June 20, 2003

Networked Transport of RTCM via Internet Protocol

Ntrip, Version 1.0

Design – Protocol – Software

Part I & II

Developed within the framework of EUREF, European Sub-Commission of Commission X (Global and Regional Geodetic Networks) of the International Association of Geodesy (IAG)

Published by Federal Agency for Cartography and Geodesy (BKG), Frankfurt, Germany

Authors

Harald Gebhard[1]

Georg Weber[2]

Contributions from

Rüdiger Kays[1]

Denise Dettmering[2]

Christian Pagels[3] (Part I)

Sebastian Pötschke[3] (Part I)

Ken Fortune[4] (Part I)

Dirk Stöcker[5] (Linux Client Example)

Affiliations

[1] Lehrstuhl für Kommunikationstechnik, Universität Dortmund, Germany

[2] Bundesamt für Kartographie und Geodäsie (BKG), Frankfurt, Germany

[3]Trimble Terrasat GmbH, Höhenkirchen, Germany

[4]Trimble New Zealand Limited, Christchurch, New Zealand

[5]Geodätisches Institut, TU Dresden, Germany

General Note

Networked Transport of RTCM via Internet Protocol (Ntrip) stands for an application-level protocol streaming Global Navigation Satellite System (GNSS) data over the Internet. Ntrip is a generic, stateless protocol based on the Hypertext Transfer Protocol HTTP/1.1. The HTTP objects are enhanced to GNSS data streams.

Ntrip is designed for disseminating differential correction data (e.g in the RTCM-104 format) or other kinds of GNSS streaming data to stationary or mobile users over the Internet, allowing simultaneous PC, Laptop, PDA, or receiver connections to a broadcasting host. Ntrip supports wireless Internet access through Mobile IP Networks like GSM, GPRS, EDGE, or UMTS.

Ntrip is implemented in three system software components: NtripClients, NtripServers and NtripCasters. The NtripCaster is the actual HTTP server program whereas NtripClient and NtripServer are acting as HTTP clients.

Ntrip is meant to be an open none-proprietary protocol. Major characteristics of Ntrip's dissemination technique are:

- Based on the popular HTTP streaming standard; comparatively easy to implement when having limited client and server platform resources available.
- Application not limited to one particular plain or coded stream content; ability to distribute any kind of GNSS data.
- Potential to support mass usage; disseminating hundreds of streams simultaneously for up to thousand users possible when applying modified Internet Radio broadcasting software.
- Considering security needs; stream providers and users don't necessarily get into contact, streams often not blocked by firewalls or proxyservers protecting Local Area Networks.
- Enables streaming over any mobile IP network because of using TCP/IP.

This Ntrip documentation will provide information to readers with diverse levels of IT knowledge. Its information is therefore given in a more narrative language, trying to avoid a less comprehensible technical diction.

This documentation describes Ntrip Version 1.0. The work done so far for Part I sections "Client Messages" and "Source-Table" has been frozen on March 3rd, 2003 in order to allow the development of Ntrip-based implementations and applications on a firm and fully anchored basis. Further Ntrip versions will be issued when necessary. Contributions from anyone to the ongoing work on Ntrip are highly appreciated.

Document History

15 Jan 2003	Draft
24 Jan 2003	Published via Internet
28 Jan 2003	Wording, format, stylistics
30 Jan 2003	Source-table edited, parameter <vrs> renamed to <nmea>
31 Jan 2003	NMEA Request Messages included
03 Feb 2003	Modification of server and client messages, screenshots updated, Downloads section included
04 Feb 2003	NtripCaster hardware configuration example and proxyserver option included
05 Feb 2003	Additional explanations in Example Implementation section
10 Feb 2003	Appendix B extended, meaning of parameter <protection> in source-table changed
20 Feb 2003	Source-table parameters <solution>, <carrier>, <authentication>, and <fee> added, parameter <protection> deleted, Appendix B split into two sections
25 Feb 2003	Wording corrected
26 Feb 2003	Appendix A, new example source-table included
27 Feb 2003	Source-table: parameter <operator> included in CAS-records, sequence of parameters in STR-records changed, meaning of parameter <carrier> changed
03 Mar 2003	Status of work on technical content of Part I frozen as “Ntrip Version 1.0”
25 Mar 2003	Appendix C, Linux Client Example added
04 Apr 2003	GNSS Internet Radio screenshots updated
19 May 2003	GNSS Internet Radio screenshots updated
18 June 2003	Co-Processor included in Broadcaster hardware recommendation section
20 June 2003	Source-Agent included in Server Message section, Downloads section extended

Table of Contents

PART I Networked Transport of RTCM via Internet Protocol

This documentation is separated in two parts. Part I describes system design and protocol details, aiming to define an HTTP streaming technique for the dissemination of GNNS data. Part II describes an example software implementation. While Part I is meant to define the mandatory communication procedures, Part II reports on an example implementation and thus contributes to software availability.

1. Introduction

Differential GPS (DGPS), or actually any Differential Global Navigation Satellite System (DGNSS) in general, improves the achievable positioning accuracy by correcting the pseudo-distances between receiver and satellites or the receiver position. Serious problems in GNSS positioning are caused by ephemeric, tropospheric, ionospheric, and clock errors. DGNSS is based on the comparison of parameters calculated from observations at a reference station and their well-known correct values. The differences can be used to calculate correction data as defined by the Radio Technical Commission for Maritime Services (RTCM). These RTCM data can be sent to a mobile GNSS rover.

With the stop of the degradation of GPS signals (Selective Availability, SA) on May 1 2000, a stationary GPS receiver without a differential correction signal will show a 10-15m average radius “random walk” pattern. The positioning error of the same receiver can be reduced to approximately ±1.0 m by receiving about 50 Byte/s of DGPS correction data in RTCM format.

The validity of corrections depend on the distance between receiver and rover (±0.5 m accuracy at a distance of 50 km). For providing a service covering a large area, a reference station network has to be introduced, which is able to generate a specific real-time correction for any region. The specific corrections can be transmitted over various communication channels, e.g. via radio transmission (LF, MF, HF, UHF), or a mobile communication network using different communication protocols.

This documentation describes a HTTP-based protocol for disseminating RTCM correction data or other kinds of GNSS streaming data to stationary or mobile receivers over the Internet, allowing simultaneous PC, Laptop, PDA, or receiver connections to a broadcasting host via IP Networks including GSM, GPRS, EDGE, or UMTS. The protocol development follows a feasibility study on real-time streaming of differential GPS corrections as described in [2]). Through the given definitions of this paper, and because a major application will be the dissemination of RTCM corrections, the system as a whole presents a format which should be referred to as the Ntrip-Format (Networked Transport of RTCM via Internet Protocol).

As far as DGNSS and Real Time Kinematic GNSS (RTK-GNSS) is concerned, the use of the popular RTCM-104 standard as the streaming data format is recommended whereby sufficient precision is obtained if given correction data are not older than a few seconds. The RTCM-104 standard is used worldwide. Most (if not all) popular GNSS receivers accept RTCM-104 differential correction messages [3].

The basic data streaming is accomplished by TCP/IP protocol stack. Several attempts based on a plain Serial-to-TCP conversion of streaming data on the reference-side (server) and TCP-to-Serial re-conversion on the rover-side (client) have shown the suitability of the TCP/IP protocol for streaming data to mobile IP clients [4].

2. System Concept

The Hypertext Transfer Protocol (HTTP) is designed as an application-level protocol for distributed collaborative hypermedia information systems but can also be used for linear streaming media. The basic unit of HTTP communication, consisting of a structured sequence of octets matching the syntax, is defined in the protocol and transmitted via a TCP/IP connection. Client and server must understand HTTP request messages and answer with adequate HTTP respond messages.

Ntrip uses HTTP. It is implemented in three programs called NtripClient, NtripServer and NtripCaster, whereby the NtripCaster is the real HTTP (splitter-)server. NtripClient and NtripServer are acting as HTTP client programs.

Concerning message format and status code, the NtripClient-NtripCaster communication is a full compatible HTTP 1.1 communication [1] whereby Ntrip uses only nonpersistent connections. The NtripServer-NtripCaster communication extends HTTP by a new message format which is “SOURCE” and a new status-code which is “ERROR - Bad Password”.

Loosing the TCP connection between communicating system-components (NtripClient-NtripCaster, NtripServer-NtripCaster) will be automatically recognized by the involved TCP-sockets. This effect can be used to trigger software events like an automated reconnection.

This Ntrip system (see Figure 1) consists of

- NtripSources which generate data streams at a specific location and
- NtripServers which transfer the data streams from a source to an
- NtripCaster, the major system component.
- NtripClients finally access data streams of desired NtripSources on the NtripCaster.

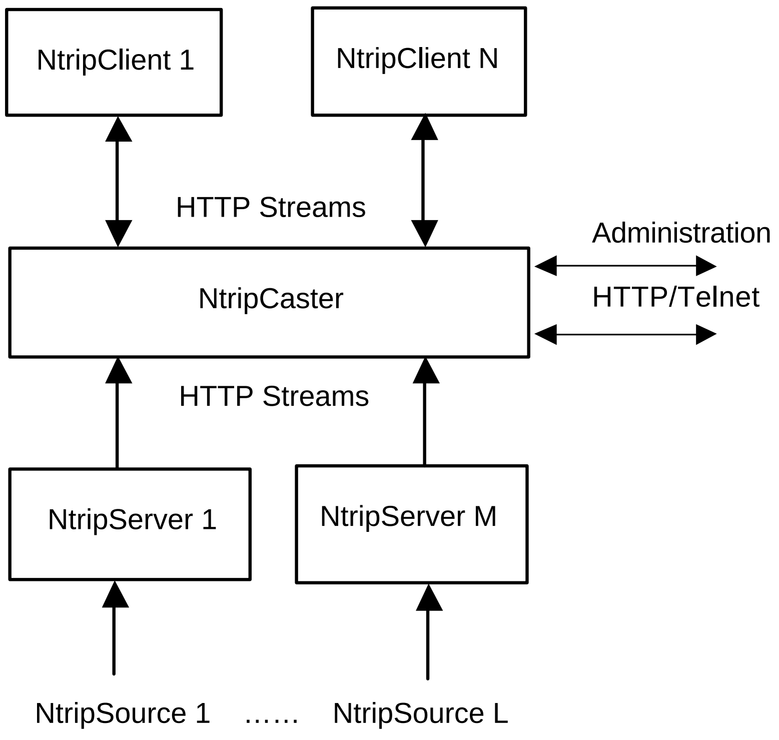


Fig. 1. Ntrip streaming system

NtripServers define a source ID called "mountpoint" for every streamed NtripSource. Several NtripClients can access data of desired NtripSources at the same time by requesting a source by its mountpoint on the NtripCaster. If implemented in the NtripCaster program, authorized personal may remote control the NtripCaster via a password-protected Telnet session or receive status information via a password-protected HTTP session using an Internet Browser.

An administrator running an NtripCaster is responsible for allowing new NtripServers to connect with new NtripSources. The administrator organizes all available NtripSources and defines all source IDs (mountpoints).

NtripClients need the possibility to choose an NtripSource by its mountpoint on the NtripCaster. Therefore a source-table is introduced into and maintained on the NtripCaster. Each record of this source-table contains parameters describing attributes of a data stream, a network of data streams, or an NtripCaster. Stream attributes (identifier, coordinates, format, nav-system, mountpoint, etc.) are defined at the NtripServer side for each NtripSource.

If an NtripClient sends an invalid or no mountpoint (no or not up-to-date source-table available for the client), the NtripCaster will upload an up-to-date source-table as a HTTP object instead of a GNSS data stream. Afterwards the NtripClient has the up-to-date source-table available and can connect to a GNSS data stream on the NtripCaster.

The Ntrip system depends on direct communication between the responsible administrators of NtripCasters and NtripServers (e.g. via email). They have to specify the parameters characterizing an NtripSource/mountpoint in the source-table.

3. System Elements

A DGNSS reference station in its simplest configuration consists of a GNSS receiver, located at a well-surveyed position. As this stationary-operated GNSS receiver knows where the satellites are located in space at any point in time, as well as its own exact position, the receiver can compute theoretical distance and signal travel times between itself and each satellite. When these theoretical values are compared to real observations, differences represent errors in the received signals. RTCM corrections are derived from these differences. Making these corrections available in real-time for mobile users is the major purpose of the Ntrip system elements although Ntrip may be used as well for transporting other types of GNSS streaming data over its system elements NtripServer, NtripCaster, and NtripClient.

NtripSource

The NtripSources provide continuous GNSS date (e.g. RTCM-104 corrections) as streaming data. A single source represents GNSS data referring to a specific location. Source description parameters as compiled in the source-table specify the format in use (e.g. RTCM 2.0, RTCM 2.1, Raw), the recognized navigation system (GPS/ GPS+GLONASS), location coordinates and other information.

Every single NtripSource needs a unique mountpoint on an NtripCaster.

NtripServer

The NtripServer is used to transfer GNSS data of an NtripSource to the NtripCaster. Before transmitting GNSS data to the NtripCaster using the TCP/IP connection, the NtripServer sends an assignment of the mountpoint.

Server passwords and mountpoints have to be defined by the administrator of the NtripCaster and handed over to the administrators of the involved NtripServers. An NtripServer in its simplest set-up is a computer program, running on a PC, which sends correction data of an NtripSource (e.g. as received via the serial communication port from a GNSS receiver) to the NtripCaster.

The Ntrip protocol may be used for the transportation of RTCM data of a virtual reference station following the so called VRS concept. Based on data from a number of reference stations, RTCM corrections are derived for a virtual point at the users approximate position. Data for these virtual reference station represent a single NtripSource which can be transmitted by an NtripServer.

NtripCaster

The NtripCaster is basically a HTTP server supporting a subset of HTTP request/response messages and adjusted to low bandwidth streaming data (50 up to 500 Bytes/sec per stream). The NtripCaster accepts request-messages on a single port from either the NtripServer or the NtripClient. Depending on these messages, the NtripCaster decides whether there is streaming data to receive or to send.

