

TSTP

Time Series Transfer Protokoll  
Schnittstellenbeschreibung

5. April 2019





# Inhaltsverzeichnis

<b>1</b>	<b>Motivation</b>	<b>1</b>
<b>2</b>	<b>Einleitung</b>	<b>1</b>
<b>3</b>	<b>TSTP starten und einrichten</b>	<b>2</b>
3.1	Starten und beenden . . . . .	2
3.2	Optionen . . . . .	3
3.2.1	Aktion nach PUT . . . . .	5
<b>4</b>	<b>Funktionsbeschreibung</b>	<b>6</b>
4.1	QUERY . . . . .	6
4.2	GET . . . . .	10
4.3	PUT . . . . .	11
4.4	Arbeitsweise der Zeitreihen-Datenbank beim Einfügen . . . . .	13
4.5	SETATTR . . . . .	14
4.6	CREATE . . . . .	14
4.7	DELETE . . . . .	15
4.8	DELETEQUAL . . . . .	15
4.9	INSPECT . . . . .	15
4.10	QNUM . . . . .	16
4.11	GETDVAL . . . . .	16
4.12	GETCOMBO . . . . .	17
4.13	UPDATE . . . . .	17
4.14	GLAMP . . . . .	17
4.15	FORMEL . . . . .	18
4.16	LAYERGET . . . . .	18
4.17	RASTERGET . . . . .	19
4.18	IMAGE . . . . .	19
<b>5</b>	<b>Format der Anfrage</b>	<b>19</b>
5.1	Authentifizierung . . . . .	19
5.2	URL-Format . . . . .	20
5.3	Zeitpunkt . . . . .	20
5.4	Zeitintervall . . . . .	21

<b>6 XML-Format</b>	<b>21</b>
6.1 Attributliste . . . . .	21
6.2 Massendaten . . . . .	22
<b>7 Binärformat</b>	<b>23</b>
7.1 Massendaten . . . . .	23

## 1 Motivation

Mithilfe der TSTP-Schnittstelle ist es möglich, Datenbestände aus der Zeitreihen-Datenbank zu erstellen, zu identifizieren, zu lesen und zu schreiben. Spezielle Methoden unterstützen den schnellen und einfachen Zugriff auf Zeitreihen ohne aufwändigen Overhead.

Jede Softwarekomponente, die zum Erstellen, zur Pflege oder zur Analyse eines Datenbestandes eingesetzt werden soll, kann über die TSTP-Schnittstelle an die Zeitreihen-Datenbank angebunden werden.

Einzige Voraussetzung für die Anbindung vorhandener Software ist eine Programmierschnittstelle zu einem TCP/IP-Socket, wodurch eine aufwändige, plattform- und sprachenabhängige Programmierschnittstelle entfällt. Die TSTP-Schnittstelle basiert auf einer zustandslosen TCP/IP-Socket-Verbindung, die nur während einer Datenübertragung besteht. Die TSTP-Daten werden in ein HTTP-Protokoll eingebettet.

## 2 Einleitung

Die Zeitreihen-Datenbank bietet folgende Funktionen:

- *Zeitreihenablage und -verwaltung*: Alle Reihen werden für die Applikation (und damit auch für den Nutzer) transparent verwaltet und abgelegt. Der Nutzer setzt sich nicht mit Dateien auseinander, sondern mit Zeitreihen. Reihen werden in einem Binärformat vorgehalten und über die ihnen zugeordneten Attribute angesprochen und geladen.
- *Attributeverwaltung*: Jede Reihe hat zwei Arten von Attributen, Identifikationsattribute und Weitere Attribute). Die Identifikationsattribute einer Reihe identifizieren sie eindeutig. Reihen können mit einem beliebigen Attributmuster gesucht werden. Die dabei gefundenen Reihen können zur Auswahl angezeigt werden. Identifikationsattribute sind:
  - Parameter (z.B. Niederschlag, Abfluss, ...)
  - Ortsbezug (Ort und SubOrt)
  - DefArt, Art des Zeitbezugs (Kontinuierlich, Intervall, Momentan)
  - Zeitdistanz (für Intervall-ZR)
  - Aussage (z.B. Summe, Mittelwert, Maximum, ...)
  - Herkunft (original, abgeleitet, transformiert, simuliert)
  - Version (z.B. für verschiedene Simulationsläufe)
  - Reihenart (Zeit, Real)

Weitere Attribute sind:

- Fehlertoleranz (zum gezielten Ausdünnen von Knickpunkten)
- Einheit (wird automatisch verarbeitet)
- Kommentar
- Lagekoordinaten (X,Y,Z)
- XEinheit (wenn die Ordinate nicht auf Zeit bezogen ist)

In den Stammdaten kann Zeitreihen eine eindeutige Kennung zugeordnet sein, eine sogenannte PV (Prozessvariable). Es ist dann möglich, die Zeitreihen statt über ihre Identifikationsattribute über diese PV zu finden.

Eine Zeitreihe umfasst immer alle Daten eines Bezugs. Zeitreihenstücke (z.B. von 1.11.1991 bis 1.11.2001) sind keine eigenen Objekte. Eine Zeitreihe kann Auskunft über den maximal vorliegenden Zeitbereich (den MaxFocus) geben.

## 3 TSTP starten und einrichten

### 3.1 Starten und beenden

Auf dem Server wird TSTP folgendermaßen gestartet:

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

Das &-Zeichen am Ende ermöglicht bei Bedarf, dass der gestartete TSTP-Prozess als „Dämon“ im Hintergrund weiterläuft.

Beim Start werden eine Reihe von Statusmeldungen ausgegeben (Beispiel):

```
>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.
>
```

Wegen der Ausführung im Hintergrund wird die Prozessnummer (30272) von der Shell ausgegeben.

Üblicherweise wird als TSTP-Port der Port 8030 genutzt. Ein Testbeispiel findet sich unter Kapitel 5.2.

Die Authentifizierung des Clients, der eine Abfrage an den TSTP-Server richtet, ist üblicherweise aktiviert (Authentication on).

Die letzten beiden Zeilen geben an, dass der Server in Betrieb ist und auf 36 Zeitreihen zugegriffen werden kann. Der Zeitreihen-Cache wird beim Start aufgebaut, was etwas Zeit erfordert.

