

1 Aufbau

Die Konfigurationsdatei ist eine einfache, ANSI-codierte Textdatei.

Sie besteht aus vier Elementen: Kommentaren, Selektoren, Attributen und Werten.

1.1 Kommentare

Kommentare werden mit einem Nummernzeichen (#) eingeleitet. Der Inhalt der Zeile rechts des Nummernzeichens bis zum nächsten CR (Carriage Return, ASCII-Code 13) wird ignoriert.

Beispiel:

```
# Sample comment
```

Möchten Sie mehrzeilige Kommentare verwenden, muß vor jeder Zeile ein Nummernzeichen stehen!

```
# Sample configuration file
board {
};
sitable {
};
modulator {
};
transportstream 1 {
};
transportstream 2 {
};
transportstream 3 {
};
transportstream 4 {
};
external program {
    video stream {
    }
    audio stream {
    }
    teletext stream {
    }
};
```

Abb. 1.a: Typischer Aufbau einer Konfigurationsdatei

1.2 Selektoren

Selektoren sind das ordnende Element in einer Konfigurationsdatei, sie unterteilen die Konfigurationsdatei in verschiedene Abschnitte, die unterschiedliche Funktionen konfigurieren. Ein Selektor beginnt stets seinem Namen, gefolgt von geschweiften Klammern und wird mit einem Semikolon abgeschlossen.

Beispiel:

```
selektor {};
```

Selektoren können auch verschachtelt werden:

Beispiel:

```
selektor1 {
    selektor1a {};;
    selektor1b {};;
};
```

Zwischen den geschweiften Klammern erfolgt die eigentliche Konfiguration, hier stehen die Attribute mit den zugehörigen Werten.

<i>Selektorname</i>	<i>gültig für</i>
board	Text für Boards, wie, wo, was und überhaupt und so. Siehe Kapitel...
sitables	
modulator	
transportstream x	
external program	
video stream	
audio stream	
teletext stream	
Abb. 1.2a: Liste gültiger Selektoren	

1.3 Attribute und Werte

Ein Attribut-Wert-Pärchen wird eingeleitet durch den Attributnamen, gefolgt von einem Gleichheitszeichen und dem Wert, abgeschlossen wird mit einem Semikolon.

Beispiel:

```
attribut1 = wert1;
```

Attribut-Wert-Pärchen erscheinen nur innerhalb der geschweiften Klammer eines Selektors und gelten auch nur innerhalb des vom Selektor vorgegebenen Rahmens. So hat ein innerhalb von `transport stream 1 {}`; stehendes Attribut keinerlei Einfluß auf `transport stream 2 {}`; trotz eines dort evtl. stehenden gleichnamigen Attributes mit anderem Wert.

2 Gültige Attribute und Wertebereiche

2.1 Abschnitt „board“

Beispiel:

```
board {  
    board = mux4;  
    clock = 60M;  
};
```

2.1.1 Parameter „board“

Der Parameter *board* legt fest, welche Hardware konfiguriert werden soll. Stimmt der hier angegebene Wert nicht mit der angeschlossenen Hardware überein, wird die Konfiguration nicht funktionieren.

Beispiel:

```
board = mod2;
```

Das Board ist ein DVB-S/C-Modulator mit 2 TS-Eingängen

Mögliche Werte:

Wert	Wirkung
mod2	DVB-S/C-Modulator mit 2 TS-Eingängen
mod4	DVB-S/C-Modulator mit 4 TS-Eingängen
mux4	Multiplexer mit 2 TS-Eingängen

Gültige Werte:

Liste der gültigen Werte	mod2, mod4, mux4
Standardwert	Der Parameter <i>board</i> hat keinen Standardwert und muß immer angegeben sein.

2.1.2 Parameter „clock“

Der Parameter *clock* legt fest, für welchen Oszillatorfrequenz die Samplingraten und Datentaktsignale berechnet werden. Der hier angegebene Wert muss mit der Frequenz des auf der Hardware gesteckten Oszillators übereinstimmen.

Beispiel:

```
clock = 58M;
```

Auf dem Board ist ein 58 MHz-Oszillator gesteckt.

Gültige Werte:

kleinster gültiger Wert	24M
größter gültiger Wert	60M
Standardwert	60M

Alle Multiplexer und die meisten Modulatoren werden mit einem 60 MHz-Oszillator ausgeliefert. Wenn auf Ihrer Hardware ein Oszillator mit 60 MHz gesteckt ist, müssen Sie den Parameter *clock* nicht setzen.