An NtripServer might be a part of the NtripCaster program. In this case only the capability of receiving NtripClient messages has to be implemented into the combined NtripCaster/NtripServer. Built-in HTTP-based remote administration capability is an optional function.

NtripClient

An NtripClient will be accepted and receive data from an NtripCaster, if the NtripClient sends the right request message (TCP connection to the specified NtripCaster IP and listening Port). Concerning message format and status code, the NtripClient-NtripCaster communication is fully compatible to HTTP 1.1, but Ntrip uses only nonpersistent connections.

4. Server Messages

The NtripServer-NtripCaster communication extends HTTP by the additional message format "SOURCE" and the additional status-code "ERROR - Bad Password". The password is not protected and therefore based like in the HTTP Basic Access Authentications scheme on the assumption, that the connection between the client and the server can be regarded as a trusted carrier.

The NtripServer has to connect to the NtripCaster using the IP and specified listening Port of the NtripCaster. This means, that the NtripCaster has to be "up and running" before any source can connect. Before transmitting GNSS data to the NtripCaster using the TCP/IP connection, the NtripServer has to send an Ntrip server message in order to get access and to specify a mountpoint.

This message is shown in Figure 2.

```
SOURCE <password> <mountpoint> <CR><LF>
Source-Agent: NTRIP<product|comment><CR><LF>
<CR><LF>
<data>
```

Fig. 2. Server message structure
<password> = Encoder password of the Caster
<mountpoint> = Caster mountpoint for the Source
<product|comment> = Information about the source agent

Example: An NtripServer sends the Ntrip server message:

```
⇒ SOURCE letmein /Mountpoint
  Source-Agent: NTRIP NtripServerCMD/1.0
```

If the password is correct, the NtripCaster will answer:

```
⇐ ICY 200 OK
```

The caster accepts data after sending the message "ICY 200 OK". The NtripCaster will only accept data from sources with the valid encoder password. An administrator has to pre-define this password on the caster. The password is coded as a plain ASCII string.

Sending a false password leads to the caster response message:

```
⇐ ERROR - Bad Password
```

The caster than automatically quits the TCP connection.

5. Client Messages

A client’s request is designed as a HTTP message similar to the server message shown in Figure 2. Each client only needs to know the mountpoint of the desired data stream. The message for requesting a data stream is defined in Figure 3. The User-Agent request-header field has to begin with the string NTRIP.

```
GET <mountpoint> HTTP/1.0 <CR><LF>
User-Agent: NTRIP<product|comment><CR><LF>
Accept: */* <CR><LF>
Connection: close <CR><LF>
<CR><LF>
```

Fig. 3. Client message structure
<mountpoint> = Caster mountpoint of requested source
<product|comment> = Information about the user agent originating the request

Example: A client sends (in the non protected case) the request message:

```
⇒ GET /BUCU0 HTTP/1.0
   User-Agent: NTRIP GNSSInternetRadio/1.2.0
```

For a valid request (desired NtripSource/mountpoint exists), the NtripCaster will answer:

```
⇐ ICY 200 OK
```

followed by the GNSS data

```
⇐ <GNSS data>
```

For an invalid request (desired NtripSource/mountpoint does not exist), the NtripCaster will answer:

```
⇐ SOURCETABLE 200 OK
```

Followed by the source-table object

```
⇐ <Source-Table>
```

Via the information from the source-table, as maintained by the NtripCaster, NtripClients are enabled to choose the desired NtripSource/mountpoint from all available NtripSources/mountpoints. An NtripClient can either store a source-table in memory (e.g. hard disc), or request a new source-table before requesting each new NtripSource. The desired NtripSource can be chosen manually (e.g. based on identifier, nav-system, and format information) or by software (e.g. based on position, format, and nav-system information).

Requesting unavailable mountpoints (not up-to-date source-table) will automatically result in the caster replying with an up-to-date source-table. Therefore, mountpoints, as a synonym for a specific NtripSource, have to be unique on an NtripCaster. Using a Four-character-ID followed by an integer value (e.g. BUCU0 for Bukarest) as mountpoint parameter is recommended.

An example for handling client messages in C programming language is given in Appendix C. This simple Linux/UNIX NtripClient program reads data from an NtripCaster and writes on standard output for further redirection of data to a file or COM-port.

5.1. Basic Authentication Scheme

For billing purposes, the NtripSources/mountpoints can be password-protected on an NtripCaster. In this protected case the HTTP communication differs from the non-protected case.

Example:

The client will send (like in the non-protected case) the request message:

```
⇒ GET /BUCU1 HTTP/1.0
   User-Agent: NTRIP GNSSInternetRadio/1.2.0
```

The HTTP-server will answer a client request according to the HTTP Protocol [5] with

```
⇐ HTTP/1.0 401 Unauthorized
```

for invalid passwords and send a second message to the client:

```
⇐ Server: NtripCaster/1.0
    WWW-Authenticate: Basic realm="/BUCU1"
    Content-Type: text/html
    Connection: close
    <html><head><title>401 Unauthorized</title></head><body bgcolor=black text=white
            link=blue alink=red>
    <h1><center>The server does not recognize your privileges to the requested entity
            y/stream</center></h1>
    </body></html>
```

The client sends a second request message for the same mountpoint including the base64 coded user:password string to the caster:

```
⇒ GET /BUCU1 HTTP/1.0
   User-Agent: NTRIP GNSSInternetRadio/1.2.0
   Authorization: Basic aHVnb2JlbjpodWdvYmVuMTIz
```

The NtripCaster will answer

```
⇐ ICY 200 OK
```

followed by the GNSS data:

```
⇐ <GNSS data>
```

If the client already knows beforehand that a mountpoint is password protected, it can send the password with the first request message.

Example:

```
⇒ GET /BUCU1 HTTP/1.0
   User-Agent: NTRIP GNSSInternetRadio/1.2.0
   Authorization: Basic aHVnb2JlbjpodWdvYmVuMTIz
```

The NtripCaster will again answer

```
⇐ ICY 200 OK
```

followed by the GNSS data:

```
⇐ <GNSS data>
```

Different from the server password, the client password is coded like the HTTP Basic Authentication Scheme [5] and allows the client access to protected content. The "basic" authentication scheme is based on the model that the user agent must authenticate with a user-ID and a password for each realm (here mountpoint). The realm value should be considered as an opaque string that can only be compared for equality with other realms on that server. The server will authorize the request only if it can validate the user-ID and password for the protected space of the requested mountpoint. There are no optional authentication parameters. To receive authorization, the client sends the user-ID and password, separated by a single colon (":") character and within a “base64” [8] encoded string in the credentials. If the user agent wishes to send the user-ID "Aladdin" and password "open sesame", it would use the following header field:

```
⇒ Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
```

The basic authentication scheme is a non-secure method of filtering unauthorized access to resources on an HTTP server. It is based on the assumption that the connection between the client and the server can be regarded as a trusted carrier. As this is not generally true on an open network, the basic authentication scheme should be used accordingly. In spite of this, clients should implement the scheme in order to communicate with servers that use it [5].

5.2. Digest Authentication Scheme

For applications in need for more security regarding user authentication, the Digest Access Authentication for HTTP specified in RFC 2069 [6] can be used as an alternative.

Like Basic, Digest access authentication verifies that both communicating parties know a shared secret (a password). Unlike Basic, this verification can be done without sending the password in the clear, which is Basic's biggest weakness.

Like Basic Access Authentication, the Digest scheme is based on a simple challenge-response paradigm. The Digest scheme challenges using a nonce value. A valid response contains a checksum (for Ntrip the MD5 [7] checksum is compulsory) of the username, the password, the given nonce value, the HTTP method, and the requested URI. Thus, the password is never sent in the clear.

The Digest Access Authentication scheme is conceptually similar to the Basic scheme. The format of the modified WWW-Authenticate header line and the Authorization header line are specified in RFC 2069. [6]

5.3. NMEA Request Messages

For some network dependent applications it is necessary to send the position of the NtripClient to the NtripCaster. That position could be used by the NtripCaster to provide a data stream for a Virtual Reference Station (VRS) or to determine the best data stream to broadcast. Ntrip allows clients to send NMEA GGA strings after the HTTP request for a data stream.

If the <nmea> parameter is set to “1” (see Source-table section), the NtripCaster waits for a NMEA GGA string to prepare the data and start sending.

Following is an example for a request of a source-table named “vrs_bayern” to get a data stream generated for a virtual reference station with the coordinates of the NMEA string:

```
⇒ GET /vrs_bayern HTTP/1.1<CR><LF>
   Accept: rtk/rtcm, dgps/rtcm<CR><LF>
   User-Agent: NTRIP Survey-Controller-15.0<CR><LF>
   <CR><LF>
   $GPGGA,165631.00,4810.8483085,N,01139.900759,E,1,05,01.9,+00400,M,,M,,*??<CR><LF>
```

6. Source-Table

The NtripCaster maintains a source-table containing information on available NtripSources, networks of NtripSources, and NtripCasters to be send to an NtripClient on request. Source-table records are dedicated to either

a) Data STReams (record type STR),
b) CASters (record type CAS), or
c) NETworks of data streams (record type NET).

This structure is expandable by introducing new record types when necessary. Older NtripClient versions might ignore newly introduced record types. All NtripClients must be able to decode record type STR. Decoding types CAS and NET is an optional feature.

The source-table is initiated by the HTTP/1.1 header fields

```
Server: <NtripCasterIdentifier>/<NtripVersion><CR><LF>
Content-Type: text/plain<CR><LF>
Content-Length: <Content-Length><CR><LF>
<CR><LF>
```

followed by the actual source-table records. The server record contains

- Before the slash: an NtripCaster identifier to be defined by the NtripCaster operator (see parameter #4 in Table 2)
- After the slash: an Ntrip version number (e.g. NtripV1.0) in order allow an NtripClient to understand which version of Ntrip is supported by a particular NtripCaster.

The content-length indicates the size of the source-table records (decimal number of octets, e.g. “Content-Length: 243”).

The end of the source-table is notified by the string: ENDSOURCETABLE

Table 1: Format and contents of source-table records describing a data stream

#	Record Parameter	Meaning	Format	Examples
1	<type> = STR	The following parameters of this record describe a data stream	3 Characters	STR
2	<mountpoint>	Caster mountpoint	Characters, 100 max.	LEIJ0 LEIJ1 WTZJ0
3	<identifier>	Source identifier, e.g. name of city next to source location	Characters, undefined lenght	Frankfurt
4	<format>	Data format RTCM, RAW, etc.	Characters, undefined length	RTCM 2.0 RTCM 2.1 RTCM 2.3 CMR CMR+ RTCM ADV RTCM SAPOS RAW RTCA

5	<format-details>	E.g. RTCM message types or RAW data format etc., update rates in parenthesis in seconds	Characters, undefined length	1(1),2(1),3(30)
				MBEN(1)

6	<carrier>	Data stream contains carrier phase information	Integer	0 1

		phase information 0 = No (e.g. for DGPS) 1 = Yes, L1 (e.g. for RTK) 2 = Yes, L1&L2 (e.g. for RTK)		2
7	<nav-system>	Navigation system(s)	Characters, undefined length	GPS GPS+GLO EGNOS
8	<network>	Network	Characters, undefined length	EUREF IGS IGLOS SAPOS GREF Misc
9	<country>	Three character country code in ISO 3166	3 Characters	GER ITA ESP
10	<latitude>	Position, latitude, north (approximate position in case of nmea = 1)	Floating point number, two digits after decimal point	40.12 -12.14
11	<longitude>	Position, longitude, east (approximate position in case of nmea = 1)	Floating point number, two digits after decimal point	10.12 357.85
12	<nmea>	Necessity for Client to send NMEA message with approximate position to Caster 0 = Client must not send NMEA message with approximate position to Caster 1 = Client must send NMEA GGA message with approximate position to Caster	Integer	0 1
13	<solution>	Stream generated from single reference station or from networked reference stations 0 = Single base 1 = Network	Integer	0 1
14	<generator>	Hard- or software generating data stream	Characters, undefined length	JPS Legacy E GPSNet
15	<compression>	Compression algorithm	Characters, undefined length	none

16	<authentication>	Access protection for this particular data stream
		N = None
		B = Basic
		D = Digest
			1 Character	N
				B
				D

17	<fee>	User fee for receiving this particular data stream
		N = No user fee
		Y = Usage is charged
			1 Character	N
				Y

18	<bitrate>	Bitrate of data stream, bits per	Integer	500

		second		5000
19	<misc>	Miscellaneous information	Characters, undefined length	none Demo

Table 2: Format and contents of source-table records describing a Caster

#	Source-table Element	Meaning	Format	Examples
1	<type> = CAS	The following parameters of this record describe a Caster	3 Characters	CAS
2	<host>	Caster Internet host domain name or IP address	Characters, 128 max.	141.74.243.11 euref-ip.ifag.de
3	<port>	Port number	Integer	80 2101
4	<identifier>	Caster identifier, e.g. name of provider	Characters, undefined lenght	ICD-V1.3 Trimble-iGate
5	<operator>	Name of institution / agency / company operating the Caster	Characters, undefined length	BKG Geo++
6	<nmea>	Capability of Caster to receive NMEA message with approximate position from Client 0 = Caster is not able to handle incoming NMEA message with approximate position from Client 1 = Caster is able to handle incoming NMEA GGA message with approximate position from Client	Integer	0 1
7	<country>	Three character country code in ISO 3166	3 Characters	GER ITA ESP
8	<latitude>	Position, latitude, north	Floating point number, two digits after decimal point	40.12 -12.14
9	<longitude>	Position, longitude, east	Floating point number, two digits after decimal point	10.12 357.85
10	<misc>	Miscellaneous information	Characters, undefined length	none

Table 3: Format and contents of source-table records describing a network of data streams

