

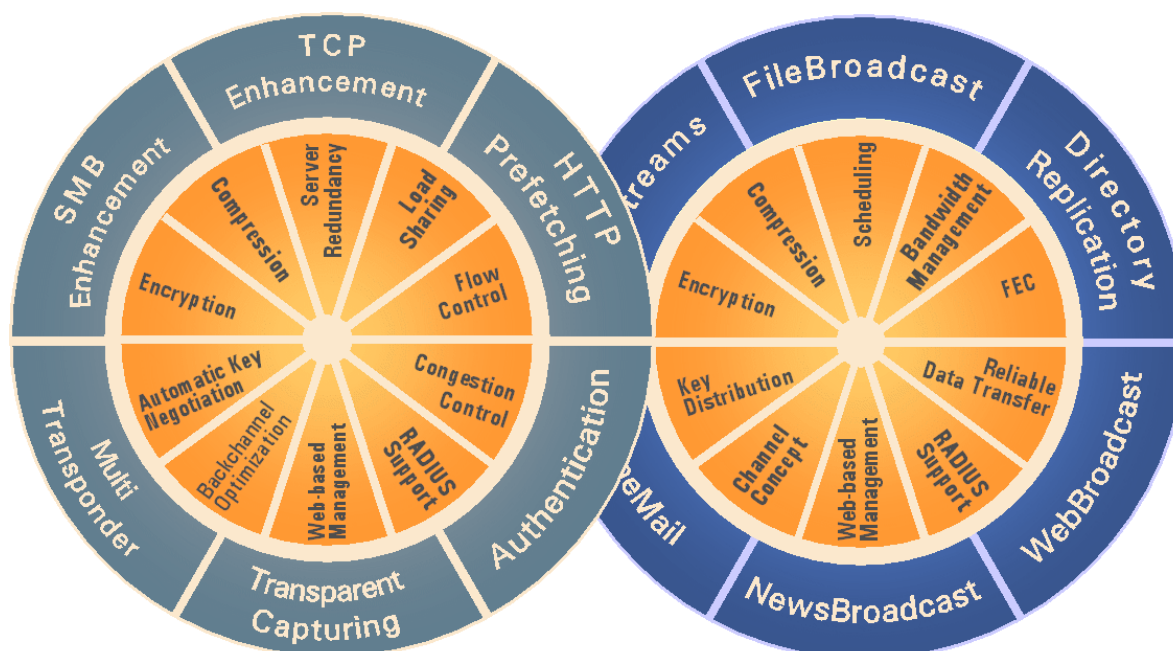
TELLINET / TELLICAST

Client Documentation

Version 2.4.2

May 9th, 2005

– confidential –



Supplemented by:

TELLINET / TELLICAST Documentation (introduction document)

TELLINET / TELLICAST Server Documentation

The distribution system TELLINET / TELLCAST described in this paper combines some of the core modules of Tellitec's multicast distribution platform TELLCAST with the high performance Internet via satellite and radio networks platform TELLINET. This combination will provide you with an ideal tool to build reliable, fast and highly economic data distribution services.

TELLINET with its profound and unmatched performance enhancement features allows efficient internet services via satellite and mobile networks and thereby serves you as one of the key elements for providing efficient network access to private and corporate users all over the world. Normally the two most widely used Internet protocols TCP and HTTP result in poor performance if being used over high latency or lossy networks. TELLINET seamlessly enhance TCP and HTTP communication allowing to set up high performance satellite and mobile data networks. At the same time TELLINET can grow along with the grows of your communications services and provides unique multi server LoadSharing functionality as well as MultiTransponder support.

The multicast distribution system TELLCAST was developed for multicast distribution of files, directories and data streams in various applications. Multicast makes it possible to reach a large number of recipients with one single transmission. It makes data transfer over networks and satellites highly economical. Therefore it can be used for a variety of services differing noticeably in functionality. In order to meet the needs of various services that may use TELLCAST for data distribution, TELLCAST was given a modular structure. Each module is designed to meet the specific needs of a special application adopted service. At the same time, the basis of all modules is the same multicast protocol, MTP/SO (based on the Internet RFC 1301) and the main system features responsible for the secure and easy to handle functionality of TELLCAST are included in all modules.

This documentation gives an overview of the technology used, and will allow the evaluation within specific communication infrastructures. The experience made with the system can serve as a starting-point to discuss the software functionalities necessary for customer-specific projects.

The aim of this paper is to describe the configuration and system features of the TELLINET / TELLCAST Client and to give instructions for the installation and operation of the system.

Please note:

1. In the following the format for directories is given as used in Unix, i.e. directories separated with slashes (home/directory/file).
If TELLINET / TELLCAST is run on a Windows system, the slashes have to be replaced with backslashes (home\directory\file).
2. Words that have to be entered into files/forms are quoted here in quotation marks for a better distinction from the text. Nevertheless the words have to be written into the files/forms without the quotation marks.
3. Pointed brackets in the description of the file/form entries indicate that the words written in the pointed brackets are not the entry itself but a description of the entry. If making the entry in the file/form, please substitute the description for the desired value without writing the pointed brackets. E.g. "<port number>" may result in an entry like "9201".
4. E.g. the file formats and configuration parameters as well as the system functionality might change for future releases.
5. Please note that individual features described within this document might require e.g. separate Application Module licenses and might be available only for certain operating systems.
6. This specification is subject to change without notice, is provided without guarantees, without engagement, and is subject to corrections.
7. All company or product names are trademarks or registered trademarks of the respective owners.
8. This document may not be reproduced, transmitted or distributed without the prior written consent of Tellitec Communications bvba.
© 2004 - 2005 Tellitec Communications bvba. All rights reserved.

1 CONTENTS

1	CONTENTS	3
2	INSTALLATION	5
2.1	SYSTEM REQUIREMENTS	5
2.2	SCOPE OF DELIVERY	5
2.3	INSTALLATION	6
2.4	TELLINET / TELICAST / TELLIVISION CONFIGURATION	6
2.4.1	Configuration	6
2.4.2	Configuration tests	6
2.5	CONFIGURATION OF THE WEB BROWSER	10
2.5.1	Proxy Configuration	10
2.6	HOST KEY DISPLAY	13
3	OPERATION	14
3.1	STARTING THE SYSTEM	14
3.2	DISPLAYING THE WEB INTERFACE OF THE TELLINET / TELICAST CLIENT	14
3.3	MULTICAST RECEPTION	15
3.3.1	Choice of TELICAST Channels to be Received	15
3.3.2	Restriction of bandwidth for TELICAST reception	15
3.3.3	File Broadcast	16
3.3.4	Web Broadcast	17
3.3.5	NewsBroadcast	17
3.3.6	Mail Notification	18
3.3.7	Data Streams	18
3.4	USING TELLINET	19
3.4.1	Changing the Speed Mode	19
3.4.2	Restricting the bandwidth for TELLINET utilisation	20
3.4.3	Using TCP Enhancement	20
3.4.4	User authentication for TELLINET	20
3.4.5	Using HTTP Prefetching	21
3.4.6	Using multi Transponder	22
3.4.7	Using Transparent Capturing	22
3.4.8	Using SMB Enhancement	23
3.5	HANDLING OF ERROR MESSAGES	25
3.6	TERMINATION	26
4	FILES FOR CONFIGURATION AND OPERATION	27
4.1	FORMAT	27
4.2	CLIENT CONFIGURATION FILE	28
4.2.1	Contents	28
4.2.1.1	General sections	28

4.2.1.2 TELLINET only sections	29
4.2.1.3 TELLCAST only sections	31
4.2.2 File entries	32
4.2.2.1 General sections	32
4.2.2.2 TELLINET only sections	51
4.2.2.3 Tellicast only sections	95
4.3 LICENSE FILE	104
4.3.1 Contents	104
4.3.2 File Entries	105
4.4 CLIENT CHANNEL FILE	111
4.4.1 Contents	111
4.4.2 File Entries	112
4.5 LOG FILE	120
5 WEB INTERFACE	122
5.1 STRUCTURE OF THE WEB INTERFACE	122
5.2 PAGE: OVERVIEW	123
5.3 SUBMENU FOR TELLINET	124
5.3.1 Page: TELLINET Statistics	124
5.3.2 Page: Connections	126
5.3.3 Page: ETCP Associations	127
5.4 SUBMENU FOR TELLCAST	129
5.4.1 Page: TELLCAST Statistics	129
5.4.2 Page: Active Channels	131
5.4.3 Page: Channel Information	132
5.5 PAGE: LICENSE	134
5.6 PAGE: LOG FILE	135
5.7 SUBMENU FOR HELP	136
5.7.1 Page: Operation	136
5.7.2 Page: Web Interface	136
5.7.3 Page: File Formats	136
5.7.4 Page: Feedback Mail	136
5.8 REFRESH	138
6 CONTACT	139
7 ABBREVIATIONS AND DEFINITIONS	140

2 INSTALLATION

2.1 SYSTEM REQUIREMENTS

Hardware: Pentium II 400 (or higher); 64 MB RAM (or more)
Possible Platforms: Windows 2000/ME/NT/98/XP, Linux, Solaris, FreeBSD
Software: Web Browser¹ (any standard Web Browser with JavaScript support, e.g. Netscape Navigator, MS Internet Explorer)

2.2 SCOPE OF DELIVERY

The distribution contains binaries for multiple operating systems. For each operating system a particular Zip archive contains the respective software. The following Zip archives are available:

- ❏ linux2.2_rh5.2-i86pc
- ❏ linux2.4_rh7.3-i86pc
- ❏ win32-i86pc
- ❏ freebsd4.9-i86pc
- ❏ irix6.5-mips
- ❏ solaris2.6-sparc
- ❏ macosx6.0-ppc

Each of these Zip archives contains a directory "<operating system>" that includes subdirectories for:

- ❏ **tc-send**
TELLINET / TELLI CAST Client including a configuration for a simple test. The binary is called "tc-send" ("tc-send.exe" on Windows systems).
- ❏ **tc-recv**
TELLINET / TELLI CAST Server including a configuration for a simple test. The binary is called "tc-recv" ("tc-recv.exe" on Windows systems).
- ❏ **tools**
 - ❏ **mcping**
Test tool for multicast connectivity.
 - ❏ **tc-keys**
Tool to generate and display host keys that are needed for the Authentication functionality of TELLINET / TELLI CAST.

The product license file is delivered separately.

¹ Optional. The browser is necessary to control the system easily over the web interface. Server and Client can work without it, however less convenient.

2.3 INSTALLATION

- ☞ Choose the top level directory corresponding to the operating system of your computer.
- ☞ Copy the directory "tc-recv" to the hard disc of your computer.
- ☞ Copy your respective License File into the working directory of the TELLINET / TELLCAST Client.

The top level directories contain software for both Server and Client. When the complete top level directory is installed it is possible to operate each computer as Server as well as Client.

Special note for the installation on Windows NT 4.0

Error-free operation of the TELLINET / TELLCAST software in combination with Service Pack 6 of Windows NT cannot be guaranteed, because this service pack causes a malfunction in the IP stack.

Service Pack 5 does not cause any problems when used with the TELLINET / TELLCAST software.

2.4 TELLINET / TELLCAST / TELLIVISION CONFIGURATION

2.4.1 CONFIGURATION

The Client can be configured with a *Client Configuration File* (recv.ini). The file has to be located in the working directory of the Client.

- ☞ Open the "recv.ini" file in the client directory with an editor.
- ☞ Change the parameters according to your demands (see chapter 4.2).
- ☞ Store the file in the working directory of the Client.

2.4.2 CONFIGURATION TESTS

Selftest of the TELLINET / TELLCAST / TELLIVISION Client can optionally be performed after configuration. It is recommended to perform these tests after each change of the configuration of the software or the network environment.

Five tests are available:

- Test 0: Availability test for configuration files
- Test 1: Test of interfaces and ports
- Test 2: Test of TELLCAST / TELLIVISION announcement channel reception
- Test 3: Test of TELLINET Server connectivity
- Test 4: Test of TELLINET login and reception

The tests are performed as follows and include the following parameters in the configuration files recv.ini, license.ini and recv-channels.ini:


Test	Action	Files/parameters involved in the test
0	It will be checked whether all configuration files are available and readable.	license.ini recv.ini recv-channels.ini
1	Sockets are opened on all relevant ports to test the availability. Directories are tested for being writable. The test is only performed for interfaces and ports used by licensed modules.	recv.ini [shell] port interface_address graphic_port [locations] file_database_directory log_file [etcp_parameters] http_listen_address socks_listen_address [etcp_port_forwarding] listen_address [parameters] interface_address [mediaondemand] http_listen_port
2	It is checked whether the Client is able to receive the announcement channel.	recv.ini [announcement_channel] address name [parameters] interface address
3	A TCP connection is opened to the TELLINET Server. This does not include any authentication or sending of any data.	recv.ini [etcp_parameters] server_address
4	A TCP connection is opened, authentication is done (if activated) and at least one UDP packet has to be received from the Server. (The UDP packets are sent automatically to the Client)	recv.ini [etcp_parameters] server_address authentication local_udp_address [recipient] user_name user_name_prefix user_key user_key_crypt host_key_x (generated automatically and not included in any file)

It is recommended to perform the tests in the given order.

Successful realisation of test 0 is precondition for all other tests.

Successful realisation of test 1 is precondition for tests 2 and 3.

Test 4 is based on test 3.

 All tests can be performed by starting the Client with the following command in the command line:

```
tc-recv -t <number of the test to be performed>
```

tc-recv will be started, the test will be performed and the program will be stopped again immediately afterwards.

The test results are not only written into the Log File but are also given with the Unix exit code for the Client and can therefore be read by other software. Under the Windows operating system the test result will be broadcast as the "iParam" of a WindowMessage. The WindowMessage's name has to be given to the Client as the argument "-m <name>".

The result of the test is given as a number. The numbers have the following meaning:

number:	meaning:	possible for test no.:
0	No error occurred.	all tests
1	Some unspecified error occurred.	currently not in use
2	The test number used in the command line does not correspond to an available test procedure.	all unsupported tests
3	The connection to a TELLCAST announcement channel failed. (Possible reasons are e.g.: wrong announcement channel, server is down, no multicast reception.)	currently not in use
4	The file database for TELLCAST could not be opened (e.g. because a directory is not writable).	1
5	The test failed because it required TELLINET authorization, which was not enabled.	4
6	The http server for the web shell could not be started (e.g. because of the utilisation of a wrong interface or port).	1
7	Errors were encountered in the configuration (details can be taken from the <i>Log File</i>).	1
8	TELLICAST could not be started for some reason.	1, 2
9	TELLINET could not be started for some reason.	1, 3, 4
11	The test timed out after 30 seconds without succeeding.	all tests
12	The TELLINET login failed (e.g. because of wrong username or wrong password)	4
13	The DNS name "localhost" could not be resolved. (The reason is normally a wrong configuration of the operating system)	1

number:	meaning:	possible for test no.:
14	The test requires TELLINET, but TELLINET is not enabled.	3, 4
15	The test requires TELLCAST, but TELLCAST is not enabled.	2
16	The multicast receive interface for TELLINET does not exist.	1, 3, 4
17	The graphic server could not be started (e.g. because of the utilisation of a wrong interface or port).	1
18	The MediaOnDemand server could not be started (e.g. because of the utilisation of a wrong interface or port).	1
19	The intended Log File could not be opened (e.g. the file or directory is not writable or not executable).	1
20	No TelliNet server could be reached (Possible reasons are e.g.: wrong address, server is down, or no Internet connectivity)	3, 4
21	The user name of the Client is not known to the Server.	4
22	The password specified at Client and at Server side are not consistent.	4
23	DNS lookup failed (the domain name could not be translated into an IP address).	3, 4
24	autoconfig_listen_address could not be opened.	1
25	Double login for the same account.	4

2.5 CONFIGURATION OF THE WEB BROWSER

The web browser used has to be configured to send requests to the TELLINET / TELLCAST Client Proxy instead of directly sending them to the web servers. This can be done manually or by using an automatic proxy configuration file.

Please note: If the TELLINET module TransparentCapturing is licensed and activated at the Server and the Client, and transparent capturing is configured, it is possible to use TELLINET without proxy configuration in the web browser. Requests sent by the web browser are then captured automatically by TELLINET transparent to the user's application.

2.5.1 PROXY CONFIGURATION

It is dependent on the web browser how to do the configuration. Please refer to the documentation or online help of the web browser for detailed information about how to perform the following settings, which are in general:

- ✎ **Change the settings of the browser to use a proxy for requests.**
- ✎ **Enter the address and portnumber "<IP address or DNS name of local host>:<configured tc-recv proxy port>"² into the settings of the browser as server for the protocols used.** It is dependent on the browser which protocols have to be configured to be working with the TELLINET / TELLCAST proxy. The proxy activation for the following protocols is recommended:

Browser	HTTP	HTTPS (SSL/Secure)	FTP	SOCKS
Microsoft Internet Explorer	X	X		X
Netscape Navigator 6.1	X		X	X
Netscape Navigator (all versions except 6.1)	X			X
Mozilla, Opera, Konqueror, others	X			X

All browsers (except Netscape Navigator version 6.1) will use the SOCKS proxy as default proxy for all other protocols (e.g. HTTPS) which gives you TELLINET TCP enhancement even for those protocols.

Please note: Always configure the HTTP proxy separately as otherwise no TELLINET HTTP Prefetching will be possible.

Please note: If an additional FTP proxy is used between the TELLINET / TELLCAST Server and the Internet (e.g. Squid), the utilisation of TELLINET / TELLCAST as proxy should be configured for FTP as well as for SOCKS independent from the browser used.

"<IP address or DNS name of local host>" can be replaced by "localhost" if

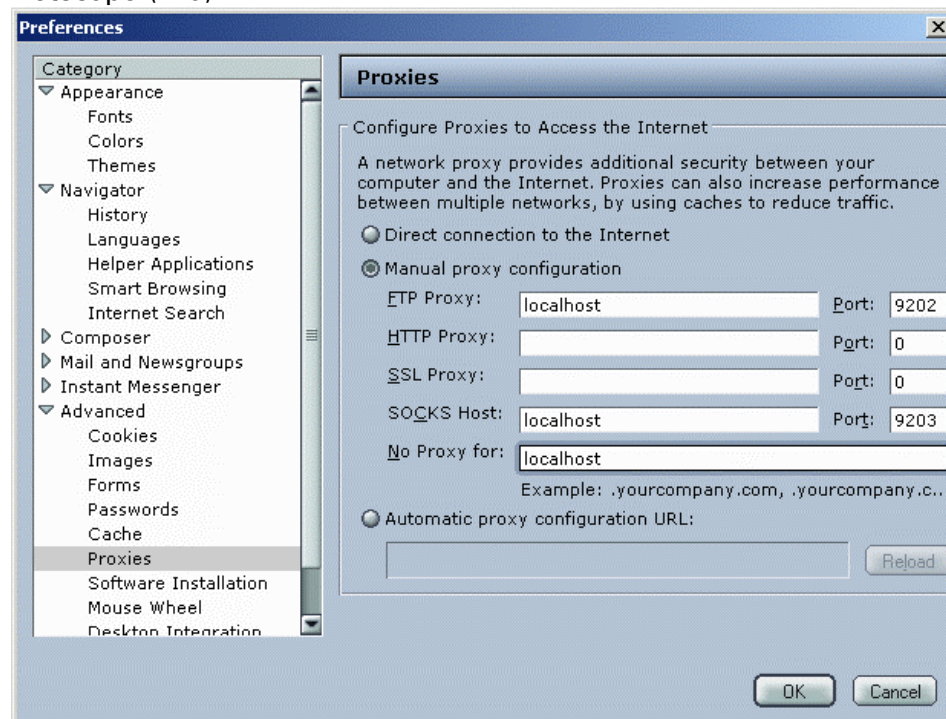
² In case tc-recv is not executed directly on the local host, please specify the IP address or DNS name of the host running tc-recv instead.

the name or IP address is not known and the TELLINET client is run on the same host as the browser used.

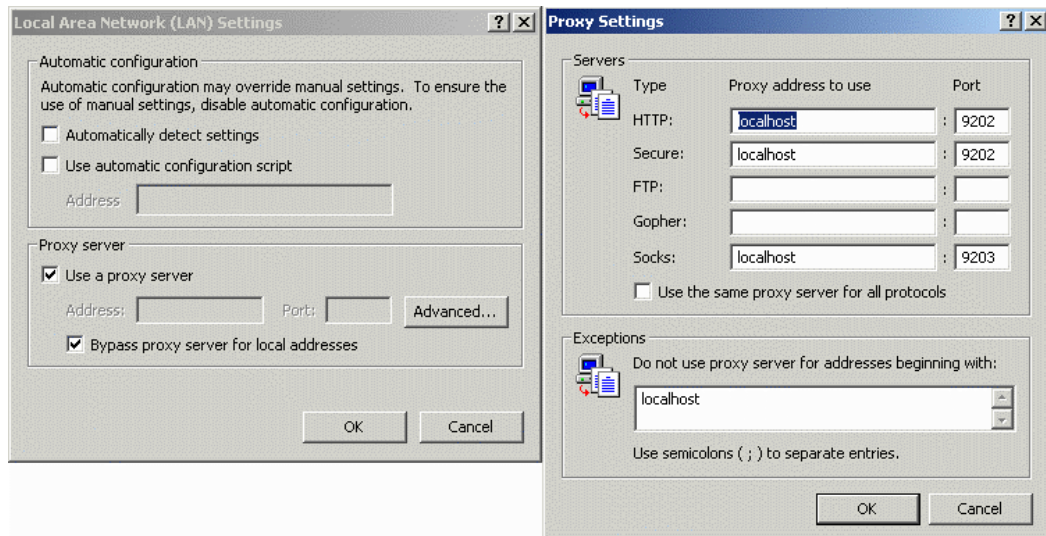
The default port number for HTTP, FTP and HTTPS (SSL/Secure) is "9202", the default port number for SOCKS is "9203". Gopher is not supported by TELLINET / TELLICAST. If your Client has been configured not to use the default port numbers please enter the port numbers configured in the *Client Configuration File*, section [etcp_parameters], entries "http_listen_address" and "socks_listen_address".

Example for the resulting Proxy settings:

Netscape (7.0):



Microsoft Internet Explorer (5.5):



 **If you have web servers in the local LAN or on the local host, please configure the browser not to use the Proxy for requests to the local LAN/host.**

2.6 HOST KEY DISPLAY

For recipient authentication for TELLINET as well as TELLICAST, hardware dependent host keys can be used. The host keys are automatically calculated by the TELLINET / TELLICAST Client and used during operation to authenticate the Client.