2.2 Abschnitt „modulator“

2.2.1 für Modulatoren

Beispiele:

```
modulator {  
    modulation = dvb-s;  
    fec = 7/8;  
    frequency = 1250M;  
    symbol rate = 27500k;  
    ptt = off;  
};
```

```
modulator {  
    modulation = dvb-c;  
    inversion = off;  
    constellation = qam32;  
    frequency = 1280M;  
    symbol rate = 6900k;  
};
```

2.2.1.1 Parameter „modulation“

Der Parameter *modulation* legt fest, mit welcher Modulation der Modulator sendet.

Beispiel:

```
modulation = dvb-s;
```

Der Modulator sendet DVB-S (QPSK).

Mögliche Werte:

Wert	Wirkung
dvb-s	Der Modulator sendet DVB-S (QPSK)
dvb-c	Der Modulator sendet DVB-C (QAM)

Gültige Werte:

Liste der gültigen Werte	dvb-c, dvb-s
Standardwert	Der Parameter <i>modulation</i> hat keinen Standardwert und muß für Modulatoren immer gesetzt sein!

Für die Modulation mit DVB-S kann im Abschnitt *modulator* zusätzlich der Parameter *fec* gesetzt werden.

Für die Modulation mit DVB-C kann im Abschnitt *modulator* zusätzlich der Parameter *constellation* gesetzt werden.

Für die Modulation mit DVB-C und DVB-S können im Abschnitt *modulator* zusätzlich die folgenden Parameter gesetzt werden: *symbol rate*, *inversion*, *frequency*, *ptt*.

2.2.1.2 Parameter „fec“

Der Parameter *fec* legt fest, mit welcher Fehlerkorrekturrate (FEC-Rate) ein DVB-S-Signal moduliert werden soll. Für die DVB-C-Modulation hat der Parameter *fec* keine Bedeutung.

Beispiel:

```
fec = 1/2;
```

Modulation mit einer FEC-Rate von 1/2.

Gültige Werte:

kleinster gültiger Wert	1/2 (stärkste Fehlerkorrektur)
größter gültiger Wert	7/8 (schwächste Fehlerkorrektur)
Liste der gültigen Werte	1/2, 2/3, 3/4, 5/6, 7/8
Standardwert	1/2

2.2.1.3 Parameter „constellation“

Der Parameter *constellation* legt fest, mit welcher Konstellation ein DVB-C-Signal moduliert werden soll. Für die DVB-S-Modulation hat der Parameter *constellation* keine Bedeutung.

Beispiel:

```
constellation = qam16;
```

DVB-C-Modulation mit QAM 16.

Gültige Werte:

Liste der gültigen Werte	qam16, qam32, qam64
Standardwert	qam16

2.2.1.4 Parameter „symbol rate“

Der Parameter *symbol rate* bestimmt die Symbolrate des modulierten Signals.

Für die **DVB-C-Modulation** muss die Symbolrate der Frequenz des auf dem Board gesteckten Oszillators geteilt durch einen Teiler zwischen 8 und 16 entsprechen. Die Bandbreite des modulierten Signals entspricht der Symbolrate multipliziert mit 1,15.

$$\text{symbolrate} = \frac{f_{osc}}{\text{Teiler}}$$

$$\text{bandbreite} = \text{symbolrate} \cdot 1,15$$

So ergibt sich bei einer Oszillatorfrequenz von z.B. 60 MHz eine Symbolrate von 7500 kSym/s und eine Bandbreite von 8625 kHz.

Die Datenrate des DVB-C-Signals kann abhängig von Symbolrate und Bits pro Symbol wie folgt berechnet werden:

$$\text{bitrate} = \text{symbolrate} \cdot \text{bits_per_symbol} \cdot \frac{188}{204}$$

Der Wert *bits_per_symbol* ergibt sich aus der DVB-C-Konstellation wie folgt:

Konstellation	bits_per_symbol
qam16	4
qam32	5
qam64	6

Für das obige Beispiel einer Symbolrate von 7500 kSym/s ergeben sich also:

Konstellation	Bitrate (@ 7500 kSym/s)
qam16	27647 kBit/s
qam32	34559 kBit/s
qam64	41471 kBit/s

Gültige Werte:

Standardwert	Der Parameter <i>symbolrate</i> hat keinen Standardwert und muss für einen Modulator immer angegeben werden.
--------------	--

Für die **DVB-S-Modulation** muss die Symbolrate der Frequenz des auf dem Board gesteckten Oszillators geteilt durch einen der im folgenden angegebenen Teiler entsprechen. Gültige Teiler sind: 2, 2,1666..., 2,25, 2,333..., 2,5, 2,666..., 2,75, 3, 3,25, 3,5, 3,75, 4, 4,5, 5, 5,5, 6, 6,5, 7, 7,5, 8, 9, 10, 11, 12, 13, 14, 15, 16.