#	Source-table Element	Meaning	Format	Examples
1	<type> = NET	The following parameters of this record describe a network	3 Characters	NET

		of data streams		
2	<identifier>	Network identifier, e.g. name of a network of GNSS permanent reference stations	Characters, undefined lenght	EPN IGLOS GREF SAPOS-THR
3	<operator>	Name of institution / agency / company operating the network	Characters, undefined length	EUREF IGS TRIMBLE ASCOS GEO++ SAPOS-THR

4	<authentication>	Access protection for data streams of the network
		N = None
		B = Basic
		D = Digest
			1 Character	N
				B
				D
				N,B

5	<fee>	User fee for receiving data streams from this network
		N = No user fee
		Y = Usage is charged
			1 Character	N
				Y
				N,Y

6	<web-net>	Web-address for network information	Characters, undefined length	http://igs.ifag.de
7	<web-str>	Web-address for stream information	Characters, undefined length	http://www.epncb.oma.be none
8	<web-reg>	Web address or mail address for registration	Characters, undefined length	euref-ip@ifag.de http://igs.ifag.de
9	<misc>	Miscellaneous information	Characters, undefined length	none

7. References

[1] Berners-Lee, T., Fielding, R., and H. Frystyk, "*Hypertext Transfer Protocol -- HTTP/1.1*", RFC 2068, January 1997

[2] Gebhard, H., Kays, R.: "*Real-Time Streaming of Differential GPS Corrections via Internet*", Feasibility Study, Informatik Centrum Dortmund, unpublished, March 2002

[3] "*Real-Time Single Difference GPS Positioning*", http://home-2.worldonline.nl/~samsvl/realtmsd.htm

[4] Weber, G.: "*Echtzeit-Übertragung von RTCM-Daten über Internet und Mobilfunk*", Beitrag zum 57. DVW-Seminar „GPS 2002: Antennen, Höhenbestimmung und RTK Anwendungen," 16.-17. Sep 2002, Karlsruhe, Schriftenreihe Deutscher Verein für Vermessungswesen, Band 44, S. 107-116, Stuttgart, 2002

[5] Berners-Lee, T., Fielding, R., and H. Frystyk: "*Hypertext Transfer Protocol -- HTTP/1.0.",* RFC 1945 MIT/LCS, UC Irvine, May 1996

[6] [32] Franks, J., Hallam-Baker, P., Hostetler, J., Leach, P., Luotonen, A., Sink, E., and L. Stewart, "*An Extension to HTTP : Digest Access Authentication*", RFC 2069, January 1997

[7] Rivest, R., "*The MD5 Message-Digest Algorithm*", RFC 1321, April 1992

[8] Borenstein, N., and N. Freed: "*MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies*", RFC 1521, Bellcore, Innosoft, September 1993

Appendix A Example Source-Table

```
SOURCETABLE 200 OK
Content-Type: text/plain
Content-Length: 7864
CAS;129.217.182.51;80;ICD;BKG;0;GER;51.5;7.5;Trial Broadcaster
NET;GREF;BKG;B;N;http://igs.ifag.de/GREF.htm;none;denise.dettmering@bkg.bund.de;none
NET;IGS-IGLOS;BKG;B;N;http://igscb.jpl.nasa.gov/projects/rtwg/;none;denise.dettmering@bkg.bund.de;none
NET;Misc;BKG;B;N;none;denise.dettmering@bkg.bund.de;none
NET;ASCOS;Ruhrgas AG;B;N;http://www.ascos.de;none;denise.dettmering@bkg.bund.de;none
NET;SAPOS-THR;LVA Thueringen;B;N;http://www.thueringen.de/vermessung/SAPOS/;none;denise.dettmering@bkg.bund.de;none
NET;SAPOS-BLN;Berliner Senatsverwaltung;B;N;http://www.stadtentwicklung.berlin.de/geoinformation/sapos/;none;denise.dettmering@bkg.bund.de;none
NET;EUREF;EUREF;B;N; http://www.epncb.oma.be/;none;euref-ip@bkg.bund.de;none
NET;SCIGN-OCRT;SOPAC,CSRC;B;N;http://www.scign.org;none;ybock@ucsd.edu;none
STR;FFMJ2;Frankfurt;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;50.12;8.68;0;1;GPSNet V1.9;none;N;N;560;Demo
STR;FFMT0;Frankfurt;RTCM 2.0;1(3),2(90),3(90),16(90);0;GPS;ALF;GER;50.02;9.02;0;0;Trimble 4000SSI;none;N;N;190;Demo
STR;FFMJ1;Frankfurt;RTCM 2.1;3(19),16(59),18(1),19(1);2;GPS;GREF;GER;50.09;8.66;0;0;GPSNet V1.9;none;N;N;2800;Demo
STR;FFMJ0;Frankfurt;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;50.09;8.66;0;0;Javad Legacy E;none;N;N;3600;Demo
STR;LEIJ0;Leipzig;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;51.33;12.37;0;0;Javad Legacy E;none;B;N;3600;none
STR;WTZJ0;Wettzell;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;49.13;12.88;0;0;Javad Legacy E;none;B;N;3600;none
STR;HELJ0;Helgoland;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;54.18;7.88;0;0;Javad Legacy E;none;B;N;3600;none
STR;TITZ0;Titz;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;51.00;6.42;0;0;Javad Legacy E;none;B;N;3600;none
STR;HUEG0;Huegelheim;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;47.82;7.62;0;0;Javad Legacy E;none;B;N;3600;none
STR;DREJ0;Dresden;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;51.05;13.73;0;0;Javad Legacy E;none;B;N;3600;none
STR;SASS0;Sassnitz;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;54.51;13.64;0;0;Javad Legacy E;none;B;N;3600;none
STR;KARJ0;Karlsruhe;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;49.01;8.41;0;0;Javad Legacy E;none;B;N;3600;none
STR;WARN0;Warnemuende;RAW;Compact(1);2;GPS+GLO;IGS-IGLOS;GER;54.17;12.10;0;0;Javad Legacy E;none;B;N;3600;none
STR;DRES0;Dresden;RTCA;none;none;EGNOS;Misc;GER;51.05;13.73;0;0;none;none;B;N;250;TU Dresden
STR;BARC0;Barcelona;RTCM 2.0;1(1),3(3),16(60);0;GPS;Misc;ESP;41.22;2.16;0;0;none;none;B;N;230;none
STR;BOCH0;Bochum;RTCM 2.1;1(1)3,16,18(1),19(1);2;GPS;Misc;GER;50.45;7.27;0;0;none;none;B;N;13000;FH Bochum
STR;BRUS0;Brussels;RTCM 2.0;1(1),3(60),16;0;GPS;Misc;BEL;50.80;4.36;0;0;Ashtech UZ-12;none;B;N;500;none
STR;CAGL0;Cagliari;RTCM 2.1;1(1),3(60),16(60),18(1),19(1);2;GPS+GLO;Misc;ITA;39.14;8.97;0;0;none;none;B;N;3900;DIST
STR;HANO0;Hannover;RTCM 2.2;none;2;GPS+GLO;Misc;GER;52.42;9.58;0;none;none;none;B;N;none;Geo++
STR;HANJ0;Hannover;RTCM 2.3;3(10),16,18(1),19(1);2;GPS+GLO;ASCOS;GER;52.37;9.73;0;none;GPSNet;none;B;N;4800;ALLSAT
STR;KOPE0;Koper;RTCM 2.1;1(5),3(69),20(1),21(1),22;2;GPS;Misc;SLO;48.40;17.03;0;0;none;none;B;N;2500;none
STR;MADR0;Madrid;RTCM 2.1;1(1),3(10),16,18(1),19(1);2;GPS;Misc;ESP;40.43;355.75;0;0;none;none;B;N;3300;none
STR;MILA0;Milano;RTCM 2.1;1(1),3(10),18(1),19(1);2;GPS;Misc;ITA;45.47;9.20;0;0;Trimble 5700;none;B;N;5000;Galileo Systems
STR;PADO0;Padova;RTCM 2.1;1(1),3(10),16(30),18(1),19(1),59;2;GPS;EUREF;ITA;45.41;11.90;0;0;none;none;B;N;3600;none
STR;ROMA0;Roma;RTCM 2.1;1(1),3(10),18(1),19(1),59;2;GPS;Misc;ITA;42.0;12.5;0;0;Trimble 4000;none;B;N;3400;Uni Roma
STR;ROMG0;Roma;RTCM2.1;1(1),3(10),18(1),19(1);2;GPS;Misc;ITA;42.0;12.5;0;0;Trimble 5700;none;B;N;5000;Galileo Sistemi
STR;TORI0;Torino;RTCM 2.1;1(1),3(60),16(30),18(1),19(1);2;GPS;Misc;ITA;45.06;7.66;0;0;none;none;B;N;12000;none
STR;WARS0;Warsaw;RTCM 2.2;1(1),3(60),18(1),19(1),22(60),31(1);2;GPS+GLO;EUREF;POL;52.10;21.03;0;0;none;none;B;N;1200;none
STR;ZIMM0;Zimmerwald;RTCM 2.1;1(1),18(1),19(1),22,23,24;2;GPS;Misc;SUI;46.88;7.47;0;1;GPSNet V1.9;none;B;N;3600;VRS
STR;ERFU0;Erfurt;RAW;(1);2;GPS;SAPOS-THR;GER;51.01;11.03;0;0;Trimble 4000SSI;none;B;N;5000;none
STR;BERL0;Berlin;RTCM AdV;20(1),21(1);2;GPS;SAPOS-BLN;GER;52.52;13.38;0;none;none;none;B;N;500;none
STR;BERJ0;Berlin;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;52.52;13.38;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;DREJ1;Dresden;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;51.05;13.73;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;EISE0;Eisenach;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;50.97;10.32;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;ERLA0;Erlangen;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;49.58;11.00;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;FLEN0;Flensburg;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;54.77;9.43;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;FREI0;Freiburg;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;47.98;7.85;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;HAMB0;Hamburg;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;53.57;10.03;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;HANN0;Hannover;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;52.37;9.73;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;KOEL0;Koeln;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;50.93;6.95;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;LEIJ1;Leipzig;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;51.33;12.37;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;MAGD0;Magdeburg;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;52.12;11.63;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;MUEN0;Muenchen;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;48.13;11.57;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;MUWF0;Muenster-WF;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;51.95;7.62;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;PASS0;Passau;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;48.57;13.45;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;ROST0;Rostock;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;54.08;12.12;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;SARB0;Saarbruecken;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;49.23;6.98;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;ULM00;Ulm;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;48.38;9.98;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;WILH0;Wilhelmshaven;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;53.52;8.10;0;1;GPSNet V1.9;none;B;N;560;VRS
STR;SBCC0;Mission-Viejo;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.55;242.34;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;SACY0;Santa-Ana;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.74;242.10;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;WHYT0;Lake-Forest;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.67;242.36;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;MJPK0;Santa-Ana-Mountains;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.71;242.45;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;CAT20;Avalon;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.31;241.67;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;SCMS0;San-Clemente;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.44;242.37;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;FVPK0;Costa-Mesa;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.66;242.06;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;OEOC0;Silverado;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.77;242.26;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;TRAK0;Irvine;RTCM 2.1;3(15),18(1),19(1),22(15),59(1);2;GPS;SCIGN-OCRT;USA-CA;33.62;242.20;0;0;Ashtech Z-XII3;none;B;N;4000;none
STR;SBCC1;Mission-Viejo;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.55;242.34;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;SACY1;Santa-Ana;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.74;242.10;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;WHYT1;Lake-Forest;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.67;242.36;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;MJPK1;Santa-Ana-Mountains;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.71;242.45;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;CAT21;Avalon;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.31;241.67;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;SCMS1;San-Clemente;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.44;242.37;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;FVPK1;Costa-Mesa;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.66;242.06;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;OEOC1;Silverado;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.77;242.26;0;0;Ashtech Z-XII3;none;B;N;7300;none
STR;TRAK1;Irvine;RAW;MBEN(1);2;GPS;SCIGN-OCRT;USA-CA;33.62;242.20;0;0;Ashtech Z-XII3;none;B;N;5000;none
ENDSOURCETABLE
```

Appendix B Format Specifications

The following is additional information on format specifications as introduced through parameter <format> and <format-details> in the source-table (see Source-table section). Currently a limited number of raw GNSS data formats and RTCM formats for differential GNSS data are considered.

Each source-table record of type “STR” contains keywords to describe the data format and format-details provided in a specific data stream (see Source-table section). The following sections define keywords and format details corresponding to these keywords.

Introducing provider-dependent other format keywords and format details is allowed.

1. RTCM Formats

Differential GNSS corrections are often available in RTCM format. RTCM stands for Radio Technical Commission for Maritime Services. Working Committees called RTCM Special Committees are chartered to address in depth radiocommunication and radionavigation areas of concern. The RTCM Special Committee 104 on Global Navigation Satellite Navigation Systems Services is responsible for defining RTCM recommended Standards for Differential GNSS. Documentations are available from http://www.rtcm.org. The following is mandatory information to be used within Ntrip that is not part of the RTCM standards.

Table 1 defines RTCM keywords to be used as <format> parameter in source-table records of type STR and the minimum message set corresponding to these keywords. A provider may add more messages. The update rate is not defined.

Table 1: RTCM Minimum Message Set

Format / Keyword	Comment	Format Details / Messages
RTCM 2.0	DGPS	1, 3
RTCM 2.1	RTK	3, 18, 19
RTCM 2.3	RTK	18, 19, 23, 24
RTCM SAPOS	SAPOS Standard	20, 21, 23, 24, 59

2. Other Formats

Most GNSS equipment vendors have their own proprietary formats. Such “raw” GNSS data may be disseminated using Ntrip. Table 2 shows examples for source-table parameters <format> and <format-details> when streaming proprietary formats.

The “Compact Measurement Record” (CMR) format is a publicly documented format accepted by several GNSS equipment vendors. The CMR format is among those providing some interoperability across vendors.