Prior to starting requests with TELLINET / reception of data with TELLICAST, the operator of the Server has to enter the host keys of the Client into a *Recipient File*. Therefore the host keys have to be read and communicated to the operator of the TELLINET / TELLICAST Server.

The keys are also displayed on the web interface of the TELLINET / TELLICAST Client on the page License (see chapter 5.5). On Windows systems they can also be read by clicking with the right mouse button on the TELLINET / TELLICAST tray icon in the system task bar and open the "About tc-recv" dialog box.

Alternatively, the tool tc-keys can be used to display the keys:

- Copy the directory tc-keys to the hard disk of your computer.
- Open a DOS command window of Microsoft Windows or a shell when using Unix and change into the directory "tc-keys".
- Start tc-keys with one of the following commands:

tc-keys -k host_key_1

The software will display "host_key_1".

tc-keys -k host_key_2 <path and file name of the TELLINET Client software>

The software will display "host_key_2".

tc-keys <path and file name of the TELLINET Client software>

The software will display "host_key_1", "host_key_2", and "host_key_3".

tc-keys -f <file name> <path and file name of the TELLINET Client software>

The software will write all host keys into the file specified with "<file name>".

tc-keys -p <user_key>

The software will display the encrypted user_key (=user_key_crypt).

3 OPERATION

3.1 STARTING THE SYSTEM

On Unix systems

- ☞ Go to the client directory (default: tc-recv).
- ☞ Type `./tc-recv`.

On Windows systems

- ☞ Double click the icon "tc-recv.exe" in the Windows Explorer.

A tray icon will appear in the task bar of the Windows screen. The tray icon informs about the status of the Client:



= The Client did not reach the TELICAST announcement channel yet or did not receive TELLINET Server announcements yet.



= The Client is connected to the Server and the status is OK.



= An error occurred.

Please note: In case TELICAST is not licensed and activated and TELLINET Server announcements are not used, the yellow icon is not used and a blue icon is displayed independent from the availability of the Server.

When the speed mode of TELLINET is on, the icons display a green inner line instead of a white one.

3.2 DISPLAYING THE WEB INTERFACE OF THE TELLINET / TELICAST CLIENT

- ☞ Enter the URL "<http://system:2517>" in your favourite browser and replace "system" with the hostname/IP address of the TELLINET / TELICAST Client host and "2517" with the selected port in the section [shell] of the *Client Configuration File* (default: 2517).

On Windows systems it is also possible to doubleclick on the product tray icon in the systemtray to open the web interface.

3.3 MULTICAST RECEPTION

3.3.1 CHOICE OF TELLI CAST CHANNELS TO BE RECEIVED

A choice among the subscribed channels can be made by creating a *Channel File*. This is optional. The Client will operate perfectly without a *Channel File* and use the given default values for reception. Without a Channel File all subscribed channels are received.

- ✎ Write a Client *Channel File* according to the information in chapter 4.4 or change the *Channel File* (recv-channels.ini) stored in the "client" directory. The file has to contain a section for each channel from which the data is received.
- ✎ Store the file in the directory of the Client.

3.3.2 RESTRICTION OF BANDWIDTH FOR TELLI CAST RECEPTION

A maximum allowed bandwidth for multicast reception with TELLI CAST can be configured within the *Client Configuration File*:

- ✎ Open the Client Configuration File "recv.ini".
- ✎ Specify the maximum allowed bandwidth with the entry "total_bandwidth" in the section [parameters].
- ✎ Store the file in the working directory of the Client.

The client will only subscribe to channels until total_bandwidth is reached.

Please note that you can assign a priority to each channel in the Client *Channel File* recv-channels.ini. In case that total_bandwidth is reached, channels with lower priority will be dropped in order to join transmissions on channels with higher priority.

3.3.3 FILE BROADCAST

All received files and directories are stored in the incoming directory of the Client. By default this is the directory "received" under the working directory of the Client.

Choice of a Target Directory

When data should be delivered to any other location than the incoming directory "received", a Client *Channel File* must be available in the Client.

Data can either be directed to any local directory or it can alternatively be forwarded to an FTP server.

- ✎ Write a Client *Channel File* according to the information in chapter 4.4. The file has to contain a section for each channel from which the data is received.
 - ✎ Include the specification of a target directory for the received data to the section of each channel that shall have a different target directory with the entry "target_directory". This directory will also be used on the FTP server in case the data is forwarded to an FTP server.
 - ✎ Only when data shall be forwarded to an FTP server: activate the forwarding with the entry "activate_ftp_forwarding=1" in the section of the respective channel.
 - ✎ Store the file in the working directory of the Client.
- Only when data shall be forwarded to an FTP server:
- ✎ Open the Client Configuration File "recv.ini".
 - ✎ Specify an FTP server by adding a section [ftp_forwarding] with at least an entry "ftp_server_address".
 - ✎ Store the file in the working directory of the Client.

Automatic Execution of Files upon Reception of Complete Transmissions

The TELLICAST Server can send files in such a way that they are executed automatically on the Client host after reception. The execution of the files requires a change in the Client *Channel File* (recv-channels.ini).

To allow the automatic execution of files after reception:

- ✎ Write a Client *Channel File* according to the information in chapter 4.4. The file has to contain a section for each channel from which the data is received.
- ✎ Include the entry "allow_execute=1" to the sections of the channels from which the automatically executed files are received.
- ✎ Store the file in the working directory of the Client.

3.3.4 WEB BROADCAST

To use the module WebBroadcast, the web browser has to be configured to use the TELLINET / TELLICAST Client Proxy as web proxy (see chapter 2.5).

The web content found during crawling by the TELLICAST Server is transmitted to the Client and stored in the directory specified with the entry "target_directory" in the Client *Channel File*. The default directory is "received/webbroadcast" under the working directory of the Client.

The Client Proxy uses that directory as a cache for quick offline access to some of the web content requested during normal operation of a browser. Only in case the URL is not found in this directory, a connection is established to the web server given by the URL (or to the TELLINET / TELLICAST Server Proxy when TELLINET Enhanced-TCP is used and the speed mode is on).

If any other than the default value is used for the target directory, the Client Proxy has to be configured to use the target directory as cache with the entry "webbroadcast_cache_directory" in the section [etcp_parameters] of the *Client Configuration File*.

3.3.5 NEWSBROADCAST

The news articles that the TELLICAST Server retrieved from a news server are transmitted to the Client and transferred to the external news server on the Client side. News articles are transferred to the external news server using NNTP, i.e. once configured, the news server cannot distinguish the TELLICAST Client from any normal news server.

For the reception of NewsBroadcast it is necessary to configure the Client to identify the receiving external news server.

- ✎ Write a *Client Configuration File* according to the information in chapter 4.2. The file has to contain a section [news] with the name of the external news server used for reception of the data.
- ✎ Store the file in the working directory of the Client.
- ✎ Configure the TELLICAST Client as sending news server in the external, receiving news server.

3.3.6 MAIL NOTIFICATION

Mail notification requires the use of a Windows system by the recipient. It is done automatically after start of the Client when a mail notification channel is received. It does not require any further changes in the mail communication program used by the recipient.

The recipient is informed about incoming mail by a red flag appearing above the envelope that the program displays in the system tray of the Windows screen of the user.

Mail is not delivered by the TELLICAST system but has to be retrieved from the mail server as usual (full mail delivery via TELLICAST and provision to the mail reader via POP3 is optional).

3.3.7 DATA STREAMS

The data stream coming from the TELLICAST Server is further distributed to the recipients by the TELLICAST Client. The default values for the address and port of the final destination of the data stream as well as the transmission protocol and transmission parameters used are configured in the Server, but it is also possible to change these values in the Client *Channel File*. When the parameters are specified in the Client *Channel File*, these parameters take priority over the parameters specified in the Server.

Changing the Parameters for the Distribution of Data Streams:

- ✎ Write a Client *Channel File* according to the information in chapter 4.4. The file has to contain a section for each channel to be received. The parameters for the final distribution of the data stream can be changed per channel (see chapter 4.4 for available parameters).
- ✎ Store the file in the directory of the Client.

Reception of DataStreams by the Recipients

- ✎ Enter the port number for DataStreams reception specified in the Client *Channel File* with the entry "stream_address" or in the Server *Job File* with the entry "destination_address" into the access configuration of the software you are using to view the data stream content. The way to change the port is dependent on the software used. Please refer to the help of your stream software to determine how to change the port.

In the simplest case use telnet to receive TCP data streams (e.g. for an initial test). Using telnet the command is normally:

"telnet <localhost> <port>"

where "<localhost>" is the DNS name or IP address of the TELLICAST Client and "<port>" is the port number specified as "stream_address" in the *Client Channel File* (default: "destination_address" in the *Server Configuration File*).

3.4 USING TELLINET

3.4.1 CHANGING THE SPEED MODE

The speed mode defines whether the TELLINET Client is connecting to the TELLINET Server or is connecting directly to the internet instead. This allows to activate or deactivate TELLINET functionality, like ETCP, HTTP Prefetching, and all other functionality requiring a connection between TELLINET Client and Server. The speed mode / non speed mode allows to deactivate or activate TELLINET functionality without altering the proxy setting of the application, e. g. web browser, used.

On Windows and Unix operating systems the speed mode can be altered by changing the parameter "default_speed_state" in the section [etcp_parameters] of the *Client Configuration File*.

An additional method for altering the speed mode is available on Windows operating systems, if an entry "use_icon = 1" is specified in the section [etcp_parameters] of the *Client Configuration File*:

On Windows Operating Systems

Toggling between speed and non speed mode can be done in the Client Proxies with the help of the product tray icon in the system tray of the Windows desktop.

The tray icon also informs about the speed mode that is currently used:



= Product tray icon used in speed mode.



= Product tray icon used in non speed mode.

To change the operation mode:

- ☞ Click with the right mouse button on the product tray icon in the system tray.
- ☞ Click on "Speed mode" in the menu.

3.4.2 RESTRICTING THE BANDWIDTH FOR TELLINET UTILISATION

Maximum allowed data rates for transmissions from TELLINET Client to TELLINET Server and from TELLINET Server to TELLINET Client can be configured in the *Client Configuration File* in the section [etcp_parameters] with the entries "max_send_data_rate" and "max_receive_data_rate".

The maximum allowed data rates can also be restricted at the TELLINET Server. In that case the smaller of the configured allowed data rates is used to specify the maximum allowed bandwidth. E. g. if the TELLINET Client is configured to allow a maximum bandwidth of 5 000 000 bit/s from Client to Server and the TELLINET Server restricts this bandwidth to 4 000 000 bit/s, the maximum allowed data rate for data sent from Client to Server will be 4 000 000 bit/s.

3.4.3 USING TCP ENHANCEMENT

The module TCP Enhancement substitutes TCP by ETCP for all communication between TELLINET Client and TELLINET Server. ETCP is especially designed to overcome the disadvantages of TCP on high latency and lossy networks.

To use ETCP, the module Enhanced TCP has to be licensed and activated at the TELLINET Server and the TELLINET Client.

It is not necessary to configure the TELLINET Server or the TELLINET Client for the utilisation of TELLINET. But it is possible to fine tune the functionality according to individual needs within the *Server and the Client Configuration File* (send.ini and recv.ini).

3.4.4 USER AUTHENTICATION FOR TELLINET

The module Authentication makes it possible to restrict the utilisation of the TELLINET functionality to a certain user or even a certain computer or DVB card/box.

User authentication for TELLINET requires that the following preconditions are met:

- The module Authentication is licensed and activated at the TELLINET Client and at the TELLINET Server
- Authentication is activated at the TELLINET Server in the Server Configuration File send.ini with the entry "activate_for_etcp" in the section [authentication].
- Authentication is activated at the TELLINET Client in the Client Configuration File recv.ini with the entry "authentication" in the section [etcp_parameters].

When Authentication is active, a Recipient File is necessary at the TELLINET Server, specifying the user data for the recipient. This can include a user name, a user key and host keys. Host keys are hardware and software dependent and are generated automatically by the Client but have to be added manually to the Recipient File at the TELLINET Server (see chapter 2.6).

The user name and the user key of the Client can be specified in the *Client Configuration File* recv.ini in the section [recipient].

In case the data on the client side is missing or is not identical to the data found at the TELLINET Server, the user is requested to enter a valid user name and password via a web interface.

If user data is submitted via that web interface, the TELLINET Client automatically erases the eventually existing user name and user key from the section [recipient] in the recv.ini file. The user name from the login via web interface is then written into the section [recipient].

The user key is only stored in the recv.ini file if "Remember Password" is set at the login interface. The user key is stored as user key crypt, i. e. in an encrypted version and further login via the login interface is not necessary for the client in the future. If "Remember Password" is not set, the user key is not stored and the user has to log in via the login interface each time he is using TELLINET.

3.4.5 USING HTTP PREFETCHING

The module HTTP Prefetching speeds up HTTP downloads by scanning of web pages for subobjects and pushing of these web objects. When requested by the web browser they can be taken rapidly from the local cache.

Utilisation of the HTTP Prefetching functionality requires that the following preconditions are met:

- The module HTTP-Prefetching is licensed and activated at the TELLINET Client and at the TELLINET Server
- HTTP Prefetching is activated at the TELLINET Client in the Client Configuration File recv.ini with the entry "use_http_prefetching" in the section [http_prefetching].

Please note: If the speed mode is deactivated, the TELLINET Client is not connecting to the TELLINET Server but directly to the web servers. In that case HTTP Prefetching can not be done even if it is activated. (For further explanation of the speed mode, see "TELLINET / TELICAST Documentation", Chapter 3.2.3 and this document, chapter 3.4.1.)

The section [http_prefetching] also allows to configure the HTTP Prefetching functionality, e. g. to restrict the cache size used for prefetching or to specify the prefetching mode for conditional requests (see chapter 4.2 for a description of possible configuration).

3.4.6 USING MULTI TRANSPONDER

The module MultiTransponder allows to switch between various transponders during web browsing when using an Internet via Satellite Service that is offered on multiple transponders.

The utilisation of the module MultiTransponder requires that the following preconditions are met:

- The module MultiTransponder is licensed and activated at the TELLINET Server and the TELLINET Client.
- The Server side of the system is configured to support the utilisation of multiple transponders (see TELLINET / TELLCAST Server Documentation).
- At least one entry "server_announcement_address" in the section [etcp_parameters] of the recv.ini file is available and is corresponding to an entry "address" in a section [server_announcement] of the send.ini file at the TELLINET Server.
- No further configuration of TELLINET external applications is necessary at client side. Whenever a new transponder is chosen, the TELLINET functionality is automatically switched to this transponder without any data loss from established connections.

3.4.7 USING TRANSPARENT CAPTURING

The module TransparentCapturing allows to use TCP enhancement by the TELLINET Client transparent to the applications used. The TELLINET Client can be configured to capture all TCP traffic that is coming in at a configurable interface and send it via the TELLINET Server, with TCP enhancement applied within Client and Server.

The utilisation of the module TransparentCapturing requires that the following preconditions are met:

- The module TransparentCapturingClient is licensed and activated at the TELLINET Server.
- The module TransparentCapturing is licensed and activated at the TELLINET Client.
- Transparent Capturing is activated at the TELLINET Client in the *Client Configuration File* recv.ini with the entry "activate" in the section [transparent_capturing]. This section can also be used to configure the transparent capturing functionality.
- A section [internal_tcp] is available in the Client Configuration File recv.ini, containing at least the entry "activate = 1".
- All TCP traffic that shall be subject to enhancement is routed to the interface specified with the entry "interface" in the section [internal_tcp] of the file recv.ini.

3.4.8 USING SMB ENHANCEMENT

SMB File Sharing is based on protocols that require a notably high number of round trip times, therefore making it necessary to apply enhancement mechanisms beyond normal TCP enhancement. TELLINET offers such SMB specific enhancement that noticeably speeds up the download of files with SMB File Sharing both from Microsoft Windows systems and Unix based SMB implementations [e. g. Samba].

SMB File Sharing is not supporting proxy utilisation. Therefore it is convenient to use the TELLINET Client transparently. This can be done by using the TransparentCapturing functionality of TELLINET. Alternatively port forwarding can be used for SMB Enhancement. It is recommended to use TransparentCapturing as port forwarding is subject to restrictions (see below).

SMB Enhancement via port forwarding

SMB Enhancement via port forwarding requires the following preconditions to be met:

- **Under Windows operating systems the TELLINET Client can not be located on the SMB Client!** Make sure that the TELLINET Client is located on another host. This is due to Windows specific restrictions. When using Samba Servers, the TELLINET Client can be located on the same host as the SMB Client.
- **Under Windows operating systems SMB File Sharing has to be deactivated at the TELLINET Client host.**

For Win NT/2000/XP this can be done as follows:

- ✓ Open your registry editor (e.g. regedit.exe) as administrator.
- ✓ Go to the registry key
"HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NetBT\Parameters\TransportBindName".
- ✓ Set the value of this key from "\Device\" to an empty string.
- ✓ Reboot the Windows machine.

Please note that after the reboot you will not be able to use SMB based services on this machine at all. It is only used as an TELLINET proxy with Enhanced-SMB support for other machines.

- The module EnhancedSMB is licensed and activated at the TELLINET Client and the TELLINET Server.
- A section [port_forwarding] is available in the Client configuration file recv.ini.
- A listen_address and a remote_address are configured in the section [port_forwarding].

Microsoft SMB Servers usually accept SMB File Sharing requests on port 445 (on Samba Servers port 139 can also be used). Therefore 445 has to be configured as port for the "listen_address" as well as the "remote_address".

The IP address to be used for the "listen_address" is the IP address of the host of the TELLINET Client.

The IP address to be used for the "remote_address" is the IP address or DNS name of the SMB Server.

- The entry "use_smb_enhancement" is activated in the section [etcp_port_forwarding].

The SMB Enhancement functionality will then be applied to all port forwarding connections to the port specified with the entry "remote_address" in this section [port_forwarding].

Please note: It is not possible to configure multiple sections [port_forwarding] with the same "listen_address". Therefore only one SMB Server can be addressed per TELLINET Client host when using TELLINET port forwarding to enhance the performance of SMB File Sharing.

In the SMB File Sharing application, the IP address or DNS name of the TELLINET client host has to be entered instead of the IP address or DNS name of the SMB Server has to be entered. E. g. when accessing the share "test" on the SMB Server "Companyremote" and using a TELLINET Client on the host "Enhancementhost", the path to the share for SMB File Sharing is

"\\Enhancementhost\test"

instead of "\\Companyremote\test".

SMB Enhancement with TELLINET TransparentCapturing

SMB Enhancement via TransparentCapturing requires the following preconditions:

- The module SMB Enhancement is licensed and activated at the TELLINET Client and the TELLINET Server.
- The module TransparentCapturingClient is licensed and activated at the TELLINET Server.
- The module TransparentCapturing is licensed and activated at the TELLINET Client.
- A section [internal_tcp] is available in the Client Configuration File recv.ini, containing at least the entry "activate = 1".
- Transparent Capturing is activated at the TELLINET Client in the Client Configuration File recv.ini with the entry "activate" in the section [transparent_capturing]. This section can also be used to configure the transparent capturing functionality.
- The port used for SMB file Sharing is configured in the section [transparent_capturing] with the entry "capture_smb".
The TELLINET Client will regard all incoming data addressed to this port as SMB File Sharing data and will apply SMB enhancement.
Microsoft SMB Servers usually accept SMB File Sharing requests on port 445 (on Samba Servers port 139 can also be used). Therefore usually 445 has to be configured as capture_smb port.

The SMB Enhancement functionality will then be applied to all captured connections addressed to the capture_smb port independent from the destination IP address. Therefore it is possible to apply SMB Enhancement to multiple SMB Servers.

3.5 HANDLING OF ERROR MESSAGES

When the TELLICAST Client encounters a problem impairing functionality, the status line in the upper right frame of the web interface will show one of the following states:

- **ERROR:** An error occurred.
- **WARNING:** A problem not as severe as an error occurred.

At the same time a new button called "Reset" appears in the upper right frame of the page.

The following steps will help to eliminate the error:

- ✓ Click on Log File in the menu of the web interface to open the *Log File*.
- ✓ Read the *Log File* to determine what kind of error occurred.
- ✓ Take the necessary steps to eliminate the error.
- ✓ Click on the button "Reset" to reset the status.

3.6 TERMINATION

On Unix systems

- ✓ Press "<Ctrl> + c" to terminate the Client.

On Windows systems

- ✓ Click with the right mouse button on the product icon in the tray bar. A menu opens.
- ✓ Click on "Exit tc-recv" in the menu.

4 FILES FOR CONFIGURATION AND OPERATION

This chapter describes the format of all files used by the Client of the TELLINET / TELLICAST system.

These files may be written and adapted to a particular situation using any standard editor. To allow the easy creation of new files through modification of existing files, examples of most files are distributed with the software.

For most entries there are default values so that the corresponding entries/sections can be left out. For these entries the default value is given with the description of the entry. If there is no default value given and the entry is mandatory, an error message will be generated if it is missing. In case of an error in a *Job File* the job will be rejected. In case of other errors the system might not start (if so please view the *Log File* or console output for information about the error that occurred).

4.1 FORMAT

The format is similar to the Windows ini-file format. You can e. g. use a standard text file editor to modify the configuration files.

The file entries are structured as sections. The general format is:

```
[<section_name>]
<parameter name> = <value>
```

e. g.:

```
[watchdog]
activate = 1
max_cpu_usage = 75
```

4.2 CLIENT CONFIGURATION FILE

The *Client Configuration File* of the TELLINET / TELLICAST Client specifies global Client parameters. It has to be named "recv.ini" and has to be located under the working directory of the Client.