Die Bandbreite des modulierten Signals entspricht der Symbolrate multipliziert mit 1,35.

$$\text{bandbreite} = \text{symbolrate} \cdot 1,35$$

So ergibt sich bei einer Oszillatorfrequenz von z.B. 60 MHz eine Symbolrate von 7500 kSym/s und eine Bandbreite von 10125 kHz.

Die Datenrate des DVB-S-Signals kann abhängig von Symbolrate und FEC-Rate (siehe Kap. 2.2.1.3) wie folgt berechnet werden:

$$\text{bitrate} = 2 \cdot \text{symbolrate} \cdot \text{FECRate} \cdot \frac{188}{204}$$

Für das obige Beispiel einer Symbolrate von 7500 kSym/s ergeben sich also:

fec	Bitrate (@ 7500 kSym/s)
7 / 8	48382 kBit/s
5 / 6	46078 kBit/s
3 / 4	41471 kBit/s
2 / 3	36863 kBit/s
1 / 2	27647 kBit/s

Tabellen mit vorberechneten Werten finden Sie im Anhang.

2.2.1.5 Parameter „inversion“

Der Parameter *inversion* legt fest, ob das vom Modulator ausgegebene Spektrum invertiert wird.

Beispiel:

<code>inversion = off;</code>	Das modulierte Spektrum ist nicht invertiert.
-------------------------------	---

Mögliche Werte:

Wert	Wirkung
off	Das modulierte Spektrum ist nicht invertiert.
on	Das modulierte Spektrum ist invertiert.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	off

2.2.1.6 Parameter „frequency“

Der Parameter *frequency* bestimmt die Mittenfrequenz des modulierten Signals.

Abhängig davon, welcher I/Q-Modulator auf das Modulatorboard gesteckt ist, sind die folgenden Frequenzbereiche möglich.

Beispiel:

<code>frequency = 1275M;</code>	Die Mittenfrequenz des modulierten Signals beträgt 1275 MHz.
---------------------------------	--

Mögliche Werte:

Wert	Wirkung
TBD? ! ?	

Gültige Werte:

kleinster gültiger Wert	?
größter gültiger Wert	?
Liste der gültigen Werte	?
Standardwert	1275M

2.2.1.7 Parameter „ptt“

Der Parameter *ptt* legt fest, ob der Modulator nach dem Start automatisch zu senden beginnt. Wird *ptt* auf *off* gesetzt, muss die Modulation über die serielle Schnittstelle eingeschaltet werden.

Beispiel:

<code>ptt = on;</code>	Der Modulator beginnt nach dem Start automatisch zu senden.
------------------------	---

Mögliche Werte:

Wert	Wirkung
off	Der Modulator beginnt nach dem Start <i>nicht</i> automatisch zu senden.
on	Der Modulator beginnt nach dem Start automatisch zu senden.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	60M

2.2.2 für Multiplexer

Beispiele:

```
modulator {  
    drive ts clock = on;  
    bit rate = 20M;  
};
```

2.2.2.1 Parameter „drive ts clock“

Der Parameter *drive ts clock* legt fest, ob der Multiplexer das Taktsignal am Ausgang selbst erzeugt oder von außen übernimmt.

Beispiel:

```
drive ts clock = off;
```

Der Multiplexer erwartet das Signal am Ausgang.

Mögliche Werte:

Wert	Wirkung
off	Der Multiplexer erwartet das Taktsignal am Ausgang.
on	Der Multiplexer erzeugt ein Taktsignal am Ausgang.

Gültige Werte:

Standardwert	off
--------------	-----

2.2.2.2 Parameter „bit rate“

Der Parameter *bit rate* legt fest, welche Nettodatenrate am Ausgang des Multiplexers ausgegeben wird, wenn der Multiplexer das Taktsignal erzeugt (drive ts clock=on).



Achtung: Wenn *drive ts clock* nicht auf *on* gesetzt wird, ist der Parameter *bit rate* bedeutungslos. Die Datenrate am Ausgang wird dann durch das von außen angelegte Taktsignal bestimmt.