Table 2: Other data formats

Format / Keyword	Comment	Format Details
RAW	Raw observation data from a GNSS receiver	Examples: - RT17 (for Trimble receiver) - MBEN (for Ashtech receiver) - COMPACT (for Topcon/Javad Receiver)
CMR	CMR Standard	none

Appendix C Client Example Source Code

The following is source code of a Linux/UNIX NtripClient program written under GNU General Public License in C programming language. The program reads data from an NtripCaster and writes on standard output for further redirection of data to a file or COM-port. Please note that this program version does not handle potentially occurring interruptions of communication or network congestion situations.

```
/*
  Easy example NTRIP client for Linux/Unix.
  Copyright (C) 2003 by Dirk Stoecker <stoecker@epost.de>

  This program is free software; you can redistribute it and/or modify
  it under the terms of the GNU General Public License as published by
  the Free Software Foundation; either version 2 of the License, or
  (at your option) any later version.

  This program is distributed in the hope that it will be useful,
  but WITHOUT ANY WARRANTY; without even the implied warranty of
  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
  GNU General Public License for more details.

  You should have received a copy of the GNU General Public License
  along with this program; if not, write to the Free Software
  Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA  02111-1307  USA
  or read http://www.gnu.org/licenses/gpl.txt
*/

/* Version history
  Please always keep revision history and the two related strings up to date!
  1.1   2003-02-24 stoecker    initial version
  1.2   2003-02-25 stoecker    fixed agent string
*/

#include <getopt.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <errno.h>
#include <string.h>
#include <netdb.h>
#include <sys/types.h>
#include <netinet/in.h>
#include <sys/socket.h>

/* The string, which is send as agent in HTTP request */
#define AGENTSTRING "NTRIP LinuxClient"

#define MAXDATASIZE 1000 /* max number of bytes we can get at once */
char buf[MAXDATASIZE];

/* CVS revision and version */
static char revisionstr[] = "$Revision: 1.2 $";
static char datestr[]     = "$Date: 2003/03/25 13:00:00 $";

struct Args
{
  char *server;
  int   port;
  char *user;
  char *password;
  char *data;
};

/* option parsing */
#ifdef NO_LONG_OPTS
#define LONG_OPT(a)
#else
#define LONG_OPT(a) a
static struct option opts[] = {
{ "data",       required_argument, 0, 'd'},
{ "server",     required_argument, 0, 's'},
{ "password",   required_argument, 0, 'p'},
{ "port",       required_argument, 0, 'r'},
{ "user",       required_argument, 0, 'u'},
{ "help",       no_argument,       0, 'h'},
{0,0,0,0}};
#endif
#define ARGOPT "d:hp:r:s:u:"

static int getargs(int argc, char **argv, struct Args *args)
{
  int res = 1;
  int getoptr;
  char *a;
  int i = 0, help = 0;
  char *t;

  args->server = "129.217.182.51";
  args->port = 80;
  args->user = "";
  args->password = "";
  args->data = 0;
  help = 0;

  do
  {
#ifdef NO_LONG_OPTS
    switch((getoptr = getopt(argc, argv, ARGOPT)))
#else
    switch((getoptr = getopt_long(argc, argv, ARGOPT, opts, 0)))
#endif
    {
    case 's': args->server = optarg; break;
    case 'u': args->user = optarg; break;
    case 'p': args->password = optarg; break;
    case 'd': args->data = optarg; break;
    case 'h': help=1; break;
    case 'r':
      args->port = strtoul(optarg, &t, 10);
      if((t && *t) || args->port < 1 || args->port > 65535)
        res = 0;
      break;
    case -1: break;
    }
  } while(getoptr != -1 || !res);

  for(a = revisionstr+11; *a && *a != ' '; ++a)
    revisionstr[i++] = *a;
  revisionstr[i] = 0;
  datestr[0] = datestr[7];
  datestr[1] = datestr[8];
  datestr[2] = datestr[9];
  datestr[3] = datestr[10];
  datestr[5] = datestr[12];
  datestr[6] = datestr[13];
  datestr[8] = datestr[15];
  datestr[9] = datestr[16];
  datestr[4] = datestr[7] = '-';
  datestr[10] = 0;

  if(!res || help)
  {
    fprintf(stderr, "Version %s (%s) GPL\nUsage: %s -s server -u user ...\n"
    " -d " LONG_OPT("--data     ") "the requested data set\n"
    " -s " LONG_OPT("--server   ") "the server name or address\n"
    " -p " LONG_OPT("--password ") "the login password\n"
    " -r " LONG_OPT("--port     ") "the server port number (default 80)\n"
    " -u " LONG_OPT("--user     ") "the user name\n"
    , revisionstr, datestr, argv[0]);
    exit(1);
  }
  return res;
}

static char encodingTable [64] = {
  'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P',
  'Q','R','S','T','U','V','W','X','Y','Z','a','b','c','d','e','f',
  'g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v',
  'w','x','y','z','0','1','2','3','4','5','6','7','8','9','+','/'
};
```

```
/* does not buffer overrun, but breaks directly after an error */
/* returns the number of required bytes */
static int encode(char *buf, int size, char *user, char *pwd)
{
  unsigned char inbuf[3];
  char *out = buf;
  int i, sep = 0, fill = 0, bytes = 0;

  while(*user || *pwd)
  {
    i = 0;
    while(i < 3 && *user) inbuf[i++] = *(user++);
    if(i < 3 && !sep)    {inbuf[i++] = ':'; ++sep; }
    while(i < 3 && *pwd)  inbuf[i++] = *(pwd++);
    while(i < 3)         {inbuf[i++] = 0; ++fill; }
    if(out-buf < size-1)
      *(out++) = encodingTable[(inbuf [0] & 0xFC) >> 2];
    if(out-buf < size-1)
      *(out++) = encodingTable[((inbuf [0] & 0x03) << 4)
               | ((inbuf [1] & 0xF0) >> 4)];
    if(out-buf < size-1)
    {
      if(fill == 2)
        *(out++) = '=';
      else
        *(out++) = encodingTable[((inbuf [1] & 0x0F) << 2)
               | ((inbuf [2] & 0xC0) >> 6)];
    }
    if(out-buf < size-1)
    {
      if(fill >= 1)
        *(out++) = '=';
      else
        *(out++) = encodingTable[inbuf [2] & 0x3F];
    }
    bytes += 4;
  }
  if(out-buf < size)
    *out = 0;
  return bytes;
}

int main(int argc, char **argv)
{
  struct Args args;

  if(getargs(argc, argv, &args))
  {
    int i, sockfd, numbytes;
    char buf[MAXDATASIZE];
    struct hostent *he;
    struct sockaddr_in their_addr; /* connector's address information */

    if(!(he=gethostbyname(args.server)))
    {
      perror("gethostbyname");
      exit(1);
    }
    if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
    {
      perror("socket");
      exit(1);
    }
    their_addr.sin_family = AF_INET;    /* host byte order */
    their_addr.sin_port = htons(args.port);  /* short, network byte order */
    their_addr.sin_addr = *((struct in_addr *)he->h_addr);
    memset(&(their_addr.sin_zero), '\0', 8);
    if(connect(sockfd, (struct sockaddr *)&their_addr,
    sizeof(struct sockaddr)) == -1)
    {
      perror("connect");
      exit(1);
    }

    if(!args.data)
    {
      i = snprintf(buf, MAXDATASIZE,
      "GET / HTTP/1.0\r\n"
      "User-Agent: %s/%s\r\n"
//      "Accept: */*\r\n"
//      "Connection: close\r\n"
      "\r\n"
      , AGENTSTRING, revisionstr);
    }
    else
    {
      i=snprintf(buf, MAXDATASIZE-40, /* leave some space for login */
      "GET /%s HTTP/1.0\r\n"
      "User-Agent: %s/%s\r\n"
//      "Accept: */*\r\n"
//      "Connection: close\r\n"
      "Authorization: Basic "
      , args.data, AGENTSTRING, revisionstr);
      if(i > MAXDATASIZE-40 && i < 0) /* second check for old glibc */
      {
        fprintf(stderr, "Requested data too long\n");
        exit(1);
      }
      i += encode(buf+i, MAXDATASIZE-i-5, args.user, args.password);
      if(i > MAXDATASIZE-5)
      {
        fprintf(stderr, "Username and/or password too long\n");
        exit(1);
      }
      snprintf(buf+i, 5, "\r\n\r\n");
      i += 5;
    }
    if(send(sockfd, buf, i, 0) != i)
    {
      perror("send");
      exit(1);
    }
    if(args.data)
    {
      int k = 0;
      while((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) != -1)
      {
        if(!k)
        {
          if(numbytes != 12 || strncmp("ICY 200 OK\r\n", buf, 12))
          {
            fprintf(stderr, "Could not get the requested data\n");
            exit(1);
          }
          ++k;
        }
        else
          fwrite(buf, numbytes, 1, stdout);
      }
    }
    else
    {
      while((numbytes=recv(sockfd, buf, MAXDATASIZE-1, 0)) != -1)
      {
        fwrite(buf, numbytes, 1, stdout);
        if(!strncmp("ENDSOURCETABLE\r\n", buf+numbytes-16, 16))
          break;
      }
    }

    close(sockfd);
  }
  return 0;
}
```

PART II Ntrip Example Implementation

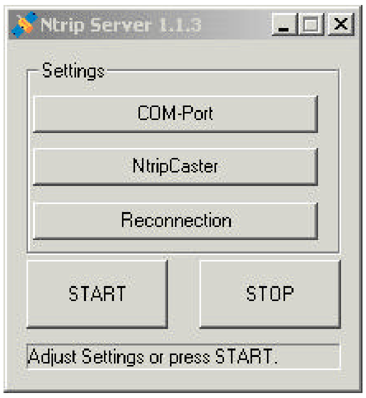
The System components described in Part I of this documentation are implemented as an example implementation under several operating systems designed with Microsoft Visual Studio C++ 6.0 including Microsoft Visual Studio 6.0 Service Pack 5 for Win32 and the gcc compiler for Linux (UNIX). A modified **Icecast** version 1.3.11 is used as the basic element: The NtripCaster. The other programs developed are called NtripServer (Windows 98/00/XP) and the “GNSS Internet Radio” (NtripClient, Windows 98/00/XP).

Note that this example implementation is not capable of handling Ntrip’s NMEA Request Messages as necessary for supporting the VRS concept. It is limited to disseminate data streams with <nmea> parameter set to “0” (see Source-table section in PART I).

1. NtripServer

The NtripServer (Windows 98/00/XP) is designed to be a server for a single NtripSource. Basically the NtripServer grabs a serial GNSS byte stream (COM-port) from a GNSS receiver at a reference station and sends it off over an Internet TCP connection to the NtripCaster. The start window is shown in Figure 4.

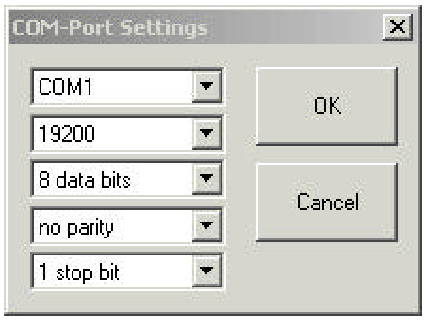
Mind that the NtripServer cannot handle a proxyserver. When using it in a proxy-protected Local Area Network (LAN), a TCP-relay has to be established connecting the proxyserver and the NtripCaster. Establishing the Internet connection for an NtripServer by using an Internet Service Provider (ISP) is an alternative.



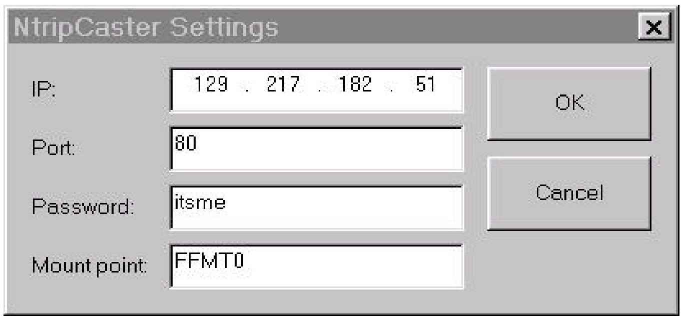
Adjust Settings or press START.

Fig. 4: NtripServer window

A server administrator needs to set all relevant parameters before the transmission will be started. The program includes setting windows to configure the serial byte stream (COM-port) from a GNSS receiver and the NtripCaster settings for the data output towards the NtripCaster (see Figure 5). Note that it is recommended to use a high baud rate (e.g. 19200) if supported by hardware and software. Server password and mountpoint are provided by the administrator of the NtripCaster.



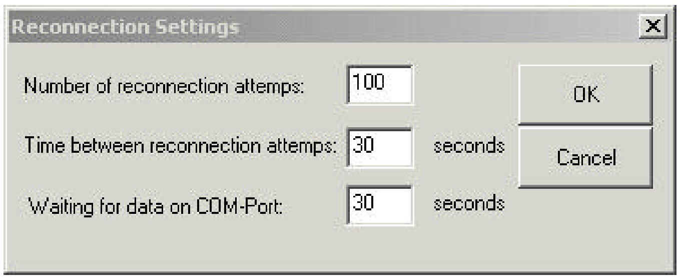
COM1
19200
8 data bits
no parity
1 stop bit



IP: 129 . 217 . 182 . 51
Port: 80
Password: itsme
Mount point: FFMT0

Fig. 5: Setting windows - press the button “NtripCaster” (right), press the button “COM-port” (left)

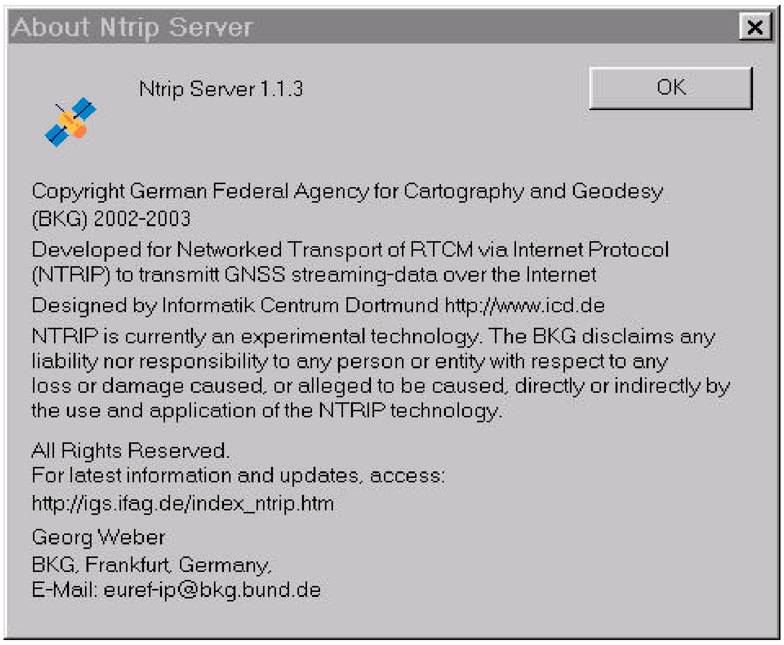
The program will automatically recognize NtripCaster failures (e.g. NtripCaster is down) und can be configured, as shown in Figure 6, to automatically reconnect a the lost NtripCaster.



