TSTP

Time Series Transfer Protocol Interface Description

5th April 2019

Contents

1 Motivation

By means of the TSTP interface, it is possible to generate, identify, read and write data from toposoft's *Time Series Management System* (`"TSMS"`). Special methods support the fast and easy access to time series without complex overhead.

Every software component to be used for generation, maintenance or analysis of a data base can be connected to the `TSMS` via TSTP interfaces.

The only requirement for the connection of existing software is a programming interface to a TCP/IP socket. An extensive platform and language dependent interface is obsolete. The TSTP interface is based on a stateless TCP/IP socket connection which only exists during a data transmission. The TSTP data is embedded in an HTTP protocol.

2 Introduction

The Time Series Management System offers the following functions:

- *Time Series Storage and Management:* All series are administrated and filed transparently for the application (and therefore also for the user). The user does not deal with files, but with time series. Series are held in a binary format. They are addressed and loaded by their attributes.
- *Attributes Management:* Each series has two kinds of attributes, identification attributes and interpretation attributes (further attributes). The identification attributes of a series define its content clearly. Series can be searched for by any (single or composed) attribute pattern. The series found hereby can be indicated for selection. Identification attributes are:
  - Parameter (e.g. `Precipitation`, `Runoff`, ...)
  - Location ID (location and sub location)
  - Time Reference (continuous, interval, momentary)
  - Time Step (for interval time series)
  - Interpretation (e.g. Measurement, Mean, Maximum,...)
  - Origin (original, derived, transformed, simulated)
  - Version (e.g. for different test runs)
  - Type of Series (time, real value)

  Further attributes are:
  - Fault Tolerance (for the specific reduction of inflexion points)
  - Unit (is processed automatically)

- Comment
- Geographical Coordinates (X,Y,Z)
- XUnit (when other than time)

A time series always comprises the whole data. Fragments therein (e.g. from `1.11.1991` to `1.11.2001`) never are handled as separate objects. A time series can inform about the maximum existing time period (the `MaxFocus`).

3 Running and Configuring TSTP

3.1 How to Run and Exit

TSTP can be run on the server by the following command:

```
tstpd [option] [&]
```

The `&` symbol at the end of the line allows running the TSTP process as a daemon.

On initializing, some status information is output depending on its startup options, e.g.:

```
>tstpd &
[1] 30272
================================================
tcp/tstp: unknown service, using 8030
using port 8030, Authentication on
29.04.2008 16:26:45 Release: 1 started.
29.04.2008 16:26:45 36 items in cache.
>
```

Due to being run as a daemon the process number (`30272`) has been output by the command shell.

By default, the TSTP port is `8030`. A sample can be found at 5.2.

Also, authentication of any inquiring client is activated by default (`Authentication on`).

Both last lines denote that the TSTP server is working and `36` time series can be accessed. On startup, the time series cache is set up, thus needing some time to complete.

If the TSTP daemon (running as background process) is to be stopped, the `kill` command is used. The process number needed here can be obtained via the `ps -edalf` command, e.g.

```
>ps -edalf
F S UID        PID [...] CMD
```

```
[...]
0 S toposoft 30902 [...] tstpd
[...]
>kill 30902
[1]    Exitcode 15                        tstpd
>
```

If all running TSTP daemons are to be finished, the `killall` command is used:

```
>killall tstpd

[1]    Exitcode 15                        tstpd
>
```

A TSTP server running in a shell as foreground process can be stopped by pressing the keys „`<Ctrl>-d`“.

3.2 Options

When running the TSTP daemon, some options can be passed in order to refine some TSTP features. The usage lines denoted below is also displayed every time `tstpd` some unknown or unsupported options are appended to the command line:

```
usage: tstpd [OPTION]
         options:
         -h, --help       : shows this list
         -v               : output version & compilation date
         -p <port>        : use port <port> rather than 8030
         -noauth          : disable authentication
         -nowrite         : disable all write access
         -pubonly         : just serve published series
         -noquery         : disable the QUERY command
         -mainonly        : just serve main series
         -timeonly        : time series only, no real series
         -minbs <rev.lvl> : only send data with <revision level> at least
         -action <action> : call <action> after PUT
         -verbose         : be more verbose on init
         -log             : logs client communication
         -logput          : logs zr data sent by PUT (implies -log)
         -enumids         : use enumerated (dynamic) ids
         -oc              : run without time series cache
-noqm            : do not merge quality stamps into value pairs
         -startdir <dir>  : starts in <dir>
```

In plain text:

- `-noauth`: No authentication:
  Disables authentication for any client requesting data. See chapter 5.1.

- `-p <port>`: Specific port:
  Determines which TCP/IP port to use for the TSTP server (instead of default port `8030`). By this, multiple TSTP servers running simultaneously and independently can be accomplished.

- `-nowrite`: No write access:
  Reduces access to „read-only“ for all series provided by the TSTP server.

