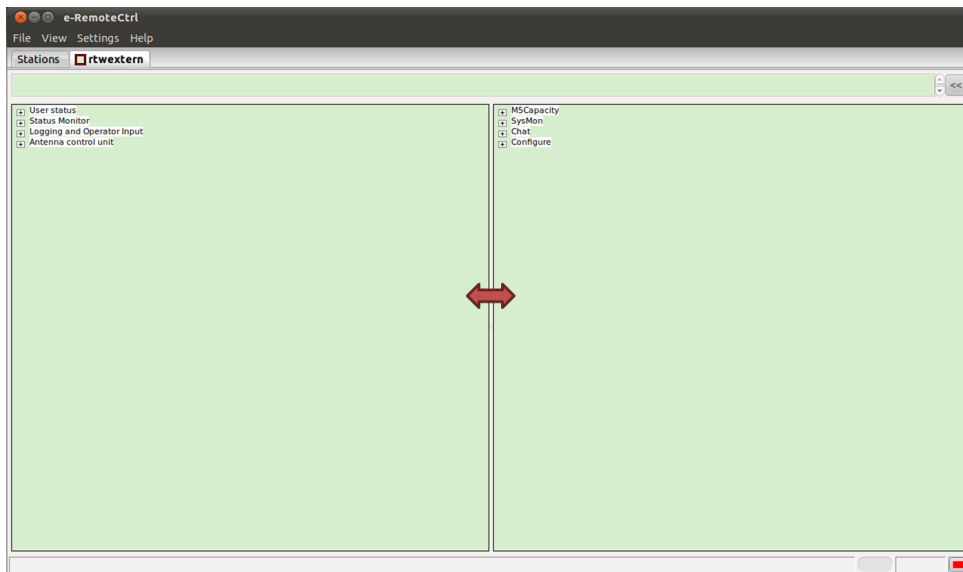
Figure 1.20: The control tab for one individual station, without an open connection.

1.5.4 Switch to another telescope station by changing the configuration

After a successful login it is directly possible to change between different remotely controlled telescopes or user accounts on-the-fly. This can be done by loading another configuration file via the configuration display. As shown in fig. 1.21, by pushing the file selection button on the lower right edge of the configuration display, a file selection dialog opens. There the folder of the new configuration file can be selected with the explorer menu on the left. After selecting the new configuration file, an user authentication is requested by an additional authentication dialog box. If the user name and password is entered correctly, the new file opens and can be used for a connection.
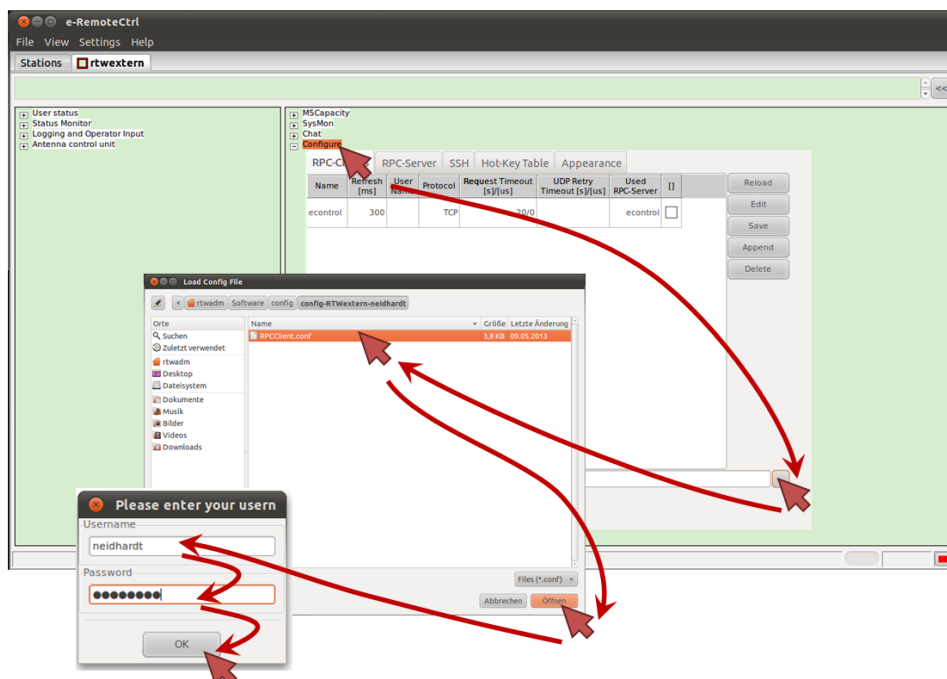
Figure 1.21: The procedure to change to another telescope control by changing the configuration.

This mechanism must currently also be used if the user must be changed for one connection. It means that each user must have its own configuration file for one individual connection to a telescope.

1.5.5   Adding additional control tabs for a parallel remote control of different telescopes

The client `GUI` can also be used to control several telescopes parallel. Therefore it is necessary to open the menu "`File`" of the main menu bar and then the sub-menu button "`Add Telescope`". It opens a file selection dialog, where the folder of an additional configuration file can be selected with the explorer menu on the left. After selecting the new configuration file an additional tab is created for the new, parallel controlled telescope. By clicking on this new folder, it opens the already described login window (see section 1.5.2 on page 25). Fig. 1.22 demonstrates the procedure to add an additional telescope for the parallel control of them.
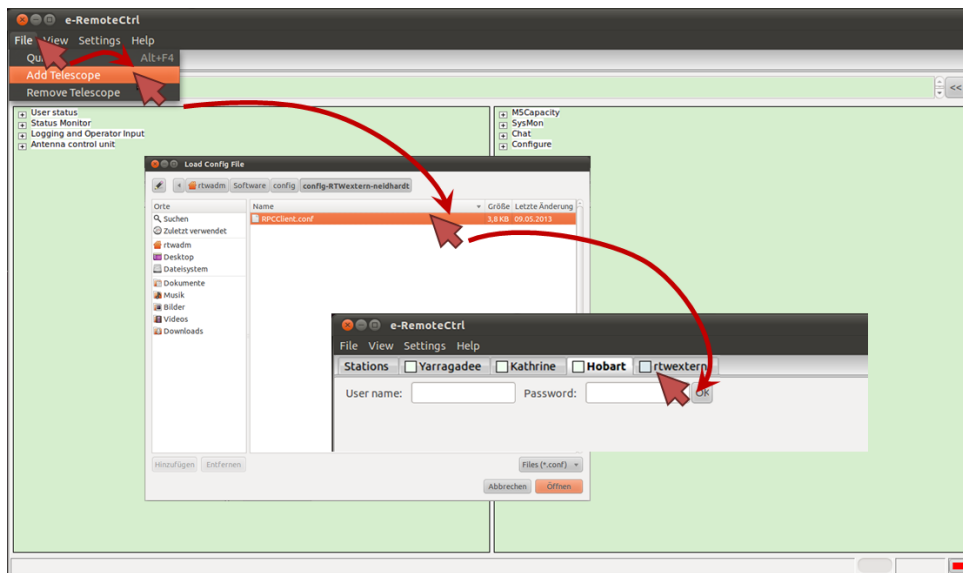
Figure 1.22: The procedure to add an additional telescope tab to control several telescopes in parallel.

After login, the new tab can be managed as described for the previous one.

1.5.6   Closing the control tab for one individual remote controls

A control tab for one individual telescope can also be closed again. Therefore it is necessary to open the menu "`File`" of the main menu bar and then the sub-menu button "`Remove Telescope`", while the control tab is selected, which should be closed.
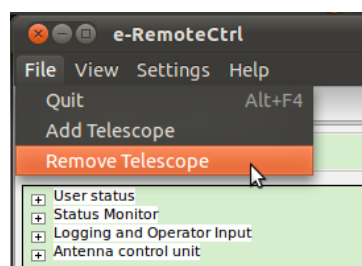
Figure 1.23: The procedure to remove a telescope tab.