Number of reconnection attemps: 100
Time between reconnection attemps: 30 seconds
Waiting for data on COM-Port: 30 seconds

Fig. 6: Reconnection window

This example server implementation is currently an experimental software. The BKG disclaims any liability nor responsibility to any person or entity with respect to any loss or damage caused, or alleged to be caused, directly or indirectly by the use and application of the Ntrip technology (see Figure 7).



Ntrip Server 1.1.3

Copyright German Federal Agency for Cartography and Geodesy (BKG) 2002-2003

Developed for Networked Transport of RTCM via Internet Protocol (NTRIP) to transmitt GNSS streaming-data over the Internet

Designed by Informatik Centrum Dortmund http://www.icd.de

NTRIP is currently an experimental technology. The BKG disclaims any liability nor responsibility to any person or entity with respect to any loss or damage caused, or alleged to be caused, directly or indirectly by the use and application of the NTRIP technology.

All Rights Reserved.
For latest information and updates, access:
http://igs.ifag.de/index_ntrip.htm

Georg Weber
BKG, Frankfurt, Germany,
E-Mail: euref-ip@bkg.bund.de

Fig 7: About Nrip Server & Copyright

2. NtripServerCMD

Like the NtripServer, the command line program NtripServerCMD (Windows 98/00/XP) is designed to be a server for a single NtripSource. Basically the NtripServerCMD grabs a GNSS byte stream via TCP/IP from an IP network address and port number and sends it off over an Internet TCP connection to the NtripCaster.

Note that the NtripServerCMD cannot handle a proxyserver. When using it in a proxy-protected Local Area Network (LAN), a TCP-relay has to be established connecting the proxyserver and the NtripCaster. Establishing the Internet connection for the NtripServerCMD program by using an ISP is an alternative.

The user has to call the program with the following arguments:

```
NtripServerCMD -q <source IP>:<source port>
               -c <caster IP>:<caster port>:<mountpoint>:<password>
               -r <attempts>:<time>
               -p <protocol file>
```

Explanations:

```
   -q <>:<> -c <>:<>:<> -r <>:<>  Obligatory arguments
    :<password>        Optional arguments
    <attempts>         Number of attempts for connection, '0' for infinite number of attempts
    <time>             Interval between connection attempts in seconds
    -p <protocol file> Optional arguments, specifies the name for protocol file,
                       no protocol is created without this argument
```

Use 'Ctrl + c' to quit the program during execution.
The argument '-h' or '/?' will display “help information” on the screen.

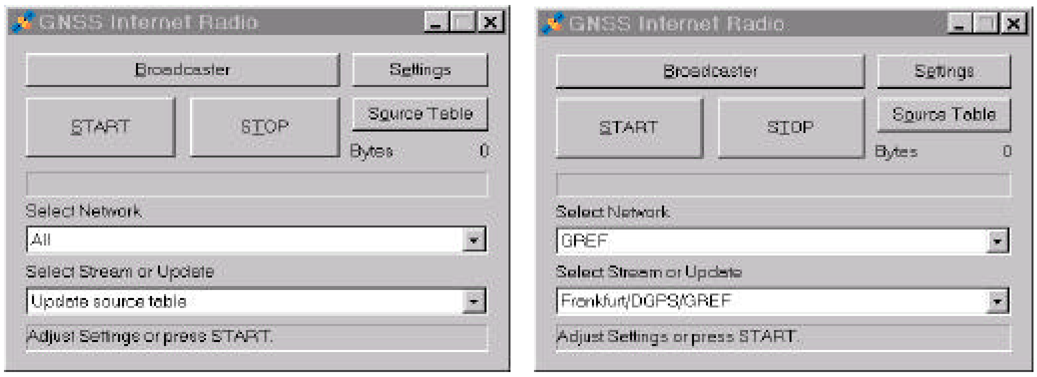
Example:

```
c:\NtripServerCMD -q 194.20.217.4:1024 -c 129.217.182.50:2101:/FFMT0:itsme -r 10:5 -p prot.txt
```

3. NtripClient (GNSS Internet Radio)

The NtripClient (named “GNSS Internet Radio”) is designed to run on a PC/Laptop (Windows 98/00/XP). It feeds a GNSS receiver, which is connected via serial port (COM-port) to the PC/Laptop. The correction data is received from an NtripCaster. The program will handle the HTTP communication and transfer the received GNSS data e.g. to the serial port. After setting up the session parameters and pressing the start button, the software connects to the desired NtripCaster and writes the received GNSS data stream e.g. to the serial port until the session is stopped.

The basic start window is shown in Figure 8. The program keeps a local file of the last session parameters (IP/port, username:password, COM-port settings, source-table etc.). As for first time uses no recorded session parameters are available, they have to be set by the user. The source-table will be automatically received from the NtripCaster when sending the first request message.

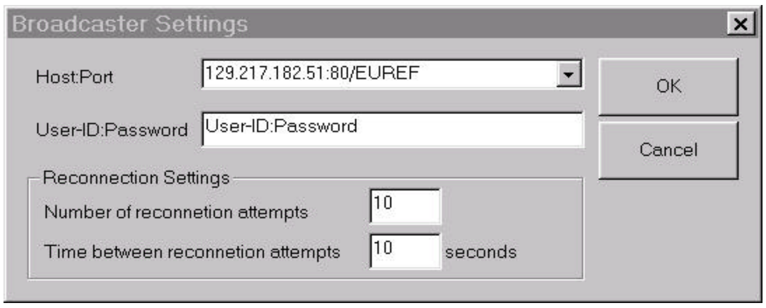


Select Network: All
Select Stream or Update: Update source table
Adjust Settings or press START.

Select Network: GREF
Select Stream or Update: Frankfurt/DGPS/GREF
Adjust Settings or press START.

Fig. 8: GNSSInternetRadio window
first time use (left), a source-table is available (right)

Broadcaster IP and Port, user-ID and password as well as reconnection settings are entered through the “Broadcaster Settings” windows (see Figure 9). The administrator of the NtripCaster provides the necessary user-ID and the password. It is generally associated to the usage of one specific data stream, but access to networks of data streams or even all data streams available from an NtripCaster can also be allowed through one single password.

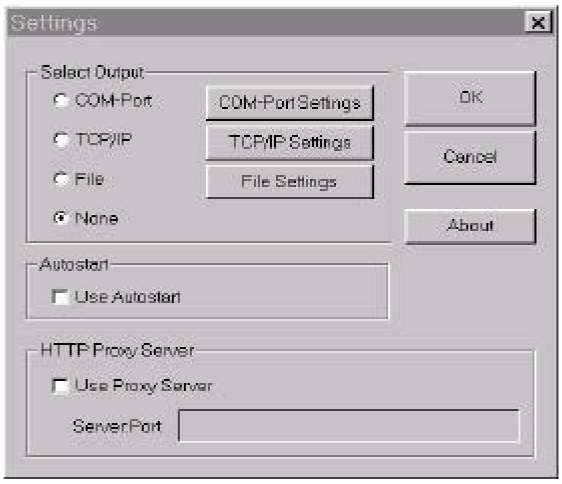


Host:Port 129.217.182.51:80/EUREF
User-ID:Password User-ID:Password
Reconnection Settings
Number of reconnetion attempts 10
Time between reconnetion attempts 10 seconds

Fig. 9: Broadcaster Settings window

The “Settings” window enables the user to select an output option. The options available are COM-port, TCP/IP, File, or None (see Figure 10). When using a proxyserver, its IP and port number can be introduced as well in this window.

Although the "GNSSInternetRadio" can be used within a proxy-protected Local Area Network (LAN) , users may still experience a blocked communication. This can especially happen if a combination of firewall, proxyserver, virus scanner etc. is in front. A work-around can be either to establish a TCP-relay (proxyserver - NtripCaster, to be arranged by a LAN administrator) or to access the Internet via an ISP.



Select Output

- COM-Port
- TCP/IP
- File
- [✓ None]

Autostart

- Use Autostart

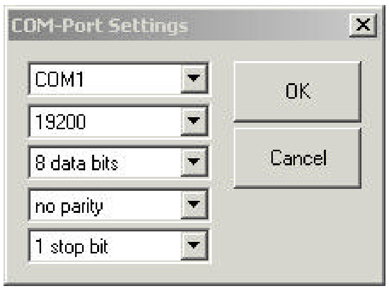
HTTP Proxy Server

- Use Proxy Server

Server:Port

Fig. 10: Settings window

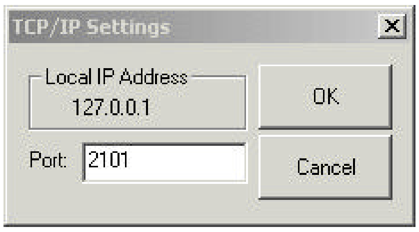
Details defining the COM-port communication are to be entered through the “COM-port Settings” window (see Figure 11). The usage of a high baud-rate is recommended (e.g. 19200) if supported by hardware and software.



COM1
19200
8 data bits
no parity
1 stop bit

Fig. 11: COM-port Settings window

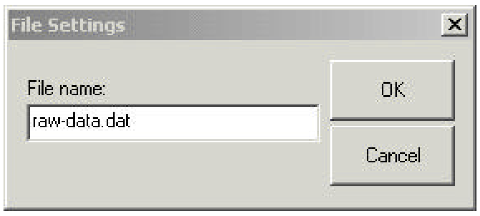
A port number can be defined through the “TCP/IP Settings” window when choosing the TCP/IP output option (see Figure 12). Note that other programs running on the same host as the “GNSS Internet Radio” can access the data stream via the Local IP Address 127.0.0.1.



Local IP Address 127.0.0.1
Port: 2101

Fig. 12: TCP/IP Settings window

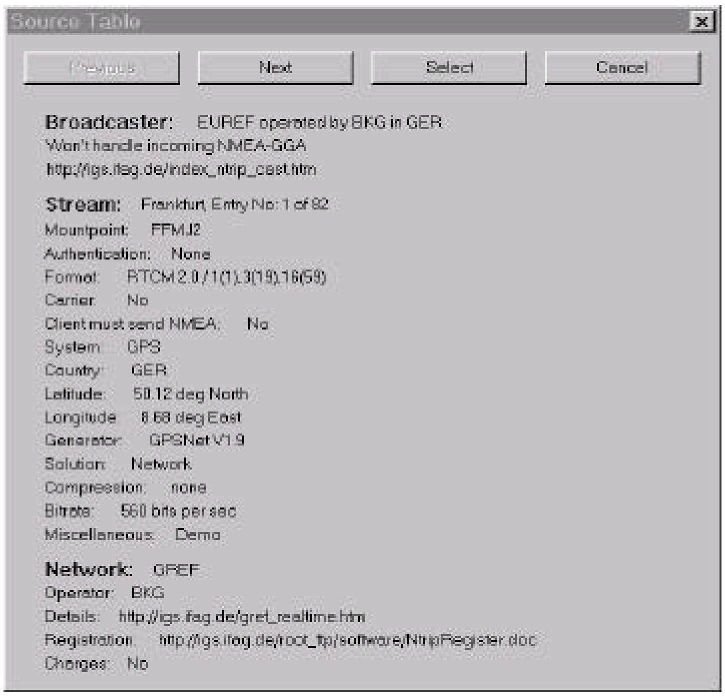
Received data may be written into a file. The “File Settings” window enables the user to define a file name (see Figure 13).



File name: raw-data.dat

Fig. 13: File Settings window

Details of data streams are available from the “Source Table” windows. You can find information here about the NtripCaster, the data format provided and the network operating the stream generator. Web-addresses or an e-mail address is given for background information or registration purposes.



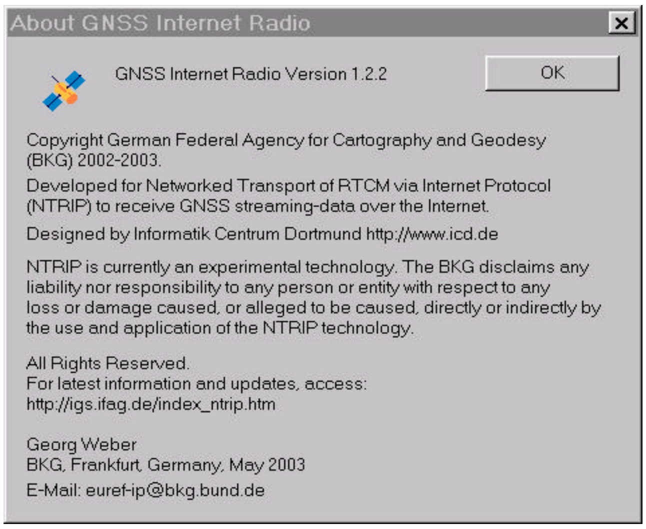
Broadcaster: EUREF operated by BKG in GER
Won't handle incoming NMEA-GGA
http://igs.ifag.de/index_ntrip_cast.htm

Stream: Frankfurt, Entry No: 1 of 82
Mountpoint: FFMJ2
Authentication: None
Format: RTCM 2.0 / 1(1),3(19),16(59)
Carrier: No
Client must send NMEA: No
System: GPS
Country: GER
Latitude: 50.12 deg North
Longitude: 8.68 deg East
Generator: GPSNet V1.9
Solution: Network
Compression: none
Bitrate: 560 bits per sec
Miscellaneous: Demo

Network: GREF
Operator: BKG
Details: http://igs.ifag.de/gref_realtime.htm
Registration: http://igs.ifag.de/root_ftp/software/NtripRegister.doc
Charges: No

Fig. 14: Source Table window

Note that this example NtripClient implementation is currently an experimental software. The BKG disclaims any liability nor responsibility to any person or entity with respect to any loss or damage caused, or alleged to be caused, directly or indirectly by the use and application of the Ntrip technology (see Figure 14).