4.2.1 CONTENTS

4.2.1.1 General sections

[recipient]	→ page 32
user_name	→ page 32
user_key	→ page 33
user_key_crypt	→ page 33
user_name_prefix	→ page 34
[shell]	→ page 35
login	→ page 35
allowed_address	→ page 35
port	→ page 36
interface_address	→ page 36
refresh	→ page 36
graphic_port	→ page 37
debug_port	→ page 37
popups_allowed	→ page 38
[watchdog]	→ page 39
activate	→ page 39
alive_check_interval	→ page 39
restart_time	→ page 40
max_memory_usage	→ page 40
max_cpu_usage	→ page 41
max_cpu_usage_period	→ page 41
max_nr_of_core_files	→ page 42
[mail]	→ page 43
mail_server_address	→ page 43
sender_address	→ page 43
default_feedback_mail_recipient	→ page 44
runtime_error_mail_recipient	→ page 44
[logging]	→ page 45
log_level	→ page 45
log_file_size	→ page 45
log_file_number	→ page 46
[locations]	→ page 47
webbroadcast_cache_directory	→ page 47
prefetching_cache_directory	→ page 47
file_database_directory	→ page 48
mail_database_directory	→ page 48
media_database_directory	→ page 48
channels_file	→ page 49

log_file	→ page 49
license_file	→ page 50

4.2.1.2 TELLINET only sections

[etcp_parameters]	→ page 51
server_address	→ page 51
server_announcement_address	→ page 52
http_listen_address	→ page 52
socks_listen_address	→ page 53
autoconfig_listen_address	→ page 53
allowed_address	→ page 54
max_send_data_rate	→ page 54
max_receive_data_rate	→ page 55
etcp_send_protocol	→ page 55
local_udp_address	→ page 56
external_udp_address	→ page 57
authentication	→ page 57
default_speed_state	→ page 58
use_icon	→ page 58
inactivity_timeout	→ page 58
packet_aggregation_timeout	→ page 59
packet_size	→ page 59
continuous_server_connection	→ page 60
http_proxy	→ page 60
target_overall_memory_usage	→ page 61
[etcp_compression]	→ page 62
compression_level	→ page 62
http_compression	→ page 62
port_forwarding_compression	→ page 63
socks_tcp_compression	→ page 63
socks_udp_compression	→ page 63
capturing_compression	→ page 64
smb_compression	→ page 64
[http_prefetching]	→ page 65
use_http_prefetching	→ page 65
conditional_prefetching	→ page 66
http_prefetching_object_size	→ page 67
http_prefetching_cache_size	→ page 67
http_prefetching_memory_cache	→ page 68
[proxy_excludes]	→ page 69
exclude	→ page 69
[etcp_port_forwarding]	→ page 70
remote_address	→ page 70
listen_address	→ page 70
use_smb_enhancement	→ page 71

[internal_tcp]	→ page 72
activate	→ page 72
interface	→ page 72
virtual_ip_address	→ page 73
tunnel_ip_address	→ page 73
execute_script	→ page 74
[transparent_capturing]	→ page 75
activate	→ page 75
capture_http	→ page 75
capture_smb	→ page 76
exclude	→ page 76
non_tcp_forward_interface	→ page 77
non_tcp_default_gw	→ page 77
http_prefetching_port_range	→ page 78
[etcp_connection_control_client]	→ page 79
activate_congestion_control	→ page 79
initial_data_rate	→ page 80
minimum_calculated_data_rate	→ page 80
low_range_bandwidth	→ page 80
low_range_increment_percent	→ page 81
low_range_increment_additive	→ page 81
low_range_decrement_percent	→ page 82
low_range_decrement_factor	→ page 82
low_range_decrement_fast_percent	→ page 83
low_range_decrement_fast_factor	→ page 83
low_range_increment_fast_threshold	→ page 84
low_range_increment_fast_factor	→ page 84
mid_range_bandwidth	→ page 85
mid_range_increment_percent	→ page 85
mid_range_increment_additive	→ page 86
mid_range_decrement_percent	→ page 86
mid_range_decrement_fast_percent	→ page 87
mid_range_decrement_fast_factor	→ page 87
mid_range_increment_fast_threshold	→ page 88
mid_range_increment_fast_factor	→ page 88
high_range_bandwidth	→ page 89
high_range_increment_percent	→ page 89
high_range_increment_additive	→ page 90
high_range_decrement_percent	→ page 90
high_range_decrement_factor	→ page 91
high_range_decrement_fast_percent	→ page 91
high_range_decrement_fast_factor	→ page 92
high_range_increment_fast_threshold	→ page 92
high_range_increment_fast_factor	→ page 93
max_window_size	→ page 93
nak_timeout	→ page 94
connect_initial_timeout	→ page 94
nr_of_tcp_connections	→ page 94

4.2.1.3 TELLI CAST only sections

[announcement_channel]	→ page 95
address	→ page 95
name	→ page 95
[parameters]	→ page 96
interface_address	→ page 96
tmp_directory	→ page 96
tmp_prefix	→ page 97
tmp_suffix	→ page 97
total_bandwidth	→ page 98
acknowledgement_proxy_address	→ page 98
packet_naks	→ page 99
allow_execute	→ page 99
file_database_size	→ page 100
[webbroadcast]	→ page 101
min_file_lifetime	→ page 101
[newsbroadcast]	→ page 102
server_address	→ page 102
connection_timeout	→ page 102
[offlinemail]	→ page 103
show_icon	→ page 103

4.2.2 FILE ENTRIES

4.2.2.1 General sections

[recipient]	file recv.ini
The section [recipient] specifies the recipient. The entries have to be identical with the entries in the Recipient File(s) (*.rcv) of the Server.	

user_name	section [recipient]	file recv.ini
Recipient name as unique! ascii string.		
syntax user_name = <string>	example user_name = miller	
<p>Name of the recipient as unique ASCII string.</p> <p>The "user_name" may be up to 50 characters long and may contain any of the following characters: lower case letters, capital letters, numbers, hyphen, underscore and "@".</p> <p>Please note that the user_name specification is case-insensitive and therefore e. g. the user names "miller" and "Miller" are regarded as identical.</p> <p>This entry has to be identical to the "user_name" specified in a Recipient File at the Server if the entry "use_name_prefix" is not used in this section. Otherwise, the "user_name" at the Recipient File of the Server has to be the sum of "user name prefix" and "user name".</p>		
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
range	related entries tc-recv - recv.ini - user_name_prefix tc-send - *.rcv - [recipient] - user_name tc-send - *.cha - [recipients] - user_name tc-send - *.job/*.str/*.crw/*.enf/*.nss - [recipients] - user_name tc-send - *.job/*.str/*.crw/*.enf/*.nss - [additional_recipients] - user_name	
default		

user_key		section [recipient]	file recv.ini
Secret recipient key used to direct encrypted transmissions to this recipient.			
syntax user_key = <string>		example user_key = topsecret	
Specifies the key of the recipient. This key is used for encrypted transmissions. It will be used to decrypt the distributed data key. The “user_key” has to be identical with the “user_key” used at Server side. This entry will be ignored if an entry “user_key_crypt” is specified in this section. The “user_key” may be up to 20 characters long and may contain any of the following characters: lower case letters, capital letters, numbers, hyphen, underscore and “@”.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default	related entries tc-end - *.rcv - [recipient] - user_key tc-recv - recv.ini - [recipient] - user_key_crypt tc-send - *.cha - [channel] - encryption		

user_key_crypt		section [recipient]	file recv.ini
Encrypted user_key (e.g. using tc-keys).			
syntax user_key_crypt = <encrypted key>		example	
<p>When using Tellinet Authentication the user can choose the option "Save password" on the login interface. The password will then be encrypted and stored in the Client Configuration File with the entry "user_key_crypt". The next time the user is logging in, a password is not required, because the password specified with the entry "user_key_crypt" is used.</p> <p>The "user_key_crypt" can also be entered manually (for TelliCast / TelliVision and TelliNet). Please use the tool "tc-keys" to manually transform a plain user key into the crypted user key. The crypted user key is a scrambled variant of the plain "user key".</p>			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default	related entries tc-send - *.rcv - [recipient] - user_key_crypt tc-recv - recv.ini - [recipient] - user_key tc-send - *.cha - [channel] - encryption		

user_name_prefix		section [recipient]	file recv.ini
Allows you to add a fixed prefix to the user_name given in recv.ini and to the user_name typed when manually logging into ETCP.			
syntax user_name_prefix = <string>		example user_name_prefix = provider	
User name prefix of the recipient as ASCII string. Only useful when using TelliNet with Authentication. When using TelliNet Authentication the user may have to enter its user name into a web based login form. If you want to make sure that, independent of the user name typed into the login form and invisible to the user, the user name always starts with a fixed prefix (e.g. company_), you can set "user_name_prefix". The resulting user name (used within the software and within the Server Recipient Files) is always the sum of "user_name_prefix" and "user_name". The "user_name_prefix" may be up to 50 characters long and may contain any of the following characters: lower case letters, capital letters, numbers, hyphen, underscore and "@".			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default	related entries tc-recv - recv.ini - [recipient] - user_name tc-send - *.rcv - [recipient] - user_name		

[shell]	file recv.ini
<p>This section allows to configure the program internal web interface.</p> <p>This web interface (which is normally used by the operators only) can be accessed locally or remotely using normal web browsers like e. g. the MS Internet Explorer or Netscape Navigator).</p>	

login	section [shell]	file recv.ini
Password protection of the web interface.		
syntax login = <name> : <password> login = none	example login = miller:mysecret login = none	
Specifies a login for the web interface. This includes a username and a password and has to be written as: "<name>:<password>". To disable the username, password protection, "login=none" has to be configured. If no parameter "login" is given, the access to the web interface is not possible.		
unit range default no login allowed and web shell is deactivated	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
related entries tc-recv - recv.ini - [shell] - allowed_address		

allowed_address	section [shell]	file recv.ini
Allows connections to the web interface only from the hosts listed (e.g. localhost).		
syntax allowed_address = <address> allowed_address = <DNS name>	example allowed_address = 127.0.0.1 allowed_address = serverhost	
Specifies the IP address or DNS name of a host that is allowed to connect to the web interface port of the Client and open the web interface. The entry can be given multiple times to allow more than one host to connect to the web interface of the Client. If "127.0.0.1" is entered, access is allowed from all available local interfaces.		
unit range default access is permitted for all hosts - but login restrictions according to the login parameter apply	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
related entries tc-recv - recv.ini - [shell] - login		

port		section [shell]	file recv.ini
port of web interface e.g. URL:http://localhost:2517			
syntax port = <port>		example port = 2517	
The port to be used for the web interface (default: 2517).			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default 2517	related entries tc-recv - recv.ini - [shell] - interface address		

interface_address		section [shell]	file recv.ini
IP address of the interface to that the web interface port is bound to.			
syntax interface_address = <ip address>		example interface_address = 172.27.1.2	
In case the host is equipped with more than one interface, this entry can be used to bind the web interface to a specific network interface only. Please note that the use of "localhost:<port>" instead of "<ip address>:<port>" in the browser is not possible to access the web interface locally when the entry "interface_address" is specifying the IP address of the local host. Instead, "localhost" or "127.0.0.1" has to be entered as interface address to use "localhost:<port>" as address of the web interface.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
range			
default bound to all local network interfaces	related entries tc-recv - recv.ini - [shell] - port		

refresh		section [shell]	file recv.ini
Time interval for automatic refreshes within various web pages of the web interface.			
syntax refresh = <seconds>		example refresh = 5	
Specifies the time interval between automatic refreshes within various web pages of the web interface.			
unit seconds range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
default 5	related entries		

graphic_port		section [shell]	file recv.ini
Port used for the web interface "bandwidth" java applet.			
syntax graphic_port = <port>		example graphic_port = 9876	
The web interface contains graphics giving an overview of the currently used bandwidth for TelliNet / TelliCast / TelliVision. The Client uses the port configured here to communicate the bandwidth values for the graphics to the web interface (default: a port is chosen randomly).			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default random port number	related entries		

debug_port		section	file
		[shell]	recv.ini
Additional terminal port for debug/statistic information in non-HTTP ASCII format.			
syntax		example	
debug_port = <port>		debug_port = 2518	
In addition to the data displayed on the web interface, statistical data is available in non-HTTP ASCII format. If the "debug_port" is set, the statistical data can be displayed using e. g. telnet (with the entry: "telnet <host ip address or host name> <debug_port>"). When the connection to the debug port is established, the command "help" can be used to display a list of available commands to access the statistic data. Please note that the parameters "login", "allowed_address" and "interface_address" in this section apply to the debug port as well as to the web interface.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
	related entries		
	tc-recv - recv.ini - [shell] - login		
deactivated	tc-recv - recv.ini - [shell] - allowed_address		
	tc-recv - recv.ini - [shell] - interface_address		

popups_allowed		section [shell]	file recv.ini
0/1 enabling of popup of login interface			
syntax popups_allowed = <value of range>		example popups_allowed = 1	
When authentication is required for the use of TelliNet, the authentication is normally done automatically by TelliNet, using the login data of the user that is specified in the Client Configuration File recv.ini. On Windows systems, in case that the user name or password in the recv.ini file is not consistent with the login data in the Recipient File of the Server, a login page is shown, that allows the user to log in manually with a correct user name and password. When the manual login is used, the user name in the recv.ini file is automatically overwritten with the user name used at the login interface. The password is deleted from recv.ini, i. e. for future TelliNet sessions a manual login is always required.			
unit range 0/1 default 1 = on	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries tc-recv - recv.ini - [etcp_parameters] - authentication tc-send - send.ini - [authentication] - activate_for_etcp tc-recv - recv.ini - [recipient] - user_name tc-recv - recv.ini - [recipient] - user_key tc-send - *.rcv - [recipient] - user_name tc-send - *.rcv - [recipient] - user key		

[watchdog]	file recv.ini
This section specifies the parameters of a built-in process of TelliNet / TelliCast / TelliVision that keeps the overall performance of the Client under surveillance.	

activate	section [watchdog]	file recv.ini
Activates or deactivates the program internal watchdog.		
syntax activate = <value of range>	example activate = 1	
Activates the program internal watchdog. Possible entries are: 1 = the watchdog is activated. 0 = the watchdog is not activated.		
unit range 0/1 default 1 = on	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable related entries tc-send - send.ini - [watchdog]	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously

alive_check_interval		section [watchdog]	file recv.ini
The "working" process has to send an alive message to the watchdog process in the specified interval.			
syntax alive_check_interval = <seconds>		example alive_check_interval = 40	
The "working" process has to send an alive message to the watchdog process in the given interval (of minimum 30 seconds). If the alive message is not received for multiple intervals, the "working" process is restarted.			
unit seconds range 30 or more default 30 seconds	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [watchdog] - activate		

restart_time		section	file
		[watchdog]	recv.ini
Force a restart at given times.			
syntax restart_time = <hh:mm> restart_time = <hh:mm ddd>		example restart_time = 05:15 restart_time = 00:00 Sun	
<p>The watchdog can be configured to restart the "working" process at a given time of each day or of a specific day per week.</p> <p>This entry specifies the UTC time of the day when the "working" process is restarted in the format "<hh:mm>" (hour:minute) or "<hh:mm ddd>" (hour:minute day). The day has to be given as the first three letters of the english name of the day, e. g. "Mon" for Monday or "Thu" for Thursday.</p> <p>If only the time is specified, the process is restarted every day.</p> <p>The entry can be given multiple times, i. e. it is possible to restart the process several times during one day.</p>			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default no forced restart	related entries tc-recv - recv.ini - [watchdog] - activate		

max_memory_usage		section [watchdog]	file recv.ini
Maximum amount of memory the "working" process is allowed to allocate.			
syntax max_memory_usage= <byte>		example max_memory_usage= 300000000	
Maximum amount of memory the process is allowed to allocate. Allocation of more memory is considered a program bug and the "working" process is restarted.			
unit byte range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
default 300000000 = 300 MByte	related entries tc-recv - recv.ini - [watchdog] - activate		

max_cpu_usage		section [watchdog]	file recv.ini
Maximum percent of CPU the "working" process is allowed to use over a period of time (see max_cpu_usage_period).			
syntax max_cpu_usage = <value of range>		example max_cpu_usage = 75	
In case the CPU usage exceeds the given amount for a period of approximately "max_cpu_usage_period" the process is restarted. Please note that it is not possible to check cpu usage when using WIN9x operating systems due to restrictions in these operating systems. Therefore this parameter is ignored under WIN9x operating systems.			
unit percent range 1-100 default 90%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries tc-recv - recv.ini - [watchdog] - activate tc-recv - recv.ini - [watchdog] - max cpu usage period		

max_cpu_usage_period		section [watchdog]	file recv.ini
Calculation period for max_cpu_usage.			
syntax max_cpu_usage_period = <seconds>		example max_cpu_usage_period = 120	
In case the CPU usage exceeds the amount given with "max_cpu_usage" for the configured period, the process is restarted by the internal watchdog. Please note that it is not possible to check cpu usage when using WIN9x operating systems due to restrictions in these operating systems. Therefore this parameter is ignored under WIN9x operating systems.			
unit seconds range default 120	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries tc-recv - recv.ini - [watchdog] - activate tc-recv - recv.ini - [watchdog] - max_cpu_usage		

max nr of core files		section [watchdog]	file recv.ini
Under Unix like operating systems the watchdog will archive the given number of possibly existing core files.			
syntax max nr of core files = <number>		example max nr of core files = 10	
This parameter restricts the storage of possibly existing core files on Unix like operating systems. If the number of core files exceeds the configured number, the oldest stored core file is deleted each time a new core file is stored. Whenever a core file is generated, the current log file is stored under the name "<log file name>.tc-send.<timestamp>" and a new log file is opened. "timestamp" is the UTC date and time when the core file and the log file have been stored.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
range			
default 10	related entries tc-recv - recv.ini - [watchdog] - activate		

[mail]	file recv.ini
<p>This section can be used to send an email in case of errors, warnings or restarts caused by the watchdog, to the user (runtime error mail) and to send an email to the operator / technical support in case the user needs help with the system (feedback mail). Runtime error mail will be sent if "mail_server_address", "sender_address" and "runtime_error_mail_recipient" are specified. To use the feedback mail it is sufficient to specify the "mail_server_address".</p>	

mail_server_address	section [mail]	file recv.ini
SMTP server for automatic and feedback mail.		
syntax mail_server_address = <ip address> mail_server_address = <dns_name> mail_server_address = <ip address> : <port> mail_server_address = <dns name> : <port>	example mail_server_address = 172.33.33.33 mail_server_address = mail.company.com mail_server_address = 172.33.33.33:1200 mail_server_address = mail.company.com:1200	
Address of the mail server that is used as SMTP-Server for outgoing mails. This entry is necessary to activate the functionality of both feedback mail and runtime error mail.		
unit	this parameter is	read by system
range	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
default	related entries tc-recv - recv.ini - [mail] - sender_address tc-recv - recv.ini - [mail] - runtime_error_mail_recipient	

sender_address	section [mail]	file recv.ini
Originator of runtime_error_mail and default originator shown in feedback mail.		
syntax sender_address = <e-mail address>	example sender_address = user@company.com	
Source email address of the sender of "feedback mail" and "runtime error mail". The entry is mandatory to activate the runtime error mail functionality. For the feedback mail functionality the entry is optional, as the sender e-mail address can also be specified directly in the feedback mail form of the web interface. The sender email address specified here is given as default value in the respective field of the form.		
unit	this parameter is	read by system
range	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
default	related entries tc-recv - recv.ini - [mail] - mail_server_address tc-recv - recv.ini - [mail] - runtime_error_mail_recipient	

default_feedback_mail_recipient		section [mail]	file recv.ini
Default recipient for feedback mail.			
syntax default_feedback_mail_recipient = <e-mail address>		example default_feedback_mail_recipient = support@company.com	
Destination email address of the recipient of the "feedback mail", which allows to communicate e. g. some comments / problem descriptions to the operator / technical support of the software via a form in the web interface. The entry can be repeated if more than one person should be notified. Recipients can also be specified directly in the feedback mail form of the web interface. The recipients specified here are given as default values in the respective field of the form.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default	related entries tc-recv - recv.ini - [mail] - mail server address		

runtime_error_mail_recipient		section [mail]	file recv.ini
Recipient of automatic error messages.			
syntax runtime_error_mail_recipient = <email address>		example runtime_error_mail_recipient = operator@company.com	
Email address of the person that should be informed by email when an error or warning occurs (runtime error mail).			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default	related entries tc-recv - recv.ini - [mail] - mail_server_address tc-recv - recv.ini - [mail] - sender address		

[logging]	file recv.ini
The section [logging] can be used to fine tune the log output.	

log_level		section	file
		[logging]	recv.ini
none			
syntax log_level = <value of range>		example log_level = normal	
Specifies the extent of logged debugging output. Possible values are: none = no debugging output quiet = only error and warning messages are logged normal = error and warning messages and information about important system functions are logged verbose = error and warning messages and detailed information about the behaviour of the program are logged			
unit range none, quiet, normal, verbose default normal	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries		

log_file_size		section [logging]	file recv.ini
maximum size of single log file in bytes			
syntax log_file_size = <bytes>		example log_file_size = 1000000	
Specifies the maximum size of the log file in bytes. When the log file gets larger than "log_file_size", it is stored as "<log_file>.1", a new log file is created for new logging and all previous log files are shifted: the old "<log_file>.1" is renamed to "<log_file>.2" etc.			
unit byte range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	default 1000000 = 1 MByte	related entries tc-recv - recv.ini - [logging] - log_file number	

log_file_number		section	file
		[logging]	recv.ini
maximum overall number of logfiles (actual + old)			
syntax		example	
log_file_number = <number of files>		log_file_number = 4	
Specifies the number of log files to store. Each time a log file is shifted (shifting described with entry for "log_file_size"), it is stored with a higher number. When the new name will be "<log_file>.<log_file number>", the file is removed.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
	related entries		
4	tc-recv - recv.ini - [logging] - log_file_size		

[locations]	file recv.ini
The section [locations] tells the Client where to find the files and directories it needs for its configuration. There are default values for all entries and thus the whole section or some entries may be left out. All locations are relative to the working directory of the Client.	