1.5.7 Open a connection

After the login, no connection is opened. The operator must actively select one of the control displays (one of the "+" marked sub-controls). Fig. 1.24 illustrates this. Here the operator has clicked on the *Status Monitor*-control (1). This opens the SSH connection-dialog (2), which shows the status of the connection establishment for the different SSH connections, defined in the configuration file.
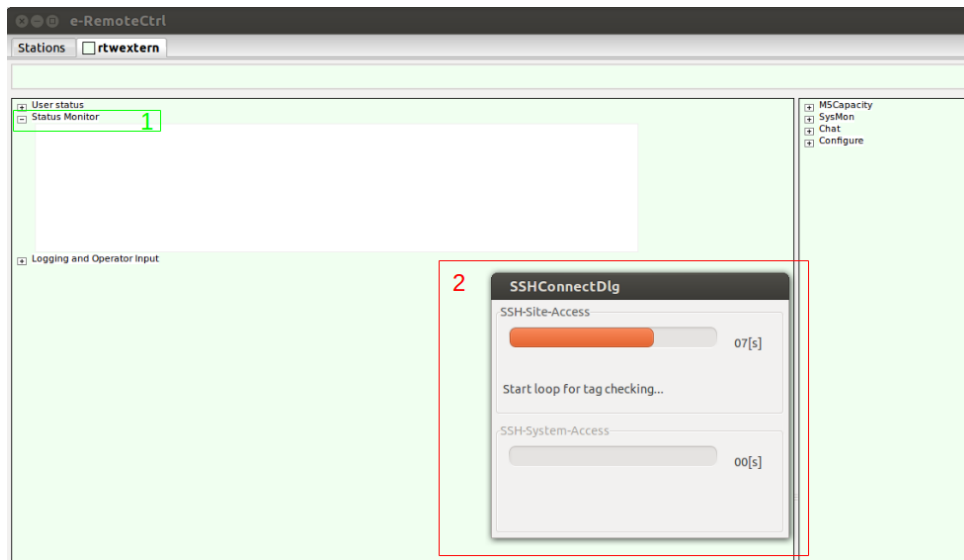
Figure 1.24: Open a connection.

The current connection state is always shown in the status-bar (at the right side, bottom). All possible states are listed in table 1.35.

	No connection: The client is currently not connected to a telescope and does not try to establish a connection.
	Connection setup and initial communication with server: The client tries to establish the data connection to the server.
	Data connection successfully established and client communicates: The data connection is established successfully and data can be requested by the client from the server.
	Server not reachable: The server is not accessible by the client, e.g. because of wrong configuration parameters (e.g. port numbers or IP addresses), network failures, routing problems, and so on.
	Version mismatch: The version numbers (*idl2rpc*-generator version and/or IDL file version) of the client and server are different.
	SSH and data connection successfully established and client communicates: After a successful connection via SSH an additional key symbol is shown to the previous status symbols (here after a successful data connection).
	SSH connection successfully established, but server is not reachable: After a successful connection via SSH an additional key symbol is shown to the previous status symbols (here after a not successful data connection).

Table 1.35: Indicators for the connection status

The status of the `SSH` connections (e.g. for a tunneled data communication or a sent remote shell command) can also be requested via the `SSH` status display. It can be opened with the small arrow on the right corner of the e-RemoteCtrl window (see fig. 1.25). It opens a list with all opened `SSH` connections to the server, which are assigned to the communication to the telescope of the current control tab. The operator can then select one of the connections with a double-click by the mouse. This opens the communication output from the `SSH` client. This status can be used to identify communication problems.
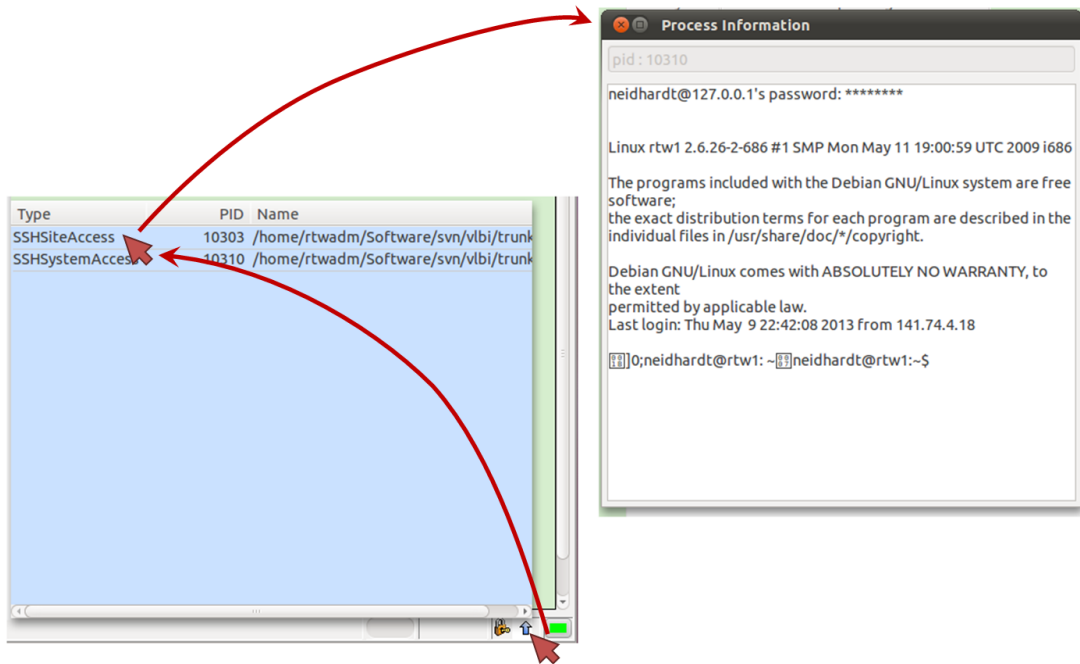
Figure 1.25: The SSH status display.

The following sections will now explain the individual control tabs in detail.

1.5.8   The "User status" display

The "user status" display shows all users, currently logged in to the telescope system. It shows two columns, where the first is for the user names and the second is for the assigned dynamic roles (see fig. 1.26).
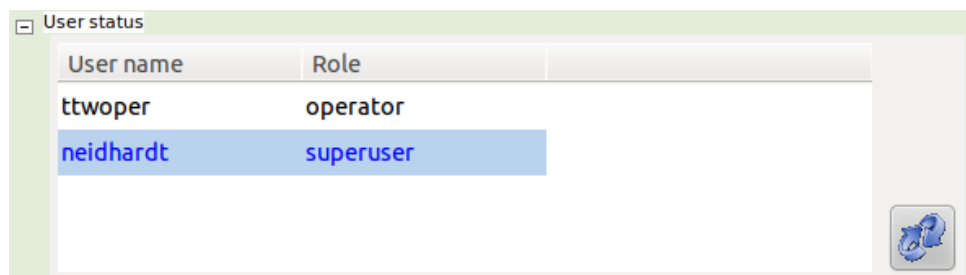
Figure 1.26: The "User status" display.