Der am Ausgang ausgegebene Takt berechnet sich wie folgt aus der festgelegten Bitrate:

$$clk_{out} = \frac{bitrate \cdot 204}{8 \cdot 188}$$

Ist der Parameter *drive ts clock* nicht auf *on* gesetzt, übernimmt der Multiplexer den Takt von außen. Die größtmögliche Nettobitrate berechnet sich dann wie folgt aus dem angelegten Takt:

$$bitrate = \frac{clk_{out} \cdot 8 \cdot 188}{204}$$

2.3 Abschnitt „sitable“

Der Abschnitt *sitable* konfiguriert den Tabellengenerator. Der Tabellengenerator ist nur aktiv, wenn der Abschnitt *sitable* in der Konfigurationsdatei vorkommt. Die PIDs unter 0x20 werden dann auf allen Eingängen automatisch blockiert, um zu verhindern, daß die vom Tabellengenerator erzeugten Tabellen durch eingehende Tabellen gestört werden.

Wenn Tabellen von einem TS-Eingang durchgereicht oder überhaupt keine Tabellen gesendet werden sollen, darf der Abschnitt *sitable* in der Konfigurationsdatei nicht vorkommen.



Achtung: Der Abschnitt *sitable* bezieht sich nur auf die PSI-Tabellen in den PIDs unter 0x20, nicht auf die PMT-Tabellen zu den einzelnen Services.

2.3.1 Parameter „psi delay“

Der Parameter *psi delay* legt den Abstand zwischen der Aussendung zweier Tabellenblöcke in Millisekunden fest.

Beispiel:

```
psi delay = 500;
```

Die PSI-Tabellen werden alle 500 ms ausgesendet.

Gültige Werte:

kleinster gültiger Wert	100
größter gültiger Wert	10000
Standardwert	500

2.3.2. Parameter „generate pat“

Der Parameter *generate pat* legt fest, ob der Tabellengenerator eine PAT (Program Allocation Table) auf der PID 0x00 erzeugt.

Beispiel:

```
generate pat = on;
```

Der Tabellengenerator erzeugt eine PAT auf der PID 0x00.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	on

2.3.3 Parameter „generate nit“

Der Parameter *generate nit* legt fest, ob der Tabellengenerator eine NIT (Network Information Table) auf der PID 0x10 erzeugt.

Beispiel:

```
generate nit = on;
```

Der Tabellengenerator erzeugt eine NIT auf der PID 0x10.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	on

2.3.4 Parameter „generate tdt“

Der Parameter *generate tdt* legt fest, ob der Tabellengenerator eine TDT (Time Definition Table) auf der PID 0x20 erzeugt.

Beispiel:

```
generate tdt = on;
```

Der Tabellengenerator erzeugt eine TDT auf der PID 0x20.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	on

2.3.5 Parameter „generate sdt“

Der Parameter *generate sdt* legt fest, ob der Tabellengenerator eine SDT (Service Definition Table) auf der PID 0x11 erzeugt.

Beispiel:

```
generate sdt = on;
```

Der Tabellengenerator erzeugt eine SDT auf der PID 0x11.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	on

2.3.6 Parameter „generate eit“

Der Parameter *generate ei* legt fest, ob der Tabellengenerator eine EIT (Event Information Table) auf der PID 0x12 erzeugt.

Beispiel:

```
generate sdt = on;
```

Der Tabellengenerator erzeugt eine SDT auf der PID 0x12.

Gültige Werte:

Liste der gültigen Werte	off, on
Standardwert	on

2.3.7 Parameter „transport stream id“

Der Parameter *transport stream id* legt fest, welche Transport-Stream-ID in die vom Tabellengenerator erzeugte PAT, NIT, SDT und EIT eingetragen wird..

Beispiel:

```
transport stream id = 0x1234;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0xFFFF
Standardwert	Wird der Parameter <i>transport stream id</i> nicht gesetzt, wird ein pseudozufälliger Wert, abhängig von Modulatorfrequenz und <i>network name</i> , als Transport-Stream-ID verwendet.

2.3.8 Parameter „network id“

Der Parameter *network id* legt fest, welche Transport-Stream-ID in die vom Tabellengenerator erzeugte PAT, NIT, SDT und EIT eingetragen wird..

Beispiel:

```
network id = 0xABCD;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0xFFFF
Standardwert	Wird der Parameter <i>transport stream id</i> nicht gesetzt, wird ein pseudozufälliger Wert, abhängig von <i>network name</i> , als Transport-Stream-ID verwendet.