Zum Beenden des im Hintergrund laufenden TSTP-Dämons muss man den `kill`-Befehl verwenden. Die Prozessnummer lässt sich mithilfe des `ps -edalf`-Befehls ermitteln. Beispiel:

```
>ps -edalf
F S UID          PID [...] CMD
[...]
0 S fgon         30902 [...] tstpd
[...]
>kill 30902
[1]      Exitcode 15          tstpd
>
```

Wenn alle laufenden TSTP-Dämonen auf einmal beendet werden sollen, ist der `killall`-Befehl anzuwenden:

```
>killall tstpd

[1]      Exitcode 15          tstpd
>
```

Läuft der `tstpd` als Vordergrund-Prozess, dann reicht der Tastendruck „<Strg>-d“ zum Beenden aus.

## 3.2 Optionen

Beim Start kann der TSTP-Dämon anhand einiger Optionen gesteuert werden. Die folgende Übersicht wird auch ausgegeben, wenn man `tstpd` mit nicht bekannten Optionen startet:

```
usage: tstpd [OPTION]
      -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 info into value pairs
    -startdir <dir> : starts in <dir>

```

Im Klartext:

- **-p <port>**: Anderer Port:  
Legt fest, welcher Port (statt 8030) für den TSTP-Server benutzt werden soll. Hierdurch sind auch mehrere spezialisierte TSTP-Server realisierbar, die zeitgleich erreichbar sind.
- **-noauth**: Keine Anmeldung:  
Mit dieser Option wird die Angabe von Benutzer und Passwort nicht mehr gefordert. Siehe Kapitel 5.1.
- **-nowrite**: Nur Lesezugriff:  
Jeder Schreibzugriff wird unterbunden.
- **-pubonly**: Nur publizierte Reihen:  
Nur die Zeitreihen, deren Attribut PUBLIZIERT auf wahr gesetzt ist, werden von TSTP berücksichtigt.
- **-noquery**: der Query-Befehl wird deaktiviert.
- **-mainonly**: Nur Haupt-Reihen:  
Nur die Zeitreihen, deren Attribut HAUPTREIHE auf wahr gesetzt ist, werden von TSTP berücksichtigt.
- **-timeonly**: Keine Realreihen:  
Die Standardfähigkeit, sowohl Zeitreihen als auch Realreihen übermitteln zu können, wird auf Zeitreihen begrenzt.
- **-minbs <rev.lv1>**: Minimalen Bearbeitungsstand festlegen:  
Für die Zeitreihen-Anfragen wird ein Mindest-Bearbeitungsstand (1 bis 5) gefordert, der über den gesamten angeforderten Zeitraum der Zeitreihendaten eingehalten sein muss. Sobald der angeforderte Zeitraum mindestens einen Teilbereich mit geringerem Bearbeitungsstand aufweist als der mittels Option **-minbs <rev.lv1>** vereinbarte, wird der gesamte Zeitraum als Lücke übermittelt.



- **-action <action>**: Aktion nach PUT:  
Nach einer erfolgreich durchgeführten PUT-Anforderung an den TSTP-Server kann hiermit eine Anwendung oder ein Skript auf dem Server ausgeführt werden, das eine den PUT-Befehl unterstützende Aktion ausführt.
- **-verbose**: Zusätzliche Ausgaben beim Start von TSTP:  
Nach dem Start von TSTP erfolgt die Erfassung aller Zeitreihen, die im Reihenverzeichnis zu finden sind. Mit dieser Option wird die Erfassung begleitet von informellen Ausgaben über die gefundenen Zeitreihen, deren Anzahl, der Liste der geprüften und akzeptierten Zeitreihen sowie deren Anzahl. Diese Option dient der gelegentlichen Kontrolle von TSTP und der Überprüfung auf Fehlerfreiheit der vorliegenden Zeitreihen.
- **-log**: Schreibe eine Logdatei:  
Die empfangenen Befehle werden mit ihren Parametern in der Datei `tstpd.log` mitgeschnitten. Standardmäßig wird keine Logdatei erzeugt.
- **-logput**: PUT-Daten mitschneiden:  
Alle Daten, die bei einem PUT übergeben werden, werden auf der Konsole und in der Logdatei im Klartext mitgeschnitten.
- **-enumids**: Vergabe von klassischen (numerischen) ZRIDs:  
Bei der Erfassung aller Zeitreihen nach dem Programmstart werden allen Zeitreihen eindeutige, persistente Kennungen zugewiesen. Standard sind hierbei MD5-Kennungen, welche auf Dauer unveränderlich sind, da diese alphanumerischen Codes durch die Zeitreihenattribute bestimmt werden.  
Mit dieser Option werden stattdessen numerische ZRIDs erzeugt, beginnend bei 1. Diese sind nicht eindeutig, da bei Neuanlage von Zeitreihen und Neustart von TSTP die Reihenfolge der Zeitreihen verändert vorliegen kann und die ZRIDs nicht gleich vergeben werden.
- **-oc**: für Zeitreihen-Folgen keinen Cache benutzen
- **-noqm**: ohne Qualitätsmerkmale  
Wenn in der Reihe Qualitätsmerkmale gespeichert sind, werden sie standardmäßig in die Binärdaten der Wertepaare integriert. Dies kann mit dieser Option verhindert werden.
- **-startdir <dir>**: direkt nach dem Start in `dir` wechseln.

### 3.2.1 Aktion nach PUT

Die Option `-action <action>` übergibt den Namen eines Programms bzw. eines Skripts, das nur dann ausgeführt wird, wenn

- das Programm oder Skript auf dem Server vorhanden ist und
- eine PUT-Anforderung an den TSTP-Server erfolgreich ausgeführt worden ist.

Es wird eine bestimmte Folge von Argumenten zur Verarbeitung übergeben, welche das Skript oder Programm `<action>` nutzen kann. Die Shell-Befehlszeile lautet dann:

```
<action> <ZRID> <URL> <von> <bis> &
```

Die Argumente bedeuten:

- **ZRID**: Die ZRID der Zeitreihe, auf die die PUT-Anweisung durchgeführt worden ist.
- **URL**: Die URL des TSTP-Servers.
- **von**: Der Startzeitpunkt des mittels PUT beeinflussten Zeitraums.
- **bis**: Der Endzeitpunkt des mittels PUT beeinflussten Zeitraums.