The users and roles are defined by the server in its configuration file (see section 1.4.2 on page 10)[11]. The user roles have different rights to request data or to command orders. Each remote user plays a

---

[11]The following is taken from the "Habilitation" thesis of A. Neidhardt, which will be published soon.

dedicated role, which is assigned to him statically during the whole time of access. With these static roles the access control can be arranged like a pyramid. The most users have only a few rights, e.g to read the current states or observation data. Then role by role the number of rights increase, while the number of users with these rights decrease. Each user on a higher level has the rights of the previous role definition plus additional permitions. Finally there are only a few users (or better only one user), who have superuser rights, to change the system itself. Following roles are possible:
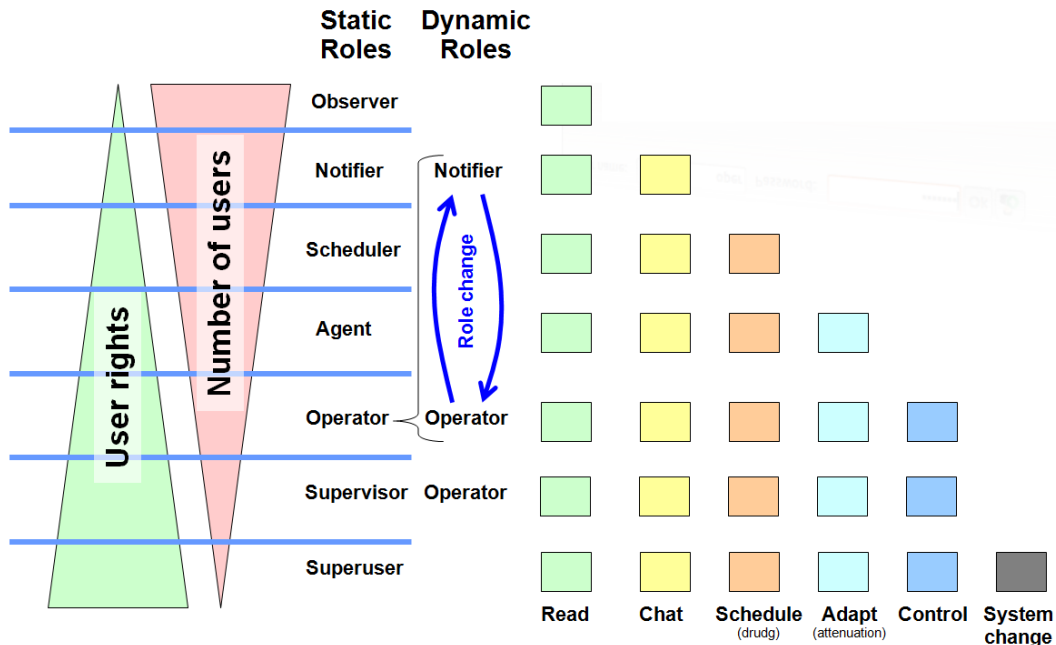
Figure 1.27: The user roles and their rights.

1. **Observer role**: An observer is the role with the least rights. He is just allowed to request data from the current observation. He cannot send any orders and is even not able to participate in chats with the other users. In principle he is just a passive listener.

2. **Notifier role**: The notifier is an observer with the additional right, to participate at the chat or to set some indicator values, to notify other users about critical states or about necessary changes.

3. **Scheduler role**: A scheduler is a notifier, which can change observation schedules. He has no direct access to the control or to the controlled sub-system, like connected hardware. But he can change parts or the whole sequence of commands in an observation schedule, to optimize the observation. These changes have an indirect impact to the controlled devices, when the commands are processed. But the commands are mostly procedures with sub-commands, where the required safety of the system is managed. This means that schedulers can't break the system safety with their changes, if locally is everything in a safe state.

4. **Agent role**: An agent is a scheduler with the additional possibility to change dedicated system parameters on-the-fly, which have no direct influence to the hardware safety. An example are attenuations, band widths are sampling rates in the back-end equipment of the `VLBI` system.

5. **Operator role**: The operator is the first role, which can get access to controlling functionalities of the system. He has all rights of an agent and additionally the possibility to send all necessary commands, to run the complete system, if he got the permissions from a local responsible operator. Therefore control rights are just temporarily for him.

6. **Supervisor role**: The supervisor is an operator, who has the rights to control the system permanently. Usually personnel staff at the telescope uses the supervisor role.

7. **Superuser role**: The superuser has all supervisor rights and is therefore usually a member of the telescope staff, but is additionally a user with root rights on the controlled system.
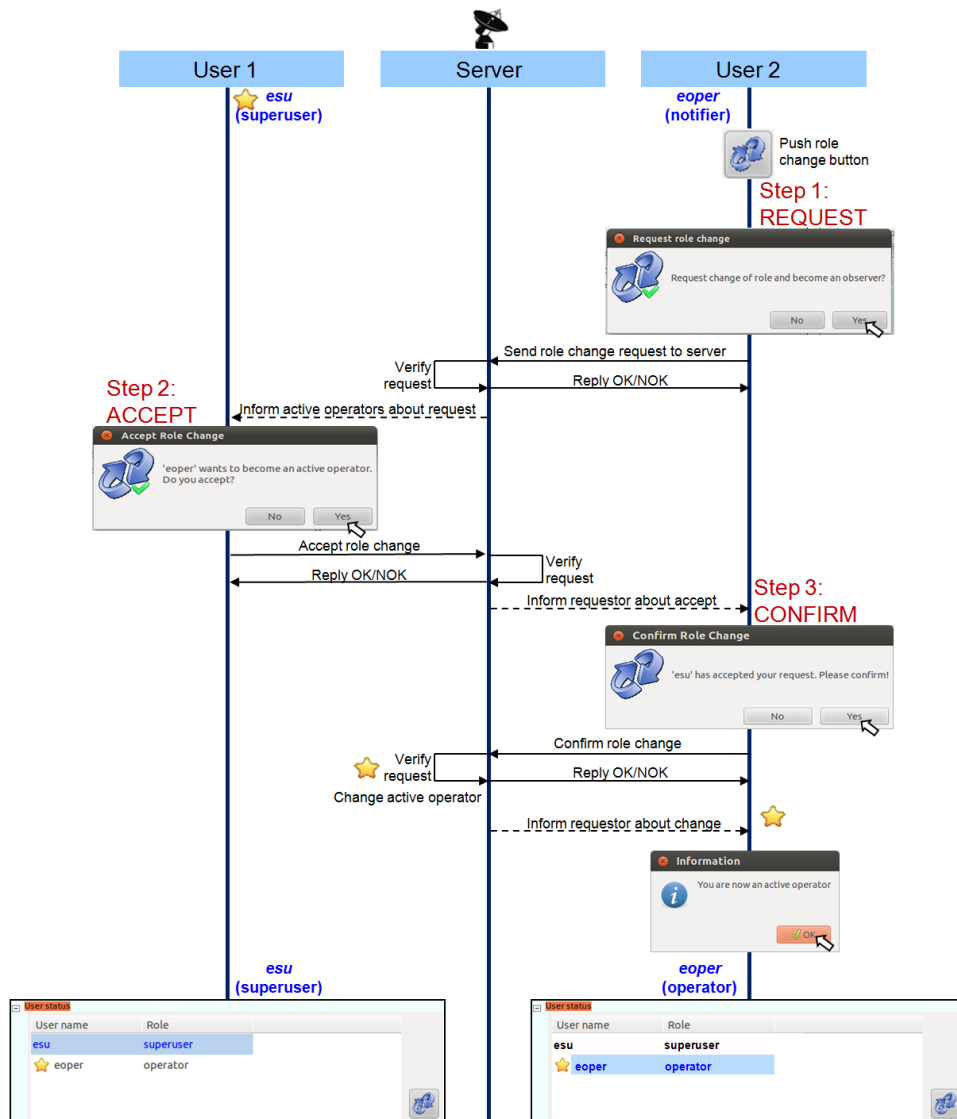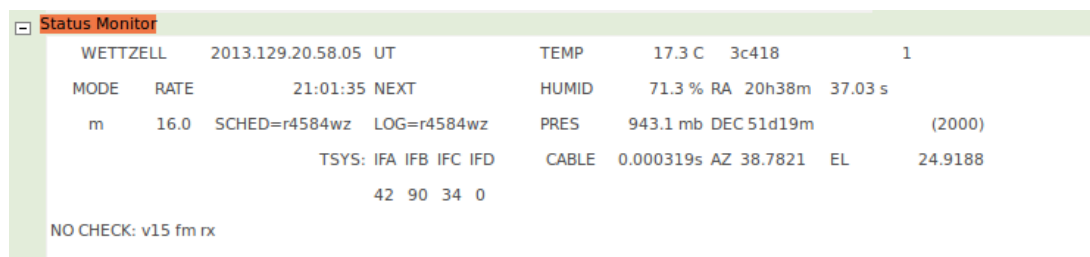
Figure 1.28: The three-way-handshake to change the dynamic role of a remote operator.