2.3.9 Parameter „network name“

Der Parameter *network name* legt fest, welcher Network Name in die vom Tabellengenerator erzeugte NIT eingetragen wird.

Beispiel:

```
network name = "Yet another Network";
```

Gültige Werte:

maximale Länge der Zeichenkette	?!? Zeichen inkl. ""
Standardwert	""

2.3.10 Unterabschnitt „ca system“

Soll der Tabellengenerator eine CAT (Conditional Access Table) auf der PID 0x01 erzeugen, muß dem Abschnitt *sitables* ein Unterabschnitt *ca system* hinzugefügt werden.

Beispiel:

```
sitables{
    ...
    ca system{
        id = 0x1234;
        pid = 0x0180;
    };
};
```

2.3.10.1 Abschnitt „id“

Der Parameter *id* im Unterabschnitt *ca system* legt fest, welche CA System ID in die vom Tabellengenerator erzeugte CAT eingetragen wird.

Beispiel:

```
id = 0x2600;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0xFFFF
Standardwert	0x2600 (die CA System ID für BISS)

2.3.10.2 Abschnitt „pid

Der Parameter *pid* im Unterabschnitt *ca system* legt fest, welche CA System EMM PID in die vom Tabellengenerator erzeugte CAT eingetragen wird.

Beispiel:

```
id = 0x1FFF;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0x1FFF
Standardwert	0x1FFF (die EMM PID für BISS)

2.4 Abschnitt „transportstream“

Die TS-Eingänge von Modulator und Multiplexer werden jeweils über einen Abschnitt *transportstream n* konfiguriert. *n* steht dabei für die Nummer des jeweiligen Eingangsports zwischen 1 und 4 (4TS-Modulator und -Multiplexer) bzw. 1 und 2 (2TS-Modulator).

Findet sich für einen TS Eingang kein Abschnitt *transportstream* in der Konfigurationsdatei, wird der Port deaktiviert. Im Beispiel rechts würde der Eingang TS Input 3 deaktiviert.

Beispiel:

```
transportstream 1 {  
    mode = extclock;  
};  
transportstream 2 {  
    mode = extclock;  
};  
transportstream 4 {  
    mode = intclock;  
};
```

2.4.1 Parameter „mode“

Der Parameter mode legt fest, in welchem Modus der TS-Eingang betrieben wird.

Beispiel:

```
mode = off;
```

Der TS-Eingang ist deaktiviert.

Mögliche Werte:

Wert	Wirkung
off	Der TS-Eingang ist deaktiviert.
encoder	Am TS-Eingang wird ein MPEG-Encoder erwartet und gesteuert. Die folgenden Parameter und Unterabschnitte dienen zur Konfiguration des Encoders: <i>video input, video gop, spatial filter, bitrate, audio bitrate, audio mode, audio sample rate, pcr pid, video pid, audio pid, pmt pid, service id, service provider, service name, event name, event text, language, ca system</i> . Andere Parameter spielen für die Konfiguration des TS-Eingangs im Modus <i>encoder</i> keine Rolle.
extclock	Am TS-Eingang wird ein Transportstream erwartet – das Taktsignal wird von außen erwartet. Die folgenden Parameter können zusätzlich angegeben werden: <i>clock edge, clock filter</i> . Andere Parameter spielen für die Konfiguration des TS-Eingangs im Modus <i>extclock</i> keine Rolle.
intclock	Am TS-Eingang wird ein Transportstream erwartet – das Taktsignal wird vom Multiplexer bzw. Modulator erzeugt. Die folgenden Parameter können zusätzlich angegeben werden: <i>clock edge</i> . Andere Parameter spielen für die Konfiguration des TS-Eingangs im Modus <i>intclock</i> keine Rolle.
dvb-s	Am TS-Eingang wird ein DVB-S-Tuner (MB86A16) erwartet und gesteuert. Die folgenden Parameter dienen zur Konfiguration des Tuners: <i>tuner frequency, tuner symrate, tuner fec, tuner polarisation, tuner sat, tuner band, tuner inversion, tuner port disable</i> . Andere Parameter spielen für die Konfiguration des TS-Eingangs im Modus <i>dvb-s</i> keine Rolle.
dvb-c	Am TS-Eingang wird ein DVB-C-Tuner (STV0297) erwartet und gesteuert. Die folgenden Parameter dienen zur Konfiguration des Tuners: <i>tuner frequency, tuner symrate, tuner modulation, tuner inversion</i> . Andere Parameter spielen für die Konfiguration des TS-Eingangs im Modus <i>dvb-c</i> keine Rolle.
dvb-t	Am TS-Eingang wird ein DVB-T-Tuner (DiBCOM 3000) erwartet und gesteuert. Die folgenden Parameter dienen zur Konfiguration des Tuners: <i>tuner frequency, tuner ofdm bandwidth, tuner ofdm mode, tuner ofdm guard</i> . Andere Parameter spielen für die Konfiguration des TS-Eingangs im Modus <i>dvb-t</i> keine Rolle.