GNSS Internet Radio Version 1.2.2

Copyright German Federal Agency for Cartography and Geodesy (BKG) 2002-2003.

Developed for Networked Transport of RTCM via Internet Protocol (NTRIP) to receive GNSS streaming-data over the Internet.

Designed by Informatik Centrum Dortmund http://www.icd.de

NTRIP is currently an experimental technology. The BKG disclaims any liability nor responsibility to any person or entity with respect to any loss or damage caused, or alleged to be caused, directly or indirectly by the use and application of the NTRIP technology.

All Rights Reserved.
For latest information and updates, access:
http://igs.ifag.de/index_ntrip.htm

Georg Weber
BKG, Frankfurt, Germany, May 2003
E-Mail: euref-ip@bkg.bund.de

Fig. 15: About GNSS Internet Radio & Copyright

4. NtripCaster

The NtripCaster as the central software component receives data streams from NtripServers as generated by NtripSources. The NtripCaster administrator is responsible for handling mountpoints for NtripSources, passwords, NtripClient access-rights, billing, statistics, etc.

The NtripCaster is based on the **Icecast** software, developed under GNU General Public License. The purpose of Icecast is to duplicate source data for up to thousand or more simultaneously connected clients accessing up to a few hundred different sources. Network connections between source, Icecast, and clients are based on HTTP/TCP/IP. The Icecast server was originally designed to stream MP3-coded data with bit rates of 32 kbit/s up to 128kbit/s. The NtripCaster is a re-designed Icecast for

- DGNSS corrections (about 0.5 kbit/s per stream) or
- RTK corrections (about 5 kbit/s per stream) or
- Raw GNSS receiver data (about 5 kbit/s per stream).

The NtripCaster does not alter the data.

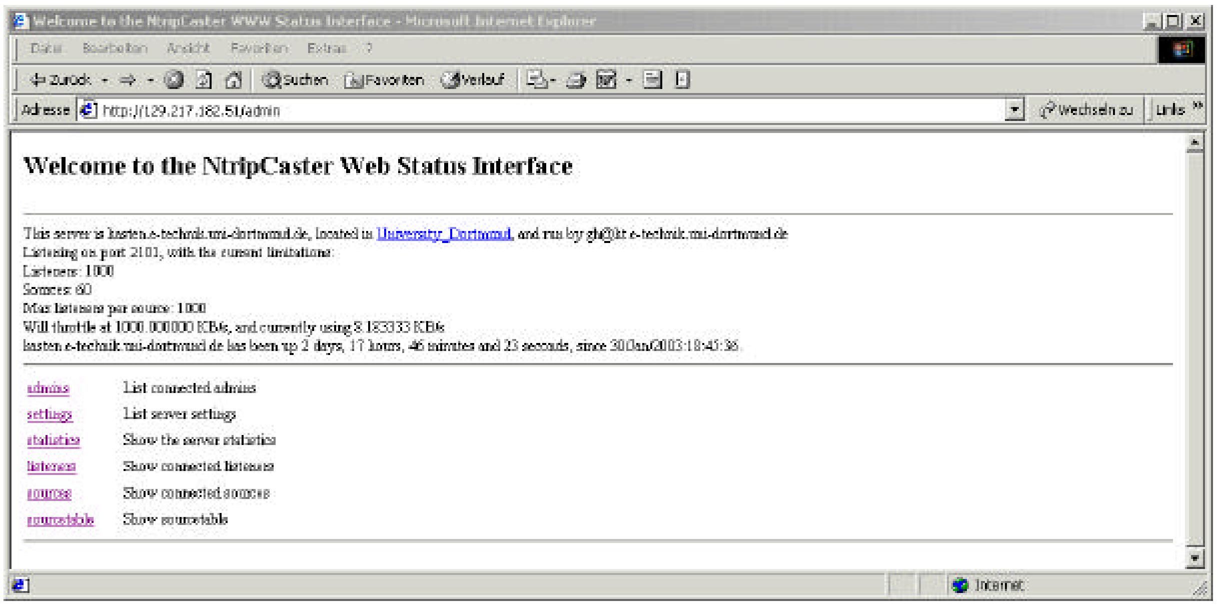
The NtripCaster can be accessed by a remote administrator using the same port (listening port) as NtripServers and NtripClients. This feature is necessary to run the NtripCaster in a professional manner, especially in an international context like EUREF. The NtripCaster provides an administration console restricted to exclusive persons [admin password], which can be accessed by Telnet. In addition, a status inquiry is possible by using a web browser. This enables to control a remote NtripCaster independent from its physical location. An NtripCaster hosted at a professional Internet Service Provider might be administrated by a person located at a completely different place. The administration functionality allows to organize the available GNSS data streams, set aliases, allow or deny clients (e.g. only accept clients from a specific country), kicking out connections to sources, clients or admins, view real-time statistics (all clients, sources) etc. while operating the NtripCaster.

To limit network costs, which are usually calculated based on the transferred data volume, the administrator can restrict the number N of simultaneous connections (e.g. N_{max}=1000). There is also a built-in bandwidth-measuring tool inside the caster. When the NtripCaster is using more than a certain bandwidth (e.g. 1 MB/s), no further clients will be accepted until the bandwidth falls below the specified limit. The NtripCaster is able to connect to GNSS data streams on a different NtripCaster and can present itself as an NtripServer. This allows e.g. the installation of a European network of NtripCasters. Therefore, Italian GNSS data streams could be made accessible via a German server, or German GNSS data streams can be made accessible via an Italian server.

Very important are the NtripCaster configuration files rtcmcast.conf and source-table.dat, which are used to set up the behavior of the HTTP-server. The administrator is able to set up the number of clients, sources, overall throughput, password for sources, password for admins, access, main behavior, and more. The *.conf and *.dat files can be edited on the local administrator PC (Windows or Unix) and uploaded to the NtripCaster via an FTP connection.

4.1. Status Information

Authorized administrators (name and password) can receive status information via a protected HTTP session (e.g. Internet Explorer). The NtripCaster Web Status Interface is shown in Figure 15.


Welcome to the NtripCaster WWW Status Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://129.217.182.51/admin
Wechseln zu Links
Internet

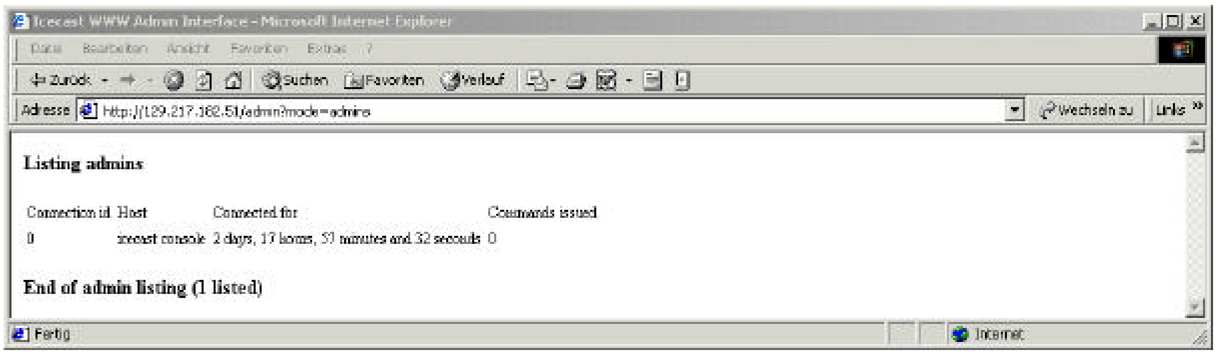
Welcome to the NtripCaster Web Status Interface

This server is kasten.e-technik.uni-dortmund.de, located in University_Dortmund, and run by gh@kt.e-technik.uni-dortmund.de
Listening on port 2101, with the current limitations:
Listeners: 1000
Sources: 60
Max listeners per source: 1000
Will throttle at 1000.000000 KB/s, and currently using 8.183333 KB/s
kasten.e-technik.uni-dortmund.de has been up 2 days, 17 hours, 46 minutes and 23 seconds, since 30/Jan/2003:18:43:36

admins	List connected admins
settings	List server settings
statistics	Show the server statistics
listeners	Show connected listeners
sources	Show connected sources
sourcetable	Show sourcetable

Fig. 16: NtripCaster Web Status Interface

More than one system administrator may work at the same time on the NtripCaster. Information on the administrators simultaneously logged into the system is available as shown in Figure 16.


Icecast WWW Admin Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://129.217.182.51/admin?mode=admins
Wechseln zu Links
Fertig
Internet

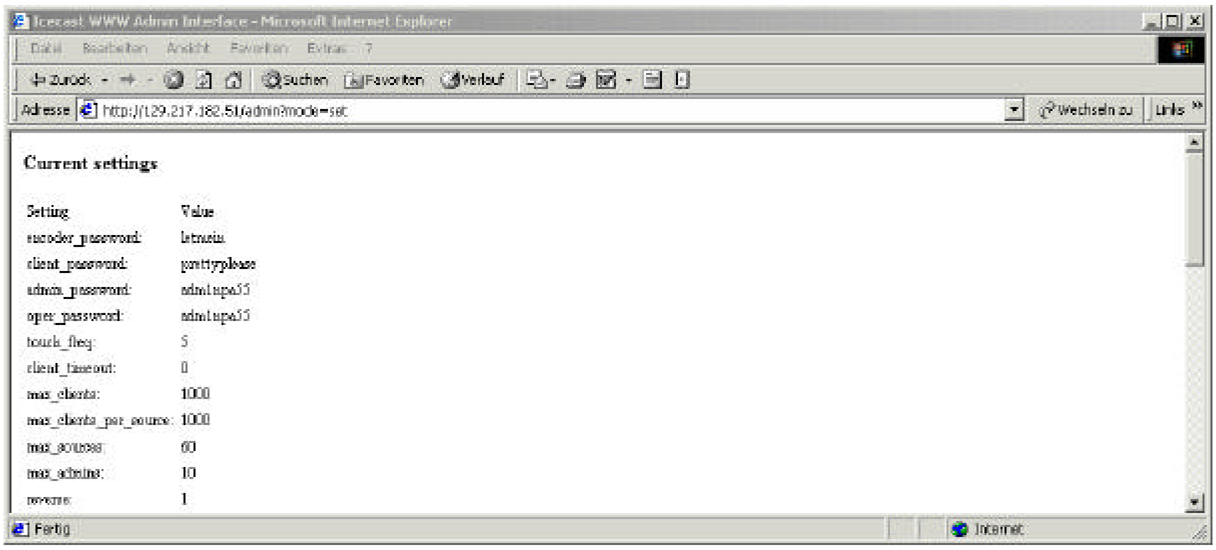
Listing admins

Connection id	Host	Connected for	Commands issued
0	icecast console	2 days, 17 hours, 57 minutes and 32 seconds	0

End of admin listing (1 listed)

Fig. 17: NtripCaster Web Status: Admins

As shown in Figure 17, the administrator can check current setting like encoder_password, client_password, admin_password, max_clients, max_clients_per_source, max_sources, max_admins, throttle (how much bandwidth the server is allowed to use before client and source access is denied), etc.


Icecast WWW Admin Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Suchen
Favoriten
Verlauf
Adresse
Wechseln zu
Links
Fertig
Internet

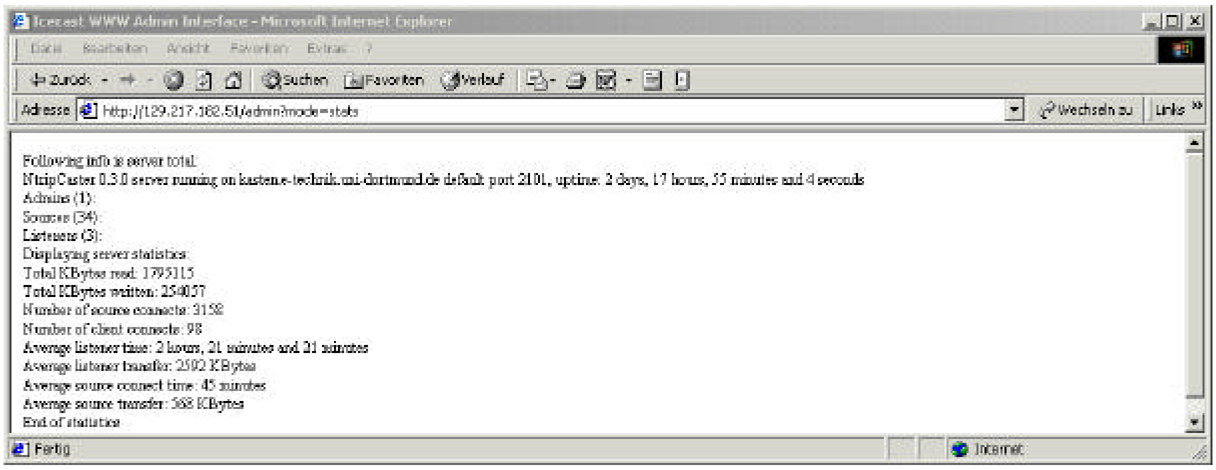
http://129.217.182.51/admin?mode=set

Current settings

Setting	Value
encoder_password:	letmein
client_password:	prettyplease
admin_password:	admin1pa55
oper_password:	admin1pa55
touch_freq:	5
client_timeout:	0
max_clients:	1000
max_clients_per_source:	1000
max_sources:	60
max_admins:	10
reverse:	1

Fig. 18: NtripCaster Web Status: Settings

Statistic information can be of interest for the administrator when operating the NtripCaster. Mean values characterizing the number of users, the number of data streams, the bandwidth in use and other important parameters are available (see Figure 18).