Additionally there is one role, which requires a separate viewing: the operator role. This is the role for all remote operators, who do not belong to the staff of the telescope. To avoid accidental interactions to the system by foreigners, which would be a safety risk, the remote operators just get temporary access to the whole functionality. Only when they really need to control the system, e.g. if they take over a remote shift from another observatory, a local operator gives them the permission to access the control functionality. This permission is only for that task of remote control, includes all necessary authorizations and can be revoked again by local staff. There is only one permitted remote operator with the control rights at a time. Without the permission the operator is just a notifier.

The handover of the control rights follow a dedicated work-flow, which is comparable to a three-way-handshake in the `TCP` . It forms the three-way-handshake to request remote access rights for control (see fig. 1.28). An operator, who has just notifier rights because of his dynamic role, requests the control rights by pushing the request button in the `GUI` . It opens the request dialog, where the operator can send the request or cancel it. The request is registered by the server on the controlled machine, where the `NASA` Field System is controlled. Immediately after the registration of the request all users with higher rights (who are supervisors or superusers and therefore members of the local staff or who are currently registered remote operators) receive a message to inform them about the request. It opens a dialog on each of these user machines, where they can accept the role change. If one of the users with higher rights accepts, a message is sent to the server. It releases the request dialog on all machines and sends a message to the operator, who requested the role change, to ask for a confirmation. Only if he confirms this message in another dialog, the server processes the role change and all users are informed about the new owner of the rights with the ⭐ symbol in front of his name. Nevertheless each supervisor or superuser can revoke the permitted rights from a remote operator with one step, by pushing the role change button, so that local staff can immediately interrupt the remote control, e.g. in case of an urgent return (no hard emergency is supported).

1.5.9 The "Status monitor" display

The "Status monitor" display is an equivalent of the display of the regular `NASA` Field System (see fig. 1.29).

Figure 1.29: The "Status monitor" display.

The display shows:

- on the first line: the station name, the current time, the current temperature, the current radio source name and the slewing status

- on the second line: the time of the next operation, the current humidity and the current right ascension

- on the third line: the current mode, rate, schedule name, log file name, the current pressure, the current declination and the year of the used reference frame

- on the fourth line: the cable delay, the azimuth pointing position and the elevation pointing position

- on the sixth line: the system temperatures of the intermediate frequencies

- on the seventh line: the shortcuts of the monitored devices

For more details see the manual of the `NASA` Field System.

The "Logging and Operator Input" display
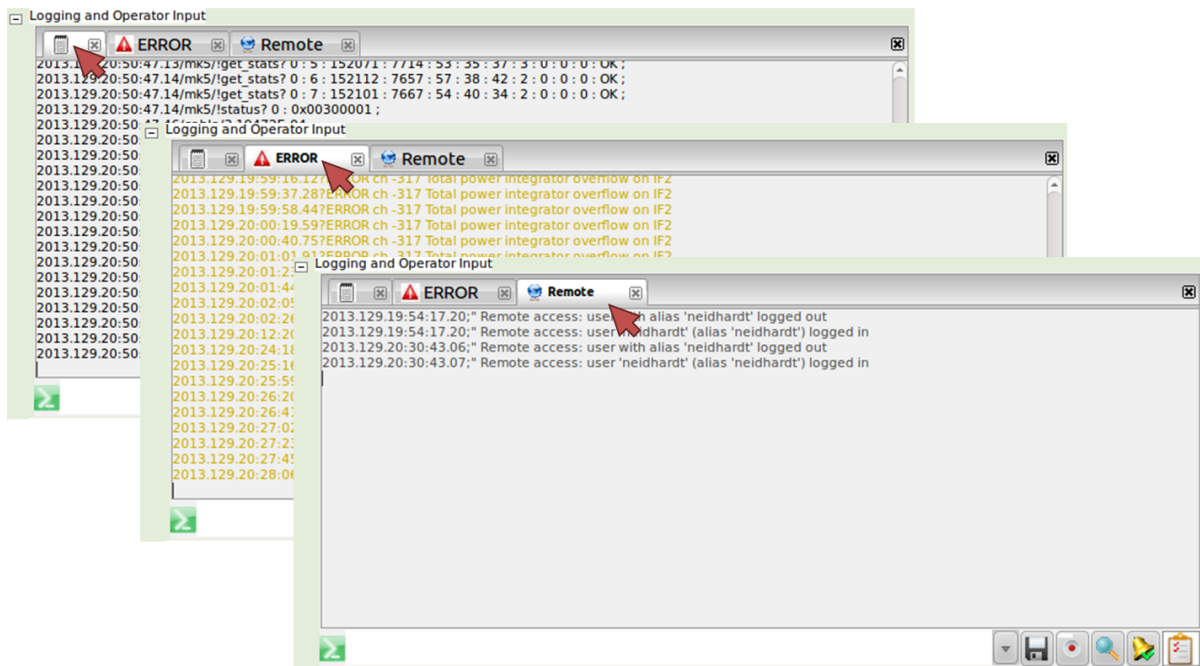
Figure 1.30: The three different static filter tabs of the "Logging and Operator Input" display.

The "Logging and Operator Input" display shows the log file output and the outputs of different pattern filters. Additionally it is the place, where the operator can enter his commands, which are then sent to the server and processed by the NASA Field System. In the standard setup the display offers three different tabs with the output of the static filter patterns (see fig. 1.30):

1. the log file output (currently, only the log file content is printed and local outputs of the NASA Field System to the standard output of the terminal are not available), which shows the latest 1000 lines of the last log entries

2. the error filter, where only the error, warning and alert messages appear

3. the remote control messages filter, where all e-RemoteCtrl relevant status messages appear

In addition to the static filters, the operator can define own dynamic filters, by clicking onto the loupe button ![loupe] (= filter button). It opens an additional input field, where a text pattern can be entered. After pushing the return button, an additional tab is opened, where only the filtered information appears (see fig. 1.31). The tabs can be closed again by pushing the ![close] button.