## 4 Funktionsbeschreibung

Für das TSTP-Protokoll werden folgende Funktionen bereitgestellt: QUERY, GET, PUT, SETATTR, INSPECT, QNUM, CREATE, DELETE, DELETEQUAL, GETDVAL, GETCOMBO, UPDATE, GLAMP, LAYERGET, RASTERGET und IMAGE.

Der Funktionsaufruf vom Client wird als Anfrage in Form eines HTTP-Requests an den TSTP-Server weitergegeben. Jeder Funktion sind dabei Argumente und Rückgabewerte zugeordnet. Die Rückgabewerte werden im XML-Format übermittelt (siehe Abbildung 1).

Massendaten (Wertepaare), die zum Server geschickt werden, werden mittels HTTP-POST versendet (z.B. bei PUT).

### 4.1 QUERY

Ein Attributmuster wird an den TSTP-Server gesendet. Alle auf das Muster passenden Zeitreihen werden als Zeitreihenliste mit allen Attributen und einer die jeweilige Zeitreihe eindeutig identifizierenden Kennung (ZRID) im XML-Format an den Client zurückgesendet.

Als URL kodiert würde die Übergabe an den TSTP-Server folgendermaßen aussehen:

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

Wird die Anfrage über das HTTP-Protokoll direkt an den Server gesendet, so muss sie mit folgendem Header versehen werden:

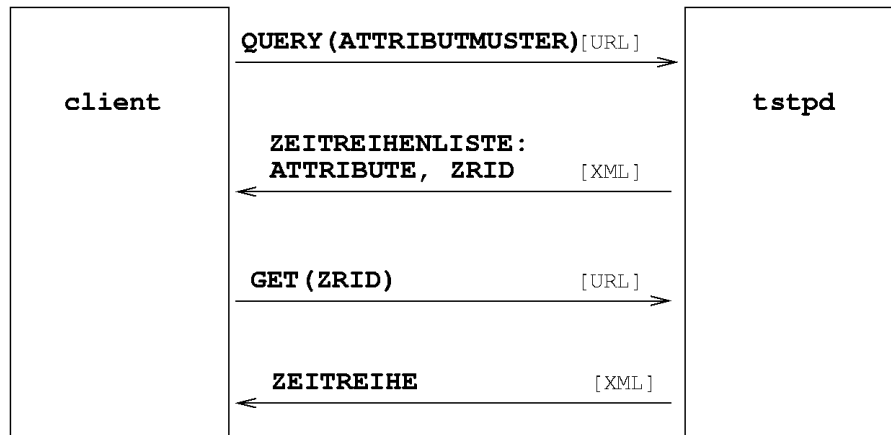


Abbildung 1: Beispiel eines Datenabrufs von einem TSTP-Server: mit der Funktion QUERY wird zunächst eine Liste mit Zeitreihenattributen angefordert, mit der Funktion GET wird eine ausgewählte Zeitreihe übertragen.

```

GET ?Cmd=Query&Parameter=Wasserstand&Ort=2400*&DefArt=K HTTP/1.0\n
Host: tstp.toposoft.de\n
Authorization: Basic <user:pass>\n
Connection: Close\n\n
  
```

wobei „<user:pass>“ ein mit Base64 kodierter String ist.

Diese Anfrage wählt alle Zeitreihen mit Parameter `Wasserstand` und `DefArt` `Kontinuierlich` aus, deren Ortsbezeichnung mit `2400` beginnt. „\*“ ist ein Wildcardzeichen. Im obigen Beispiel bewirkt es, dass alle Zeitreihen ausgewählt werden, deren Ort mit `2400` beginnt.

Folgende Attribute sind in der QUERY-Anfrage möglich:

- Parameter,
- Ort,
- SubOrt,
- DefArt,
- Aussage,
- XDistanz,
- XFaktor,
- Herkunft,
- Reihenart,

- Version
- Quelle sowie
- ParMerkmal

Sie werden auch als „Identifikationsattribute“ bezeichnet. Eine Liste mit möglichen Werten aller Attribute ist im Anhang zusammengestellt.

Wenn die Stammdaten entsprechend eingerichtet sind, können Zeitreihen auch statt mit einem Attributemuster mit einer PV (Prozessvariablen) aufgesucht werden. Beispiel:

<http://www.tstp.net:8030/?Cmd=Query&PV=BW001S090T56>

Das Ergebnis ist eine XML-Seite, die von einem TSQ-Element eingeschlossen wird. Darin enthalten ist eine Liste von TSATTR-Elementen (pro passender Zeitreihe eins), die wiederum alle Attribute der Zeitreihe enthält. Pro Attribut wird ein XML-Node ausgegeben, der `<Attributname>Wert</Attributname>` enthält.

Beispiel:

Als Ergebnis der Anfrage von oben wird geliefert:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSQ RELEASE="1">
  <TSATTR>
    <ZRID>PK8n4XrPPUfYpndH6GLH6A</ZRID>
    <MAXFOCUS-Start>1960-05-06T12:02:00Z</MAXFOCUS-Start>
    <MAXFOCUS-End>2001-08-01T00:00:00Z</MAXFOCUS-End>
    <MAXQUAL>1</MAXQUAL>
    <PARAMETER>Wasserstand</PARAMETER>
    <ORT>24004501</ORT>
    <DEFART>K</DEFART>
    <AUSSAGE></AUSSAGE>
    <XDISTANZ>E</XDISTANZ>
    <XFAKTOR>1</XFAKTOR>
    <HERKUNFT>0</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>J7v2xiGggfJNPSCJNaTtFw</ZRID>
  <MAXFOCUS-Start>1950-05-29T11:33:00Z</MAXFOCUS-Start>
  <MAXFOCUS-End>1999-06</MAXFOCUS-End>
  <MAXQUAL>3</MAXQUAL>
  <PARAMETER>Wasserstand</PARAMETER>
  <ORT>24006008</ORT>
  <DEFART>K</DEFART>
  <AUSSAGE></AUSSAGE>
  <XDISTANZ>E</XDISTANZ>
  <XFAKTOR>1</XFAKTOR>
  <HERKUNFT>0</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</FTOLREL>
  <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>

```

Alternativ reicht eine Angabe der Zeitreihen-ID als Parameter. Über diese Abfrage kommt man an die zu einer Zeitreihe gehörenden Attribute.

Als URL kodiert würde die Übergabe an den TSTP-Server folgendermaßen aussehen:

```
http://www.tstp.net:8030/?Cmd=Query&ZRID=pYMb9a2wMHkKtNf7ElfW0g
```

Zurückgeliefert wird dann eine XML-Seite, welches die angefragte Zeitreihe enthält.

## 4.2 GET

Holt Daten (Wertepaare, Zeit - Zahl). Texte (Zeit - Text) werden mit GETCOMBO geholt. Eine Zeitreihe wird mithilfe einer ZRID angefordert. Zusätzlich muss ein Zeitintervall mit Von=<Zeitpunkt> und Bis=<Zeitpunkt> angegeben werden, optional kann eine Qualitätsstufe mit Qual=[0..47] angegeben werden. Als Default wird die höchste vorliegende Qualitätsstufe ausgewählt. Geliefert wird ein Base64-kodierter Binärblock, der die Liste der Wertepaare enthält. Als XML-Attribute des DEF-Elements werden „DefArt“, „Reihenart“ und „Einheit“ ausgegeben.

Die Übertragung kann alternativ in ASCII-Form erfolgen, was über den Parameter Typ=Asc gewählt wird.

URL-Beispiel:

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

Das Attribut ANZ gibt die Zahl der Wertepaare bei der ASCII-Übertragung an, das Attribut LEN die Länge des Base64-dekodierten Datenblocks bei der Binärübertragung.

Beispiel 1:

Bei Übermittlung im ASCII-Format ist die Antwort folgendermaßen strukturiert:

```

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

Die Daten werden durch ein einfaches Linefeed (ASCII-Char 10) getrennt.

Beispiel 2:

Bei Übermittlung im Binärformat hingegen folgendermaßen:

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

<base64-daten> steht hierbei für die eigentlichen, Base64-kodierten Daten.

Die Daten werden fortlaufend nach Tabelle 1 verschlüsselt übermittelt. Zeitpunkte (zzzzzzzz) belegen 8 Bytes, Realzahlen (ffff) 4 Bytes. Anschließend wird dieser Binärblock Base64-kodiert, damit die Daten XML-konform werden. Der so entstandene Base64-kodierte Block wird dann nach jeweils 60 Zeichen mit einem LineFeed (ASCII-Char 10) versehen, um die Länge der Zeilen kurz zu halten. Dies ist wichtig, im Hinblick auf eine XML-Validierung, bei der die Zeilenlänge beschränkt ist. Außerdem erhöht es die Lesbarkeit in einem Browser.

Die Wertepaare im Binärblock schließen nahtlos aneinander an. Je nach Anfrage kann der Binärblock also durchaus ein paar hundert Kilobyte groß sein.

### 4.3 PUT

Eine Zeitreihe wird mithilfe einer ZRID ausgewählt. Optional kann eine Qualitätsstufe angegeben werden. In einer per POST an den TSTP-Server übergebenen XML-Nachricht werden dann die Daten übertragen und auf dem Server eingegliedert.

Beispiel:

Im HTTP-Header müssen folgende Parameter enthalten sein:

```
POST /?Cmd=PUT&ZRID=123&QUAL=2 HTTP/1.0\n
Host: www.tstp.net:8030\n
Authorization: Basic <user:pass>\n
Content-Length: <Länge des kompletten XML-Anhangs>\n
Connection: Close\n\n
```

Der darauf folgende <Binaerblock> muss folgendermaßen aussehen:

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

<base64-daten> steht hierbei für die eigentlichen, Base64-kodierten Daten.

Die Rückmeldung erfolgt in Form einer XML-Meldung. Als Bestätigung wird

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

übermittelt, im Fall eines Fehlers

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
```

<Fehlermeldung> steht hierbei für den Text der Meldung.

### Spezialfall Niederschlag

Das Übertragen von Niederschlagsdaten an den Server stellt eine Besonderheit dar. Die Zeitreihen werden als Intensitäten in mm/h gespeichert. Oft liegen die Daten auf der Seite des Clients aber als Summenlinie oder als Impulse vor. Daten dieser Art können verschickt werden, wenn deren Art bekannt gemacht wird. Das erfolgt mit dem Attribut MESAUS des DEF-Element. MESAUS kann vier Zustände haben:

- **SUMLIN** die Werte in wpl stellen eine Summenlinie dar, sind also monoton steigend. Der erste Wert muss nicht 0 sein.
- **SUML0** die Werte in wpl stellen eine Summenlinie dar, die an beliebiger Stelle wieder auf 0 zurückfallen kann. Die 0 selbst muss aber nicht in den Daten vorkommen. Das heißt, wenn ein Wert  $w$  kleiner als der vorherige ist, wird ein Nullsetzen und ein daran anschließendes Ansteigen bis  $w$  angenommen. Der erste Wert muss nicht 0 sein. Wenn zwischen dem Ende des letzten Transfers und dem Anfang der Daten Niederschlag gefallen ist, so muss wpl links um den letzten Wert des letzten Transfers ergänzt werden.
- **DELTA** die Werte stellen Differenzen zum vorherigen Wert dar.
- **INTENS** die Werte stellen Intensitäten in mm/h dar.

**Alle** Angaben (außer für INTENS) müssen in mm erfolgen!

Das Binärformat für Niederschlag sieht z.B. so aus:

```
<?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="SUMLO"/>
  <DATA><! [CDATA[<base64-daten>]]></DATA>
</TSD>
```



### Spezialfall Abfluss in Wasserstand umrechnen

Es ist möglich, Abflussdaten an eine Wasserstandsreihe zu schicken. Die Abflussdaten werden mithilfe der inversen Abflusskurven in Wasserstand zurück gerechnet. Die Abfolge der Abflusskurven, die in der Zeitreihe Abflusskurven bzw. Pegelschlüssel definiert ist, wird beachtet. Stauwerte und das Eta-Verfahren werden nicht angewendet. Alle beteiligten Reihen werden automatisch herangezogen. Das Attribut Q2W des DEF-Elements muss auf True stehen. Die Einheit der Abflussdaten muss mit denen der Abflusskurven übereinstimmen. Beispiel:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" LEN="240" ANZ="20" Q2W="True"/>
  <DATA><! [CDATA[<base64-daten>]]></DATA>
</TSD>
```

## 4.4 Arbeitsweise der Zeitreihen-Datenbank beim Einfügen

Zeitreihen besitzen keinen Start und kein Ende, sie liegen von „-Unendlich“ bis „+Unendlich“ vor. Die zugehörigen Zeitpunkte sind +Infty und -Infty. Bereiche, auf denen die Daten nicht bekannt sind, enthalten den Wert Luecke. Generell sind Lücken reguläre Y-Werte, die keine Sonderrolle spielen.