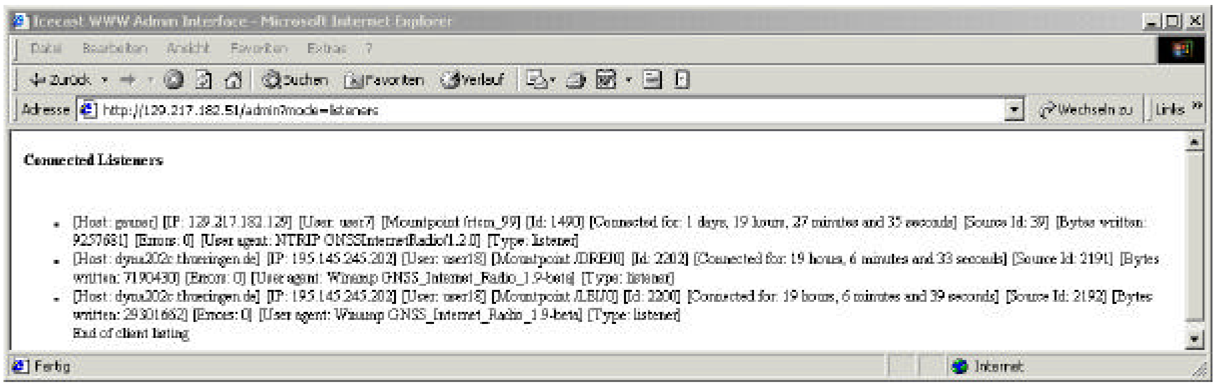

Icecast WWW Admin Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück
Suchen
Favoriten
Verlauf
Adresse
Wechseln zu
Links
Fertig
Internet

http://129.217.182.51/admin?mode=stats

```
Following info is server total:
NtripCaster 0.3.0 server running on kasten.e-technik.uni-dortmund.de default port 2101, uptime: 2 days, 17 hours, 55 minutes and 4 seconds
Admins (1):
Sources (34):
Listeners (3):
Displaying server statistics:
Total KBytes read: 1795115
Total KBytes written: 254057
Number of source connects: 3152
Number of client connects: 98
Average listener time: 2 hours, 21 minutes and 21 minutes
Average listener transfer: 2592 KBytes
Average source connect time: 45 minutes
Average source transfer: 568 KBytes
End of statistics
```

Fig. 19: NtripCaster Web Status: Statistics

The system administrator has real-time control over client parameters like listen IP, username, chosen source/mountpoint, connection time, bytes written, errors und used NtripClient (Fig. 19). This information is kept in log files for statistical or billing purposes. NtripClient usernames are only logged and displayed for password-protected mountpoints. For open mountpoints the username "null" is displayed/recorded. Unwanted users can be identified in real-time and kicked out by the administrator using the admin console command window.

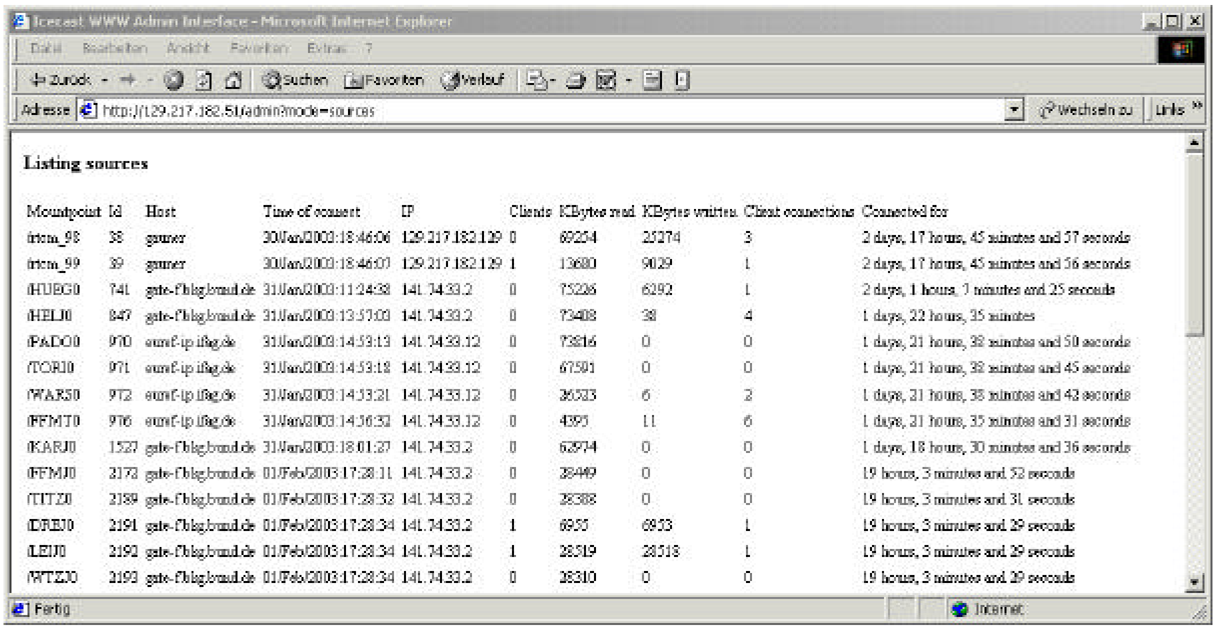

Icecast WWW Admin Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://129.217.182.51/admin?mode=listeners
Wechseln zu Links
Fertig
Internet

Connected Listeners

- [Host: gruner] [IP: 129.217.182.129] [User: user7] [Mountpoint /rtcm_99] [Id: 1490] [Connected for: 1 days, 19 hours, 27 minutes and 35 seconds] [Source Id: 39] [Bytes written: 9257681] [Errors: 0] [User agent: NTRIP GNSSInternetRadio/1.2.0] [Type: listener]
- [Host: dyna202c.thueringen.de] [IP: 195.145.245.202] [User: user18] [Mountpoint /DREJ0] [Id: 2202] [Connected for: 19 hours, 6 minutes and 33 seconds] [Source Id: 2191] [Bytes written: 7190430] [Errors: 0] [User agent: Winamp GNSS_Internet_Radio_1.9-beta] [Type: listener]
- [Host: dyna202c.thueringen.de] [IP: 195.145.245.202] [User: user18] [Mountpoint /LEIJ0] [Id: 2200] [Connected for: 19 hours, 6 minutes and 39 seconds] [Source Id: 2192] [Bytes written: 29301662] [Errors: 0] [User agent: Winamp GNSS_Internet_Radio_1.9-beta] [Type: listener]
End of client listing

Fig. 20: NtripCaster Web Status: Listener

The system administrator can also view all currently connected NtripSources (Fig. 20).

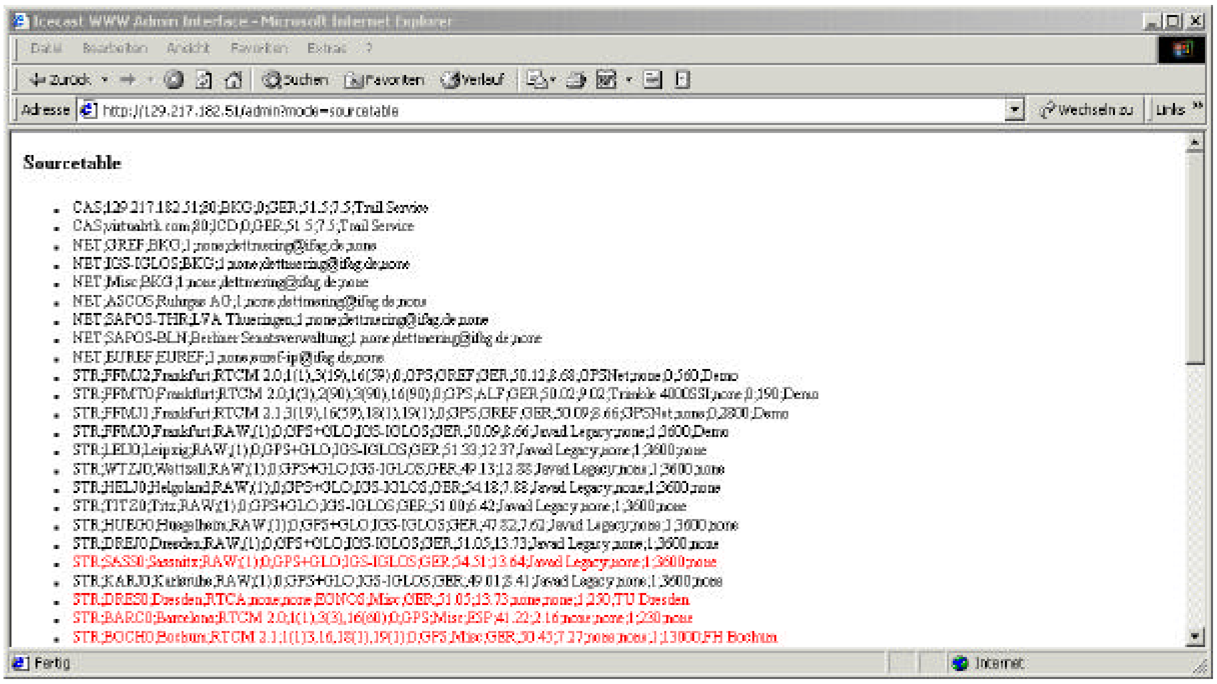

Icecast WWW Admin Interface - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Favoriten Extras ?
Zurück Suchen Favoriten Verlauf
Adresse http://129.217.182.51/admin?mode=sources
Wechseln zu Links
Fertig
Internet

Listing sources

Mountpoint	Id	Host	Time of connect	IP	Clients	KBytes read	KBytes written	Client connections	Connected for
/rtcm_98	38	gruner	30/Jan/2003:18:46:06	129.217.182.129	0	69254	25274	3	2 days, 17 hours, 45 minutes and 57 seconds
/rtcm_99	39	gruner	30/Jan/2003:18:46:07	129.217.182.129	1	13680	9029	1	2 days, 17 hours, 45 minutes and 56 seconds
/HUEG0	741	gate-f.bkg.bund.de	31/Jan/2003:11:24:38	141.74.33.2	0	75226	6292	1	2 days, 1 hours, 7 minutes and 25 seconds
/HELJ0	847	gate-f.bkg.bund.de	31/Jan/2003:13:57:03	141.74.33.2	0	73408	38	4	1 days, 22 hours, 35 minutes
/PADO0	970	eurof-ip.ifag.de	31/Jan/2003:14:53:13	141.74.33.12	0	73816	0	0	1 days, 21 hours, 38 minutes and 50 seconds
/TORI0	971	eurof-ip.ifag.de	31/Jan/2003:14:53:18	141.74.33.12	0	67591	0	0	1 days, 21 hours, 38 minutes and 45 seconds
/WARS0	972	eurof-ip.ifag.de	31/Jan/2003:14:53:21	141.74.33.12	0	26523	6	2	1 days, 21 hours, 38 minutes and 42 seconds
/FFMT0	976	eurof-ip.ifag.de	31/Jan/2003:14:56:32	141.74.33.12	0	4395	11	6	1 days, 21 hours, 35 minutes and 31 seconds
/KARJ0	1527	gate-f.bkg.bund.de	31/Jan/2003:18:01:27	141.74.33.2	0	62974	0	0	1 days, 18 hours, 30 minutes and 36 seconds
/FFMJ0	2172	gate-f.bkg.bund.de	01/Feb/2003:17:28:11	141.74.33.2	0	28449	0	0	19 hours, 3 minutes and 52 seconds
/TITZ0	2189	gate-f.bkg.bund.de	01/Feb/2003:17:28:32	141.74.33.2	0	28388	0	0	19 hours, 3 minutes and 31 seconds
/DREJ0	2191	gate-f.bkg.bund.de	01/Feb/2003:17:28:34	141.74.33.2	1	6955	6953	1	19 hours, 3 minutes and 29 seconds
/LEIJ0	2192	gate-f.bkg.bund.de	01/Feb/2003:17:28:34	141.74.33.2	1	28519	28518	1	19 hours, 3 minutes and 29 seconds
/WTZJ0	2193	gate-f.bkg.bund.de	01/Feb/2003:17:28:34	141.74.33.2	0	28310	0	0	19 hours, 3 minutes and 29 seconds

Fig. 21: NtripCaster Web Status: Sources

Red characters when displaying the source-table (see Figure 21) indicate currently unavailable data streams.