The display is also used, to send commands to the NASA Field System. The commands can be entered into the text field behind the ![icon] sign and must be confirmed by pressing return. Allowed are all known NASA Field System commands and the defined station commands on the Field System machine. Only these commands, which open a local display on the remote NASA Field System machine should be avoided or are directly blocked by the server. Additionally all entered commands are registered in a command history. Clicking into the command text box and pushing the up/down arrow keys on the keyboard allows the manual selection of historic, already entered commands. Using the list button behind the text box opens the history command list, in which a previous command can be selected with the mouse or with the arrow keys (see fig. 1.32). Internally, the last 50 commands are stored (even if the program is
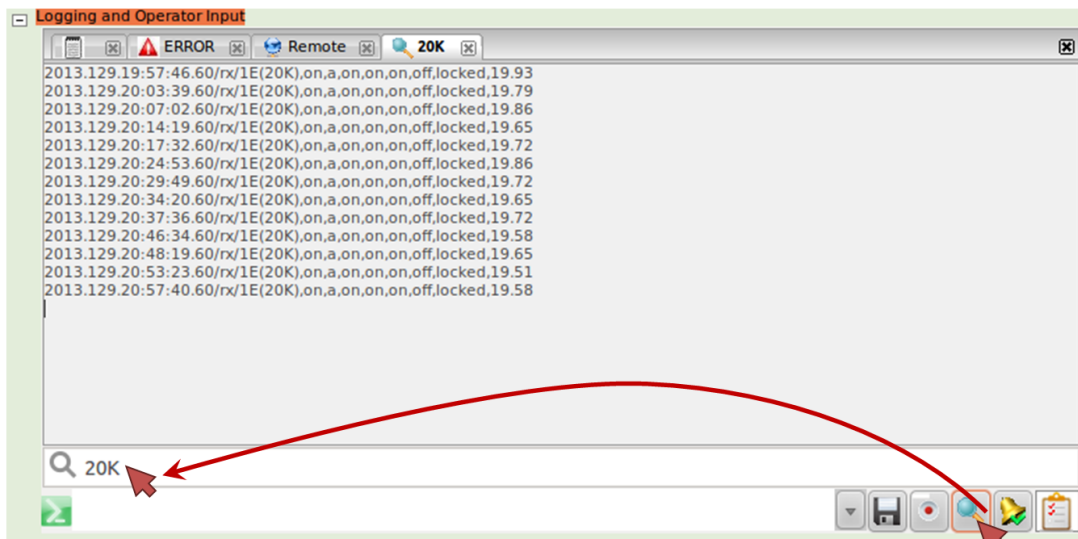
Figure 1.31: The usage of additional, dynamic text filters.
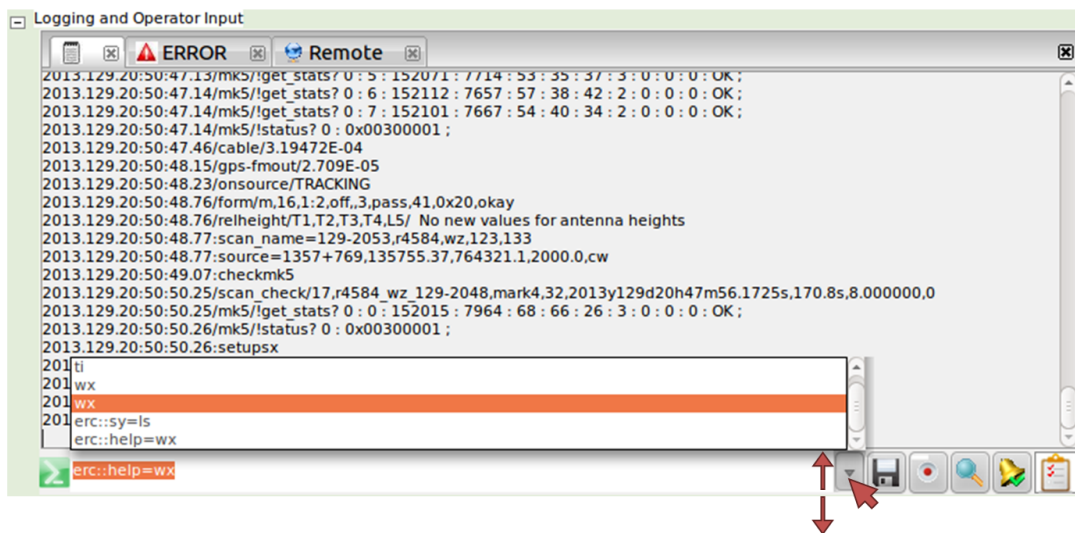
Figure 1.32: The input of commands and the usage of the command history.

restarted).

In addition to the standard Field System commands, e-RemoteCtrl defines additional commands. These are:

- "*erc::help=<command>*": It opens another display, which shows the known help pages of the NASA Field System in a HTML version. Additionally all defined HTML help pages in the e-RemoteCtrl server can be requested (see fig. 1.33).
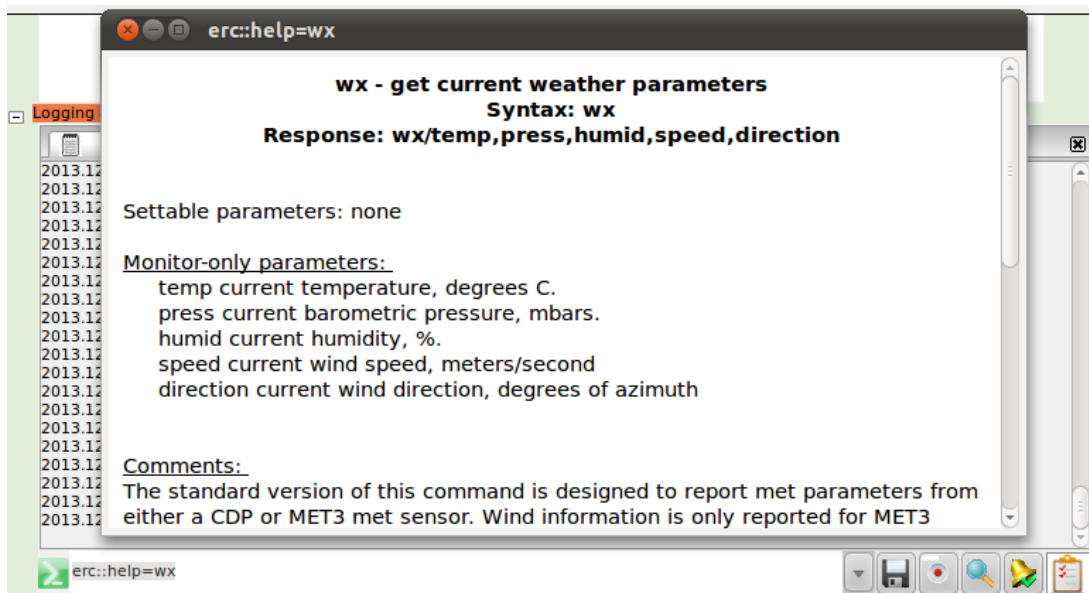
Figure 1.33: The help command.

- "*erc::sy=<command>*": This is the replacement of the system command functionality, known from the NASA Field System, which opens a SSH connection to run a remote command in a shell on the Field System computer, which display is forwarded to the client machine, on which the operator started the command. The shell output of the started program appears in an additional tab of the "Logging and Operator Input" display, similar to the output of search filters (see fig. 1.34).

Figure 1.34: The system call command.

- "*erc::checklist=before*", "*erc::checklist=during*" or "*erc::checklist=after*": These commands allow to open checklists, which are defined in the configuration file of the server (see section 1.4.2 on page 12). The checklists are defined to check activities before, during and after the observation. The checklists are usually opened automatically, according to dedicated patterns in the log file or after defined time intervals. But with the command they can be activated on demand by the operator. Alternatively, the checklist button can be used, which opens a drop-down list with the commands, where one can be selected with the mouse (see fig. 1.35).