webbroadcast cache_directory		section [locations]	file recv.ini
TelliCast WebBroadcast only. Source directory holding pushed web data.			
syntax webbroadcast_cache_directory = <directory name>		example webbroadcast- cast cache_directory = received/webbroadcast	
Directory used as web cache containing WebBroadcast data. The directory has to be identical with the directory used as incoming directory for the web content, which is sent via WebBroadcast. Therefore this entry has to be made when another than the default value is used for the "target_directory" specified in the Client Channel File.			
unit range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
default received/webbroadcast	related entries tc-recv - recv-channels.ini - [<channel name>] - target_directory		

prefetching_cache_directory		section [locations]	file recv.ini
TelliNet HTTP-Prefetching only. Temporary directory holding HTTP prefetching data.			
syntax prefetching_cache_directory = <directory name>		example prefetching_cache_directory = receiving/pushed_data	
The directory used as web cache containing HTTP-Prefetching data.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default receiving/pushed data	related entries tc-recv - recv.ini - [http prefetching]- use http prefetching		

file_database_directory		section [locations]	file recv.ini
TelliCast only (all modules). Temporary directory to store partially received data.			
syntax file_database_directory = <directory name>		example file_database_directory = receiving/tmp	
The directory for caching data fragments. The path is relative to the working directory of the Client.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default receiving/tmp	related entries		

mail_database_directory		section [locations]	file recv.ini
TelliCast OfflineMail (Push) only: temporary directory to store received (pushed) e-mails.			
syntax mail database directory = <directory>		example mail database directory = receiving/mail	
The directory for temporary storage of received (pushed) e-mails. The path is relative to the working directory of the Client.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default receiving/mail	related entries		

media_database_directory		section [locations]	file recv.ini
TelliVision MediaOnDemand only: storage for media content			
syntax media_database_directory = <directory name>		example media_database_directory = receiving/media	
Directory for the storage of media content.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default receiving/media	related entries		

channels_file		section [locations]	file recv.ini
TelliCast and TelliVision: Receiver channel subscription file			
syntax channels_file = <file name>		example channels_file = recv-channels.ini	
Specifies the Client Channel File containing the parameters for the reception of the TelliCast data channels.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default recv-channels.ini	related entries		

log_file		section [locations]	file recv.ini
Location/name of log file for status/error/debug output.			
syntax log_file = <file name> .log log_file = <directory> / <file name> .log log_file = syslog log_file = <memory>		example log_file = recv.log log_file = ../logging/receiving/recv.log log_file = syslog log_file = <memory>	
<p>The log file that is created automatically at startup to write the debugging information into it during operation.</p> <p>On Linux operating systems the log messages can also be redirected to the local syslog daemon. In this case please specify "log_file = syslog".</p> <p>To write log messages to the memory of the local host instead on storing it into a log file please specify "log_file = <memory> " (in this case the variable do not have to be exchanged by the according value but has to be written as a variable in pointed brackets). If log message are written into the memory, they can only be read from the software internal web interface.</p>			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default recv.log	related entries tc-recv - recv.ini - [logging]		

license_file		section	file
		[locations]	recv.ini
Client License File			
syntax license_file = <file name> license_file = <directory>/<file name>		example license_file = license.ini license_file = ../administration/license.ini	
The name of the license file containing the product license. This can include a path to the file in case it is not stored in the working directory.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default license.ini	related entries		

4.2.2.2 TELLiNET only sections

[etcp_parameters]	file recv.ini
The section [etcp_parameters] specifies the parameters for enhanced TCP and web content push.	

server_address	section [etcp_parameters]	file recv.ini
IP address/DNS name:port specifying the server proxy (tc-send) this TelliNet client should connect to.		
syntax server_address = <address> server_address = <address> : <port> server_address = <dns name> server_address = <dns name> : <port>	example server_address = 172.27.1.2	
Name or numeric address of the Server Proxy host and port at the Server Proxy host that the Client Proxy will connect to. This entry has to be identical with the "external_listen_address" or, if this is not specified, with the "listen_interface_address" and "listen_port" specified in the section [etcp_parameters] in the Server Configuration File of the Server Proxy. If a host name is entered and the DNS system is configured to assign more than one address to this name, multiple hosts can be addressed with one entry. It is also possible to address more than one host by adding multiple entries. If more than one Server Proxy host is specified, a Server Proxy is chosen randomly for every new connection. This entry is optional. The TelliNet Server announces its address and port for incoming Client connections to the TelliNet Clients in regular intervals (in case of etcp load balancing the Server will announce all listen addresses and ports of all Servers that are known to him). The address specified with this entry is only used if the Client is not receiving server announcements. Please note that the entry becomes mandatory for the use of TelliNet in case that the entry "server_announcement_address" is set to "none", i. e. the Client will not receive the server announcements.		
unit range default any interface, port 9200	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	related entries tc-send - send.ini - [etcp_parameters] - external_listen_address tc-send - send.ini - [etcp_parameters] - listen_port tc-send - send.ini - [etcp_parameters] - listen_interface_address tc-recv - recv.ini - [etcp_parameters] - server announcement address	

server_announcement_address		section [etcp_parameters]	file recv.ini
Multicast Address and optionally port number to listen for TelliNet server and transponder announcements.			
syntax server_announcement_address = <address> : <port>		example server_announcement_address = 229.1.0.1:2513	
<p>Multicast address and optionally port number to listen for TelliNet server announcements and optionally transponder-ID announcements.</p> <p>If "none" is specified, the Client does not listen for any server announcements. In that case the specification of the entry "server_address" in this section becomes mandatory to use TelliNet. The entry "server_announcement_address" has to be identical with the "address" in the section [server_announcement] in the Server Configuration File send.ini of the TelliNet Server. If that section is not specified, the default values are valid for server announcements.</p> <p>For each transponder used, there has to be a section [server_announcement] in the Server Configuration File send.ini, each containing a different address. Therefore the entry "server_announcement_address" can also be given multiple times to specify all of these addresses at the Client.</p>			
unit ip:port range ip:port / none default 229.1.0.1:2513	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	related entries tc-send - send.ini - [server_announcement] - address tc-recv - recv.ini - [etcp_parameters] - server address		

http_listen_address		section	file
		[etcp_parameters]	recv.ini
Interface and local port or interface:port for browsers to connect to client proxy using HTTP.			
syntax http_listen_address = <address>:<port> http_listen_address = <port>		example http_listen_address = 172.1.1.34:9202 http_listen_address = 9202	
Local address and port for incoming connections from the browser. Please note: if the [internal_tcp] functionality is active (i. e. when the entry "activate = 1" is available in the section [internal_tcp]), the "http_listen_address" will also include the "virtual_ip_address" (as well as any other available interface) unless you specify different interface explicitly using "http_listen_address=interface:port".			
unit range default 9202	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_parameters] - socks_listen_address tc-recv - recv.ini - [etcp_parameters] - capturing_listen_address tc-recv - recv.ini - [etcp_parameters] - autoconfig_listen_address		

socks_listen_address		section [etcp_parameters]	file recv.ini
Local interface or port or interface:port for browsers to connect to client proxy using SOCKS			
syntax socks_listen_address = <address>:<port> socks_listen_address = <port>		example socks_listen_address = 9203	
Local address and port for incoming connections from SOCKS clients in the format "<address>:<port>" or "<port>". Please note: When the [internal_tcp] functionality is active (i. e. when the entry "activate = 1" is available in the section [internal_tcp]), the "socks_listen_address" will also include the "virtual_ip_address" (as well as any other available interface) unless you specify another interface explicitly using "socks_listen_address=interface:port".			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range	related entries tc-recv - recv.ini - [etcp_parameters] - http_listen_address tc-recv - recv.ini - [etcp_parameters] - capturing_listen_address tc-recv - recv.ini - [etcp_parameters] - autoconfig_listen_address		
default 9203			

autoconfig_listen_address		section [etcp_parameters]	file recv.ini
local port or interface:port for browsers to connect to client proxy for browser proxy autoconfiguration.			
syntax autoconfig_listen_address = <port>		example autoconfig_listen_address = 9204	
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default 9201	related entries tc-recv - recv.ini - [etcp_parameters] - http_listen_address tc-recv - recv.ini - [etcp_parameters] - socks_listen_address tc-recv - recv.ini - [etcp_parameters] - capturing_listen_address		

allowed_address		section	file
		[etcp_parameters]	recv.ini
Allow only client programs (e.g. web browsers) from the listed hosts (IP addresses) to use this tc-recv process as proxy.			
syntax allowed_address = <address>		example allowed_address = 127.0.0.1	
This entry can be used to restrict the right to connect to the Client. It specifies the address of the interface allowed to be used for connections to the Client. The entry can be given multiple times to allow more than one host to connect to the Client. (Default: connections are accepted from any host)			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default connections are accepted from any host	related entries		

max_send_data_rate		section	file
		[etcp_parameters]	recv.ini
Maximum data rate used from client proxy to Server Proxy.			
syntax		example	
max receive data rate = <bits>		max send data rate = 1000000	
Maximum data rate per ETCP Association in bit/s for outgoing traffic. If a maximum data rate is also configured for the respective Client Proxy in a Recipient File of the Server (entry "max_receive_data_rate" in the section [recipient]) or in the Configuration File send.ini of the Server as "max_receive_data_rate" (section [etcp_parameters]), the smallest of the configured data rates is used.			
unit	this parameter is	read by system	
bit/s	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
	<input type="checkbox"/> repeatable		
default	related entries		
100 000 000 = 100 Mbit/s	tc-send - send.ini - [etcp_parameters] - max_receive_data_rate		
	tc-send - *.rcv - [recipient] - max receive data rate		

max_receive_data_rate		section	file
		[etcp_parameters]	recv.ini
Maximum data rate received from server proxy - please note: this rate might be limited at the server proxy as well.			
syntax		example	
max_send_data_rate= <bits>		max_receive_data_rate=4000000	
Maximum data rate per ETCP Association in bit/s for incoming traffic. If a maximum data rate is also configured for the respective Client Proxy in a Recipient File of the Server (entry "max_send_data_rate" in the section [recipient]) or in the Configuration File send.ini of the Server as "max_send_data_rate" (section [etcp_parameters]), the smallest of the configured data rates is used.			
unit	this parameter is	read by system	
bit/s	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
	<input type="checkbox"/> repeatable		
default	related entries		
100 000 000 = 100 Mbit/s	tc-send - send.ini - [etcp_parameters] - max_send_data_rate		
	tc-send - *.rcv - [recipient] - max_send_data_rate		

etcp_send_protocol		section [etcp_parameters]	file recv.ini
Defines which underlying transport protocol (TCP vs. UDP) is used to send user data to the Server Proxy.			
syntax etcp_send_protocol = <value of range>		example etcp_send_protocol = 1	
This entry defines, which protocol is used to send data from the Client Proxy to the Server Proxy. Possible entries are: 0 = TCP is used 1 = UDP is used to send the data Please note that a TCP control connection is always used from Client Proxy to the Server Proxy, even if the data is sent via UDP, to keep a connection state.			
unit range 0 = TCP, 1 = UDP default 0 = TCP	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_parameters] - local_udp_address tc-send - send.ini - [etcp_parameters] - max_receive_data_rate tc-recv - recv.ini - [etcp_parameters] - max_send_data_rate		

local_udp_address		section	file
		[etcp_parameters]	recv.ini
The Client Proxy will listen for UDP packets sent by the Server Proxy on the IP address and port number given.			
syntax local_udp_address = <ip address> local_udp_address = <ip address>:<port> local_udp_address = <port> local_udp_address = <dns name> local_udp_address = <dns name>:<port>		example local_udp_address = 172.27.1.51 local_udp_address = 172.27.1.51:4711 local_udp_address = 4711 local_udp_address = outgoing.company.com local_udp_address = outgoing.company.com:4711	
<p>The own IP address that the Client Proxy will report to the Server Proxy. The Server Proxy will sent all UDP packets to this IP address when addressing this Client Proxy.</p> <p>As default (if local_udp_address is not specified) the Client Proxy will use the "external_udp_address", i. e. it will report its IP address used for the TCP connection to the Proxy Server. Depending on the IP routing and IP/MAC address mapping (IP/DVB gateway) it may be necessary to send the UDP packets to a different client interface / IP address (e.g. the DVB card). In the latter case you can define the corresponding UDP destination IP address by using "local_udp_address".</p> <p>In case that the Client Proxy is configured to send data packets to the Server Proxy via UDP (see section [etcp_parameters], entry "etcp_send_protocol"), the address and port are also used as source for outgoing packets.</p> <p>Please note: In case that some firewall or address translation is in place, it might be necessary to define "external_udp_address" (section [etcp_parameters] as well.</p>			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default equals external_udp_address	related entries tc-recv - recv.ini - [etcp_parameters] - external_udp_address tc-recv - recv.ini - [etcp_parameters] - etcp_send_protocol		

external_udp_address		section [etcp_parameters]	file recv.ini
The Server Proxy will send UDP packets (ETCP) to this IP address.			
syntax external_udp_address = <ip address> external_udp_address = <ip address> : <port> external_udp_address = <port> external_udp_address = <dns name> external_udp_address = <dns name> : <port>		example external_udp_address = 172.27.1.51 external_udp_address = 172.27.1.51:7900 external_udp_address = 7900 external_udp_address = outgoing.company.com external_udp_address = outgoing.company.com:7900	
The IP address that the Client Proxy will report to the Server Proxy. The Server Proxy will direct all UDP packets to this IP destination address when addressing this Client Proxy (speed mode). If the Client is located behind a firewall that is working with Network Address Translation (NAT), the Server has to send UDP packets to the firewall instead of directly to the Client in order to pass the firewall. This entry specifies the address of the firewall and the port used for the connections. If the entry is available, UDP packets are sent to the firewall. Please note: The firewall has to be configured to forward the UDP packets to the Client (NAT). The IP address used at the Client has to be specified with the entry "local_udp_address".			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default interface according to route to server proxy with dynamic port number	related entries tc-recv - recv.ini - [etcp_parameters] - local_udp_address		

authentication		section	file
		[etcp_parameters]	recv.ini
activates TelliNet authentication			
syntax		example	
authentication = <value of range>		authentication = 1	
Activates the user authentication for TelliNet. Possible entries are: 1 = authentication is activated. 0 = authentication is deactivated.			
unit	this parameter is	read by system	
range 0/1	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	related entries		
0 = off	tc-send - send.ini - [authentication] - activate for etcp		

default_speed_state		section	file
		[etcp_parameters]	recv.ini
Initial speed state.			
syntax default_speed_state = <value of range>		example default_speed_state = 1	
Specifies the speed mode at the start of the Client Proxy. Possible entries are: 0 = the Client Proxy is not in the speed mode when started, 1 = the Client Proxy is in the speed mode when started.			
unit range 0/1 default 1	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_parameters] - toggle speed listen port		

use_icon		section	file
		[etcp_parameters]	recv.ini
display speed (toggle) icon on/off			
syntax		example	
use icon = <value of range>		use icon = 1	
Specifies, whether the Speed Button should be displayed in the task bar on Windows systems. Possible entries are: 1 = the Speed Button is displayed 0 = the Speed Button is not displayed			
unit	this parameter is	read by system	
range 0/1	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default 0 = off	related entries		
	tc-recv - recv.ini - [etcp_parameters] -toggle_speed_listen_port		
	tc-recv - recv.ini - [etcp_parameters] - default speed state		

inactivity_timeout		section	file
		[etcp_parameters]	recv.ini
Idle timeout for server connections (ETCP associations). (0 = off)			
syntax inactivity_timeout = <seconds>		example inactivity_timeout = 180	
'Timeout in seconds of an ETCP Association in case no data is transmitted. There is a corresponding entry existing within the Server Configuration File as well. The smaller of both values will be used.			
unit seconds range 0/1 default 0 = value configured at the server will be used	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-send - send.ini - [etcp_parameters] - inactivity_timeout		

packet_aggregation_timeout		section [etcp_parameters]	file recv.ini
Maximum time TelliNet will aggregate data to compose bigger packets.			
syntax packet_aggregation_timeout = <value of range>		example packet_aggregation_timeout = 50	
<p>TelliNet is aggregating small packets to be send via ETCP to compose bigger packets. The aggregation is completed when a maximum packet size, configured with the parameter packet_size (in this section), is reached.</p> <p>In order to avoid long waiting times, a timeout for the aggration can be configured. This entry specifies the maximum time that TelliNet is waiting for more incoming data after receiving the first data for a packet.</p> <p>The aggregation can be deactivated by configuring a value of "0".</p>			
unit milliseconds range 0 - 200 default 30 milliseconds	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries tc-recv - recv.ini - [etcp_parameters] - packet_size		

packet_size		section	file
		[etcp_parameters]	recv.ini
Maximum packet size used by TelliNet packet aggregation.			
syntax		example	
packet_size = <value of range>		packet_size = 1400	
The configured size includes a typical IP header, but does not include other headers (e. g. MAC layer, IPSEC, ...)			
Please note: The configured packet size only applies to ETCP UDP packets. For those types of data configured to be sent as TCP, the TCP layer will modify the packet sizes.			
unit	this parameter is	read by system	
bytes	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
750 - 64000	<input type="checkbox"/> repeatable		
default	related entries		
1500 bytes	tc-recv - recv.ini - (etcp_parameters) - packet aggregation timeout		

continuous_server_connection		section [etcp_parameters]	file recv.ini
if activated, the Client will try to hold a permanent association to one Server Proxy open			
syntax continuous_server_connection = <value of range>		example continuous_server_connection = 21600	
<p>If this entry is set, the Client will try to hold a permanent ETCP association to one of the Server Proxies open. In case the association is terminated for some reason the Client will immediately try to open a new association. If no new association is possible at the moment, the Client will retry in 60 second intervals</p> <p>In order to better distribute the Clients between the active TelliNet Servers (when LoadSharing is used) the parameter "continuous_server_connection" defines an interval after which the Client will reconnect to one of the active TelliNet Servers whenever the association is idle (see recv.ini [etcp_parameters] inactivity_timeout).</p> <p>Please note: It is not recommended to activate this setting unless [transparent_capturing] is activated on the server! side.</p> <p>Please note: on server side a timeout can be configured for idle associations with the parameter iinactivity_timeout in the section [etcp_parameters] of the Server Configuration File send.ini. This setting is completely independent of the continuous_server_connection parameter. So you should make sure to set send.ini [etcp_parameters] inactivity_timeout to a value higher than continuous_server_connection to prevent the closing of the connection caused by the idle timeout.</p>			
unit seconds range 0-700,000 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
		related entries tc-send - send.ini - [etcp_parameters] - idle timeout	

http_proxy		section [etcp_parameters]	file recv.ini
use this web proxy to retrieve web pages when not using the server proxy.			
syntax http_proxy = <adress>:<port>		example http_proxy = 172.27.1.2:80	
In case a caching web proxy should be used for the connection to HTTP servers if the speed mode is off, the TelliNet Client Proxy can be configured to connect to the proxy instead of directly to the HTTP server. This entry specifies the address and port of the intermediate caching web proxy that the Client Proxy is connecting to.			
unit	range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
default none		related entries tc-recv - recv.ini - [etcp_parameters] - ftp proxy	

target overall memory usage		section [etcp_parameters]	file recv.ini
Allows to delay the processing of new incoming client connections if the given target maximum memory usage is reached.			
syntax target overall memory usage = <bytes>		example target overall memory usage = 10000000	
Specifies the maximum allowed memory usage. If the memory utilisation has reached this value, the processing of new incoming client connections (e.g. from web browsers) is delayed. Please note: Only the processing of additional connections coming from the clients is delayed. E. g. additional push prefetching connections from the server or TelliCast usage can result in the allocation of additional memory as well. Therefore e. g. for set top box integration you should set "target overall memory usage" to a value some MByte lower than the actual maximum.			
unit bytes range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
default unlimited	related entries tc-recv - recv.ini - [etcp_parameters] - http prefetching memory cache		

[etcp_compression]	file recv.ini
The section [etcp_compression] specifies, whether data, that is sent from the TelliNet Client to the TelliNet Server is compressed.	

compression_level	section [etcp_compression]	file recv.ini
Compression level for compression of data sent from the TelliNet Client to the TelliNet Server.		
syntax compression_level = <value of range>	example compression_level = 6	
The level of compression for data sent from the TelliNet Client to the TelliNet Server. "1" is the lowest compression level, "9" is the highest compression level. A higher level will usually result in better compression, but in parallel in higher memory and CPU usage.		
unit range 1 - 9 default 2	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	related entries tc-recv - recv.ini - [etcp_compression] tc-send - send.ini - [etcp_compression] - compression_level	

http_compression		section [etcp_compression]	file recv.ini
Turns compression on for HTTP data sent from the TelliNet Client to the TelliNet Server.			
syntax html_compression = <value of range>		example html_compression = 1	
Specifies, whether HTTP data sent via the HTTP proxy of TelliNet Client to the TelliNet Server is compressed. Possible entries are: 1 = HTTP data is compressed. 0 = HTTP data is not compressed.			
unit range 0/1 default 0	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
related entries			

port_forwarding_compression		section	file
		[etcp_compression]	recv.ini
Specifies if port forwarding traffic is compressed.			
syntax port_forwarding_compression = <value of range>		example port_forwarding_compression = 1	
Specifies, whether port forwarding traffic is compressed or not compressed. Possible entries are: 1 = port forwarding traffic is compressed. 0 = port forwarding traffic is not compressed.			
unit range 0/1 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries		

socks_tcp_compression		section	file
		[etcp_compression]	recv.ini
Specifies if SOCKS TCP traffic is compressed.			
syntax socks_tcp_compression = <value of range>		example socks_tcp_compression = 1	
Specifies, whether SOCKS TCP traffic is compressed or not compressed. Possible entries are: 1 = SOCKS TCP traffic is compressed. 0 = SOCKS TCP traffic is not compressed.			
unit range 0/1 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries		

socks_udp_compression		section	file
		[etcp_compression]	recv.ini
Specifies if SOCKS UDP traffic is compressed.			
syntax socks_udp_compression = <value of range>		example socks_udp_compression = 1	
Specifies, whether SOCKS UDP traffic is compressed or not compressed. Possible entries are: 1 = SOCKS UDP traffic is compressed. 0 = SOCKS UDP traffic is not compressed.			
unit range 0/1 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries		

capturing_compression		section	file
		[etcp_compression]	recv.ini
Turns compression on for captured data sent from the TelliNet Client to the TelliNet Server.			
syntax		example	
capturing_compression = <value of range>		capturing_compression = 1	
Specifies, whether data sent via the TelliNet Client "capturing_listen_address" is compressed. Possible entries are: 1 = the data is compressed. 0 = the data is not compressed.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
default	<input type="checkbox"/> repeatable		
	related entries		
0 = off	tc-recv - recv.ini - [etcp parameters] - capturing_listen_address		

smb_compression		section	file
		[etcp_compression]	recv.ini
Turns compression on for data send to the TelliNet Server for SMB FileSharing connections.			
syntax		example	
smb_compression = <value of range>		smb_compression = 1	
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
0/1	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
0 = off	related entries		
	tc-recv - recv.ini - [transparent_capturing] - capture_smb		
	tc-recv - recv.ini - [etcp_port_forwarding] -		
	use smb enhancement		

[http_prefetching]	file recv.ini
This section allows to configure and tune the HTTP Prefetching functionality.	