- `-pubonly`: Published series only:
  Reduces access to series that are marked as „published“ (regarding the series attribute `PUBLIZIERT`).

- `-noquery`: Disable the QUERY command.

- `-mainonly`: main series only:
  Reduces access to series that are marked as „main ts“ (regarding the series attribute `HAUPTREIHE`).

- `-timeonly`: Time series only:
  Reduces access to time series only, leaving real series out of scope.

- `-minbs <rev.lvl>`: Minimum revision level:
  Raises the required minimum revision level of series to be accessed from `0` to the one passed as option argument (in range `1` to `5`, the latter one meaning „deployed“).
  When a client requests series data over a specified time interval, all in-interval data's revision level has to be at least of the level determined on startup. Otherwise, the series data would consist of one big gap.
  If the time series have not been edited to higher revision levels, its default level is `0`. So, if you determined `0` to be the minimum level to acquire series data, it would be the trivial case, anyway - and you would get all the data.

- `-action <action>`: Action after PUT:
  After performing a PUT request successfully (modifying series data on the TSTP server), an action may be processed, if desired. This option enables the TSTP server to call a shell script or an application with some parameters.

- `-verbose`: additional output when starting TSTP:
  The TSTP browses through all time series when it is started. This option causes that info about each recognised time series is outputted before it is processed. This helps finding corrupted time series.

- `-log`: write a log file:
  All received commands and their parameters are logged into `tstpd.log`. By default no logging is performed.
- `-logput`: log PUT data:
  All data the client sends via the PUT command is logged.
- `-startdir <dir>`: change to dir at start.

3.2.1 Action after PUT

The option `-action <action>` passes the name `<action>` of an application or a script to the TSTP server that will be run

- if the application ort the script exists and
- after a PUT request has been served successfully.

A well defined sequence of arguments is passed to the application or the script for being used as context of the processing. The command line contains the following items:

```
<action> <TSID> <URL> <start> <end> &
```

The arguments refer to the preceding PUT request and mean in detail:

- `TSID`: the identification number of the time series having been affected,
- `URL`: the URL of the TSTP server involved,
- `start`: Start time of the interval regarded and
- `end`: End time of the interval regarded.

4 Commands

The following commands are provided for the TSTP protocol: `QUERY`, `GET`, `PUT`, `SETATTR`, `INSPECT`, `QNUM`, `CREATE`, `DELETE`, `DELETEQUAL`, `GETDVAL` und `GETCOMBO`. All functions are documented below.

The call of the client is transmitted as a query in the form of an HTTP request to the TSTP server. Each function is assigned with arguments and return values. The return values are transmitted in XML format (see figure 1).

Mass data (time and value pairs) sent to the server is transmitted via HTTP-POST (see `PUT`).

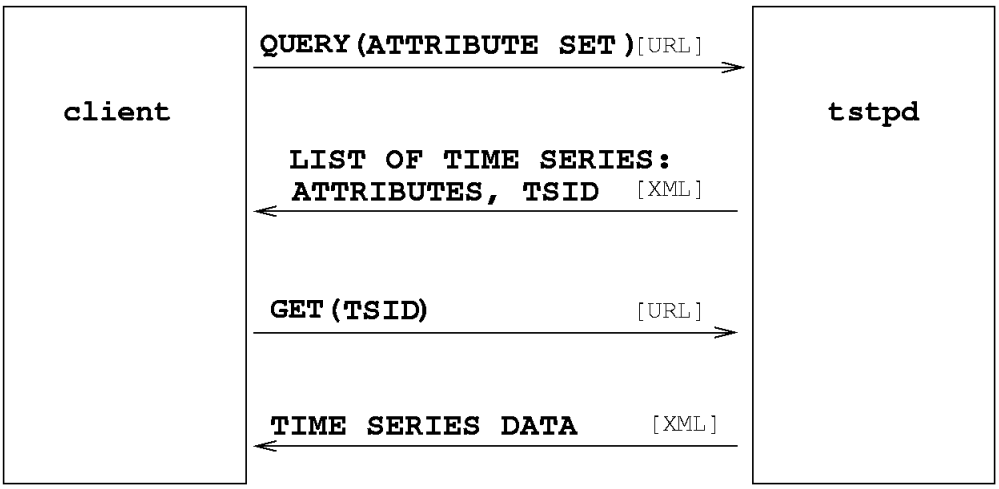

Figure 1: Example for a data call from a TSTP server: QUERY requests a list of time series and their attributes; GET inquires transmission of one specific time series.

4.1 QUERY

An attribute pattern is sent to the TSTP server. All time series matching the pattern are returned to the client as a time series list together with all attributes and a number identifying it clearly (`TSID`) in XML format.