Note: The input of commands is only allowed for users, having the necessary rights. In the case that
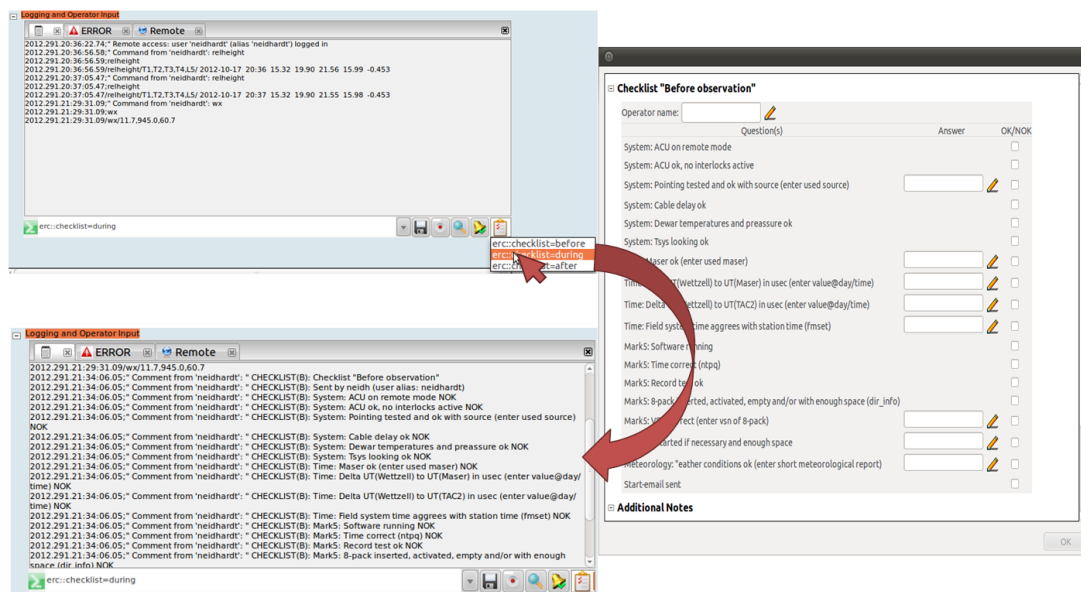
Figure 1.35: The checklist command.

an user without the required permissions tries to send a command, it will be rejected and nothing happens.

On the right lower corner of the display are additional buttons, which support dedicated operator functionalities (see fig. 1.36):

Figure 1.36: The control buttons of the "Logging and Operator Input" display.

- The first button opens a file system dialog, where a filename can be entered, to which the complete current content of the log file in the display is written to.

- The second button opens a file system dialog, where a filename can be entered, to which the content of the log file in the display is permanently recorded to (then the light turns to bright red).

- The third button opens the already described search pattern matching line.

- The fourth button activates the sound for alarms ("beep") according to the definitions in the server configuration file (see section 1.4.2 on page 12). If the sound is on, a green check-mark will appear on the right, lower corner of the bell-symbol.

- The fifth button activates the checklist drop-down list, as already described.

1.5.10 The "MK5 Capacity" display

The "MK5 Capacity" display is an equivalent of the display of the regular NASA Field System, if it is opened with the classic view (see fig. 1.37). It shows the status of the two 8-packs in the Mark5 recording system with the 8-pack number (VSN), the remaining recording time, the remaining recording space, the filled volume in percent and the time of the last check.

Figure 1.37: The classic view of the "MK5 Capacity" display.

The display additionally offers a more graphical view of the Mark 5 with pie charts (see fig. 1.38). The operator can individually switch between those two views, using the small tabs in the lower left corner of the display (Classic FS View Graphical View).
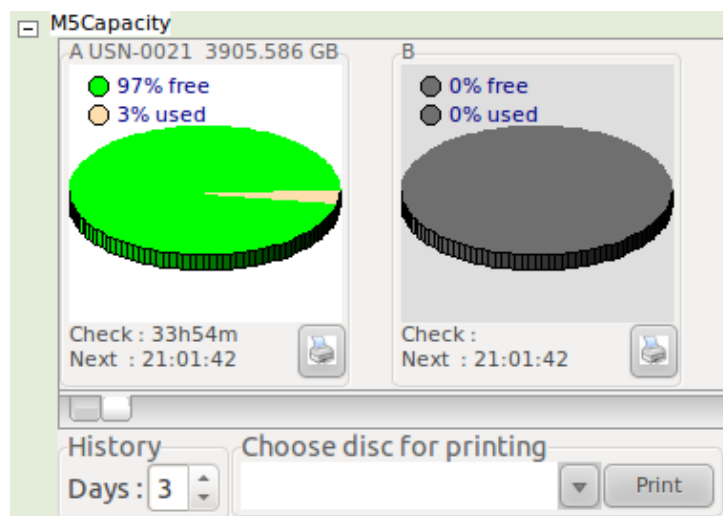
Figure 1.38: The more graphical view of the "MK5 Capacity" display.

The display also allows the printing of labels for the 8-packs, similar to the command "`drudg <logfilename>`". To print a label for a dedicated experiment, it is necessary to select a number of days in the log file history filter on the lower left side of the display. After this filtering it is possible to select a log file from the drop-down list (only log files in the NASA Field System folder "`/usr2/log/`" are shown). By pushing the print button on the right, the label is printed on location of the NASA Field System machine. The complete process is shown in fig. 1.39.
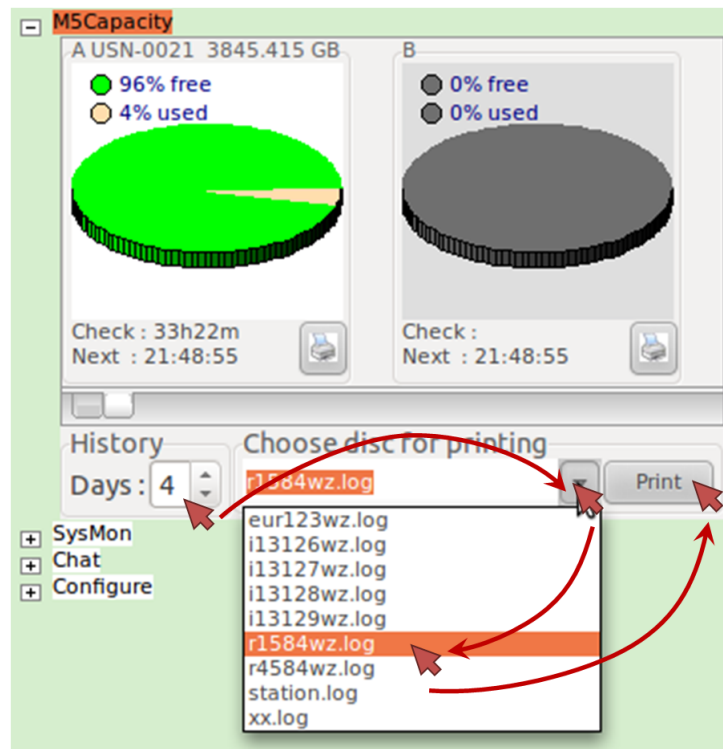
Figure 1.39: Printing labels with the "MK5 Capacity" display.

1.5.11 The "SysMon" display
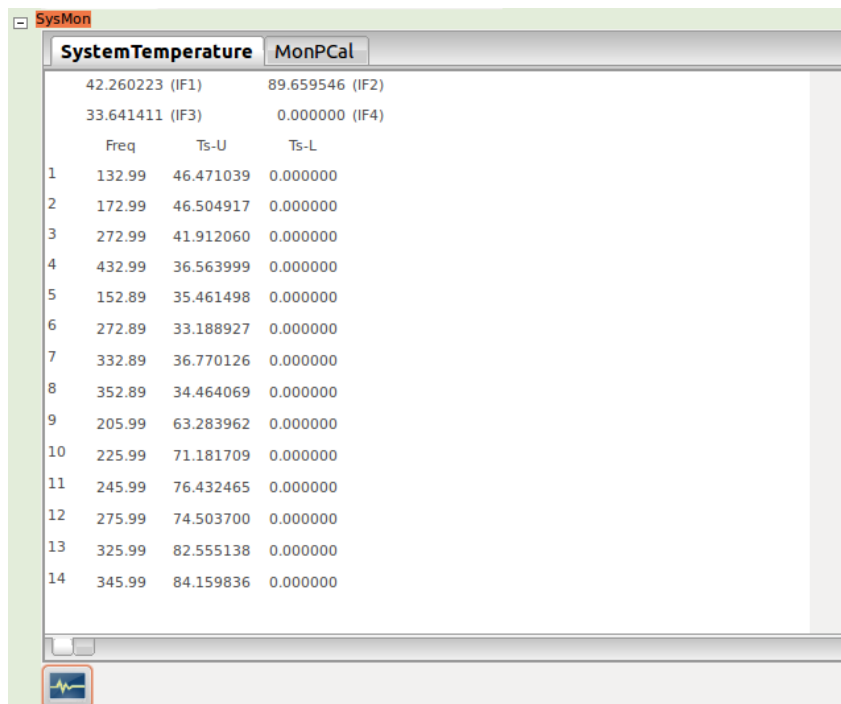
Figure 1.40: The "System Temperatures" tab of the "SysMon" display with the classic view.