use_http_prefetching	section [http_prefetching]	file recv.ini
0/1 activation flag for HTTP Prefetching.		
syntax use http_prefetching = <value of range>	example use http_prefetching = on	
This entry specifies, whether HTTP-Prefetching is used. Possible entries are: "on" or "1" = HTTP-Prefetching is used, "off" or "0" = HTTP-Prefetching is not used.		
unit range 1/0 or on/off default on = activated	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries	

conditional_prefetching		section [http_prefetching]	file recv.ini
Defines which conditional_prefetching_mode will be used for Conditional HTTP GET requests. Please note that the server might be configured to allow just a subset of these modes.			
syntax conditional_prefetching = <value of range>		example conditional_prefetching = normal	
<p>This entry specifies, which conditional prefetching mode will be used for conditional HTTP GET requests.</p> <p>Please note that the server might be configured to allow only a subset of the possible modes. The mode for conditional Prefetching can be:</p> <p>“off”:</p> <p>HTTP Prefetching is deactivated for conditional HTTP requests of the respective Client.</p> <p>“normal”:</p> <p>The TelliNet Server saves the Last Modified Date of the request. It then sends the request to the web server without the Conditional Header and without the Entity Tag. Prefetching is done. At the TelliNet Server, the Last Modified Dates of the prefetched web objects are compared to the Last Modified Date sent with the conditional request. Only the web objects that differ in the Last Modified Date are sent to the Client. For all other web objects only the status "not changed" is transmitted to the Client.</p> <p>Using this mode, the Tellinet Server is not sending the web objects to the Client with the Last Modified Date received from the web server. Instead, the Last Modified Date is set to the current system time and date of the TelliNet Server system.</p> <p>The conditional prefetching mode "normal" thereby very often avoids to send unnecessarily objects to the Client that are likely to be still in (internal) caches e. g. of the used web browser.</p> <p>“more”:</p> <p>This mode is the same as the mode "normal", but the Last Modified Dates are not changed. With this method it is likely that some web objects which are already in the cache of the client’s browser are transmitted.</p> <p>“all”:</p> <p>The TelliNet Server ignores all conditions given in HTTP requests. Requests to the Web Server are sent without the "if-modified-since" Header and without the Entity Tag and normal prefetching is done.</p> <p>With this mode, the Client receives the pages in short time, but it is very likely that HTTP Pre-fetching will result in additional bandwidth consumption towards the TelliNet Server.</p>			
unit range off, normal, more, all default normal	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries tc-recv - recv.ini - [http parameters] - use http_prefetching		

http_prefetching_object_size		section [http_prefetching]	file recv.ini
Maximum number of bytes prefetched per object. If an object exceeds the configured size, only the initial bytes will be prefetched - the transmission of the following bytes will be delayed until the client (e.g. web browser) explicitly requests the corresponding object.			
syntax http_prefetching_object_size = <bytes>		example http_prefetching_object_size = 100000	
Maximum number of bytes prefetched per object. If an object exceeds the configured size, only the initial bytes will be prefetched. The transmission of the following bytes will be delayed until the client (e. g. the web browser) explicitly requests the corresponding object. The parameter can be deactivated with the entry of "0".			
unit bytes range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
default 100000 bytes = 100 kBytes	related entries tc-recv - recv.ini - [http_prefetching] - use http_prefetching		

http_prefetching_cache_size		section	file
		[http_prefetching]	recv.ini
Limits the cache size for HTTP objects to the given maximum value.			
syntax		example	
http_prefetching_cache_size = <bytes>		http_prefetching_cache_size = 2000000	
In case the size of the cache exceeds the limit given here, old HTTP objects are deleted from the cache. If this is not sufficient to keep the cache size to the maximum value, newly prefetched documents are also deleted and directly transmitted files are not cached. Please note: depending on the parameter "http_prefetching_memory_cache" the cache for HTTP objects will reside within the system memory or in the file system. Per default it resides in the file system.			
unit	this parameter is	read by system	
bytes	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
	<input type="checkbox"/> repeatable		
default	related entries		
unlimited	tc-recv - recv.ini - [http_prefetching]		
	http_prefetching_memory_cache		
	tc-recv - recv.ini - [http_prefetching] - use http_prefetching		

http_prefetching_memory_cache		section	file
		[http_prefetching]	recv.ini
Can be enabled to hold the complete HTTP prefetching cache in memory.			
syntax http_prefetching_memory_cache = <value of range>		example http_prefetching_memory_cache = 1	
Per default the HTTP cache used for HTTP Prefetching is located in the file system. This entry can be used to store cache data into the memory instead of into the file system. Possible entries are: 0 = the cache data is mainly stored in the file system and only a small part of the prefetching cache resides in memory. 1 = the cache data is stored completely in memory. In that case it is recommended to adjust the parameter "http_prefetching_cache_size" in this section.			
unit range 0/1 default 0=off - cache is stored in file system	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries tc-recv - recv.ini - [http_prefetching] - http_prefetching_cache_size tc-recv - recv.ini - [http_prefetching] - target_overall_memory_usage tc-recv - recv.ini - [http_prefetching] - use http_prefetching		

[proxy_excludes]	file recv.ini
The [proxy_excludes] section allows to specify a set of URLs/hosts for that the Tellinet Client will handle connections locally instead of forwarding the requests to the TelliNet Server.	

exclude	section [proxy_excludes]	file recv.ini
will result in the TelliNet client handling requests for URL/host locally without forwarding the request to the TelliNet server.		
syntax exclude = [<protocol> ://] <host> [: <port>] [/ <url-start>]	example exclude = localhost exclude = http://intranet exclude = http://localhost:80/data/ exclude = 172.27.*.* exclude = *.company.com	
Specifies an URL or host. The TelliNet Client will handle requests for that URL/host locally without forwarding the request to the TelliNet Server and connect directly to the web server. The syntax of the entry is: [<protokoll> :\\] <host> [: <port>] [/ <url-start>] e.g.: exclude = localhost exclude = http://intranet exclude = http://localhost:80/*.html exclude = 172.27.*.100 exclude = *.company.com		
unit	this parameter is	read by system
range	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
default default	related entries	

[etcp_port_forwarding]	file recv.ini
<p>The section [etcp_port_forwarding] is used to set up parameters for port_forwarding (e. g. POP3 mail reception, SMB File Sharing with TelliNet). In case data is received or sent to multiple different ports, this section has to be repeated for every local port number used.</p> <p>This section is read continuously, i. e. changes in the section will become valid immediately without restart of the TelliNet Client.</p>	

remote_address	section [etcp_port_forwarding]	file recv.ini
Remote IP address/DNS name:port to which incoming connections will be forwarded.		
syntax remote_address = <address> : <port> remote_address = <dns name> : <port>	example remote_address = 172.27.1.2:110	
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input type="checkbox"/> at startup only <input checked="" type="checkbox"/> continuously
range		
default	related entries tc-recv - recv.ini - [etcp_port_forwarding] - listen_address	

listen_address	section	file
	[etcp_port_forwarding]	recv.ini
Interface:port local applications can connect to.		
syntax listen_address = <address>:<port> listen_address = <dns name>:<port>	example listen_address = 127.0.0.1:110	
<p>Local interface and port at the TelliNet Client Proxy. The local application (e. g. e-mail client) has to connect to this port to receive or send data.</p> <p>The incoming connections will be forwarded to the remote_address given within this section.</p> <p>Please note: When the [internal_tcp] functionality is active (i. e. when the entry "activate = 1" is available in the section [internal_tcp]), the "listen_address" will also include the "virtual_ip_address" (as well as any other available interface) unless you specify another interface explicitly using "listen address=interface:port".</p>		
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input type="checkbox"/> at startup only <input checked="" type="checkbox"/> continuously
range		
default localhost:9210	related entries tc-recv - recv.ini - [etcp_port_forwarding] - remote address	

use smb enhancement		section [etcp port forwarding]	file recv.ini
Activates SMB enhancement for this ETCP port forwarding connection.			
syntax use smb enhancement = <value of range>		example use smb enhancement = 1	
Please note that on the server and client side the TelliNet module Enhanced-SMB has to be li- censed and activated to activate SMB enhancement for the port forwarding connection with this entry.			
unit range 0/1 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries		

[internal_tcp]	file recv.ini
<p>The section [internal_tcp] defines the parameter for activating and using the TelliNet internal TCP implementation. The internal TCP implementation is currently only available for Linux 2.2, 2.4, 2.6.</p> <p>Please note: When activating the TelliNet internal TCP stack to be used on a physical interface please disable Linux ip forward.</p> <p>When using a Tunnel interface (tun0-3) please make sure to activate Linux ip_forward.</p>	

activate	section [internal tcp]	file recv.ini
activates the TelliNet internal TCP implementation		
syntax activate = <value of range>	example activate = 1	
Activates or deactivates the TelliNet internal TCP implementation. Possible entries are: 0 = deactivated, 1 = activated.		
unit range 0/1 default 0 = deactivated	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries tc-recv - recv.ini - [internal tcp]	

interface		section	file
		[internal tcp]	recv.ini
defines the interface used			
syntax interface = <interface name> interface = <ip address> interface = <dns name> interface = <tunnel interface>		example interface = eth0 interface = 172.27.1.51 interface = external.company.com interface = tun0	
Specifies the interface used for the internal TCP implementation. This can be specified as an interface name, an IP address, a DNS name or a tunnel interface. The definition of tunnel interfaces is restricted to the utilization of tun0, tun1, tun2 and tun3. Please note: when a tunnel interface is specified, TelliNet will create the corresponding tunnel interface.			
unit range default eth0	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
		related entries tc-recv - recv.ini - [internal tcp] - activate	

virtual ip address		section [internal tcp]	file recv.ini
virtual ip address to be used by the TelliNet internal TCP implementation			
syntax virtual_ip_address = <ip address/subnet> virtual_ip_address = <dns name>		example virtual_ip_address = 172.27.1.248 virtual_ip_address = 172.27.1.248/24 virtual_ip_address = virtual.company.com	
The TelliNet internal TCP implementation acts like a second virtual protocol stack on an existing (physical or virtual) network interface. In order not to interfere with the existing protocol stacks it is mandatory to define a new local IP address as virtual_ip_address which is not used by the local or any other computer on the network. All outgoing user data will be sent using this virtual ip address.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default 172.27.1.248/32 - in case no subnet size is defined /32 is assumed	related entries tc-recv - recv.ini - [internal_tcp] - activate tc-recv - recv.ini - [internal_tcp] - interface tc-recv - recv.ini - [internal_tcp] - tunnel_ip_address tc-recv - recv.ini - [transparent capturing] - restore ip address		

tunnel_ip_address		section [internal_tcp]	file recv.ini
IP address for created tunnel interface			
syntax tunnel_ip_address = <ip address/subnet> tunnel_ip_address = <dns name>		example tunnel_ip_address = 172.27.1.249 vtunnel_ip_address = 172.27.1.249/24 tunnel_ip_address = virtual12.company.com	
This parameter is only used when a tunnel is specified with the parameter "interface" in this section. It specifies the IP address for the tunnel interface used.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default virtual_ip_address + 1 - in case no subnet size is defined /32 is assumed	related entries tc-recv - recv.ini - [internal_tcp] - activate tc-recv - recv.ini - [internal_tcp] - virtual_ip_address		

execute_script		section	file
		[internal tcp]	recv.ini
defines the path/filename (relative to the active working directory) of a shell script to be executed after activating the TelliNet internal TCP stack			
syntax		example	
execute_script = <script location (path and name)>		execute_script = ./tun-route.sh	
<p>Specifies a script that will be executed once after activating the TelliNet internal TCP stack. This shell script might be useful to adjust routing or iptable configurations to a new tunnel interface generated e.g. by interface=tun0.</p> <p>The script has to be specified as the path to and the name of the script. The path has to be relative to the working directory.</p> <p>Please note: The script will be executed with the following parameters: {start stop} <interface> <ip address> <interface> is the interface specified in this section and <ip address> is the virtual IP address specified in this section or, if a tunnel is used, the IP address of the tunnel interface.</p>			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
	related entries		
tc-recv - recv.ini - [internal tcp] - activate			

[transparent_capturing]	file recv.ini
<p>The section [transparent_capturing] specifies parameters for the configuration of TransparentCapturing. It can only be used when the module TransparentCapturing is licensed and activated at the Client and the module TransparentCapturingClient is licensed and activated at the Server. Please note that the TelliNet internal TCP stack needs to be activated in order to use TransparentCapturing (Please see the section [internal_tcp].)</p>	

activate	section [transparent_capturing]	file recv.ini
activates Transparent Capturing on the interface defined as [internal_tcp] interface		
syntax activate = <value of range>	example activate = 1	
Activates the TransparentCapturing functionality. Possible entries are: 0 = deactivated 1 = activated The interface specified for internal tcp is used for TransparentCapturing. Therefore the section [internal_tcp] has to be activated as well to use Transparent Capturing.		
unit range 0/1 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries tc-recv - recv.ini - [internal_tcp] - activate tc-recv - recv.ini - [internal_tcp] - interface tc-recv - recv.ini - [transparent_capturing]	

capture_http		section [transparent capturing]	file recv.ini
Defines a port number or range of port numbers to be considered as HTTP connections.			
syntax capture_http = <port> capture_http = <port> - <port>		example capture_http = 80 capture_http = 80-88	
In case that TelliNet HTTP Prefetching is activated connections with the destination port numbers specified with this entry will be handled as HTTP connections including HTTP prefetching and HTTP specific compression.			
unit range default 80	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries tc-recv - recv.ini - [transparent capturing] - activate		

capture_smb		section [transparent capturing]	file recv.ini
Defines a port number or range of port numbers to be considered as SMB FileSharing connections.			
syntax capture_smb = <port number> capture_smb = <port number> -<port number>		example capture_smb = 445 capture_smb = 445-447	
Please note that this entry is only valid if the TelliNet module Enhanced-SMB has to be licensed and activated on the server and client side.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default 445	related entries		

exclude		section	file
		[transparent_capturing]	recv.ini
all packets/connections with the corresponding destination addresses will be excluded from being captured			
syntax		example	
exclude = <ip address>		exclude = 10.10.10.1	
exclude = <ip address> - <ip address>		exclude = 10.0.0.0-10.255.255.255	
When TransparentCapturing is activated, all connections from the configured interface will be captured by the TelliNet Client except of the ones directed to local IP addresses and the complete IP subnet defined with the entry "virtual_ip_address" in the section [internal_tcp]. It is possible to exclude packets/connections with other destination addresses as well. The corresponding destination addresses have to be specified with this entry. Please note: By defining exclude statements you can only extend and not override the default.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
default	<input checked="" type="checkbox"/> repeatable		
all local IP addresses and the complete IP subnet defined by [internal_tcp] virtual ip address	related entries		
	tc-recv - recv.ini - [transparent_capturing] - activate tc-recv - recv.ini - [internal_tcp] - virtual_ip_address		

non_tcp_forward_interface		section	file
		[transparent_capturing]	recv.ini
activates non tcp forwarding and specifies an interface to which non tcp packets are forwarded			
syntax		example	
non_tcp_forward_interface= <interface>		non_tcp_forward_interface=eth1	
Normally TelliNet TransparentCapturing handles captured TCP traffic only. In case that you need to forward non-TCP packets from the interface defined with the entry "interface" in the section [internal_tcp] to another interface, you can activate non-TCP forwarding with this entry. An interface has to be specified with this entry. TelliNet will then capture all non-TCP packets (unless they are excluded by the entry "exclude" in this section) and write them to the "non_tcp_forward_interface". Please note: you have to specify a "non_tcp_default_gw" as well.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
	related entries		
eth1	tc-recv - recv.ini - [transparent_capturing] - activate		
	tc-recv - recv.ini - [transparent capturing] - non_tcp_default_gw		

non_tcp_default_gw		section	file
		[transparent_capturing]	recv.ini
IP or DNS address of default gateway to forward all captured non-TCP traffic			
syntax		example	
non_tcp_default_gw = <ip address>		non_tcp_default_gw = 10.1.1.2	
non_tcp_default_gw = <dns name>		non_tcp_default_gw = outgoing.company.com	
In case that non TCP data is handled by Tellinet TransparentCapturing, i. e. a "non_tcp_forward_interface" is specified in this section, it is mandatory to specify the IP address or DNS name of the default gateway for non-TCP traffic with this entry. All captured non TCP traffic is forwarded to this gateway.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
	related entries		
	tc-recv - recv.ini - [transparent_capturing] - activate		
	tc-recv - recv.ini - [transparent_capturing] -		
	non_tcp forward interface		

http_prefetching_port_range		section	file
		[transparent_capturing]	recv.ini
port range used by the TelliNet Server for HTTP-Prefetching connections towards the internet			
syntax http_prefetching_port_range = <port> - <port>		example http_prefetching_port_range = 10000-25000	
<p>Specifies the port range that the TelliNet Server will use for HTTP-Prefetching connections towards the internet in case that the Client IP address is used as source address for HTTP-Prefetching.</p> <p>This entry is only valid if the utilisation of the Client address as source address for HTTP-Prefetching is specified at Server side in the Server Configuration File send.ini in the section [transparent_capturing] with the entries "activate = 1" and "prefetching_with_client_address = 1". Please note: it is important that the configured port range is not used by the local operating system of the TelliNet Client (e.g. configured as reserved port range). It is also important that the port range is large enough to allow many prefetching connections without reusing the port number very often (e. g. 10,000 to 30,000 ports should be specified).</p>			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default none	related entries tc-recv - recv.ini - [transparent_capturing] - activate tc-send - send.ini - [transparent_capturing] - activate tc-send - send.ini - [transparent_capturing] - prefetching with client address		

[etcp_connection_control_client]	file recv.ini
<p>The section [etcp_connection_control] allows to adjust parameters needed for connection control from the TelliNet Clients towards the TelliNet Server (e.g. for congestion control, flow control). This section is not mandatory. For all entries there are default values that allow a good regulation of the connections. Therefore it is recommended to leave this section out and use the default values.</p> <p>Any changes in this section are unnecessary in normal operation mode and should be made carefully!</p> <p>Please note: All congestion control and flow control parameters of this section are overruled by send.ini [etcp_connection_control_client] parameter values unless "allow_client_parameters" in the section [etcp_connection_control_client] of the send.ini file is set to 1 (on).</p>	

activate_congestion_control	section [etcp connection control client]	file recv.ini
Activates/deactivates congestion control (rate adaptation to experienced packet losses).		
syntax activate_congestion_control = <value of range>	example activate_congestion_control = 1	
<p>Congestion control is accomplished by checking the loss rate of data packets. When the loss rate is very low, the send rate is incremented. When the packet loss rate increases, no further increment of the send rate will take place.</p> <p>Further increase of the packet loss rate will lead to decrementing the send rate. The extend of the decrement is dependent on the degree of packet loss.</p> <p>This entry specifies whether transmission is done with or without congestion control, i. e. the reduction of bandwidth when packet loss occurs. Possible entries are:</p> <p>1 = congestion control is activated.</p> <p>0 = congestion control is not activated.</p> <p>Please note: An activation will only take effects for Clients sending data as UDP packets to the proxy server (when the Client is configured with etcp_send_protocol = 1)</p>		
unit boolean	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
range 0/1	related entries tc-recv - recv.ini - [etcp parameters] - etcp_send_protocol	
default 1 = on		

initial_data_rate		section	file
		[etcp_connection_control_client]	recv.ini
Defines the initial per Client send data rate (Client sending to Server proxy) for a new established Client to Server association.			
syntax initial_data_rate = <bits>		example initial_data_rate = 64000	
While exchanging data from the Client to the TelliNet Server the data rate will be adjusted automatically according to the overall congestion control parameters. This entry specifies the start value for the minimum send data rate of a new established Client to Server association.			
unit bit/s range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
default 64000 = 64kbit/s	related entries tc-recv - recv.ini [etcp_connection_control_client] - activate_congestion_control tc-send - send.ini - [etcp_connection_control_client] - initial_data_rate		

minimum_calculated_data_rate	section [etcp connection control client]	file recv.ini
Defines the per Client minimum send data rate (Client sending to Server proxy).		
syntax minimum_calculated_data_rate = <bits>	example minimum_calculated_data_rate = 64000	
Specifies the minimum per Client send data rate in bit/s.		
unit bit/s range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
default 64000 = 64kbit/s	related entries tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - minimum_calculated_data_rate	

low_range_bandwidth		section [etcp connection control client]	file recv.ini
Defines the bandwidth at and below of which the low_range parameters apply.			
syntax low_range bandwidth = <bit/s>		example low_range bandwidth = 200000	
For a higher bandwidth sliding calculated values interpolated between the "low_range" and "mid_range" parameters are use.			
unit factor range 0-100000000 default 80000 = 80000bits/s	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		

low_range_increment_percent		section	file
		[etcp_connection_control_client]	recv.ini
If the experienced recent packet loss rate is lower than the given percentage, the calculated data rate will be increased by "low_range_increment_additive".			
syntax		example	
low_range_increment_percent = <percentage>		low_range_increment_percent = 3	
Applies to the "low_range_bandwidth".			
unit	this parameter is	read by system	
percent	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
0-50%	<input type="checkbox"/> repeatable		
default	related entries		
3 = 3%	tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_increment_additive		
	tc-recv - recv.ini - [etcp_connection_control_client] - low_range_increment_percent		

low_range_increment_additive		section [etcp_connection_control_client]	file recv.ini
Congestion control additive increment value.			
syntax low_range_increment_additive = <bits>		example low_range_increment_additive = 10000	
Applies to the "low_range_bandwidth". Please see "low_range_increment_percent" as well.			
unit bit/s range 1-1000000 default 10000 = 10 kbit/s	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control tc-send - send.ini - [etcp_connection_control_client] - low_range_increment_percent tc-recv - recv.ini - [etcp_connection_control_client] - low_range_increment_additive		

low_range_decrement_percent		section	file
		[etcp_connection_control_client]	recv.ini
If the experienced recent packet loss rate is higher than the given percentage, the calculated data rate will be decreased by "low_range_decrement_factor".			
syntax		example	
low_range_decrement_percent = <percentage>		low_range_decrement_percent = 10	
Applies to the "low_range_bandwidth".			
unit	this parameter is	read by system	
percent	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
1-50%	<input type="checkbox"/> repeatable		
default	related entries		
10 = 10%	tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_factor		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_fast_percent		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_fast_factor		
	tc-recv - recv.ini - [etcp_connection_control_client] - low_range_decrement_percent		

low_range_decrement_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative decrement value.			
syntax low_range_decrement_factor = < value of range >		example low_range_decrement_factor = 0.96	
Applies to the "low range bandwtdh". Please see "low range decrement percent" as well.			
unit range 0.00 - 1.00 default 0.96 = 96%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_percent tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_fast_percent tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_fast_factor tc-recv - recv.ini - [etcp_connection_control_client] - low range decrement factor		