Encoded as URL, the handoff to the TSTP server would read as follows:

```
http://www.tstp.net:8030/?Cmd=Query&Parameter=Runoff&Ort=2400*&DefArt=K
```

If the query is sent directly to the server via HTTP protocol, it must be tagged with the following header:

```
GET /?Cmd=Query&Parameter=Runoff&Ort=2400*&Type of series=K HTTP/1.0\n
Host: tstp.toposoft.de\n
Authorization: Basic? {\it user:pass}\verb?\n
Connection: Close\n\n
```

whereas „`user:pass`“ is a string encoded with Base64.

This query selects all time series with the parameter `Runoff` and the time reference `continuous` whose station ID starts with `2400`. `*` is a wildcard. In the above example, it ensures that all time series stations starting with `2400` (literally, not numerically) are selected.

The following attributes are allowed in the query:

- `Parameter` (parameter),
- `Ort` (location),

- `SubOrt` (sub-location),
- `DefArt` (time reference),
- `Aussage` (interpretation),
- `XDistanz` (time step),
- `XFaktor` (time step factor),
- `Herkunft` (origin),
- `Reihenart` (type of series),
- `Version` (version) as well as
- `Quelle` (source)

These are also called „identification attributes“. A list with possible values of all attributes can be found in the addendum.

The result is an XML page embedded in a `TSQ` element. This contains a list of `TSATTR` elements (one for each matching time series), which again contains all attributes of the time series. An XML node containing `<attributename>value</attributename>` is output for each attribute.

Example:

The result of the above query would return as:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSQ RELEASE="1">
  <TSATTR>
    <ZRID>5</ZRID>
    <MAXFOCUS-Start>1960-05-06T12:02:00Z</MAXFOCUS-Start>
    <MAXFOCUS-End>2001-08-01T00:00:00Z</MAXFOCUS-End>
    <MAXQUAL>1</MAXQUAL>
    <PARAMETER>Runoff</PARAMETER>
    <ORT>24004501</ORT>
    <DEFART>K</DEFART>
    <AUSSAGE></AUSSAGE>
    <XDISTANZ>E</XDISTANZ>
    <XFAKTOR>1</XFAKTOR>
    <HERKUNFT>O</HERKUNFT>
    <REIHENART>Z</REIHENART>
    <VERSION>0</VERSION>
    <X>2526320</X>
    <Y>5640320</Y>
    <GUELTVON></GUELTVON>
```

```
    <GUELTBIS></GUELTBIS>
    <EINHEIT>cm</EINHEIT>
    <MESSGENAU>0.0000</MESSGENAU>
    <FTOLERANZ>1.0000</FTOLERANZ>
    <FTOLREL>False</FTOLREL>
    <NWGRENZE>0.0000</NWGRENZE>
    <SUBORT></SUBORT>
    <KOMMENTAR></KOMMENTAR>
    <HOEHE>83</HOEHE>
    <YTYP></YTYP>
    <XEINHEIT></XEINHEIT>
    <QUELLE></QUELLE>
    <PUBLIZIERT>F</PUBLIZIERT>
 <PARMERKMAL/>
 <HAUPTREIHE>T</HAUPTREIHE>
 <MAXTEXTFOCUS-Start>1990-05-29T11:33:00Z</MAXTEXTFOCUS-Start>
    <MAXTEXTFOCUS-End>1995-06</MAXTEXTFOCUS-End>
  </TSATTR>
  <TSATTR>
    <ZRID>9</ZRID>
    <MAXFOCUS-Start>1950-05-29T11:33:00Z</MAXFOCUS-Start>
    <MAXFOCUS-End>1999-06</MAXFOCUS-End>
    <MAXQUAL>3</MAXQUAL>
    <PARAMETER>Runoff</PARAMETER>
    <ORT>24006008</ORT>
    <DEFART>K</DEFART>
    <AUSSAGE></AUSSAGE>
    <XDISTANZ>E</XDISTANZ>
    <XFAKTOR>1</XFAKTOR>
    <HERKUNFT>O</HERKUNFT>
    <REIHENART>Z</REIHENART>
    <VERSION>0</VERSION>
    <X>2519320</X>
    <Y>5649600</Y>
    <GUELTVON></GUELTVON>
    <GUELTBIS></GUELTBIS>
    <EINHEIT>cm</EINHEIT>
    <MESSGENAU>0.0000</MESSGENAU>
    <FTOLERANZ>0.0500</FTOLERANZ>
    <FTOLREL>0.0500</FTOREL>
    <NWGRENZE>0.0000</NWGRENZE>
    <SUBORT></SUBORT>
    <KOMMENTAR></KOMMENTAR>
    <HOEHE>57</HOEHE>
```