Jeder Schreibvorgang auf Zeitreihen ist daher ein Einfügen. Dieses wird im Folgenden beschrieben.

Intervalle erstrecken sich **immer** vom Knickpunkt zurück zum vorherigen Knickpunkt. Tageswerte sind beispielsweise am Ende des Tages abgelegt.

Dem Einfügevorgang liegt der Gedanke zugrunde, dass die einzufügenden Daten sich auf der Zeitspanne, die durch sie definiert ist, vollständig durchsetzen und **alle** anderen Zeitbereiche nicht berührt werden.

Zeitreihen speichern keine Wertepaare, sondern Polygonzüge. An jedem Zeitpunkt, ob er nun Knickpunkt ist oder nicht, ist ein Y-Wert definiert. Die Knickpunkte sind linear (kontinuierliche ZR) oder sprunghaft (Intervall-ZR) verbunden. In Momentan-ZR sind die Y-Werte nicht verbunden, bilden aber dennoch ein geschlossene Einheit. Beim Einfügen werden alle bestehenden Knickpunkte auf dem Bereich gelöscht.

Wenn die einzufügenden Daten links oder rechts nicht genau auf einen Knickpunkt fallen, werden die Rändern angeglichen. Das geschieht auf folgende Weise:

- *kontinuierliche ZR*: Es wird ein Sprung mit der Breite 5s erzeugt. Der Polygonzug wird dazu an der Nahtstelle interpoliert, dieser Wert dann 5s vor/nach dem Einfügebereich eingefügt.
- *Intervall-ZR*: Der erste Altwert rechts vom Einfügebereich wird an Stelle des ersten Neuwerts eingefügt. Der Y-Wert des ersten Wertepaars wird also verworfen. Ledig-

lich der Zeitpunkt ist von Belang. Implizit ändert sich ggf. das dem Einfügebereich folgende Intervall dadurch, das der neue vorherige Zeitpunkt ein anderer wird.

- *Momentan-ZR*: Es findet kein Ausgleich statt.

## 4.5 SETATTR

Ein Attribut der durch ZRID ausgewählten Zeitreihe wird gesetzt. Identifikationsattribute können **nicht** gesetzt werden. Neben den Attributen der Reihe können auch andere Eigenschaften der Reihe gesetzt werden, nämlich INFO und LEBENSLAUF.

URL-Beispiel:

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

Als Bestätigung wird

```
<TSR>confirm</TSR>
```

übermittelt, im Fall eines Fehlers

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
```

<Fehlermeldung> steht hierbei für den Text der Meldung.

## 4.6 CREATE

Vorgegeben wird ein kompletter, z.B. als URL kodierter Attributesatz. Zurückgeliefert wird die ZRID der Zeitreihe, die angelegt (oder im Datenbestand gefunden) wurde. Die ZRID erhält den Wert 0, falls das Anlegen fehlschlug.

URL-Beispiel:

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

Zurückübermittelt wird die ZRID als

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

oder im Fehlerfall

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

<Fehlermeldung> steht hierbei für den Text der Meldung.

## 4.7 DELETE

Vorgegeben wird die ZRID der zu löschenden Zeitreihe.

URL-Beispiel:

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

Zurückgeliefert wird

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

um das erfolgreiche Löschen zu bestätigen oder

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
```

falls das Löschen fehlschlug. <Fehlermeldung> steht hierbei für den Text der Meldung.

## 4.8 DELETEQUAL

Vorgegeben werden eine ZRID, ein Zeitintervall und eine Qualitätsstufe. Die Qualität wird auf dem Intervall gelöscht.

URL-Beispiel:

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

Zurückgeliefert wird entsprechend

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

oder

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
```

<Fehlermeldung> steht hierbei für den Text der Meldung.

## 4.9 INSPECT

Vorgegeben werden eine ZRID und optional ein Zeitintervall. Geliefert werden die maximale Qualität und die maximale physikalische Qualität auf dem Intervall, der Lebenslauf, die permanent gespeicherte Info der Zeitreihe und der Zeitpunkt der letzten Änderung an der Reihe.

Ohne Angabe eines Zeitintervalls werden die maximalen Qualitäten des MaxFocus' der Reihe geliefert.

URL-Beispiel:

```
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>
  <LEBENS LAUF><![CDATA[Base64-Daten]]></LEBENS LAUF>
  <INFO><![CDATA[Base64-Daten]]></INFO>
  <TIMESTAMP>2009.11.10T20:03:23Z</TIMESTAMP>
</TSR>
```

oder

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
<Fehlermeldung> steht hierbei für den Text der Meldung.
```

## 4.10 QNUM

Vorgegeben werden eine ZRID und optional ein Zeitintervall. Geliefert wird die Anzahl Quanten (Werte) der Reihe auf dem Zeitintervall. Ohne dessen Angabe wird die Gesamtanzahl geliefert.