The system monitoring display "SysMon" shows all system relevant system states. This includes the system temperatures and the phase calibration data (MonPCal) from the NASA Field System, which are

displayed in the same way as in the Field System, if the classic view is selected (see fig. 1.40). It shows the set frequencies and the system temperatures of the upper and lower band of the 14 video or baseband converters. Additionally the system temperatures of the four intermediate frequencies (IF1 - IF4) are shown.

The display additionally offers a more graphical view of system temperatures with a plot of the temperature values over time (see fig. 1.41). The operator can individually switch between those two views, using the small tabs in the lower left corner of the display (Classic FS View ➡ 🔲 ⬅ Graphical View).

After selecting the graphical view, the display for the system temperatures is split into two parts: the output selection panel on the left and the output plot on the right. All possible system temperatures can be selected for plotting in the selection panel. All selected channels appear as plot line in the right plot panel. The different outputs can be differed by its assigned colors.

To enable a better view on the graph, it can be expanded to the complete width of the display, using the expanding button `<<`. The view can be shrunken again to the original situation, using the shrink button `>>` (see fig. 1.41).
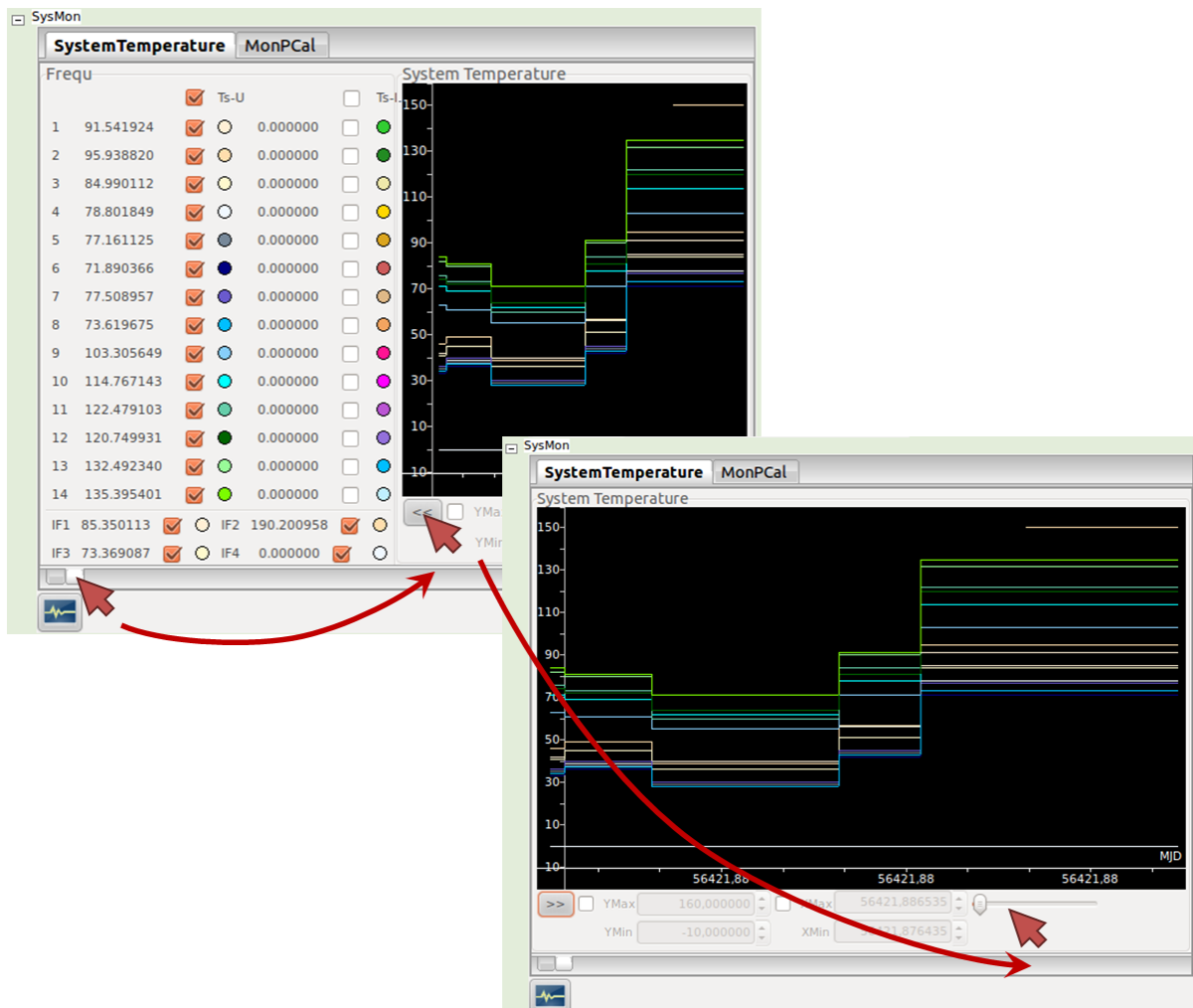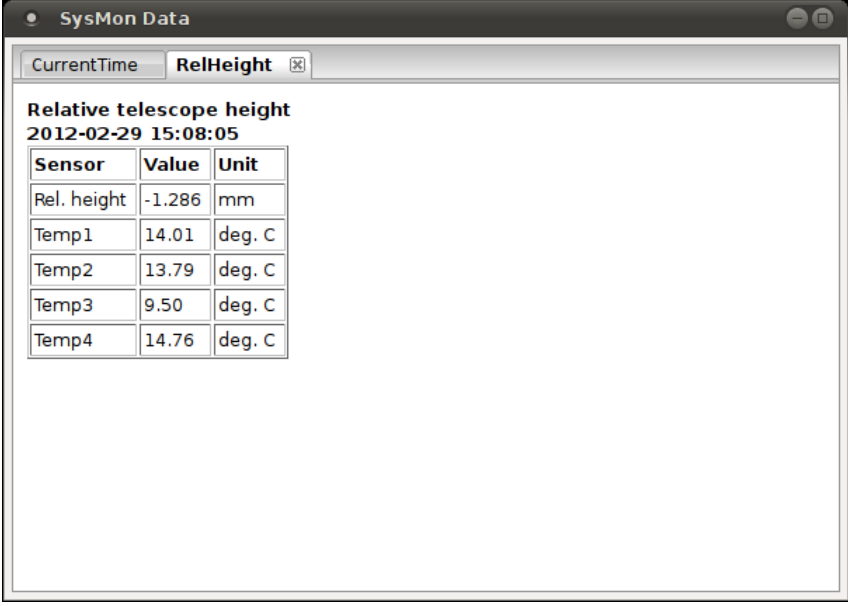
Figure 1.41: The "System Temperatures" tab of the "SysMon" display with the graphical view.