```
    <YTYP></YTYP>
    <XEINHEIT></XEINHEIT>
    <QUELLE></QUELLE>
    <PUBLIZIERT>T</PUBLIZIERT>
 <PARMERKMAL/>
 <HAUPTREIHE/>
    <MAXTEXTFOCUS-Start>1990-05-29T11:33:00Z</MAXTEXTFOCUS-Start>
    <MAXTEXTFOCUS-End>1995-06</MAXTEXTFOCUS-End>
  </TSATTR>
</TSQ>
```

Alternatively, the specification of the time series ID suffices as parameter. The following query leads to the attributes belonging to a time series.

Encoded as URL, the handoff to the TSTP server would read as follows:

`http://www.tstp.net:8030/?Cmd=Query&ZRid=J7v2xiGggfJNPSCJNaTtFw`

An XML containing the requested time series is then sent back.

4.2 GET

Gets data from a time series (value pairs, time - number). Uses GETCOMBO to get Texts (time - string). A time series is requested by passing a `TSID` as argument. In addition, a time interval with `Von=<Point in time>` and `Bis=<Point in time>` (from and to) must be specified, optionally a quality level with `Qual=[0..47]` can be specified. The highest quality level used is selected as default. A Base64 encoded binary block containing the list of the time value pairs is delivered. „`Time reference`“, „`type of series`“ and „`unit`“ are put out as XML attributes of the `DEF` element.

Alternatively, the transmission can be carried out in ASCII form which is selected via the parameter `Typ=Asc`.

URL example:

```
http://www.tstp.net:8030/?Cmd=Get&ZRID=1234&Von=2003.08.11T12:35:21Z
&Bis=2003.08.31T11:55:01Z&Qual=1
```

The attribute `ANZ` defines the number of time value pairs for the ASCII transmission, the attribute `LEN` defines the length of the Base64 decoded data block during the binary transmission.

Example 1:

When transmitted in the ASCII format, the returned data is structured as follows:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
```

```
    DEFART="K" EINHEIT="m^3/s" LEN="0" ANZ="5" />
  <DATA><![CDATA[2003-01-01T17:30:20Z 45.89
2003-01-01T17:35:10Z 0
2003-04-01T17:30:20Z -34.009
2003-05-01T17:30:00Z 12.34
2003-05-01T18:30:20Z 3.141592654]]></DATA>
</TSD>
```

The data is seperated by a simple linefeed (ASCII character 10)

Example 2:

When transmitted in the binary format, however, as follows:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" EINHEIT="m^3/s" LEN="24" ANZ="2"/>
  <DATA><![CDATA[base64-data]]></DATA>
</TSD>
```

`<base64-data>` is a wildcard for the Base64 coded data therein.

The data is submitted successively according to chart 1 (see below). Points in time (`zzzzzzzz`) occupy 8 bytes, real values (`ffff`) 4 Bytes. Subsequently, this binary block is encoded in Base64 so that the data is XML conformable. The so originated Base64 encoded block is then supplied with a linefeed after every 60 characters in order to keep the lines short. This is important with regard to an XML validation where the line length is limited. Besides, it enhances the readability in a browser.

The time value pairs in the binary block connect to each other seamlessly. Depending on the query, the capacity of the binary block can thus be a few hundred kilobytes.

4.3 PUT

A time series is selected by means of a `TSID`. Optionally, a quality level can be specified. The data is then submitted to the TSTP server by POST in an XML message and integrated on the server.

Example:

The HTTP header must contain the following parameters:

```
POST /?Cmd=PUT&ZRID=123&QUAL=2 HTTP/1.0\n
Host: www.tstp.net:8030\n
Authorization: Basic <user:pass>\n
Content-Length: <length of the complete XML appendix>\n
Connection: Close\n\n
```

The proximate binary block must read as follows:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" EINHEIT="m^3/s" LEN="24" ANZ="2"/>
  <DATA><![CDATA[<base64-data>]]></DATA>
</TSD>
```

`<base64-data>` is a wildcard for the Base64 coded data therein. The response is an XML message.

`<TSR RELEASE="1">confirm</TSR>`

is submitted as confirmation and

`<TSR RELEASE="1"><ERR><error message></ERR></TSR>`

in case of failure.

`<error message>` is a wildcard for the error message text therein.

An AXML document is returned. Confirmation is signaled by

`<TSR RELEASE="1">confirm</TSR>`

an error is signaled by

`<TSR RELEASE="1"><ERR><error message></ERR></TSR>`

`<error message>` is the plain text of the error.

Special case Precipitation

Transferring precipitation data to the server is a special case. Time series are stored as intensities in mm/h. Often the data on the client side are total graphs or impulses. You may send this kind of data if you declare their type. This is done using the attribute `MESAUS` of the `DEF`-element. MESAUS can take four states:

- SUMLIN the values represent a total graph, are hence montonous increasing. The first value does not need to be 0.
- SUML0 the values represent a total graph that may drop back to 0 at any time. But the 0 itself may not be appear in the data. This means that if a value v is smaller than the previous one the drop back to 0 and then a rise to v is assumed. The first value does not have to be 0. If precipitation occured between the last transfer an the actual transfer then this amount of precipitation must be prefixed to the actual transferred data.
- DELTA the values are deltas to the previous value.
- INTENS the values are intensities in mm/h.

Alle All values (save of INTENS) have to be in mm!