low_range_decrement_fast_percent		section	file
		[etcp_connection_control_client]	recv.ini
If extreme packet loss rate is experienced that is higher than the given percentage, the calculated data rate will be decreased by "low_range_decrement_fast_factor".			
syntax		example	
low_range_decrement_fast_percent = <percentage>		low_range_decrement_fast_percent = 30	
Applies to the "low_range_bandwidth".			
unit percent range 1-99% default 30 = 30%	this parameter is	read by system	
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		
	tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_fast_factor tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_percent tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_factor tc-recv - recv.ini - [etcp_connection_control_client] - low_range_decrement_fast_percent		

low_range_decrement_fast_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative decrement value for extremely high packet loss rates.			
syntax		example	
low_range_decrement_fast_factor = <value of range>		low_range_decrement_fast_factor = 0.80	
Applies to the "low_range_bandwidth". Please see "low_range_decrement_percent as well".			
unit factor range 0.00-1.00 default 0.80 = 80%	this parameter is	read by system	
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		
	tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_fast_percent		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_percent		
	tc-send - send.ini - [etcp_connection_control_client] - low_range_decrement_factor		
	tc-recv - recv.ini - [etcp_connection_control_client] - low_range_decrement_fast_factor		

low_range_increment_fast_threshold		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative increment parameter.			
syntax low_range_increment_fast_threshold = <value of range>		example low_range_increment_fast_threshold = 5	
After the given number (threshold) of consecutive additive send rate increases the system will switch into a fast increase mode. The calculated data rate will be increased by multiplying it with "low_range_increment_fast_factor".			
unit counter range 0-20 default 5	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - active_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - low_range_increment_fast_factor tc-send - send.ini - [etcp_connection_control_client] - low_range_increment_fast_threshold		

low_range_increment_fast_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative increment value.			
syntax low_range_increment_fast_factor = <value of range>		example low_range_increment_fast_factor = 1.3	
Applies to the "low_range bandwidth". Please see "low_range increment fast threshold" as well.			
unit factor range 1.01-10.00 default 1.3 = 130%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - active_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - low_range_increment_fast_threshold tc-send - send.ini - [etcp_connection_control_client] - low_range_increment_fast_factor		

mid_range_bandwidth		section	file
		[etcp connection control client]	recv.ini
Defines the bandwidth at which the mid_range parameters apply.			
syntax		example	
mid_range bandwidth = <bit/s>		mid_range bandwidth = 200000	
For a higher bandwidth sliding calculated values interpolated between the "mid_range" and "low_range" parameters are use. For a lower bandwidth sliding calculated values interpolated between the "mid_range" and "high_range" parameters are use.			
unit factor range 0-100000000 default 500000	this parameter is	read by system	
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		

mid_range_increment_percent		section	file
		[etcp_connection_control_client]	recv.ini
If the experienced recent packet loss rate is lower than the given percentage, the calculated data rate will be increased by "mid_range_increment_additive".			
syntax mid_range_increment_percent = <percentage>		example mid_range_increment_percent = 3	
Applies to the "mid_range bandwidth".			
unit percent range 0-50% default 3 = 3%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_increment_additive tc-send - send.ini - [etcp_connection_control_client] - mid_range_increment_percent		

mid_range_increment_additive		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control additive increment value.			
syntax mid_range_increment_additive = <bits>		example mid_range_increment_additive = 10000	
Applies to the "mid_range_bandwidth". Please see mid_range_increment_percent as well.			
unit bit/s range 1-1000000 default 50000 = 50kbit/s	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_increment_percent tc-send - send.ini - [etcp_connection_control_client] - mid_range_increment_additive		

mid_range_decrement_percent		section	file
		[etcp_connection_control_client]	recv.ini
If the experienced recent packet loss rate is higher than the given percentage, the calculated data rate will be decreased by "mid_range_decrement_factor".			
syntax		example	
mid_range_decrement_percent = <percentage>		mid_range_decrement_percent = 10	
Applies to "mid_range_bandwidth".			
unit	this parameter is	read by system	
percent	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
1-50%	<input type="checkbox"/> repeatable		
default	related entries		
10 = 10%	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_factor		
	tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_fast_percent		
	tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_fast_factor		
	tc-send - send.ini - [etcp_connection_control_client] - mid_range_decrement_percent		

mid_range_decrement_fast_percent		section	file
		[etcp_connection_control_client]	recv.ini
If an extreme packet loss rate is experienced that is higher than the given percentage, the calculated data rate will be decreased by "mid_range_decrement_fast_factor".			
syntax		example	
mid_range_decrement_fast_percent = <percentage>		mid_range_decrement_fast_percent = 30	
Applies to the "mid range bandwidth".			
unit percent range 1-99% default 30 = 30%	this parameter is		read by system
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	related entries		
	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_fast_factor		
	tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_percent		
	tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_factor		
	tc-send - send.ini - [etcp_connection_control_client] - mid_range_decrement_fast_percent		

mid_range_decrement_fast_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative decrement value for extremely high packet loss rates.			
syntax mid_range_decrement_fast_factor = <value of range>		example mid_range_decrement_fast_factor = 0.80	
Applies only to the "mid range bandwidth". Please see "mid range decrement percent" as well.			
unit range range 0.00-1.00 default 0.80 = 80%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_fast_percent tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_percent tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_decrement_factor tc-send - send.ini - [etcp_connection_control_client] - mid range decrement fast factor		

mid_range_increment_fast_threshold		section [etcp_connection_control_client]	file recv.ini
Congestion control multiplicative increment parameter.			
syntax mid_range_increment_fast_threshold = <value of range>		example mid_range_increment_fast_threshold = 5	
After the given number (threshold) of consecutive additive send rate increases the system will switch into a fast increase mode. The calculated data rate will be increased by multiplying it with "mid_range_increment_fast_factor".			
unit counter range 0-20 default 5	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - active_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_increment_fast_factor tc-send - send.ini - [etcp_connection_control_client] - mid_range_increment_fast_threshold		

mid_range_increment_fast_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative increment value.			
syntax mid_range_increment_fast_factor = <value of range>		example mid_range_increment_fast_factor = 1.3	
Applies to the "mid range bandwidth". Please see "mid range increment fast threshold" as well.			
unit factor range 1.01-10.00 default 1.3 = 130%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - active_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - mid_range_increment_fast_threshold tc-send - send.ini - [etcp_connection_control_client] - mid range increment fast factor		

high_range_bandwidth		section	file
		[etcp_connection_control_client]	recv.ini
Defines the bandwidth at and above of which the high_range parameters apply.			
syntax		example	
high_range_bandwidth = <bit/s>		high_range_bandwidth = 200000	
For a higher bandwidth sliding calculated values interpolated between the "high_range" and "mid_range" parameters are use.			
unit factor range 0-100000000 default 1500000	this parameter is	read by system	
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		

high_range_increment_percent		section	file
		[etcp_connection_control_client]	recv.ini
If the experienced recent packet loss rate is lower than the given percentage, the calculated data rate will be increased by "high_range_increment_additive".			
syntax		example	
high_range_increment_percent = <percentage>		high_range_increment_percent = 3	
Applies to the "high_range_bandwidth".			
unit percent range 0-50% default 3 = 3%	this parameter is	read by system	
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		
	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_increment_additive		
	tc-send - send.ini - [etcp_connection_control_client] - high_range_increment_percent		

high range increment additive		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control additive increment value.			
syntax		example	
high range increment additive = <bits>		high range increment additive = 100000	
Applies to the "high range bandwidth". Please see "high range increment percent" as well.			
unit bits/s range 1-1000000 default 100000 = 100kbit/s	this parameter is	read by system	
	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries		
	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - high_range_increment_percent tc-send - send.ini - [etcp_connection_control_client] - high range increment additive		

high_range_decrement_percent		section	file
		[etcp_connection_control_client]	recv.ini
If the experienced recent packet loss rate is higher than the given percentage, the calculated data rate will be decreased by "high_range_decrement_factor".			
syntax		example	
high_range_decrement_percent = <percentage>		high_range_decrement_percent = 10	
Applies to the "high range bandwidth".			
unit	this parameter is	read by system	
percent	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
1-50%	<input type="checkbox"/> repeatable		
default	related entries		
10 = 10%	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_factor		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_fast_percent		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_fast_factor		
	tc-send - send.ini - [etcp_connection_control_client] - high_range_decrement_percent		

high_range_decrement_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative decrement value.			
syntax high_range_decrement_factor = <value of range>		example high_range_decrement_factor = 0.92	
Applies to the "high range bandwidth". Please see "high range decrement percent" as well.			
unit factor range 0.00-1.00 default 0.92 = 92%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_percent tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_fast_percent tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_fast_factor tc-send - send.ini - [etcp_connection_control_client] - high_range_decrement_factor		

high_range_decrement_fast_percent		section	file
		[etcp_connection_control_client]	recv.ini
If an extreme packet loss rate is experienced that is higher than the given percentage, the calculated data rate will be decreased by "high_range_decrement_fast_factor".			
syntax		example	
high range decrement fast percent = <percentage>		high range decrement fast percent = 30	
Applies to the "high range bandwidth".			
unit	this parameter is	read by system	
percent	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
1-99%	<input type="checkbox"/> repeatable		
default	related entries		
30 = 30%	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_fast_factor		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_percent		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_factor		
	tc-send - send.ini - [etcp_connection_control_client] - high_range_decrement_fast_percent		

high_range_decrement_fast_factor		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative decrement value for extremely high packet loss rates.			
syntax		example	
high_range_decrement_fast_factor = <value of range>		high_range_decrement_fast_factor = 0.75	
Applies to the "high range bandwidth". Please see "high range decrement percent" as well.			
unit	this parameter is	read by system	
factor	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
0.00-1.00	<input type="checkbox"/> repeatable		
default	related entries		
0.75 = 75%	tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_fast_percent		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_percent		
	tc-recv - recv.ini - [etcp_connection_control_client] - high_range_decrement_factor		
	tc-send - send.ini - [etcp_connection_control_client] - high_range_decrement_fast_factor		

high_range_increment_fast_threshold		section	file
		[etcp_connection_control_client]	recv.ini
Congestion control multiplicative increment parameter.			
syntax high_range_increment_fast_threshold = <value of range>		example high_range_increment_fast_threshold = 5	
After the given number (threshold) of consecutive additive send rate increases the system will switch into a fast increase mode. The calculated data rate will be increased by multiplying it with "high range increment fast factor".			
unit counter range 0-20 default 5	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - active_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - high_range_increment_fast_factor tc-send - send.ini - [etcp_connection_control_client] - high range increment fast threshold		

high_range_increment_fast_factor		section [etcp_connection_control_client]	file recv.ini
Congestion control multiplicative increment value.			
syntax high_range_increment_fast_factor = <value of range>		example high_range_increment_fast_factor = 1.3	
Applies to the "high_range_bandwidth". Please see "high_range_increment_fast_threshold" as well.			
unit factor range 1.01-10.00 default 1.3 = 130%	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-recv - recv.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - high_range_increment_fast_threshold tc-send - send.ini - [etcp_connection_control_client] - high_range_increment_fast_factor		

max_window_size		section [etcp_connection_control_client]	file recv.ini
Maximum allowed Client window size for ETCP transmissions to the proxy server.			
syntax max_window_size = <bytes>		example max_window_size = 7500000	
unit bytes range default 4000000 Byte = 4 MByte	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	related entries tc-send - send.ini - [etcp_connection_control_client] - activate_congestion_control tc-recv - recv.ini - [etcp_connection_control_client] - max_window_size

nak_timeout		section	file
		[etcp connection control client]	recv.ini
The client proxy will report missing packets (NAKs) after waiting for nak_timeout.			
syntax		example	
nak_timeout = <milliseconds>		nak_timeout = 100	
Retardation of the sending of negative acknowledgements (NAK) in milliseconds (default: 100 milliseconds). It is possible that packets are delivered in the wrong order. In that case the late coming packet will be regarded as missing and a NAK will be sent out to cause resending before the original packet is delivered. As a result the Client Proxy will receive the packet two times. To avoid this redundant sending, the Client Proxy is configured to wait for a couple of milliseconds to make sure the packet is actually not delivered before sending out the NAK.			
unit	this parameter is	read by system	
milliseconds	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
50 or more	<input type="checkbox"/> repeatable		
default	related entries		
100 ms	tc-recv - recv.ini - [etcp connection control] - nak_timeout tcp		

connect_initial_timeout		section	file
		[etcp connection control client]	recv.ini
Initial timeout when connecting to a server proxy.			
syntax		example	
connect initial timeout = <milliseconds>		connect initial timeout = 2000	
Specifies the period of time a Client Proxy is trying to connect to a Server Proxy in milliseconds (default: 10000 milliseconds). If the connection can not be established in the period of time specified with this entry, the Client Proxy is trying to connect to another Server Proxy.			
unit	this parameter is	read by system	
milliseconds	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
range	<input type="checkbox"/> mandatory	<input type="checkbox"/> continuously	
	<input type="checkbox"/> repeatable		
default	related entries		
10000 ms = 10 seconds			

nr_of_tcp_connections		section	file
		[etcp connection control client]	recv.ini
Allows to use multiple TCP connections to the server proxy.			
syntax nr_of_tcp_connections = <number of connections>		example nr_of_tcp_connections = 1	
Specifies the number of TCP connections used for one ETCP association from the Client to the Server.			
unit boolean	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range 0/1			
default 1 = on	related entries		

4.2.2.3 TelliCast only sections

[announcement_channel]	file recv.ini
The announcement channel is constantly received by the Client to get the invitations for reception and the data keys in case of encrypted data transmissions. The name and parameters of the announcement channel have to be the same as used in the Server.	

address	section [announcement_channel]	file recv.ini
IP multicast address:port for the announcement channel		
syntax address = < address > address = < address > : < port > address = : < port >	example address = 229.1.1.1:2511	
Specifies the multicast address and port to be used for the announcements.		
unit range default 229.1.1.1:2511	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	related entries tc-send - send.ini - [announcement_channel] - address	

name	section [announcement_channel]	file recv.ini
name of the announcement channel (additional distinction beside the ip multicast address)		
syntax name = <string>	example name = TSL Announcement Channel	
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
range	related entries tc-send - send.ini - [announcement_channel] - name	
default TSL Announcement Channel		

[parameters]	file recv.ini
The section [parameters] contains general configuration parameters for TELICAST functionality.	

interface_address		section [parameters]	file recv.ini
local interface to receive multicast data.			
syntax interface_address = <address>		example interface_address = 172.27.1.51	
Address of the interface that is used to receive multicast data packets with TelliCast. Please note that this parameter has to be specified when using some DVB card types.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default	related entries		

tmp_directory	section [parameters]	file recv.ini
If this parameter is given tc-recv will internally store/compose all received files within the given directory and later move them to the target directory.		
syntax tmp_directory = <directory>	example tmp_directory = receiving/tmp/files	
<p>The delivery of big files from the Client cache database directly to the target directory needs time and therefore may cause problems because the user may open the files while the storing process is still not complete. To avoid that, the files can be stored in a temporary directory. After the files are completely available, they are moved to the target directory.</p> <p>The value given here is the default value for all received channels. You may overwrite this default setting using the per channel "tmp_directory" entry within recv-channels.ini.</p> <p>Please note: alternatively, the parameters "tmp_prefix" or "tmp_suffix" can be used to avoid the utilization of still incomplete files. Please do not set more than one of the parameters "tmp_directory", "tmp_prefix" and "tmp_suffix".</p>		
unit range default undefined = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	related entries tc-recv - recv-channels.ini - [<channel name>] - tmp_directory tc-recv - recv.ini - [parameters] - tmp_prefix tc-recv - recv.ini - [parameters] - tmp_suffix	

tmp_prefix		section [parameters]	file recv.ini
Like tmp_directory, but files are composed within the final target directories but stored with the given prefix in front of the normal filenames until the files are fully written.			
syntax tmp_prefix = <file name start>		example tmp_prefix = .	
The delivery of big files from the Client cache database directly to the target directory needs time and therefore may cause problems because the user may open the files while the storing process is still not complete. To avoid that a prefix can be specified for the files. When a file is moved to the target directory, it is stored under the file name plus the prefix. The prefix is deleted by the Client as soon as the storing of the file is completed. This mechanism allows the user to see from the name of the files in the target directory whether the files are complete. The value given here is the default value for all received channels. You may overwrite this default setting using the per channel "tmp_prefix" entry within recv-channels.ini. Please note: alternatively, the parameters "tmp_directory" or "tmp_suffix" can be used to avoid the utilization of still incomplete files. Please do not set more than one of the parameters "tmp_directory", "tmp_prefix" and "tmp_suffix".			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default undefined = off	related entries tc-recv - recv-channels.ini - [<channel name>] - tmp_prefix tc-recv - recv.ini - [parameters] - tmp_directory tc-recv - recv.ini - [parameters] - tmp_suffix		

tmp_suffix		section [parameters]	file recv.ini
Like tmp_prefix, but adding a suffix to the normal filenames until the files are fully written.			
syntax tmp_suffix = <file suffix>		example tmp_suffix = .tmp	
<p>The delivery of big files from the Client cache database directly to the target directory needs time and therefore may cause problems because the user may open the files while the storing process is still not complete. To avoid that a suffix can be specified for the files. When a file is moved to the target directory, it is stored under the file name plus the suffix. The suffix is deleted by the Client as soon as the storing of the file is completed. This mechanism allows the user to see from the name of the files in the target directory whether the files are complete.</p> <p>The value given here is the default value for all received channels. You may overwrite this default setting using the per channel "tmp_suffix" entry within recv-channels.ini.</p> <p>Please note: alternatively, the parameters "tmp_directory" or "tmp_prefix" can be used to avoid the utilization of still incomplete files. Please do not set more than one of the parameters "tmp directory", "tmp prefix" and "tmp suffix".</p>			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default undefined = off	related entries tc-recv - recv-channels.ini - [<channel name>] - tmp_suffix tc-recv - recv.ini - [parameters] - tmp_prefix tc-recv - recv.ini - [parameters] - tmp_directory		

total bandwidth		section [parameters]	file recv.ini
maximum bandwidth this receiver is expected to be able to receive simultaneously.			
syntax total bandwidth = <bits>		example total bandwidth = 10000000	
This entry specifies the maximum allowed bandwidth for multicast reception with TelliCast. The client will only subscribe to channels until "total_bandwidth" is reached. Please note that you can assign a priority to each channel in the Client Channel File "recv-channels.ini". In case that "total_bandwidth" is reached, channels with lower priority will be blocked in order to join transmissions on channels with higher priority.			
unit bits/s range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
default 100000000 = 100 Mbit/s	related entries tc-recv - recv-channels.ini - [<channel name>] - priority		

acknowledgement_ proxy_address		section [parameters]	file recv.ini
to send acknowledgement indirectly via an HTTP proxy (ip:port)			
syntax acknowledgement_proxy_address = <address>:<port>		example acknowledge- ment_proxy_address = 172.27.1.2:80	
Address and port number of the caching web porxy host that will be used to send acknowledgements in case that the acknowledgements should pass a proxy. This entry is necessary if proxy utilisation is required for internet access.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
range			
default	related entries tc-send - *.job/*.crw - [scheduling] - re- quest acknowledgements		

packet_naks		section [parameters]	file recv.ini
Activates packet based multicast retransmission at transport layer (MTP/SO).			
syntax packet_naks = <value of range>		example packet_naks = 1	
Specifies whether NAKs are sent to the TelliCast Server at the transport layer (MTP/SO) in case that single packets of a transmission are missing. Possible entries are: 0 = no NAKs are sent. 1 = NAKs are sent via UDP unicast. 2 = NAKs are sent via UDP multicast. Please note that at the Server the entry "allow_packet_naks" has to be activated for the corresponding channel as well (Channel File *.cha; section [channel]).			
unit	this parameter is	read by system	
range 0/1/2 default 0 = off	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries tc-send - *.cha - [channel] - allow_packet_naks tc-send - *.rcv - [recipient] - receive_only		

allow_execute		section [parameters]	file recv.ini
Allows sender to automatically execute received files on this client.			
syntax allow_execute = <value of range>		example allow_execute = 1	
<p>This entry determines, whether the automatic execution of files on the Client host after reception is accepted. Possible entries are:</p> <p>1 = execution is accepted, 0 = execution is refused, i. e. executable files are only stored in the target directory without being executed</p> <p>When Clients are configured to refuse the execution, they will receive the executable files, but the files are not executed automatically.</p> <p>This value can be overwritten for individual channels in the recv-channels.ini file.</p>			
unit	this parameter is	read by system	
range 0/1	<input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	<input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	default 0 = not allowed	related entries tc-recv - recv-channels.ini - [<channel name>] - allow_execute tc-send - *.job - [execute]	

file database size		section [parameters]	file recv.ini
maximum size of temporary fragment storage within the client file system.			
syntax file database size = <bytes>		example file database size = 262144000	
Specifies the maximum size of the TelliCast directory for caching data fragments. The location of the directory is specified in the section [locations] with the entry "file database directory".			
unit bytes range default 262144000 = 250 MByte	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	related entries tc-recv - recv.ini - [locations] - file database directory		

[webbroadcast]	file recv.ini
The [webbroadcast] section specifies the configuration for the WebBroadcast functionality of TelliCast.	

min_file_lifetime	section [webbroadcast]	file recv.ini
Minimum life time in seconds for web content in the local proxy directory.		
syntax min file lifetime = <seconds>	example min file lifetime = 86400	
Specifies the minimum life time for web content in the webbroadcast cache directory of the Client. Normally files in that directory are automatically erased when either the expire date sent with the file by the web server or the "max_file_lifetime" configured in the TelliCast Server is reached. In case the recipient wants to make sure that the files are kept in the cache directory for a certain time, even if the expire date or the "max_file_lifetime" are reached, (e. g. because in the meantime a connection to the internet is not possible and the recipient is dependent on cache content to see the web page), the "min file lifetime" can be set.		
unit seconds range	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
default 0 = not set	related entries tc-send - *.crw - [crawl parameters] - max file lifetime	

[newsbroadcast]	file recv.ini
The [newsbroadcast] section specifies the configuration necessary for NewsBroadcast.	

server_address		section [newsbroadcast]	file recv.ini
destination news server			
syntax server_address = <address>		example server_address = 172.27.1.51	
Address or host name and port of the external news server receiving the articles transmitted by TelliCast. The entry has to be repeated for every news server that should receive articles from the TelliCast Client.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input checked="" type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range	related entries		
default default			

connection_timeout		section [newsbroadcast]	file recv.ini
close connection to news server when no articles are received via multicast			
syntax connection_timeout = <seconds>		example connection_timeout = 300	
Specifies the time that the Client is keeping a connection to the news server open after the last activity. When no further article to be transmitted is coming in during this period, the connection is closed until the next transmission.			
unit seconds	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default 300 s	related entries		

[offlinemail]	file recv.ini
The section [offlinemail] specifies the parameters for the TelliCast module OfflineMail.	

show_icon		section [offlinemail]	file recv.ini
Indicate new email (notification) using a "flag like" additional tray icon (Windows operating system only).			
syntax show_icon = <value of range>		example show_icon = 1	
This entry specifies, whether a "flag like" additional tray icon is displayed under Windows operating systems to notify about incoming e-mail. Possible entries are: 1 = The tray icon for e-mail notification is shown. 0 = No tray icon is shown.			
unit range 0/1 default default1 = on	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	related entries tc-send - offlinemail.olm		

4.3 LICENSE FILE

This file contains the product license. It has to be put in the working directory of the Client program and has to be named as specified in the *Client Configuration File*, section [locations], with the entry license_file (default: license.ini). A product license is necessary for the program to run.