URL-Beispiel:

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

Zurückgeliefert wird in der Form:

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

oder

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
<Fehlermeldung> steht hierbei für den Text der Meldung.
```

## 4.11 GETDVAL

Vorgegeben werden eine ZRID, ein Zeitintervall mit Von=<Zeitpunkt> und Bis=<Zeitpunkt>, eine Intervallbreite mit IB=<Intervallbreite>, eine Aussage mit Aussage=<Aussage> (z.B. Mit - siehe Tabelle 3 in Anhang A) und optional eine Qualitätsstufe mit Qual=[0..50].

URL-Beispiel:

```
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
```

Geliefert wird eine Zeitreihe mit entsprechenden Werten (Intervallwerte, kontinuierliche Werte oder Momentanwerte) für das Zeitintervall als Binärblock (siehe GET). Die gewählte Intervallbreite IB legt die Attribute X**Faktor** und X**Distanz** der zurückgelieferten Zeitreihe fest.

## 4.12 GETCOMBO

In der Zeitreihenmaschine werden drei Quantenfolgen vorgehalten: die Quantenfolge für Zahlen, die für Texte und die Null-Quantenfolge (für isolierte Punkte). Mit GETCOMBO können diese mit einem einzigen Befehl geholt werden. GETCOMBO wird wie GET benutzt. Die Rückgabe besteht jedoch aus drei TSD-Elementen, die hintereinander ausgegeben werden. Das erste enthält die Zahl-Quantenfolge, darauf folgt die Text-Quantenfolge und dann die Null-Quantenfolge.

Zusätzlich zu den Feldern des Befehls GET (VON, BIS, QUAL) gibt es das Feld READMODE, das die Werte INTERPOLIERT, INNEN oder AUSSEN annehmen kann. Voreinstellung ist INTERPOLIERT.

## 4.13 UPDATE

Vorgegeben wird eine ZRID. Der Server liest die Zeitreihe neu ein und aktualisiert den MaxFocus. Wenn die Zeitreihe geändert wurde, ohne dass der Server dies mitbekommen hat, muss dieser Befehl angewendet werden.

URL-Beispiel:

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

Zurückgeliefert wird

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

um das erfolgreiche Update zu bestätigen oder

```
<TSR RELEASE="1"><ERR><Fehlermeldung></ERR></TSR>
```

falls das Update fehlschlug (z.B. wegen falscher ZRID oder fehlender Zugriffsrechte).

<Fehlermeldung> steht hierbei für den Text der Meldung.

## 4.14 GLAMP

Berechnet gleitende Amplituden. Vorgegeben werden eine ZRID, ein Zeitintervall mit Von=Zeitpunkt und Bis=Zeitpunkt, sowie eine Intervallbreite mit IB=Intervallbreite

URL-Beispiel:

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

Geliefert wird eine kontinuierliche Zeitreihe mit den gleitenden Amplituden der Ausgangsreihe, also den Differenzen der gleitenden Maxima und gleitenden Minima. Diese gleitenden Werte werden berechnet, indem jeder Wert durch das Minimum bzw. Maximum der Zeitreihe innerhalb eines Intervalls (im Beispiel 30 Minuten) zentriert um den jeweiligen Zeitpunkt ersetzt wird.

## 4.15 FORMEL

Eine Formel wird an den TSTP-Server gesendet, mit der Zeitreihen umgerechnet und verknüpft werden können. Die Formel wird dabei als Parameter **TERM** übergeben, Die zu verknüpfenden Zeitreihen werden durch Komma getrennt mit dem Parameter **ZRID** in der gleichen Reihenfolge übergeben, wie sie in der Formel auftauchen.

Beispiel:

```
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
```

Optional können auch noch die Parameter **EINHEIT** und **TYP** (z.B. für **Image**) angegeben werden.

Geliefert werden die durch die Verknüpfung entstandenen Zeitreihenwerte.

## 4.16 LAYERGET

Dem Server wird ein Geolayer, der Name einer Raster-ZR-Datenbank, der gewünschte Parameter, der Fokus und ggf. der Feldname und die Länge der ersten Datei (**LEN1**) (für **SHP**, siehe unten) übermittelt. Für alle Polygone aus dem Layer wird jeweils aus der Raster-ZRDB aus dem angegebenen Parameter eine Quantenfolge generiert. Ist Parameter leer, wird der erste Parameter genommen.

Jeder Quantenfolgenblock enthält im Header auch den Parameter und den Ort. Der Parameter wird der Raster-ZRDB entnommen, der Ort dem Namen des jeweiligen Polygons.

Für Flächenpolygone wird aus den überdeckten Kacheln der Mittelwert gebildet, angeschnittene Kacheln werden anteilig berücksichtigt. Für Punktpolygone wird ermittelt, in welcher Kachel sich der Punkt befindet, deren Wert dann herangezogen wird. Linienpolygone werden nicht unterstützt.

Der GeoLayer kann in den Formaten **BIN**, **GEO**, **AI** oder **SHP** vorliegen. **SHP**-Layer werden aus zwei Dateien zusammengesetzt, der **shp**-Datei mit Koordinaten und der **dbf**-Datei mit Attributen pro Polygon. In der **dbf**-Datei gibt es pro Polygon mehrere Felder. Aus einem davon wird der Name der Polygone gebildet. Dessen Name wird mit in der URL übergeben. Im Feld **LEN1** wird die Länge der ersten Datei in Bytes übermittelt.

## 4.17 RASTERGET

Dem Server werden die Koordinaten X und Y, der Name einer Raster-ZR-Datenbank, der Fokus und ggf. der gewünschte Parameter übermittelt. Für den Punkt (X,Y) wird aus der Raster-ZRDB aus dem angegebenen Parameter eine Quantenfolge generiert. Ist Parameter leer, wird der erste Parameter genommen. Optional kann mit dem Parameter ENTKOL gesteuert werden, ob die Daten entkollinearisiert werden sollen, ENTKOL=True. Standard ist es, nicht zu entkollinearisieren, ohnehin geschieht dies nur bei kontinuierlichen Raster-Datenbanken.

## 4.18 IMAGE

Der Server produziert eine Grafik im PNG-Format. Die Funktionsweise ist ansonsten so wie beim Befehl GET. Die zusätzlichen, optionalen Felder XSIZE und YSIZE legen die Größe des Bildes fest. Die Voreinstellung ist 900x640.

# 5 Format der Anfrage

## 5.1 Authentifizierung

Im HTTP-Header einer Anfrage werden in einer Zeile Benutzername und Password im Format „Authorization: Basic username:password“ übergeben, wobei die Zeichenkette „username:password“ nach Base64 (RFC 3548, Kapitel 3) kodiert wird.

Beispiel:

```
GET ?Cmd=Query&Parameter=Wasserstand&Ort=2400*&DefArt=K HTTP/1.0
Authorization: Basic dXNlcm5hbWU6cGFzZ3dvcmlQ
```

Vom TSTP-Server wird der Benutzer mit dem angegebenen Passwort überprüft. Sollte das Passwort falsch sein oder der Benutzer nicht existieren, wird eine „HTTP-401“-Rückmeldung generiert. In einem Browser führt dies zu einer Login-Maske.

Den Benutzern sind Rechte zum Benutzen von Zeitreihen zugeteilt. Ein Benutzer kann entweder

- nur Lesen,
- Lesen/Schreiben oder
- Lesen/Schreiben/Anlegen/Löschen.

## 5.2 URL-Format

Die Kodierung der in einer URL-Anfrage übergebenen Parameter wird mit einem Fragezeichen eingeleitet:

URL-Beispiel:

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

Gefolgt von der Funktion:

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

URL-Beispiel:

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

Zur URL-Kodierung von Attributen wird zwischen Attributnamen und Attributwert das Zeichen = eingefügt, z.B. `Ort=24003100`. Weitere Paare von Attributnamen und -werten werden durch das Zeichen & abgetrennt.

Beispiel:

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

Die möglichen Werte der Attribute `DefArt`, `Aussage`, `XDistanz`, `Herkunft`, `Reihenart` und `Quelle` sind in den Tabellen 2 bis 6 im Anhang A zusammengestellt.

## 5.3 Zeitpunkt

Nach ISO 8601, dem internationalen Standard für Datum und Uhrzeit, wird hier dieses Format verwendet. Das TSTP-Protokoll gibt nur dieses Format aus.

Komplettes Datum plus Stunden, Minuten und Sekunden:

`YYYY-MM-DDThh:mm:ssZ` (z.B. `1997-07-16T19:20:30Z`)

Beschreibung:

YYYY	=	4-stelliges Jahr
MM	=	2-stelliger Monat (01=Januar, usw.)
DD	=	2-stelliger Tag des Monats (01 bis 31)
'T'	=	Trennzeichen Datum/Uhrzeit
hh	=	2-stellige Stunden des Tages (00 bis 23)
mm	=	2-stellige Minuten der Stunde (00 bis 59)
ss	=	2-stellige Sekunden der Minute (00 bis 59)
'Z'	=	Zeitzone (in TSTP immer 'Z')

Alternativ wird das folgende Format als Eingabe erkannt:

Zeitpunkte werden in der Form

`<Tag>.<Monat>.<Jahr>[_<Stunde>:<Minute>[:<Sekunde>]]`

kodiert, d.h. die Angabe der Uhrzeit, sowie die Angabe der Sekunden innerhalb der Uhrzeit sind optional.



## 5.4 Zeitintervall

Bei Zeitintervallen wird mit `Von=<Zeitpunkt>` der Startzeitpunkt und mit `Bis=<Zeitpunkt>` der Endzeitpunkt angegeben.

## 6 XML-Format

XML-kodierte Antworten des TSTP-Servers beginnen mit einem HTTP-Header, gefolgt von den XML-kodierten Daten.

Beispiel:

```
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"?>
<TSR RELAESE="1">
  ...
</TSR>
```

Die Zeilen sind mit einem Linefeed (ASCII-Char 10) abgeschlossen.

### 6.1 Attributliste

Als Antwort auf eine QUERY-Anfrage wird das Wurzelement `<TSQ RELEASE="1">` zurückgesendet. Innerhalb des `<TSQ>`-Elements ist für jede Zeitreihe ein `<TSATTR>`-Element enthalten. Die einzelnen Attribute sind Unterelemente im `<TSATTR>`-Element:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSQ RELEASE="1">
  <TSATTR>
    <ZRID>dN28eSEVIMADyNScIH5BRA</ZRID>
    <MAXFOCUS_START>1964-06-01T00:00:00Z</MAXFOCUS_START>
    <MAXFOCUS_END>2005-11-01T00:00:00Z</MAXFOCUS_END>
    <MAXQUAL>0</MAXQUAL>
    <PARAMETER>Abflusskurven</PARAMETER>
```

```

    <ORT>24004002</ORT>
    ..
  </TSATTR>
  <TSATTR>
    <ZRID>PBtPwnBsFZb9jawtmc61bg</ZRID>
    ..
  </TSATTR>
</TSQ>

```

## 6.2 Massendaten

Massendaten sind im Wurzelement `<TSD RELEASE="1">` eingeschlossen. Im Wurzelement wird die Version kodiert als `RELEASE="<Versionsnummer>"`. Weiterhin enthalten ist ein `<DEF>`-Element mit den Header-Attributen der Zeitreihe sowie eine zeilenweise Liste von Zeitpunkten und Y-Werten oder ein Binärblock.

Die Header-Attribute enthalten

- die Reihenart kodiert als `REIHENART="<Reihenart>"` (z.B. Z),
- ein Flag, das angibt ob Textattribute mitgeliefert werden (z.B. `TEXT="Nein"`),
- die Definitionsart kodiert als `DEFART="<Definitionsart>"` (z.B. K,I,M),
- die physikalische Einheit kodiert als `EINHEIT="<Einheit>"` (z.B.  $m^3/s$ ) sowie
- die Anzahl der Wertepaare kodiert als `LEN="<Anzahl>"`.

Standardmäßig werden Massendaten binär übertragen (siehe unten). Es ist aber (z.B. zum Testen) möglich, die Daten im lesbaren ASCII-Format anzufordern. In der Wertepaarliste wird der Zeitpunkt im Format `YYYY.MM.DDThh:mm:ssZ` angegeben. (Siehe Kapitel 5.3, „Zeitpunkt“.)

Bei den Y-Werten wird das Zeichen ‘.’ als Dezimalpunkt verwendet. Zeilen werden mit dem ASCII-Zeichen 10 (Linefeed) abgeschlossen.