Precipitation data looks like:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" EINHEIT="mm/h" LEN="240" ANZ="20" MESAUS="SUML0"/>
  <DATA><![CDATA[<base64-daten>]]></DATA>
</TSD>
```

4.4 Functioning Of The Time Series Data Base On Insert

Time series have neither beginning nor end, they are available from „-Infinite“ to „+Infinite“. The associated points of time are `+Infty` und `-Infty`. Periods for which the data is not known have the value `GAP`. In general, gaps are regular Y values which don't play a special role.

Therefore, each writing process on time series is an insert. This is described in the following.

Intervals **always** run from the inflexion point back to the previous inflexion point. Daily values, for example, are filed at the end of each day.

The basic idea is that the data to be inserted fully penetrates the time period which is defined by it and **all** other time periods are not touched.

Time series do not store time value pairs but traverses. At every point in time, be it an inflexion point or not, a Y value is defined. The inflexion points are connected linearly (continuous TS) or in steps (interval TS). In momentary TS, the Y values are not connected. On insert, all existing inflexion points are deleted in that range.

When the data to be inserted does not exactly meet an inflexion point on left or right, the margins are adjusted. This takes place in the following way:

- *continuous TS:* A skip by the width of 5 seconds is generated. For this purpose, the traverse is interpolated on the conjunction, this value is then inserted 5 seconds before/after the insert range.

- *interval TS:* The first existing value on the right of the insert range is inserted instead of the first replacement value. Thus, the Y value of the first time value pair is purged, solely the point in time is of importance. Implicitly the interval following the insert range may change to the new previous point in time.

- *momentary TS:* An adjustment does not take place.

4.5 SETATTR

An attribute of the time series selected by TSID is set. Identification attributes can **not** be set. Besides attributes you may pass INFO und LEBENSLAUF to set this fields of the time series.

URL example:

```
http://www.tstp.net:8030/?Cmd=SetAttr&ZRID=1234&Attr=Kommentar
&Wert=ABCDEF
```

```
<TSS>confirm</TSS>
```

is submitted as confirmation and

```
<TSS RELEASE="1"><ERR>?<error message></ERR></TSS>
```

in case of failure.

<error message> is a wildcard for the error message text therein.

4.6 CREATE

A complete attribute set encoded as URL is submitted. A TSID of the series which was created (or found in the database) is returned. In case the creation failed, the TSID obtains the value 0.

URL example:

```
http://www.tstp.net:8030/?Cmd=Create&Parameter=Precipitation
&Ort=24003123&SubOrt=0&DefArt=K&Aussage=Sum&Herkunft=0
&Reihenart=Z&Version=0&Quelle=S
```

The TSID is returned as

```
<TSR RELEASE="1"><TSATTR>ZRID=1234</TSATTR></TSR>
```

or in case of failure

```
<TSR RELEASE="1"><TSATTR>ZRID=0</TSATTR><ERR><error message></ERR></TSR>
```

<error message> is a wildcard for the error message text therein.

4.7 DELETE

The TSID of the time series to be deleted is submitted.

URL example:

```
http://www.tstp.net:8030/?Cmd=Delete&ZRID=1234
```

```
<TSR RELEASE="1">confirm</TSR>
```

is returned in order to confirm the successful deletion or

`<TSR RELEASE="1"><ERR><error message></ERR></TSR>`

in case the deletion failed.

`<error message>` is a wildcard for the error message text therein.

4.8 DELETEQUAL

A `TSID`, a time interval and a quality level are submitted. The quality is deleted in the interval range.

URL example:

```
http://www.tstp.net:8030/?Cmd=DeleteQual&ZRID=1234
&Von=2003.02.01T11:14:00Z&Bis=2003.23.22T12:31:00Z&Qual=2
```

Accordingly returned is

`<TSQ RELEASE="1">confirm</TSQ>`

or

`<TSQ RELEASE="1"><ERR><error message></ERR></TSQ>`

`<error message>` is a wildcard for the error message text therein.

4.9 INSPECT

Supply a time series id `ZRID` and optionally a focus. Returned is the maximal quality and the maximal physical quality on focus, the vita (Lebenslauf), the INFO field of the time series and the time of the last modification.

Without a focus the maximal quality and maximal physical quality refer to the maxfocus of the time series.

Example URL:

```
http://www.tstp.net:8030/?Cmd=Inspect&ZRID=1234
&Von=2003.01.01T00:00:00Z&Bis=2004.01.01T00:00:00Z
```

Zurückgeliefert wird z.B.:

```
<TSR RELEASE="1">
   <MAXQUAL>2</MAXQUAL>
   <MAXPHYSQUAL>2</MAXPHYSQUAL>
   <LEBENSLAUF><![CDATA[Base64-Daten]]></LEBENSLAUF>
   <INFO><![CDATA[Base64-Daten]]></INFO>
   <TIMESTAMP>2009.11.10T20:03:23Z</TIMESTAMP>
</TSR>
```

oder

`<TSR RELEASE="1"><ERR><error message></ERR></TSR>`

`<error message>` plain text of the error message.

4.10 QNUM

Supply a time series id `ZRID` and optionally a focus. Returned is the number of quants (values) of the time series on focus. Without focus the total number is returned.

Example URL:

```
http://www.tstp.net:8030/?Cmd=QNUM&ZRID=1234
&Von=2003.01.01T00:00:00Z&Bis=2004.01.01T00:00:00Z
```

Returned is:

```
<TSR RELEASE="1">
   <ANZ>2654</ANZ>
</TSR>
```

ANZ means number of. or

`<TSR RELEASE="1"><ERR><error message></ERR></TSR>`

`<error message>` plain text of the error message.

4.11 GETDVAL

A `TSID`, a time interval with `Von=<point in time>` and `Bis=<point in time>`, an interval width with `IB=<interval range>`, an interpretation with `Aussage=<interpretation>` (e.g. `Mit`(mean) - see chart 3 in appendix A) and optionally a quality layer with `Qual=[0..50]` are submitted.

URL example:

```
http://www.tstp.net:8030/?Cmd=GetDVal&ZRID=1234&Von=2003.02.01T11:15:00Z
&Bis=2003.03.22T12:30:00Z&IB=5Min&Aussage=MIT
```

A time series with corresponding values (interval values, continuous values or momentary values) is returned for the time interval as binary block (see `GET`). The selected interval width `IB` defines the attributes `time step factor` and `time step` for the returned series.

4.12 GETCOMBO

The time series machine stores three data lists: the list with numbers, the list with texts and the list with isolated points. GETCOMBO gets all three data lists in one single request. Uses GETCOMBO like GET. The return comprises three consecutive TSD elements. The first is for numbers, the second for texts, the third for isolated points.

In addition to the parameters of the GET command (VON, BIS, QUAL) there is READMODE, that can take INTERPOLIERT (interpolated), INNEN (inside) or AUSSEN (outside). The default is INTERPOLIERT.

4.13 UPDATE

A `TSID` is submitted. The server reads in the time series once again and updates the `MaxFocus`. In case the time series has been modified without the server having noticed this, this command must be applied.

URL example:

`http://www.tstp.net:8030/?Cmd=Update&ZRID=1234`

`<TSR RELEASE="1">confirm</TSR>`

is returned in order to confirm the successful update, or

`<TSR RELEASE="1"><ERR><error message></ERR></TSR>`

in case the update failed (e.g. due to incorrect `TSID` or missing access rights).

`<error message>` is a wildcard for the error message text therein.

4.14 GLAMP

Calculates floating amplitutes (difference between the floating maxima and the floating minima). Supply a time series id `ZRID`, a focus with `Von=`time and `Bis=`time, and the width of the interval that is used to calculate teh floating values (e.g. 1h): `IB=`width (like 1h, 3d, 50min ...)

Example URL:

`http://www.tstp.net:8030/?Cmd=GlAmp&ZRID=1234&Von=2003.02.01T11:15:00Z`
`&Bis=2003.03.22T12:30:00Z&IB=30Min`

Floating extrema are computed by replacing each value by the minumum ore maximum of all values inside an interval centered at the time of teh value.

4.15 FORMEL (formula)

By sending a formula to the tstp server time series are calculates based on one or more other time series.

Pass the formula as `TERM`. Pass the time series using `ZRID` and their MD5 sums, separated by commas. Reihenfolge übergeben, wie sie inder Formel auftauchen.

Example:

```
ttp://www.tstp.net:8030/?Cmd=Formel&TERM=Schlauch(ZR1,ZR2)&ZRID=1234,9876
&Von=2003.02.01T11:15:00Z&Bis=2003.03.22T12:30:00Z
```

The resulting time series data are returned.

4.16 LAYERGET

Use this command to retrieve raster data for given polygones.

Pass the binary data of a layer (for example in SHAPE format) followed by the binary data of the tabular data (DBF file). Use LEN1 to pass the length of the shape binary data. The tstp server splits the passed binary block at LEN1 and reads the rest as dbf.

Time series data is created for each polygon found in the shape. Use PARAMETER to select a specific channel. If PARAMETER is not given the first channel is selected.

If the polygone is the outline of an area all raster tiles inside and all tiles intersecting the outline (proportional) are combined.

4.17 IMAGE

Returns graphics of time series (PNG images). Use XSIZE/YSIZE to define the number of X resp. Y pixels of the images. Default is 900x640.

5 Format Of The Query

5.1 Authentication

The username and password are transmitted in one line in the format „`Authorization: Basic username:password`“ in the HTTP header of a query, whereas the string „`username:password`“ is encoded according to Base64 (RFC 3548, Chapter 3).

Example:

```
GET ?Cmd=Query&Parameter=Runoff&Ort=2400*&DefArt=K HTTP/1.0
Authorization: Basic dXNlcm5hbWU6cGFzc3dvcmQ
```

The user and his indicated password are checked by the TSTP server. Should the password be wrong or the user not exist, an „HTTP-401“ response is generated. In a browser this leads to a login mask.

All Users are assigned individual rights to use time series. A user can either have the right to

- read only,
- read and write or
- read, write, create and delete time series.

5.2 URL Format

The encoding of the parameters submitted in an URL query is preceded with a question mark:

URL example:

`http://www.tstp.net:8030/?`

Followed by the function:

`Cmd=Query|Get|Put|SetAttr|Create|Delete|DeleteQual|GetDVal`

URL example:

`http://www.tstp.net:8030/?Cmd=Create`

The character `=` is inserted between the attribute name and the attribute value for the URL encoding of attributes, e.g. `Ort=24003100`. Further pairs of attribute names and values are separated by the character `&`.

Example:

`http://www.tstp.net:8030/?Cmd=Query&Ort=2400*`

The possible values of the attributes „Time reference“, „Interpretation“, „Time step“, „Origin“, „Type of series“ and „Source“ are compiled in the charts 2 to 6 in appendix A.

5.3 Point in time

According to ISO 8601, the international standard for date and time, this format is used here. The TSTP protocol outputs only this format.

Complete date including hours, minutes and seconds:

`YYYY-MM-DDThh:mm:ssZ` (e.g., `1997-07-16T19:20:30Z`)

Description:

YYYY	=	4 digit year
MM	=	2 digit month (01=january, etc.)
DD	=	2 digit day of the month (01 to 31)
'T'	=	separator between date and time
hh	=	2 digit hours of the day (00 to 23)
mm	=	2 digit minutes of the hour (00 to 59)
ss	=	2 digit seconds of the minute (00 to 59)
'Z'	=	timezone (in TSTP always 'Z')

Alternatively, the following format is recognised as input:

Points in time are encoded in form of

`<day>.<month>.<year>[_<hour>:<minute>[:<second>]]`

i.e. the specifications of the time as well as of the seconds within the time are optional.

5.4 Time Interval

The beginning time is specified with `Von=<point in time>` and the ending time with `Bis=<point in time>` for time intervals.

6 XML Format

XML encoded return messages of the TSTP server start with an HTTP header followed by the XML encoded data.

Example:

```
HTTP/1.0 200 OK
Date: Tue, 28 Oct 2003 13:49:10 GMT
Server: TSTPD/1.0 (UNIX)
Last-Modified: Tue, 28 Oct 2003 13:49:10 GMT
Expires: Tue, 28 Oct 2003 13:49:10 GMT
Cache-Control: max-age=0
Connection: close
Content-Length: 6733
Content-Type: text/plain; charset=ISO-8859-1

<?XML version="1.0" encoding="ISO-8859-1"?>
<TSQ RELAESE="1">
 ...
</TSQ>
```

The lines end with a linefeed (ASCII character 10).

6.1 Attribute List

The root element `<TSQ RELEASE="1">` is returned as reply to a query. A `<TSATTR>` element is included for each time series within the `<TSQ>` element. The single attributes are sub-elements in the `<TSATTR>` element:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSQ RELEASE="1">
  <TSATTR>
    <ZRID>1</ZRID>
    <MAXFOCUS_START>1964-06-01T00:00:00Z</MAXFOCUS_START>
    <MAXFOCUS_END>2005-11-01T00:00:00Z</MAXFOCUS_END>
    <MAXQUAL>0</MAXQUAL>
    <PARAMETER>Rating curves</PARAMETER>
    <ORT>24004002</ORT>
    ..
  </TSATTR>
  <TSATTR>
    <ZRID>2</ZRID>
    ..
  </TSATTR>
</TSQ>
```

6.2 Mass Data

Mass data is enclosed in the root element `<TSD RELEASE="1">`, where the version is encoded as `RELEASE="<release number>"`. Also included is a `<DEF>` element with the header attributes of the time series as well as a line-by-line list of points in time and Y values or a binary block.

The header attributes enclose

- the type of series encoded as `REIHENART="<type of series>"` (e.g., `Z`),
- a flag indicating whether text attributes are delivered as well (`TEXT="No"`),
- the time reference encoded as `DEFART="<time reference>"` (e.g., `K,I,M`),
- the physical unit encoded as `EINHEIT="<unit>"` (e.g., `m^3/s`) as well as
- the number of time value pairs `LEN="<number>"`.

By default, mass data is transferred binary-coded (see below). However, it is also possible to request data in human-readable ASCII format (e.g., for testing). The point in time is indicated in the list of time value pairs in the format `YYYY.MM.DDThh:mm:ssZ`. (See chapter 5.3)

The character '.' is used as decimal point for the Y values. Lines end with the ASCII character 10 (linefeed).