4.3.1 CONTENTS

[license]	→ page 105
license_class	→ page 105
expiry_date	→ page 105
oem_license	→ page 106
restriction	→ page 106
product_license	→ page 106
operating_system	→ page 107
description	→ page 107
http_skin	→ page 108
min_version	→ page 108
max_version	→ page 109
license_id	→ page 109
[activations]	→ page 110

4.3.2 FILE ENTRIES

[license]	file license.ini
The license section is necessary for the program to run. It contains the license key and information about the validity of the license. It has to be entered in exactly the same way (including the order of the lines) as it is given by the distributor.	

license_class		section [license]	file license.ini
The license class of the product.			
syntax license_class = <license classification>		example license_class = evaluation copy	
This entry gives a classification for the license, e. g. whether it is a commercial copy or an evaluation license.			
unit range default	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	further description		
	related entries		

expiry_date	section [license]	file license.ini
The date at which the license is expired.		
syntax expiry_date = <yyyy.mm.dd>	example expiry_date = 2004.10.01	
An expiry date is only given when the license has a time limit (e. g. for evaluation licenses). The format of the time is "year.month.day".		
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
range		
default	further description	
	related entries	

oem_license		section [license]	file license.ini
Informs, whether the license is an oem license (product can be used as integrated part of other products).			
syntax oem_license = <company name> oem_license = none		example oem_license = MyCompany oem_license = none	
The name of the company is specified here when the license is an oem license. Otherwise the entry is "none".			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default	further description		
	related entries		

restriction	section [license]	file license.ini	
Any restrictions for the license.			
syntax restriction = <text>	example restriction = valid in conjunction with service or evaluation agreement only!		
This entry specifies the restrictions for the use of the software with this license (e. g. that it can only be used for evaluation purposes).			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default			further description
			related entries

product_license		section	file
		[license]	license.ini
Specifies a module that can be used with the given license.			
syntax		example	
product_license = Client/<module>		product_license = Client/TelliNet	
A module is not necessarily the whole software but each part/functionality of the software that can be licensed separately is called module. When the license is valid for more than one module, the entry is repeated for each module.			
unit	this parameter is	read by system	
range	<input checked="" type="checkbox"/> optional	<input checked="" type="checkbox"/> at startup only	
	<input type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input checked="" type="checkbox"/> repeatable		
	further description		
	related entries		

operating_system		section [license]	file license.ini
Specifies the operating system and processor necessary to run the version valid with the given license key.			
syntax operating_system = Client/ < OS >		example operating_system = Client/Windows	
unit range default	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	further description		
	related entries		

description		section [license]	file license.ini
A text giving any description or comments.			
syntax description = < text >		example description = License purchased for test in customer environments.	
This section is optional. It will not be given by the distributor but can be entered by the operator of the Client according to his demands. Any text can be entered without effect on the validity of the license.			
unit range default	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
	further description		
	related entries		

http_skin		section [license]	file license.ini
Corporate design of the web interface.			
syntax http_skin = <company name>		example http_skin = CustomerCompany	
The web interface design can be adapted to the corporate design of the customer (by including customer logo, colors etc.). These changes in the web interface skin can only be made by the software producer on request of the customer. This entry specifies the company design that will be displayed for the web interface of the licensed product.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously	
range			
default		further description	
		related entries	

min_version		section [license]	file license.ini
Oldest version of the software that can be used with this license file.			
syntax min_version = <version number>		example min_version = 2.3.0	
The license is valid for a range of versions of the software. This value gives the lower boundary of the range of versions that can be used with this license.			
unit	this parameter is <input type="checkbox"/> optional <input checked="" type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default		further description	
	related entries tc-recv - license.ini - [license] - max_version		

max_version		section [license]	file license.ini
Newest version of the software that can be used with this license file.			
syntax max_version = <version number>		example max_version = 2.4.1	
The license is valid for a range of versions of the software. This value gives the upper boundary of the range of versions that can be used with this license.			
unit	this parameter is <input type="checkbox"/> optional <input checked="" type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range			
default	further description		
	related entries tc-recv - license.ini - [license] - min_version		

license_id		section [license]	file license.ini
The license key as given by the distributor.			
syntax		example	
unit	this parameter is <input type="checkbox"/> optional	read by system <input checked="" type="checkbox"/> at startup only	
range	<input checked="" type="checkbox"/> mandatory	<input type="checkbox"/> continously	
default	<input type="checkbox"/> repeatable		
	further description		
	related entries		

[activations]

file
license.ini

The section [activations] can be included in the License File if the functionality of the Client should be limited to less modules than included in the [license] section. The advantage of the deactivation of non used modules is a reduction in the utilisation of processor resources and a better overview, because the web interface only displays the information for the activated modules.

When no section [activations] is existent, all modules are activated.

A module is deactivated with the entry "Client/<module> =off" and activated with the entry "Client/<module> =on". "<module>" is the name of the module as given in the section [license] with the entry "product_license".

4.4 CLIENT CHANNEL FILE

The Client *Channel File* is used to configure the received multicast channels. If there is no such file the Client will receive all channels it is invited to and store them in the directory "received". The file must have the name "recv-channels.ini" and has to be stored under the working directory of the Client.

4.4.1 CONTENTS

[*]	→ page 112
priority	→ page 112
target_directory	→ page 112
activate_ftp_forwarding	→ page 113
allow_execute	→ page 113
tmp_directory	→ page 114
tmp_prefix	→ page 115
tmp_suffix	→ page 116
stream_protocol	→ page 116
stream_address	→ page 117
stream_interface	→ page 117
allowed_address	→ page 118
stream_ttl	→ page 118
stream_buffer	→ page 119

4.4.2 FILE ENTRIES

[*]	file recv-channels.ini
<p>The section [<channel name>] describes one channel to be received. If more than one channel should be received, the section has to be repeated for each channel. If no such section is found then all channels to which the Client is invited will be received. An asterisk (*) can be used as wildcard at the end of the <channel name>. E. g. the channels 'de.test.company.data', 'de.test.company.content' and 'de.test.company.more' can be received with the entry "[de.test.company.*]". All channels are received with the entry "[*]".</p>	

priority	section [*]	file recv-channels.ini
Priority of this channel/channel group reception in case the reception would cause reception bandwidth sum to exceed "recv.ini [parameters] total_bandwidth". 0 = highest; 1000 = lowest		
syntax priority = <value of range>	example priority = 500	
Specifies the priority of the channel. When the bandwidth of all incoming transmissions exceeds the maximum bandwidth configured in the recv.ini file (i. e. it is not possible to receive all incoming data), transmissions on the channels with the highest priorities are received. "0" is the maximum priority, "1000" is the minimum priority.		
unit range 0-1000 default 500	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	further description	
	related entries	

target_directory		section [*]	file recv-channels.ini
Directory for received files. The target_directory name is relative to the current working directory.			
syntax target_directory = <directory>		example target_directory = received	
All incoming files transmitted on this channel are stored under the directory specified with this entry.			
unit range default received	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	further description		
	related entries		

activate ftp forwarding		section [*]	file recv-channels.ini
Will activate FTP forwarding for the correctly received files according to the recv.ini [ftp_forwarding] parameters.			
syntax activate ftp forwarding = <value of range>		example activate ftp forwarding = 1	
This entry specifies, whether the correctly received files are stored on the local host or forwarded to an FTP server. Possible entries are: 0 = The files are stored on the local host in the directory specified with the entry "target_directory" in this section. 1 = The files are forwarded to the FTP server specified in the section [ftp_forwarding] of the recv.ini file. The directory name for the directory to store the correctly received files specified with the entry "target_directory" in this section is also used on the FTP server.			
unit range 0/1 default 0 = off	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	further description		
	related entries tc-recv - recv.ini - [ftp_forwarding] - ftp_server_address tc-recv - recv-channels.ini - [<channel name>] - target_directory		

allow_execute		section [*]	file recv-channels.ini
Enables/disables execution of programs/scripts after job reception if specified within the server job file.			
syntax allow_execute = <value of range>		example allow_execute = 1	
This entry determines, whether the automatic execution of files on the Client host after reception is accepted. Possible entries are: BR>"1" execution is accepted, and with the entry of "0" execution is refused, i. e. executable files are only stored in the incoming directory without being executed.			
unit range 0/1 default 0 = not allowed, but default value may be redefined in recv.ini	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	further description		
	related entries		

tmp_directory		section [*]	file recv-channels.ini
If this parameter is given tc-recv will internally store/compose all received files within the given directory and later move them to the target directory.			
syntax tmp_directory = <directory>		example tmp_directory = receiving/tmp/files	
The delivery of big files from the Client cache database directly to the target directory needs time and therefore may cause problems because the user may open the files while the storing process is still incomplete. To avoid that, the files can be stored in a temporary directory during the storing process. After the files are completely available, they are moved to the target directory. This value overwrites the default value for all received channels given in the Client Configuration File recv.ini in the section [parameters]. Please note: alternatively, the parameters "tmp_prefix" or "tmp_suffix" can be used to avoid the utilization of still incomplete files. Please do not set more than one of the parameters "tmp_directory", "tmp_prefix" and "tmp suffix".			
unit range default undefined = off, but default value may be redefined in recv.ini	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	further description		
	related entries tc-recv - recv.ini - [parameters] - tmp_directory tc-recv - recv-channels.ini - [<channel name>] - tmp_prefix tc-recv - recv-channels.ini - [<channel name>] - tmp_suffix		

tmp_prefix		section [*]	file recv-channels.ini
Like tmp directory, but files are composed within the final target directories but stored with the given prefix in front of the normal filenames until the files are fully written.			
syntax tmp_prefix = < file name start >		example tmp_prefix = .	
<p>The delivery of big files from the Client cache database directly to the target directory needs time and therefore may cause problems because the user may open the files while the storing process is still not complete. To avoid that a prefix can be specified for the files. When a file is moved to the target directory, it is stored under the file name plus the prefix. The prefix is deleted by the Client as soon as the storing of the file is completed. This mechanism allows the user to see from the name of the files in the target directory whether the files are complete.</p> <p>This value overwrites the default value for all received channels that is set with the entry "tmp_prefix" in the client Configuration File recv.ini in the section [parameters].</p> <p>Please note: alternatively, the parameters "tmp_directory" or "tmp_suffix" can be used to avoid the utilization of still incomplete files. Please do not set more than one of the parameters "tmp_directory", "tmp_prefix" and "tmp_suffix".</p>			
unit range default undefined = off, but default value may be redefined in recv.ini	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	further description		
	related entries tc-recv - recv.ini - [parameters] - tmp_prefix tc-recv - recv-channels.ini - [<channel name>] - tmp_directory tc-recv - recv-channels.ini - [<channel name>] - tmp_suffix		

tmp_suffix		section [*]	file recv-channels.ini
Like tmp_prefix, but adding a suffix to the normal filenames until the files are fully written.			
syntax tmp_suffix = < file suffix >		example tmp_suffix = .tmp	
<p>The delivery of big files from the Client cache database directly to the target directory needs time and therefore may cause problems because the user may open the files while the storing process is still not complete. To avoid that a suffix can be specified for the files. When a file is moved to the target directory, it is stored under the file name plus the suffix. The suffix is deleted by the Client as soon as the storing of the file is completed. This mechanism allows the user to see from the name of the files in the target directory whether the files are complete.</p> <p>This value overwrites the default value for all received channels set in the Client Configuration File recv.ini in the section [parameters].</p> <p>Please note: alternatively, the parameters "tmp_directory" or "tmp_prefix" can be used to avoid the utilization of still incomplete files. Please do not set more than one of the parameters "tmp_directory", "tmp_prefix" and "tmp_suffix".</p>			
unit range default undefined = off, but default value may be redefined in recv.ini	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously
	further description		
	related entries tc-recv - recv.ini - [parameters] - tmp_suffix tc-recv - recv-channels.ini - [<channel name>] - tmp_directory tc-recv - recv-channels.ini - [<channel name>] - tmp_prefix		

stream_protocol		section [*]	file recv-channels.ini
for DataStreams channels only: protocol used to further distribute the data stream by the receiver			
syntax stream_protocol = < value of range >		example stream_protocol = tcp	
unit range udp, tcp, file default destination_protocol specified in the Job File at the Server	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable		read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continuously
	further description		
	related entries		

stream_address		section [*]	file recv-channels.ini
For DataStreams channels only: target IP address and port or filename to further distribute the data stream by the receiver. When using UDP a multicast ip address may be given.			
syntax stream_address = <address>:<port> stream_address = <port>		example stream_address = 172.27.1.51:9999	
Specifies the final destination of the data stream. The Server is transmitting the incoming data stream to the Clients. The Clients further distribute the streams to the recipients. The address is dependent on the mode of transmission used: If the "destination_protocol" used is TCP, it is not allowed to specify an address. Only a port has to be specified. Client applications can connect to the port to receive the data stream. If the "stream_protocol" is UDP and multicast distribution is used, a multicast address and a port have to be specified (default: source_address of the Server Job File). If the "stream_protocol" is UDP and unicast transmission is used, the address or name of the recipient host and the port used have to be specified (default: localhost). If the "stream_protocol" is file, a file name has to be specified.			
unit	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
range	further description related entries tc-send - *.str - [stream] - destination address		
default			

stream_interface		section [*]	file recv-channels.ini
For DataStreams channels only: source interface to further distribute the data stream by the receiver.			
syntax stream interface = <address>		example stream interface = 172.27.1.51	
unit range default automatically chosen interface	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	further description		
	related entries tc-recv - recv-channels.ini - stream_address tc-recv - recv-channels.ini - stream_protocol		

allowed_address		section [*]	file recv-channels.ini
for DataStreams channels only: Allow connections to the TCP stream port only from the hosts listed (e.g. localhost).			
syntax allowed_address = <address> allowed_address = <DNS-Name>		example allowed_address = 172.27.1.51	
This parameter does only apply to streams provided via TCP.			
unit range default access is permitted for all hosts	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input checked="" type="checkbox"/> at startup only <input type="checkbox"/> continously	
	further description		
	related entries tc-recv - recv-channels.ini - [<channel_name>] - stream_protocol tc-recv - recv-channel.ini - [<channel name>] - steam address		

stream_ttl		section [*]	file recv-channels.ini
For DataStreams channels only: only relevant when locally sending via multicast TTL used in stream packets generated to further distribute the data stream by the receiver.			
syntax destination_ttl = <value of range		example destination_ttl = 1	
Range of multicast distribution. This entry specifies the maximum number of routers that a multi-cast packet can pass between the Client and the recipient.			
unit range 0-255 default destination_ttl in the Job File at the Server	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input type="checkbox"/> at startup only <input checked="" type="checkbox"/> continuously	
	further description		
	related entries		

stream_buffer		section [*]	file recv-channels.ini
For DataStreams channels only: defines how much data will be stored for this channel within the tc-recv in case that it can not immediately be distributed by the TelliCast client.			
syntax stream_buffer = <bytes>		example stream_buffer = 5000	
Defines how much data will be stored for this channel within the Server in case that the data can not immediately be distributed to the TelliCast Clients. Please configure a small buffer space when using low bit-rate channels that should rather discard data instead of waiting to deliver all data completely.			
unit bytes range 1600-100000 default 100000 = 100 kByte	this parameter is <input checked="" type="checkbox"/> optional <input type="checkbox"/> mandatory <input type="checkbox"/> repeatable	read by system <input type="checkbox"/> at startup only <input checked="" type="checkbox"/> continously	
	further description		
	related entries		

4.5 LOG FILE

The *Log File* is used for the automatic documentation of the status of the transmission process. Therefore data in the *Log File* is generated automatically during transmission and should not be changed by the user.

The *Log File* is called "recv.log" and is located in the working directory of the Client, if not otherwise configured in the section [locations] of the *Client Configuration File*.

The *Log File* can be used to check what error or warning has occurred when the "Status" is not "OK". To control the behaviour of the system, open the *Log File* with an editor and read the information given about the system behaviour. The *Log File* is also shown on the web interface of TELLINET / TELLCAS.

In case a *Log File* is specified in the *Configuration File* and the log messages cannot be written into that file, e.g. because the directory is not existent or the permission to write into that directory is denied, the TELLINET / TELLCAS Client is writing the log messages into the default *Log File* "recv.log". When this also fails, the TELLINET / TELLCAS Client tries to use the following files in descending order to write the log messages in:

- Under Unix systems:
 - <system tmp directory>/recv.log
 - /tmp/recv.log
 - <stderr>
- Under Windows systems:
 - \$TMP\recv.log
 - \$TEMP\recv.log
 - c:\temp\recv.log
 - c:\tmp\recv.log
 - c:\recv.log

(\$TMP means the value of the environment variable TMP. The same applies for other directory names including "\$").

When any other than the specified *Log File* is used, a log message is written that informs about the new location of the *Log File*. The *Log File* is automatically displayed at the web interface of the TELLINET / TELLCAS Client. Therefore the information about the location of the *Log File* can be taken from the web interface. Log messages written into <stderr> cannot be displayed at the web interface. However a message is displayed that informs about the utilisation of <stderr>.

Each line has the format:

<Level>:< Date Time (UTC)>:<Message>

The level can be one of the following:

- ERR: An error occurred.
- WRN: A problem not as severe as an error occurred (failure of non-essential functions).
- INF: Information about the program status.
- MSG, VRB: Additional details about the behaviour of the program. In the VRB (verbose) mode messages are more detailed than in the MSG (message) mode.

It can be configured in the *Client Configuration File* which severity levels are written to the *Log File*. ERR, WRN and INF messages are displayed in all output levels except "none", MSG is included in the "normal" output and all messages are displayed in the output level "verbose".

Date/Time: UTC time and date when the error or warning message was written into the *Log File* in the format <year-month-day hours:minutes:seconds: milliseconds>.

Message: HTML format text describing the problem.

In the web interface only the latest information about the Client activity is shown. In case the *Log File* contains very much information, the oldest status informations are suppressed. On top of the status lines information on the number of suppressed lines is given. These lines can only be read directly from the *Log File*.

5 WEB INTERFACE

5.1 STRUCTURE OF THE WEB INTERFACE

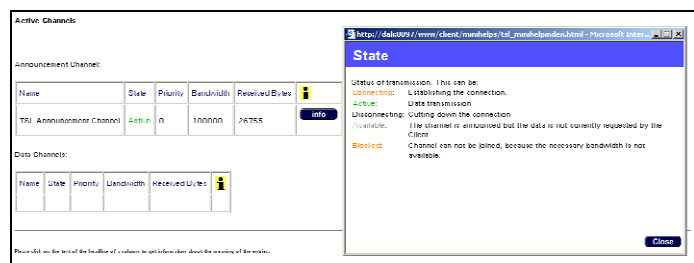
The web interface is designed to allow simple monitoring of the Client.

The web interface consists of various pages that give information about the activity of the TELLINET / TELLCAST Client. Submenus are available for information on TELLCAST, information on TELLINET and the online help. A choice between the pages can be made using the menu on the left side of the main area.

In the head of the page the following information is given independent from the page chosen for the main area:

Status:	<p>The status of the Client which can be</p> <ul style="list-style-type: none"> • OK Everything is okay. • Starting The Client is starting. • Not Connected: The Client is not connected to a Server. • Finishing The Client is shutting down. • WARNING A warning message was written to the <i>Log File</i>. • ERROR An error message was written to the <i>Log File</i>. • Config. Error An error message concerning a false configuration was written to the <i>Log File</i>. <p>Error or warning messages are usually the result of one or more configuration problems. The error status will not be reset automatically even if the error condition is resolved. In case of an error or warning condition the steps described in chapter 3.5 should be followed.</p>
Host:	The host on which the TELLINET / TELLCAST Client is run.
Date:	Current UTC time and date at the Client in the format „year-month-day hour:minutes:seconds“.

In addition to the online help on the pages of the Submenu “Help”, help for the meaning of individual entries is available. Clicking on the entries written in colour on the main pages opens a new, small window that provides help for the meaning of the specific entry.



5.2 PAGE: OVERVIEW

This page gives an overview of the Client state, including the TELLICAST and TELLINET activity. It will be refreshed periodically.

Data of TELLINET

Speed State:	Specifies whether the speed mode is turned on (ON) or off (OFF).
Client Connections:	Number of current connections to clients (e.g. web browsers, mail clients).
Server Connections:	Number of current connections to Server Proxies.
Throughput:	Currently used bandwidth in kbit/s.

The data for TELLINET is given separately for HTTP/FTP, SOCKS and port forwarding connections.

Data of TELLICAST

Active Data Channels:	The number of data channels which are currently received.
Connecting Channels:	The number of channels to which the Client is currently connecting.
Disconnecting Channels:	The number of channels from which the Client is currently disconnecting.
Blocked Channels:	The number of channels which are currently blocked, i.e. the Client is invited for this channel but reception of the channel is not possible because the necessary bandwidth is not available at the moment. Possible reasons are that the maximum bandwidth configured in the Client for this channel is exceeded or that too many channels are received at the same time and the maximum bandwidth for the reception of channels is exceeded.
Throughput:	Currently used bandwidth in kbit/s.

Graphic Overview

Under the line giving this information a graphic overview of the transmission throughput of TELLICAST and TELLINET is displayed. The data transfer rate is shown in bits/s. The scale of the y axis is changed dynamically to make sure that all peaks can be displayed completely.

5.3 SUBMENU FOR TELLINET

5.3.1 PAGE: TELLINET STATISTICS

This page gives an overview of the proxy activity. It will be refreshed periodically. When reasonable, the following statements are specified separately for HTTP/FTP connections and port forwarding connections.