Achtung: die Modi *extclock*, *dvb-s*, *dvb-c* und *dvb-t* können ausschließlich an den TS-Eingängen 1 und 2 benutzt werden – für die TS-Eingänge 3 und 4 ist das nicht möglich!

Gültige Werte:

Liste der gültigen Werte	off, encoder, extclock, intclock, dvb-s, dvb-c, dvb-t
Standardwert	? ! ?

2.4.2 Parameter „clock edge“

Der Parameter *clock edge* legt fest, auf welcher Taktflanke der Modulator bzw. Multiplexer Daten am TS-Eingang übernimmt. Der Standardwert *rising* sollte nur in Ausnahmefällen geändert werden.

Beispiel:

```
clock edge = rising;
```

 Daten werden an steigender Taktflanke übernommen.

Gültige Werte:

Liste der gültigen Werte	rising, falling
Standardwert	rising

2.4.3 Parameter „clock filter“

Der Parameter *clock filter* legt fest, ob und wie stark das eingehende Taktsignal am TS Eingang gefiltert werden soll. Der Standardwert *1* sollte nur in Ausnahmefällen geändert werden..

Beispiel:

```
clock filter = 1;
```

Gültige Werte:

kleinster gültiger Wert	1
größter gültiger Wert	4
Liste der gültigen Werte	1, 2, 3, 4
Standardwert	1

2.4.4 Parameter „video input“

Der Parameter *video input* konfiguriert den Videoeingang des Encoders. Dazu müssen Bildgröße, Videonorm und Eingangsport konfiguriert werden.

Beispiel:

```
video input = <size> <mode> <port>
```

Mögliche Werte für <size>:

<size>	Wirkung
d1	
hd1	
sif	
qsif	

Mögliche Werte für <mode>:

<mode>	Wirkung
ntsc	
pal	
secam	
pal60	
paln	
ntsc_443	

Mögliche Werte für <port>:

Wert	Wirkung
composite	
svideo	
yc	
ycbcr	
rgb	
scart	
sdi	

2.4.5 Parameter „video gop“

Der Text.

Beispiel:

video gop = ?;	TBD
----------------	-----

Mögliche Werte:

Wert	Wirkung

Gültige Werte:

kleinster gültiger Wert	
größter gültiger Wert	
Liste der gültigen Werte	
Standardwert	

2.4.6 Parameter „spatial filter“

Der Text.

Beispiel:

spatial filter = ?;	TBD
---------------------	-----

Mögliche Werte:

Wert	Wirkung

Gültige Werte:

kleinster gültiger Wert	
größter gültiger Wert	
Liste der gültigen Werte	
Standardwert	

2.4.7 Parameter „bitrate“

Der Parameter *bitrate* legt die Systembitrate des MPEG-Encoders fest. Die Systembitrate ist die Summe aus *video bitrate* und *audio bitrate*.

Beispiel:

```
bitrate = 3500k;
```

 Encoder Systembitrate: 3500 kBit/s

Gültige Werte:

kleinster gültiger Wert	1000k
größter gültiger Wert	16000k

2.4.8 Parameter „audio bitrate“

Der Parameter *audio bitrate* legt die Audiobitrate des MPEG-Encoders fest. Abhängig vom Wert des Parameters *audio mode* sind unterschiedliche Werte möglich.

Beispiel:

```
audio bitrate = 192k;
```

 Encoder-Audiobitrate wird auf 192 kBit/s gesetzt.