Beispiel:

```

<?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 Binärformat

### 7.1 Massendaten

Massendaten werden mit dem gleichen Header wie ASCII-Daten verschickt. Die in `<DATA><![CDATA[ und ]]></DATA>` eingeschlossenen Daten sind jedoch binär. `LEN` gibt die Anzahl der Bytes des Binärblocks an, `ANZ` die Anzahl der Wertepaare.

Beispiel:

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<TSD RELEASE="1">
  <DEF REIHENART="Z" TEXT="Nein"
    DEFART="K" EINHEIT="mm/h" LEN="24" ANZ="2" />
  <DATA><![CDATA[Base64-Daten]]></DATA>
</TSD>
```

Die Anordnung der Bytes innerhalb eines 32-bit-Wortes entspricht der „network byte order“ (3-2-1-0).

Die Daten bestehen aus einer Folge von Wertepaaren, die nahtlos aneinander anschließen. Ein Wertepaar besteht aus `XY`, wobei `X` einem 64-bit Zeit- oder Realpunkt und `Y` einer 32-bit IEEE-Fließkommazahl oder einem String entspricht. Der `Y`-Wert `4E+37` (hexadezimal `C2 BD F0 7D` - nicht Base64!) repräsentiert eine Lücke.

Anhand des für alle Wertepaare des Blocks geltenden Attributs `TEXT` in der Kennung `DEF` wird festgelegt, ob die `Y`-Werte Real oder String sind.

Texte werden folgendermaßen abgelegt: im ersten Byte steht die Variante des Stringaufbaus. 8 steht dabei für einen Leerstring, dem keine weiteren Daten folgen. Variante 6 leitet einen kurzen String ein, dessen Länge im nächsten Byte abgelegt ist, und Variante 7 dient für längere Texte, deren Länge in den nächsten vier Bytes abgelegt ist (network byte order). Bei den Varianten 6 und 7 folgt direkt hinter der Längenangabe der eigentliche Text.

Die Zeit- bzw. Realpunkte sind entsprechend Tabelle 1 kodiert, es werden drei Modi unterschieden: Zeitpunkt (0), Realpunkt (1) und mikrosekundengenauer Zeitpunkt (2).

Binärformat Zeit-/Realpunkt															
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						
<b>0:</b> Zeitpunkt	Qualitätsmerkmal	0: normal 1: $-\infty$ 2: $+\infty$	Jahr	Jahr	Jahr	Jahr	Monat		Tag	Stunde	Minute	Sekunde			
<b>1:</b> Realpunkt		0: normal 1: $-\infty$ 2: $+\infty$						32-bit IEEE float							
<b>2:</b> MilliZP	Qualitätsmerkmal	ms	Jahr	Jahr	Jahr	Jahr	Monat	ms	Tag	Stunde	Minute	Sekunde			

Tabelle 1: Kodierung der Zeit- bzw. Realpunkte im Binärformat.

Die Angabe ms steht für Millisekunden und umfasst insgesamt 10bit (Bit 9-2 in Byte 6 und Bit 1-0 in Byte 3)

## Anhang A

DefArt		
Erlaubter Wert	Bedeutung	URL-Kodierung
K	Kontinuierlich	DefArt=K
I	Intervall	DefArt=I
M	Momentan	DefArt=M

Tabelle 2: Mögliche Werte des Attributs „Definitionsart“.

Aussage			
Erlaubter Wert	Bedeutung	URL-Kodierung	Mit GetDVal nutzbar
Mes	(Hand-)Messungen	Aussage=Mes	
Sum	Summen	Aussage=Sum	Ja
Mit	Mittelwerte	Aussage=Mit	Ja
Max	Maxima	Aussage=Max	Ja
Min	Minima	Aussage=Min	Ja
DMax	Maxima-Zeitpunkte	Aussage=DMax	Ja
DMin	Minima-Zeitpunkte	Aussage=DMin	Ja
Dif	Intervall-Differenzen (rechts-links))	Aussage=DIF	Ja
Abl	Ableitungen	Aussage=Abl	
Int	Integral	Aussage=Int	
Dau	Dauerlinie	Aussage=Dau	
PDF	Wahrscheinlichkeits-Dichte	Aussage=PDF	
VF	Verteilungsfunktion	Aussage=VF	
Lck	Lückenreihe	Aussage=Lck	Ja
Ant	Lückenanteile	Aussage=Ant	
OS	Obere Schranke	Aussage=OS	
US	Untere Schranke	Aussage=US	
pSe	Partielle Serie	Aussage=pSe	
jSe	Jährliche Serie	Aussage=jSe	
pVP	Partielle Verteilungspar.	Aussage=pVP	
jVP	Jährliche Verteilungspar.	Aussage=jVP	
ERG	Ereignisse	Aussage=ERG	
NER	NEreignisse	Aussage=NER	
jSN	J. Ser. N.	Aussage=jSN	
pSN	Part. Ser. N.	Aussage=pSN	
Kon	Kontrollwerte	Aussage=Kon	
XSy	X-Synchropunkte	Aussage=XSy	
YSy	Y-Synchropunkte	Aussage=YSy	

Tabelle 3: Mögliche Werte des Attributs „Aussage“. Die Aussagen DMin und DMax liefern eine Momentan-Zeitreihe mit den Zeitpunkten, an denen die Extremwerte erreicht wurden.

Herkunft		
Erlaubter Wert	Bedeutung	URL-Kodierung
O	original	Herkunft=O
T	transformiert	Herkunft=T
S	simuliert	Herkunft=S
A	abgeleitet	Herkunft=A
M	temporär	Herkunft=M
B	trendbereinigt	Herkunft=B
F	ZRFolge	Herkunft=F
P	periodisch	Herkunft=P

Tabelle 4: Mögliche Werte des Attributs „Herkunft“.

Reihenart		
Erlaubter Wert	Bedeutung	URL-Kodierung
Z	Zeitreihe	Reihenart=Z
R	Realreihe	Reihenart=R

Tabelle 5: Mögliche Werte des Attributs „Reihenart“.

Quelle		
Erlaubter Wert	Bedeutung	URL-Kodierung
L	Datenlogger	Quelle=L
S	Schreibstreifen	Quelle=S
H	Handeingabe	Quelle=H
P	Produktion	Quelle=P
T	Testversion	Quelle=T
	(leer)	Quelle=

Tabelle 6: Mögliche Werte des Attributs „Quelle“.