Active Since:	The UTC time when the the TELLINET Client was started in the format "<year-month-day hour:minutes:seconds>".
Speed State:	Specifies whether the speed mode is turned on (ON) or off (OFF).
Sent Data Packets:	Number of data packets, excluding control data packets, sent to TELLINET Servers. This includes retransmitted data packets.
Sent Retransmitted Data Packets:	Number of data packets, excluding control data packets, that are retransmitted to TELLINET Servers to adjust packet losses.
Received Data Packets:	Number of data packets, excluding control data packets, received from TELLINET Servers. This includes retransmitted data packets.
Received Retransmitted Data Packets:	Number of data packets, excluding control data packets, that are retransmitted to this TELLINET Client from TELLINET Servers to adjust packet losses.
Client Connections:	Number of current connections to clients (e.g. web browsers, mail clients).
Server Connections:	Number of current connections to Server Proxies.
Incoming Connections:	Overall number of incoming connections since start of the Client Proxy.
Accepted Connections:	Overall number of accepted connections since start of the Client Proxy.
Push Connections:	Number of current connections for the reception of prefetched data (only for HTTP).
Accepted Push Connections:	Overall number of accepted connections for the reception of prefetched data since the start of the Client Proxy (only for HTTP).
Bytes Received from Clients:	Overall number of bytes received from clients (e.g. web browsers) since the start of the Client Proxy.
Bytes Sent to Clients:	Overall number of bytes sent to clients (e.g. web browsers) since the start of the Client Proxy.

Bytes Received from Servers:	Overall number of bytes received from Server Proxies since the start of the Client Proxy.
Bytes Sent to Servers:	Overall number of bytes sent to Server Proxies since the start of the Client Proxy.
Throughput:	Currently used bandwidth in kbit/s.
Cache Hits:	Web content that has been delivered by the Server Proxy to the Client Proxy during HTTP-Prefetching and is available in the Client Proxy cache when requested by the client, i.e. web browser (only for HTTP).
Late Cache Hits:	Number of requests for web content that has been delivered to the Client Proxy during HTTP-Prefetching shortly after the client, i.e. web browser, has requested the content (only for HTTP). That means that the transmission of the request and the push transmission for the requested web content are overlapping. Data is taken out of the cache of the Client Proxy and the request is not answered by the Server Proxy, i.e. the data is not sent again.
Cache Misses:	Number of requests for web content that was not available through HTTP-Prefetching (only for HTTP).
Overall HTTP Requests:	Overall number of HTTP and FTP requests.

At the bottom of the page the button "Reset statistics" allows to set all values on this page to zero to start the statistical calculation at a specific time. Reset of the statistic will also reset the statistical values on the page "Overview".

5.3.2 PAGE: CONNECTIONS

This page gives an overview of TELLINET transmission processes at the TELLINET / TELICAST Client Proxy. Pull, push and port forwarding connections are listed in different tables. Pull connections are connections caused by HTTP requests from the client, i.e. the web proxy, and push connections are transmissions of web content by pre-fetching. Port forwarding connections are e.g. used for connections to mail servers.

Tables of the following format give information about HTTP/FTP pull, HTTP/FTP push, SOCKS, port forwarding and transparent capturing connections respectively:

User Name	Client Address	Bytes received from client	Bytes sent to client	Server Address	Bytes received from server	Bytes sent to server
Mickey	172.1.1.2	388	0	www.heise.de	0	415
Mickey	172.1.1.2	334	0	www.heise.de	0	383



Please note that for HTTP/FTP push connections there is no connection to the client. Therefore the table does not contain sections for the information on client connections. For each current request/push there is a line giving the following information about it.

User Name	Specifies the name of the user that sent the request.
Client Address:	The IP address of the client, e.g. the web browser host.
Bytes received from client:	Number of bytes that the Client Proxy received from the client, e.g. for requests.
Bytes sent to client:	Number of bytes that the Client Proxy sent to the client, e.g. data of the target URL or e-mail.
Server Address:	Address of the web server from which the data is requested. If the data is taken from the cache, "cache" is stated in addition to the server address.
Bytes received from server:	Number of bytes that the Client Proxy received from the Server Proxy, e.g. data of the target URL or e-mail.
Bytes sent to server:	Number of bytes that the Client Proxy sent to the Server Proxy, e.g. for requests.

5.3.3 PAGE: ETCP ASSOCIATIONS


This page gives detailed information about ETCP sessions within current ETCP associations. The ETCP association “bundles” all logical connections to the Server Proxy. E.g. the reception rate given is the accumulated reception rate of all active HTTP / FTP / port forwarding connections.

A table of the following format gives information about each ETCP association:

Nr.	User Name	Server IP	Server MC	Calc. Send Rate	Send Rate	Sent Pkts	Sent Naks	Retr. Pkts	Last Sent Pkt	Oldest Ret. Pkt	Recv. Rate	Recv. Pkts	Recv. Naks	Obsolete Naks	Dup. Pkts	Last Del. Pkt	First Wait. Pkt	RTT-UDP	RTT-TCP	Calc. TCP Send Rate	
1	Mickey	172.27.1.2/14760	228.30.30.0/3750	0	4872	544	6	0	2	0	998400	7886	0	0	0	7705	0	2000	100	470	

For each current ETCP association there is a line giving the following information about it.

Nr:	Number of the ETCP association.
User Name	Specifies the name of the user that sent the request.
Server IP:	IP address of the TELLINET / TELLICAST Server Proxy and the number of the port used.
Server MC:	Multicast address of the current ETCP association in case the Client is sending data requested by the Server.
Calc. Send Rate:	Maximum bandwidth of this ETCP association in bit/s. The actual bandwidth may be smaller, because in case the overall bandwidth is smaller than the sum of the maximum bandwidths of all current ETCP associations, the bandwidth for each ETCP association is reduced.
Send Rate:	Current bandwidth for outgoing connections (in bit/s).
Sent Pkts:	Number of sent data packets.
Sent Naks:	Number of NAK packets (negative acknowledgements) sent by the Client Proxy.
Retr. Pkts:	Number of data packets that are sent again to adjust packet losses.
Last Sent Pkt:	Sequence number of the last data packet received.
Oldest Ret. Pkt:	Sequence number of the last packet that has been stored temporarily to be available for potential retransmission.
Recv. Rate:	Current bandwidth for incoming connections (in bit/s).
Recv. Pkts:	Number of received data packets.
Recv. Naks:	Number of NAK packets (negative acknowledgements) received by the Client Proxy.
Obsolete Naks:	Number of NAKs that already led to repeated transmission of the lost data packets.
Dup. Pkts:	Number of packets that are received in duplicate.

Last Del. Pkt:	Sequence number of the last delivered data packet.
First Wait. Pkt:	Sequence number of the first data packet in the queue for sending. A packet is queued when packets with lower sequence numbers, that have to be sent earlier, are not available yet.
RTT-UDP:	Current round trip time for UDP packets in milliseconds in case the Client is sending data requested by the Server.
Recent Worst RTT-UDP:	The worst, i. e. longest, round trip time measured for the association for UDP packets in milliseconds in case the Client is sending data requested by the Server.
RTT-TCP:	Current round trip time for TCP packets in milliseconds in case the Client is sending data requested by the Server.
Calc. TCP Send Rate:	Maximum bandwidth used to send TCP packets in bit/s in case the Client is sending data requested by the Server.
	A button to immediately terminate the ETCP association.

5.4 SUBMENU FOR TELLICAST

5.4.1 PAGE: TELLICAST STATISTICS

This page gives an overview of the state of TELLICAST. It will be refreshed periodically.

Active Since:

Recipient(s): Name of the recipient(s) as specified in the *Client Configuration File*.

Active Announcement Channels: The number of announcement channels which are currently received.

Active Data Channels: The number of data channels which are currently received.

Connecting Channels: The number of channels to which the client is currently connecting.

Disconnecting Channels: The number of channels from which the client is currently disconnecting.

Blocked Channels: The number of channels which are currently blocked, i.e. the Client is invited for this channel but the data is not received because the necessary bandwidth is not available at the moment. Possible reasons are that the maximum bandwidth configured in the Client for this channel is exceeded or that too many channels are received at the same time and the maximum bandwidth for the reception of channels is exceeded.

Maximum Bandwidth: Maximum allowed bandwidth for multicast reception (specified with the parameter "total_bandwidth" in the *Client Configuration File*).

Assigned Bandwidth: Sum of bandwidth used by all channels currently received.

Server Address: IP Address of the TELLINET / TELLICAST Server currently connected to the TELLINET / TELLICAST Client.

Announcement Channel Name(s): Name(s) of the currently received announcement channel(s).

Announcement Channel Address(es): Multicast address(es) of the currently received announcement channel(s).

Received Data Packets: Number of data packets received since start of the TELLICAST Client or since reset of statistics.

Missed Data Packets before FEC: Number of transmitted data packets that were not received by the TELLICAST Client. The Client uses forward error correction (FEC) to recover the lost data packets. The number of data pack-

ets recovered by FEC is also displayed at the page "TELLICAST Statistics" (see: Recovered Data Packets).

The received data is completely available at the Client if the number of data packets missed before FEC does not exceed the number of recovered data packets.

Recovered Data Packets:

Number of data packets that were not received by the TELLICAST Client but have been recovered by forward error correction (FEC). The overall number of data packets missed before FEC is done is also displayed at the page "TELLICAST Statistics" (see: Missed Data Packets before FEC).

The received data is completely available at the Client if the number of data packets missed before FEC does not exceed the number of recovered data packets.

Received User Bytes:


Number of data bytes received since start of the TELLICAST Client or since reset of statistics. This value includes bytes from the transmitted data and from TELLICAST control information but excludes bytes for multicast headers.

At the bottom of the page the button "Reset statistics" allows to set all values on this page to zero to start the statistical calculation at a specific time. Reset of the statistic will also reset the statistical values on the page "Overview".

5.4.2 PAGE: ACTIVE CHANNELS

This page gives an overview of all currently received channels along with the most important parameters. This page will automatically be updated periodically.

The announcement channel and the data channels are listed in different tables, which both have the following format:

Name	State	Priority	Bandwidth	Received Bytes	
File Channel	Active	1	200000	2066704	Info

For each currently received channel there is a line giving the following information:

Name: Name of the channel.

State: Status of transmission. This can be:

- **Connecting:** Establishing the connection
- **Active:** Data transmission
- **Disconnecting:** Cutting down the connection
- **Available:** The channel is announced but the data is not currently requested by the Client
- **Blocked:** The Client is invited for this channel but reception of the channel is not possible because the necessary bandwidth is not available at the moment. The reason for that may be that the maximum bandwidth configured in the Client Channel File for this channel is exceeded or that too many channels are received at the same time and the maximum bandwidth for the reception of channels (as configured in the Client Configuration File) is exceeded.

Priority: Priority of the channel as specified in the Client *Channel File*.

Bandwidth: Bandwidth of the channel.

Received Bytes: Number of bytes received on the channel.



A link to the page "Channel Information", which shows more information about the channel.

5.4.3 PAGE: CHANNEL INFORMATION

The page is not included in the menu and can be opened by clicking the button "Info" in the table on the page [Active Channels](#).

A table is displayed, showing general transmission parameters for the channel.

Name	Value
Name	TSL Announcement Channel
State	Active
Address	229.1.1.1/2511
Priority	100
Bandwidth	100000
Received Bytes	812

Name: Name of the channel.

State: Status of transmission. This can be:

- **Connecting:** Establishing the connection
- **Active:** Data transmission
- **Disconnecting:** Cutting down the connection
- **Available:** The channel is announced but the data is not currently requested by the Client
- **Blocked:** The Client is invited for this channel but reception of the channel is not possible because the necessary bandwidth is not available at the moment. The reason for that may be that the maximum bandwidth configured in the Client Channel File for this channel is exceeded or that too many channels are received at the same time and the maximum bandwidth for the reception of channels (as configured in the Client Configuration File) is exceeded.

Address: Multicast address which is used for the transmission on the channel.

Priority: Priority of the channel as specified in the Client *Channel File*.

Bandwidth: Bandwidth of the channel.

Received Bytes: Number of bytes received on the channel.

For channels currently transmitting FileBroadcast, the following additional information is given in a separate table:

Name	Value
Target Directory	received
Received Files	95
Received Bytes	1024098

Target Directory The target directory where the received files are stored.
 Received Files Number of completely received files.
 Received Bytes Number of bytes of the completely received files.

For channels currently transmitting DataStreams, the following additional information is given in a separate table:

Name	Value
Protocol	tcp
Address	8405
Interface	172.24.24.24
TTL	1
Received Bytes	17864

After reception of the data stream from the TELLINET / TELLCAST Server, the TELLINET / TELLCAST Client further transmits the data stream to the final destination, i.e. the recipient. Most of the particulars given in this table refer to the final distribution of the data from the TELLINET / TELLCAST Client to the recipient.

Protocol Protocol used for the final distribution of the data to the recipients.
 Address Address and port number used by the recipient to receive the data from the TELLINET / TELLCAST Client.
 Interface IP address of the interface at the Client host used to provide the data stream to the recipients.
 TTL Maximum number of routers that a data packet can pass between the Client and the recipient.
 Received Bytes Number of bytes received on the channel.

5.5 PAGE: LICENSE

This page lists all available product modules of TELLINET / TELLICAST and indicates whether the modules are licensed /activated.

The functionality of the present product is limited to the licensed modules.

Licensed modules can be activated or deactivated by addition of a section [activations] in the *License File* (see 4.3).

When no section [activations] is available, all licensed modules are active.

Deactivation of modules that are not used at the moment has the following advantages:

- Processor load is reduced.
- The overview of the product handling and control is enhanced, as the web interface only displays the information necessary for the operation of activated modules.

The page also gives information about the version of the software used and details about the license of the Server currently connected to the Client. The Server transmits the license information every 30 seconds. Until the first transmission of the license information "not yet received" is displayed for the Server license.

The information on the Client identification can be used to register the Client for the transmission service or to update the registration information about the Client at the Server (in case this is not done automatically).

Next to the user name, the host keys_of the Client are given when a license for TELLICAST is available and / or TELLINET authentication is activated in the *Client Configuration File*. These keys are generated automatically by the software out of specific host parameters to restrict the utilisation of the software to a specific host. The host keys have to be communicated to the operator of the Server who has to enter them into the *Recipient File* of the Server in order to allow the Server to identify the Client as entitled user of the service. When hardware and/or mayor software changes are made at the Client host, the host keys are changed and the entries at the Server have to be updated.

5.6 PAGE: LOG FILE

This page shows the current *Log File* as a text. The *Log File* can be used to check what error or warning has occurred when the status of the Client is not "OK".

The file format is described in chapter 4.5.

On top of the lines giving the status information, a form is displayed that allow to alter the level of the log output:

- none: nothing is written to the Log file.
- quiet: only error, warning and status messages are written to the Log File.
- normal: error, warning and status messages are written to the Log File together with information on the program behaviour.
- verbose: detailed information about the program behaviour is given.

5.7 SUBMENU FOR HELP

The online help of the Client differs from this documentation as it does not provide all possible product information but is limited to information on the product modules actually licensed.

5.7.1 PAGE: OPERATION

This page describes how to operate the Client.

5.7.2 PAGE: WEB INTERFACE

This page gives explanation about all pages of the web interface.

5.7.3 PAGE: FILE FORMATS

This page describes the possible entries for the files that are necessary to run the licensed modules.

5.7.4 PAGE: FEEDBACK MAIL

This page allows to send an e-mail to the operator / support to make any comments on the product or to get help in case problems occurring during operation cannot be solved by the user. The following entries are allowed in the form:

Sender*:	<input type="text" value="sender@company.com"/>
Recipient(s)*:	<input type="text" value="operator@company.com"/>
Subject:	<input type="text" value="feedback mail"/>
Observed Behaviour*:	<div></div>
How to repeat*:	<div></div>
Comments:	<div></div>
Attachments:	<div><div>all attachments</div><div>no attachments</div><div><input type="checkbox"/> The executable, 9424045 Bytes</div><div><input checked="" type="checkbox"/> The main configuration file, 25773 Bytes</div><div><input checked="" type="checkbox"/> The license file, 1491 Bytes</div><div><input checked="" type="checkbox"/> The central log file, 19559 Bytes</div></div>
<div>send mail</div>	

Sender:	<p>E-mail address of the person / company sending the mail. When a sender e-mail address is configured in the section [mail] of the <i>Client Configuration File</i> that e-mail address is displayed as default value.</p> <p>Please note that it is not possible to enter more than one sender e-mail address. When more than one address is entered, the first e-mail address is used and all other entries are ignored.</p>
Recipient(s):	<p>E-mail addresses of the recipients for the feedback mail as configured in the section [mail] of the <i>Client Configuration File</i>, e.g. the e-mail of the operator / support.</p> <p>When the e-mail should be send to more than one recipient, the e-mail addresses of the recipients have to be separated by commas, semi-colons or spaces.</p>
Subject:	<p>Subject of the e-mail. This entry will appear in the subject heading of the e-mail.</p>
Observed Behaviour:	<p>A description of the observed behaviour.</p>
How to repeat:	<p>A description of the necessary actions to be taken to repeat the behaviour described above. This entry should allow the operator / support to reproduce the behaviour.</p>
Comments:	<p>Any remarks or comments.</p>
Attachments:	<p>In order to find a reason for the observed behaviour described in the e-mail it is helpful for the support to know the exact entries in the files involved. Setting a flag at one of the file names will cause the Client to send the file content attached with the "feedback mail".</p> <p>Clicking on the button "All Attachments" will include all possible files as attachment.</p> <p>Clicking on the button "No Attachment" will exclude all possible files from being sent as e-mail attachment.</p> <p>The size of the files is given as additional information. Files without content are not included in the list.</p>

The e-mail recipient will receive all information entered in the form and the following additional information:

- product name,
- version of the software,
- date of the last modification of the Software (UTC time),
- size of the software in byte,
- working directory of the Client,
- uptime (time elapsed since the start of the Client process),
- UTC time when the Client was started,
- operating system of the Client host,

- size of the working memory (RAM) of the Client host in MB,
- file index listing all attachments with the path (relative to the working directory, unless otherwise specified in the *Client Configuration File*), the file size in byte and the date of the last modification (UTC time) of each file.

5.8 REFRESH

Clicking on this button will update the content of the main area.

6 CONTACT

TELLINET / TELICAST is distributed by

Tellitec Communications bvba
Laarstraat 5
B-9100 Belgium
Tel.: + 32.3.780-6545
Fax: + 32.3.780-6546
e-mail: tellitec@tellitec.be
www.tellitec.be

7 ABBREVIATIONS AND DEFINITIONS

A

ACK	Acknowledgement
announcement channel	Logical channel created by the TELLICAST software for the distribution of invitations and <i>data keys</i> .
ASCII	American Standard Code for Information Interchange.
association	ETCP connection that includes one or more TCP channels and optionally a unidirectional UDP channel from the Server Proxy to the Client Proxy of TELLINET.
atomicity	Transmission of a bundle of files as a unit. Files are cached invisibly for the recipient in the Client system until all files are available. They become visible for the recipient only after complete reception of all files. If one or more files are missing, no files will be transferred to the recipient.

B

bit/s	bits per second
-------	-----------------

C

cache	Device for temporary storage of web objects (such as HTML documents) for later retrieval.
customer	Person who orders the transmission of specific data to a number of recipients.
channel	Here: logical channel created by the TELLICAST software for the distribution of data or announcements and data keys.
client	The client part of a client-server architecture. Typically, a client is an application that runs on a personal computer or workstation and relies on a server to perform some operations.
Client	Software of the TELLINET / TELLICAST system responsible for data reception.
congestion control	Avoidance of packet loss during high transmission frequency, e.g. through transmission slow down.
crawler	Tool for the automatic search and collection of data in the internet.

D

data key	Key for the encryption of the data of a job.
----------	--

E

e.g.	example given
ETCP	Enhanced Transmission Control Protocol developed by Tellitec to enhance the speed of TCP transmission over satellite connections.

F

FEC	Forward Error Correction; Reconstruction of lost data packets with the aid of redundancy packets.
FTP	File Transfer Protocol. Standard application for TCP/IP, allowing file transfer but no file access.

H

HTML	HyperText Markup Language, the authoring language used to create documents on the World Wide Web.
HTTP	HyperText Transport Protocol; Protocol for data transmission inside the World Wide Web.

I

i.e.	id est, that is
invitation	Signal send by the TELLICAST Server through the announcement channel informing the Client about incoming data and initialising data reception.
IP	Internet Protocol, specifying the format of packets and the addressing scheme.
IP address	Identifier for a computer or device on an IP network.

J

job	Transmission of a specific set of data to specific recipients by TELLICAST. A job includes the whole process including all transmissions necessary for safe data transfer, reception of acknowledgements and writing of accounting information.
-----	---

K

kbit/s	kilobits per second
--------	---------------------

M

MB	Megabytes
Mbit/s	Megabits per second
MByte	Megabyte
module	TELLINET / TELLICAST software adapted to a specific application.
multicast	Transmission of a message to a select group of recipients (Point to multipoint transmission).

N

NAK	Negative Acknowledgement. Transmission control by sending of acknowledgements when data packets are not received.
-----	---

P

POP3	Post Office Protocol Version 3. A protocol used to retrieve e-mail from a mail server.
Port-Forwarding	Forwarding of data packets with a specified port number.
prefetching	Patent pending technology included in TELLINET, that allows the Server Proxy to scan all HTML files passed to the clients for referenced subobjects, retrieve this objects from the web servers and push them to the client before the clients requests the subobjects.
proxy	Here: proxy server. A server that sits between a client application, such as a web browser, and a real server. It intercepts all requests to the real server to see if it can fulfill the requests itself. If not, it forwards the request to the real server.
private key	Password assigned to a recipient, used for safe distribution of data keys.
pull	Data transmission caused by client requests.

	push	Data transmission without client requests.
Q		
	QoS	Quality of Service
R		
	recipient	Person receiving the data sent with TELLINET / TELLICAST.
S		
	server	A computer or device on a network that manages network resources.
	Server	Software of the TELLINET / TELLICAST system responsible for data sending/transmission.
	session	ETCP connection, part of an ETCP association.
	SOCKS	A protocol for handling TCP traffic through a proxy server.
T		
	TCP	Transmission Control Protocol; protocol for reliable data transport.
U		
	URL	Uniform Resource Locator; address of internet files, mainly used in the world wide web.
	UTC	Universal Time Coordinated. UTC is equal to the Greenwich Mean Time (GMT).
W		
	web interface	HTML pages designed for the supervision/control of the Server and Client of the TELLINET / TELLICAST system.
Y		
	yyyy-mm-dd hh:mm:ss	year year year year – month month – day day hour hour : minute minute : second second; Format of the entry for date and time, giving the number of digits for the entry of each component.