Mögliche Werte:

	Audiobitrate													
<i>audio mode</i>	32 kBit/s	48 kBit/s	56 kBit/s	64 kBit/s	80 kBit/s	96 kBit/s	112 kBit/s	128 kBit/s	160 kBit/s	192 kBit/s	224 kBit/s	256 kBit/s	320 kBit/s	384 kBit/s
stereo				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
joint stereo				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
dual channel				✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
mono	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				

Gültige Werte:

kleinster gültiger Wert	32k
größter gültiger Wert	384k
Liste der gültigen Werte	32k, 48k, 56k, 64k, 80k, 96k, 112k, 128k, 160k, 192k, 224k, 256k, 320k, 384k
Standardwert	384k

2.4.9 Parameter „audio mode“

Der Parameter *audio mode* legt fest, in welcher Form der MPEG-Encoder das Audiosignal codiert.

Beispiel:

```
audio mode = mono;
```

 Das Audiosignal wird im Mono-Modus codiert.

Gültige Werte:

Liste der gültigen Werte	stereo, joint stereo, dual channel, mono
Standardwert	stereo

2.4.10 Parameter „audio sample rate“

Der Parameter *audio sample rate* legt fest, mit welcher Frequenz das Audiosignal abgetastet wird.

Beispiel:

```
audio sample rate = 44100;
```

 Das Audiosignal wird mit 44,1 kHz abgetastet.

Gültige Werte:

kleinster gültiger Wert	32000
größter gültiger Wert	48000
Liste der gültigen Werte	32000, 44100, 48000
Standardwert	44100

2.4.11 Parameter „audio pid“

Der Parameter *audio pid* legt fest, mit welcher PID die Audiodaten vom Encoder erzeugt werden. Soll für den Service kein Audio erzeugt werden, kann der Parameter *audio pid* auf *off* gesetzt werden.



Achtung: die für einen Service vergebene Audio-PID darf nicht mit den PMT und Video-PIDs des Services oder PIDs anderer Services übereinstimmen. Jeder Service benötigt eigene PIDs für PMT, Audio und Video.

Beispiel:

```
audio pid = 0x121;
```

Gültige Werte:

kleinster gültiger Wert	0x0020
größter gültiger Wert	0x1FFE
Liste der gültigen Werte	off und der Wertebereich von 0x0020 bis 0x1FFE
Standardwert	TS-Eingang 1: 0x121 TS-Eingang 2: 0x221 TS-Eingang 3: 0x321 TS-Eingang 4: 0x421

2.4.12 Parameter „video pid“

Der Parameter *video pid* legt fest, mit welcher PID die Videodaten vom Encoder erzeugt werden. Soll für den Service kein Video erzeugt werden (Radiokanal), kann der Parameter *video pid* auf *off* gesetzt werden.



Achtung: die für einen Service vergebene Video-PID darf nicht mit den PMT und Audio-PIDs des Services oder PIDs anderer Services übereinstimmen. Jeder Service benötigt eigene PIDs für PMT, Audio und Video.

Die PCR (Program Clock Reference) für den konfigurierten Service wird automatisch auf der gleichen PID wie die Videodaten erzeugt. Wird *video pid* auf *off* gesetzt (Radiokanal), muß mit dem Parameter *pcr pid* eine separate PID für die PCR-Daten eingestellt werden.

Beispiel:

```
video pid = 0x120;
```

Gültige Werte:

kleinster gültiger Wert	0x0020
größter gültiger Wert	0x1FFE
Liste der gültigen Werte	off und der Wertebereich von 0x0020 bis 0x1FFE
Standardwert	TS-Eingang 1: 0x120 TS-Eingang 2: 0x220 TS-Eingang 3: 0x320 TS-Eingang 4: 0x420

2.4.13 Parameter „pmt pid“

Der Parameter *pmt pid* legt fest, mit welcher PID die PMT (Program Map Table) für den vom Encoder codierten Service erzeugt wird. Soll für den Service keine PMT erzeugt werden, kann der Parameter *pmt pid* auf *off* gesetzt werden.



Achtung: die für einen Service vergebene PMT PID darf nicht mit den Video- und Audio-PIDs des Services oder PIDs anderer Services übereinstimmen. Jeder Service benötigt eigene PIDs für PMT, Audio und Video.

Beispiel:

```
pmt pid = 0x122;
```

Gültige Werte:

kleinster gültiger Wert	0x0020
größter gültiger Wert	0x1FFE
Liste der gültigen Werte	off und der Wertebereich von 0x0020 bis 0x1FFE
Standardwert	TS-Eingang 1: 0x122 TS-Eingang 2: 0x222 TS-Eingang 3: 0x322 TS-Eingang 4: 0x422

2.4.14 Parameter „pcr pid“

Der Parameter *pcr pid* legt fest, mit welcher PID die PCR (Program Clock Reference) vom Encoder erzeugt wird. Der Parameter *pcr pid* muß nur gesetzt werden, wenn der Parameter *video pid* auf *off* gesetzt wurde.