Example:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" EINHEIT="m" LEN="0" ANZ="5" />
  <DATA><![CDATA[2003-01-01T17:30:20Z 45.89
2003-01-01T17:35:10Z 0
2003-04-01T17:30:20Z -34.009
2003-05-01T17:30:00Z 12.34
2003-05-01T18:30:20Z 3.141592654]]></DATA>
</TSD>
```

7 Binary Format

7.1 Mass Data

Mass data is transferred with the same header as ASCII data. However, the data enclosed in `<DATA><![CDATA[` and `]]></DATA>` is binary. `LEN` specifies the number of bytes of the binary block, `ANZ` specifies the number of time value pairs.

Example:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" EINHEIT="m3/s" LEN="24" ANZ="2" />
  <DATA><![CDATA[base64-data]]></DATA>
</TSD>
```

The byte ordering within a 32-bit word complies with the „network byte order“ (3-2-1-0).

The data is composed of a sequence of value pairs which are linked to each other seamlessly. A time value pair is composed of `XY`, whereas `X` corresponds to a time or real point and `Y` to a 32-bit IEEE floating-point number or a string. The Y value `4E+37` (`C2 BD F0 7D`) represents a gap.

The attribute s `TEXT` in the `DEF` tag that is valid for all value pairs determines whether the y values are numbers or strings.

Strings are stored in the following way: the first byte contains the variant of the string. 8 denotes an empty string (no further data follow). 6 being for small strings (1-255 characters) is followed by the string length. 7 is for long strings and followed by the length in

four bytes (network byte order). strings of variant 6 and 7 follow seamlessly behind the length.

The time or real points are encoded in 64 bits according to chart 1, three modes are discerned: point in time (0), real point (1) and point in time in microsecond accuracy (2).

binary format time/real												
byte 7		byte 6		byte 5		byte 4		byte 3		byte 2	byte 1	byte 0
bit 4-5	bit 0-3	bit 4-7	bit 0-3	bit 4-7	bit 0-3	bit 4-7	bit 0-3	5-7	Bit 0-4			
year **0: time**	quality stamp	0: regular 1: $-\infty$ 2: $+\infty$	Jahr	year	year		month		day	hour	minute	second
1: Realpoint		0: regular 1: $-\infty$ 2: $+\infty$						32-bit IEEE float				
2: Millitime	quality-stamp	ms		year	year	year	month	ms	day	hour	minute	second

Table 1: Binary format of time and realp points.

ms means milli seconds and has 10bits (bit 9-2 in byte 6 and bit 1-0 in byte 3)

Appendix A

Time reference		
Allowed value	Type	URL encoding
K	continuous	DefArt=K
I	interval	DefArt=I
M	momentary	DefArt=M

Table 2: Possible values for the attribute „Time reference“ (`DefArt`).

Interpretation		
Allowed value	Type	URL encoding
Mes	Manual measurements	Aussage=Mes
Sum	Sums	Aussage=Sum
Mit	Means	Aussage=Mit
Max	Maxima	Aussage=Max
Min	Minima	Aussage=Min
DMax	Maxima points in time	Aussage=DMax
DMin	Minima points in time	Aussage=DMin
Abl	Derivatives	Aussage=Abl
Int	Integral	Aussage=Int
Dau	Duration curve	Aussage=Dau
PDF	Probability density	Aussage=PDF
VF	Distribution function	Aussage=VF
Lck	Gap series	Aussage=Lck
Ant	Gap quota	Aussage=Ant
OS	Upper limit	Aussage=OS
US	Lower limit	Aussage=US
pSe	Partial series	Aussage=pSe
jSe	Annual series	Aussage=jSe
pVP	Partial distribution par.	Aussage=pVP
jVP	Annual distribution par.	Aussage=jVP
ERG	Events	Aussage=ERG
NER	Rainfall events	Aussage=NER
jSN	Annual series Rainfall	Aussage=jSN
pSN	Partial series Rainfall	Aussage=pSN
Kon	Control values	Aussage=Kon
XSy	X-synchro points	Aussage=XSy
YSy	Y-synchro points	Aussage=YSy

Table 3: Possible values for the attribute „Interpretation“. The interpretations `DMin` and `DMax` deliver a momentary time series with those points in time at which the extreme values have been reached.

Origin		
Allowed value	Type	URL encoding
O	original	Herkunft=O
T	transformed	Herkunft=T
S	simulated	Herkunft=S
A	derived	Herkunft=A
M	temporary	Herkunft=M
B	reassessed	Herkunft=B
F	TS sequence	Herkunft=F
P	periodic	Herkunft=P

Table 4: Possible values for the attribute „Origin“.

Type of series		
Allowed value	Type	URL encoding
Z	time series	Reihenart=Z
R	real series	Reihenart=R

Table 5: Possible values for the attribute „Type of series“.

Source		
Allowed value	Type	URL encoding
L	Data logger	Quelle=L
S	charts	Quelle=S
H	Manual entry	Quelle=H
P	Production	Quelle=P
T	Test version	Quelle=T
	(blank)	Quelle=

Table 6: Possible values for the attribute „Source“.