Sourcetable

- CAS;129.217.182.51;80;BKG;0;GER;51.5;7.5;Trail Service
- CAS;virtualrtk.com;80;ICD;0;GER;51.5;7.5;Trail Service
- NET;GREF;BKG;1;none;dettmering@ifag.de;none
- NET;IGS-IGLOS;BKG;1;none;dettmering@ifag.de;none
- NET;Misc;BKG;1;none;dettmering@ifag.de;none
- NET;ASCOS;Ruhrgas AG;1;none;dettmering@ifag.de;none
- NET;SAPOS-THR;LVA Thueringen;1;none;dettmering@ifag.de;none
- NET;SAPOS-BLN;Berliner Senatsverwaltung;1;none;dettmering@ifag.de;none
- NET;EUREF;EUREF;1;none;euref-ip@ifag.de;none
- STR;FFMJ2;Frankfurt;RTCM 2.0;1(1),3(19),16(59);0;GPS;GREF;GER;50.12;8.68;GPSNet;none;0;560;Demo
- STR;FFMT0;Frankfurt;RTCM 2.0;1(3),2(90),3(90),16(90);0;GPS;ALF;GER;50.02;9.02;Trimble 4000SSI;none;0;190;Demo
- STR;FFMJ1;Frankfurt;RTCM 2.1;3(19),16(59),18(1),19(1);0;GPS;GREF;GER;50.09;8.66;GPSNet;none;0;2800;Demo
- STR;FFMJ0;Frankfurt;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;50.09;8.66;Javad Legacy;none;1;3600;Demo
- STR;LEIJ0;Leipzig;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;51.33;12.37;Javad Legacy;none;1;3600;none
- STR;WTZJ0;Wettzell;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;49.13;12.88;Javad Legacy;none;1;3600;none
- STR;HELJ0;Helgoland;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;54.18;7.88;Javad Legacy;none;1;3600;none
- STR;TITZ0;Titz;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;51.00;6.42;Javad Legacy;none;1;3600;none
- STR;HUEG0;Huegelheim;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;47.82;7.62;Javad Legacy;none;1;3600;none
- STR;DREJ0;Dresden;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;51.05;13.73;Javad Legacy;none;1;3600;none
- STR;SASS0;Sassnitz;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;54.51;13.64;Javad Legacy;none;1;3600;none
- STR;KARJ0;Karlsruhe;RAW;(1);0;GPS+GLO;IGS-IGLOS;GER;49.01;8.41;Javad Legacy;none;1;3600;none
- STR;DRES0;Dresden;RTCA;none;none;BONOS;Misc;GER;51.05;13.73;none;none;1;230;TU Dresden
- STR;BARC0;Barcelona;RTCM 2.0;1(1),3(3),16(60);0;GPS;Misc;ESP;41.22;2.16;none;none;1;230;none
- STR;BOCH0;Bochum;RTCM 2.1;1(1)3,16,18(1),19(1);0;GPS;Misc;GER;50.45;7.27;none;none;1;13000;FH Bochum

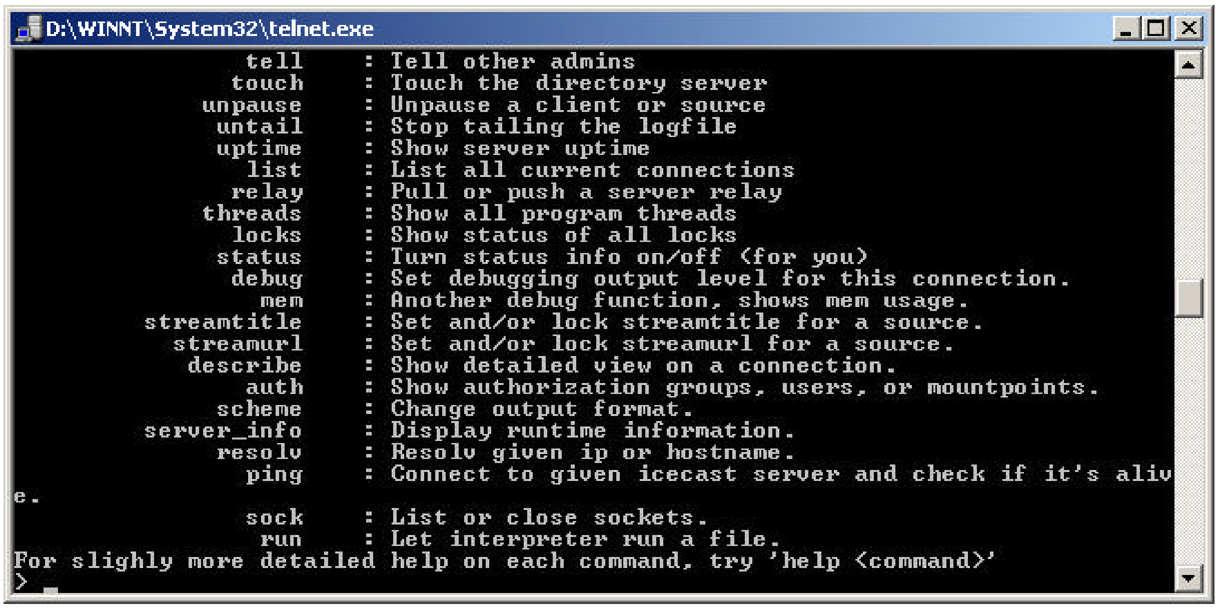
Fig. 22: NtripCaster Web Status: Sourcetable

4.2. Administration

The NtripCaster can be remote controlled via the admin console. To access the admin console, a password-protected telnet connection to the NtripCaster (e.g. through communications/listening port 80 or ports 80 and 2101) can be used. After connecting, no prompt will be displayed. Then 'ADMIN [admin password]' followed by two new lines (press enter twice) has to be entered. Substitute [admin password] by the admininistrator password set in ntripcast.conf or enter it as an NtripCaster command line parameter (see following example).

```
telnet euref-ip.ifag.de 80
ADMIN adm1npa55
Return
Return
```

The admin console command window is shown in Figure 22.



```
              tell     : Tell other admins
             touch     : Touch the directory server
           unpause     : Unpause a client or source
            untail     : Stop tailing the logfile
            uptime     : Show server uptime
              list     : List all current connections
             relay     : Pull or push a server relay
           threads     : Show all program threads
             locks     : Show status of all locks
            status     : Turn status info on/off (for you)
             debug     : Set debugging output level for this connection.
               mem     : Another debug function, shows mem usage.
       streamtitle     : Set and/or lock streamtitle for a source.
         streamurl     : Set and/or lock streamurl for a source.
          describe     : Show detailed view on a connection.
              auth     : Show authorization groups, users, or mountpoints.
            scheme     : Change output format.
       server_info     : Display runtime information.
            resolv     : Resolv given ip or hostname.
              ping     : Connect to given icecast server and check if it's aliv
e.
              sock     : List or close sockets.
               run     : Let interpreter run a file.
For slighly more detailed help on each command, try 'help <command>'
> _
```

Fig. 23. NtripCaster administration console

The admin console defines following basic commands [*Icecast 1.3.11,* Manual, http://www.icecast.org, 2002]:

alias

The alias command is used for two things. It can be used to add an alias for a local mountpoint so that a stream can be accessed from two mountpoints. Or it can be used to add an alias for a remote NtripCaster stream.

Syntax:

alias add <mountpoint> <newmountpoint>
alias del <mountpoint>
alias list

allow and deny

This adds a hostmask to an internal access lists (acl). It specifies that this hostmask should allow or deny access. A type for this acl can be specified, which can be either all, client, source or admin. It only has affect on the specified connection type.

Syntax:

allow|deny <client|source|admin|all> add <hostmask>
allow|deny <client|source|admin|all> del <hostmask>
allow|deny <client|source|admin|all> list

Examples:

allow client add *.se
allow all del *.netcom.com
deny admin add *.se
deny admin del ap.*.com
allow admin list

admins

The „admins“ command is used to list the admins connected to the server. It will display one admin connection per line, like:

Admin 341 <d66.ryd.student.liu.se> connected for 1 hours, 8 minutes and 14 seconds. 0 commands issued

A host-mask as an argument can be supplied, and only the admins who match the mask, will be shown.

Syntax:

admins [hostmask]

Example:

admins *.ifag.de

dump

This command can be used to dump a connected stream to a file on disk.

If the filename specified is „close“, then the file will be closed.

Syntax:

dump <source id> <filename|close>

Example:

dump 230 /tmp/stream.rpcm

help

Without arguments, the command „help“ prints a list of all commands and what they are used for. With a command as an argument, a slightly longer description of what the command does, is printed.

Syntax:

help [command]

Examples:

help kick
help

kick

This command can be used to get rid of some listeners, admins, or sources. A single connection may be kicked out as well as all connections of a certain type matching a hostmask.

Syntax:

kick <id>
kick <-acs> <hostmask>
kick <hostmask>

Examples:

kick 314
kick -a *.com
kick *.badsource.com

listeners

This command displays a list of all connected listeners. The output can be restricted to those listeners matching a specific hostmask.

Syntax:

listeners [hostmask]

Example:

listeners *.ifag.de

oper

Most commands on the admin console are restricted to NtripCaster operators. To obtain operator privileges, the „oper“ command with the NtripCaster operator password as argument can be used.

Syntax:

oper <operator password>

Example:

oper cooloperpass

pause

The data flow to a client or from a source can be cut with this command.

Syntax:

pause <client or source id>

Example:

pause 314

quit

The „quit“ command is used to leave the NtripCaster admin console. This can only be used, when the console is accessed through the remote interface (e.g. using telnet). An optional argument will be used as a signoff message, e.g. displayed to other connected admins.

Syntax:

quit [message]

Example:

quit Bye Bye

rehash

This command can be used after changing parameters in the NtripCaster configuration file (ntripcast.conf or source-table.dat). When used with an argument, it will understand this argument as a filename and parse this as a configuration file.

Syntax:

rehash [filename]

Example:

rehash /tmp/newrtcmast.conf

select

„select“ is useful when a listeners should be transferred from one stream to another. It can either transfer all clients connected to source A, to source B, or all clients connected to source A matching a pattern, to source B, or just one single client with a specified id, to a new source.

Syntax:

```
select <-a> <source source id> <target source id>
select <client id> <target source id>
select <source source id> <hostmask> <target source id>
```

Examples:

```
select 514 23
select -a 43 23
select 23 *.se 43
```

sources

„sources“ is used just like „listeners“ to view all the connected sources. It can be used with an optional argument to limit the list to only the sources matching the specified hostmask. It also accepts a large number of options. By default the output is defined by the variable default_sourceopts.

The options are as follows.

i - Connection id
s - Socket descriptor
t - Time of connect
p - Connection ip
h - Hostname
c - Connection state
y - Source type
c - Number of clients
d - Dumpfile
r - Mountpoint priority
M - Source mountpoint
R - Bytes read
W - Show bytes written
C - Total client connects
T - Time connected

Syntax:

```
sources [hostmask] [-options]
```

Example:

```
sources *.apan.com -aCW
```

shutdown

This command is used to shutdown the server. An optional argument can be introduced which will be used as a shutdown message broadcasted to all connected admins.

Syntax:

```
shutdown [message]
```

Example:

```
shutdown caster_goes_down
```

set

This command is used without arguments. „set“ will display a list of all NtripCaster variables and their current values. It can be used also to change any of those variables if the name is specified and the new value is entered as arguments. Typing „help settings“ will give a short description of all the settings, and a longer description for the settings is available here.

Syntax:

```
set
set <variable name>
set <variable name> <new value>
```

Examples:

```
set client_timeout
set max_clients 580
```

stats

The „stats“ command will print some information about the current number of connections, averages on client connections, bytes read and written, etc. It looks a lot like the output available from the statistics file. As an argument, „hourly“ or „daily“ can be put. Then the averages and totals for the last hour or day will be printed.

Syntax:

```
stats [hourly|daily]
```

Example:

```
stats hourly
```

tail

The „tail“ command can be issued by an administrator to display messages written to the logfile. It's similar to the unix tail command.

Syntax:

```
tail
```

Example:

```
tail
```

tell

An admin can send a message to all other connected admins using the „tell“ command. Any argument will be send to all admins.

Syntax:

```
tell <message>
```

Example:

```
tell new NtripSource
```

unpause

The „unpause“ command unfreezes a client or stream previously paused with the „pause“ command.

Syntax:

unpause <client or source id>

Example:

unpause 314

uptime

The „uptime“ command is similar to the Unix uptime command. It prints the number of days, hours, minutes and seconds the NtripCaster is up and running.

Syntax:

uptime

Example:

uptime

list

This command is used to list all connections, sources, admins, or clients.

Syntax:

list [hostmask]

Example:

list *.toomanyarguments.com

relay

To mount a remote stream on the NtripCaster, or to mount one of your local streams on a remote server, the „relay“ command should be used.

Syntax:

relay pull [-m <localmount>] <url>

Example:

relay pull ifag.de:2101/BUCU0

auth

This command is used to display authorization users, mounts and groups.

Syntax:

auth <groups|users|mounts>

Example:

auth groups
auth users

4.3. Hardware

Various Linux distributions and hardware platforms may be used to run the NtripCaster example implementation. The following configuration has been tested in January 2003 and may thus be understood as one possible example.

- Server PowerEdge 1650
- Basic Server Installation Packet, Linux Redhat 7.3
- Intel® Pentium® III Processor 1,40 GHz
- Second Processor, PIII P1650
- 8x DVD-ROM
- Rack Kit for 19" Racks
- Redundant power supply (2x)
- PCI Slots - PE 1650 Riser: 2x 64 Bit/66 MHz
- 1 GB PC133 ECC SDRAM (2x 512 MB)
- 2x18 GB SCSI Hard Disc, 10.000 RPM Ultra3/160
- PERC3/DC PCI RAID Controller 128 MB
- RAID Level 1
- 2x Intel® Gigabit Ethernetadapter Onboard (10/100/1000)
- ERA - Embedded Remote Access onboard integrated

Downloads

The BKG maintains a Web-side in order to keep the interested community up to date about Ntrip developments. This information is available through

- http://igs.ifag.de/index_ntrip.htm.

The latest version of this document as well as the programs which are part of the example implementation are available through

- NtripDocumentation (latest version of this document)
  http://igs.ifag.de/root_ftp/software/NtripDocumentation.zip
- NtripClient for Windows (GNSS Internet Radio)
  http://igs.ifag.de/root_ftp/software/NtripGNSSInternetRadioWindows.zip
- NtripClient for Windows CE (GNSS Internet Radio)
  http://igs.ifag.de/root_ftp/software/NtripGNSSInternetRadioWindowsCE.zip
- NtripClient for Linux, simple programming example
  http://igs.ifag.de/root_ftp/software/NtripLinuxClient.zip
- NtripServer (reading from serial port)
  http://igs.ifag.de/root_ftp/software/NtripServerWindows.zip
- NtripServerCMD (command-line version of the NtripServer, reading from TCP/IP port)
  http://igs.ifag.de/root_ftp/software/NtripServerWindowsCMD.zip
- NtripCaster
  The NtripCaster developed under GNU General Public License will become available in July 2003.