In the expanded view it is possible to manipulate the plotted graph. With the slider on the right (see fig. 1.41), the operator can individually zoom into the graph. If it is necessary to select a dedicated interval in time or in the values, the x- and the y-axis can be limited with a minimum and a maximum value, after the individual option is selected (see the setting elements on the left of the zoom slider).

The display also offers the output of station specific, individual system monitoring data in text and HTML format. The station specific system monitoring output can be opened with the ⊞ button. This opens an additional window with the station specific outputs (see fig. 1.42). Each individual, station specific monitoring point is represented as one single tab. The programming of such individual system monitoring points in the e-RemoteCtrl server is explained in section 2.1 on page 46. If no station specific monitoring point is programmed, only the demo output with the Unix time of the NASA Field System computer is available.

Figure 1.42: The output of station specific system monitoring points with the example of the relative height and radio telescope tower temperatures at the 20m radio telescope Wettzell.

The "Chat" display

The "Chat" display can be used, to communicate with other operators of the same system. It consists of a text panel with all received chat messages as communication log. Messages can be sent, by entering a text messages behind the ▶ sign. A message is sent after pressing the return key on the keyboard. With the check box behind the text field for the message input, the sent messages can also be added to the log file. Usually this box is activated, as all messages should also be logged like in a "flight recorder" of an aircraft. For the short periods of a "private" communication, the writing to the log file can be deactivated.

Each time a new message is received, a sound is released to notice the message arrival to the operator. Each message in the communication text panel starts with the user name, from whom the message comes from. Own messages are in blue color and the user name is marked with an "*".

Local operators in front of the classic NASA Field System can also participate to the chat. If they enter a comment with the structure "" `erc::chat: <chatmessage>`" the server filters the text messages and sends it to the chat display with the user name "`oper«local»`" (see fig. 1.43). Entering chat
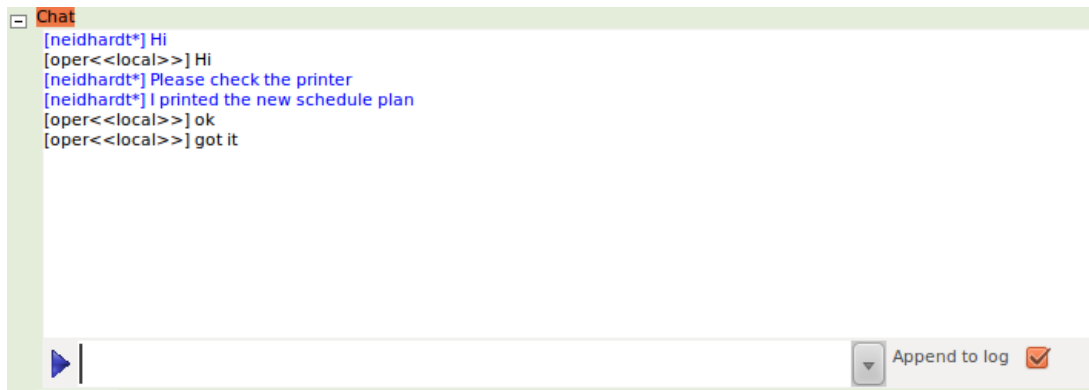
Figure 1.43: The "Chat" display.

messages in that way to the remote display is not permitted, so that those messages are not committed.

1.5.12 Close a connection

The connection can be closed by collapsing all of the control displays with the "+" in front of the display names or by removing a control tab for a dedicated telescope (see section 1.5.6 on page 28).

1.5.13 Starting of shell programs with the defined shortcuts
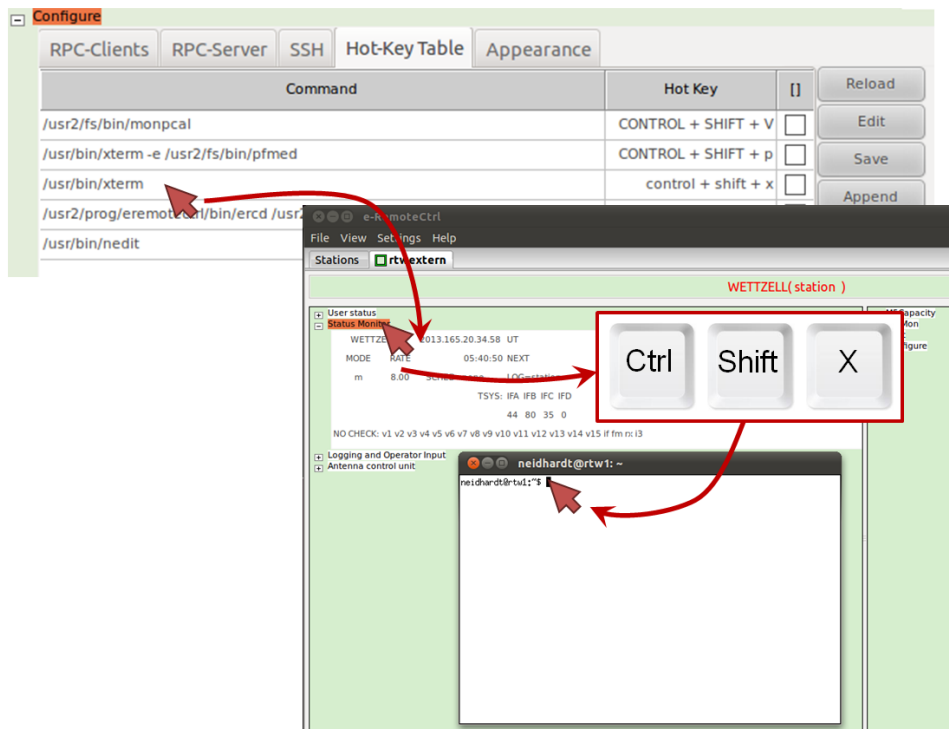
Figure 1.44: The procedure to start a program in a remote shell.

After defining shortcuts in the configuration of the GUI client, it is possible to start commands in a remote shell. Click onto one of the control displays and press the defined shortcut key combination

(see fig. 1.44, where a "xterm" terminal is opened on the NASA Field System machine with the key combination "Ctrl"+"Shift"+"X" and forwarded to the GUI client machine).

1.5.14 Logging of the round trip delay

This functionality is currently not available.

1.5.15 Automatic adaption of the data transfer volume and automatic reconnections

The data transfer volume is adaptive to the opened displays. By closing one display, no data for that display is transferred anymore. This can be used to safe bandwidth on weak communication lines, where only the really needed displays should be opened.

If the data communication is interrupted, the client automatically tries to restart the connection as long as the data communication is not re-established again and as long as it was opened once before. If this reconnection does not work, close the connection manually (see section 1.5.12 on page 42) and restart it again (see section 1.5.7 on page 29).

Chapter 2

Station programming

Abstract

This chapter describes the programming of station specific parts of the remote control software for `VLBI` telescopes. A first focus is on the system specific system monitoring data, which can be injected to the server and shown on each client.

2.1 Programming of the injection of station specific system monitoring points

This section is under construction!

List of Figures

List of Tables

List of acronyms

ASD Automatic Safety Device

CLI Command Line Interface

DNAT Destination Network Address Translation

FTP File Transfer Protocol

GUI Graphical User Interface

HTML Hypertext Markup Language

IDL Interface Definition Language

IP Internet Protocol

NASA National Aeronautics and Space Administration

NAT Network Address Translation

ONC RPC Open Network Computing Remote Procedure Call

PC Personal Computer

RPC Remote Procedure Call

SLR Satellite Laser Ranging

SSH Secure Shell

TCP Transmission Control Protocol

UDP User Datagram Protocol

VLBI Very Long Baseline Interferometry

XDR External Data Representation

XML Extensible Markup Language

Bibliography

[1] idl2rpc-User Manual

Index