Achtung: die für einen Service vergebene PCR PID darf nicht mit den PMT- und Audio-PIDs des Services oder PIDs anderer Services übereinstimmen. Jeder Service benötigt eigene PIDs für PMT, Audio und Video.

Beispiel:

```
pcr pid = 0x120;
```

Gültige Werte:

kleinster gültiger Wert	0x0020
größter gültiger Wert	0x1FFE
Standardwert	TS-Eingang 1: 0x120 TS-Eingang 2: 0x220 TS-Eingang 3: 0x320 TS-Eingang 4: 0x420

2.4.15 Parameter „service id“

Der Parameter *service id* legt fest, welcher Service-ID für den vom Encoder erzeugten Service in PAT, NIT, SDT und PMT eingetragen wird.

Beispiel:

```
service id = 0x11;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0xFFFF
Standardwert	TS-Eingang 1: 0x11 TS-Eingang 2: 0x12 TS-Eingang 3: 0x13 TS-Eingang 4: 0x14

2.4.16 Parameter „service provider“

Der Parameter *service provider* legt fest, welcher Service-Provider-Name für den vom Encoder erzeugten Service in die SDT eingetragen wird. Wenn in der Konfigurationsdatei kein Abschnitt *sitables* vorkommt, oder der Parameter *generate sdt* auf *off* gesetzt wurde, spielt der Parameter *service provider* keine Rolle.

Beispiel:

```
service provider = "Yet another Service Provider";
```

Gültige Werte:

Standardwert	" "
--------------	-----

2.4.17 Parameter „service name“

Der Parameter *service name* legt fest, welcher Service-Name für den vom Encoder erzeugten Service in die SDT eingetragen wird. Wenn in der Konfigurationsdatei kein Abschnitt *sitables* vorkommt, oder der Parameter *generate sdt* auf *off* gesetzt wurde, spielt der Parameter *service name* keine Rolle.

Beispiel:

```
service name = "Yet another Service Name";
```

Gültige Werte:

Standardwert	" "
--------------	-----

2.4.18 Parameter „event name“

Der Parameter *event name* legt fest, welcher Event-Name für den vom Encoder erzeugten Service in die EIT eingetragen wird. Wenn in der Konfigurationsdatei kein Abschnitt *sitables* vorkommt, oder der Parameter *generate eit* auf *off* gesetzt wurde, spielt der Parameter *event name* keine Rolle.

Beispiel:

```
event name = "Yet another Event Name";
```

Gültige Werte:

Standardwert	" "
--------------	-----

2.4.19 Parameter „event text“

Der Parameter *event text* legt fest, welcher Event-Text für den vom Encoder erzeugten Service in die EIT eingetragen wird. Wenn in der Konfigurationsdatei kein Abschnitt *sitables* vorkommt, oder der Parameter *generate eit* auf *off* gesetzt wurde, spielt der Parameter *event text* keine Rolle.

Beispiel:

<code>service name = "Yet another Event Text";</code>

Gültige Werte:

Standardwert	" "
--------------	-----

2.4.20 Parameter „language“

Der Text.

Beispiel:

<code>language = ;</code>	TBD
---------------------------	-----

Mögliche Werte:

Wert	Wirkung

Gültige Werte:

kleinster gültiger Wert	
größter gültiger Wert	
Liste der gültigen Werte	
Standardwert	

2.4.21 Unterabschnitt „ca system“

Soll der Tabellengenerator eine CA-Section (Conditional Access Section) in die PMT zu dem auf diesem Encoder erzeugten Service einfügen, muß ein Unterabschnitt *ca system* in die Konfigurationsdatei aufgenommen werden.

Beispiel:

```
transportstream 1 = {  
    ...  
    ca system{  
        id = 0x1234;  
        pid = 0x0180;  
    };  
};
```

2.4.21.1 Parameter „id“

Der Parameter *id* im Unterabschnitt *ca system* legt fest, welche CA System-ID in die PMT eingetragen wird.

Beispiel:

```
id = 0x2600;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0xFFFF
Standardwert	0x2600 (die CA System-ID für BISS)

2.4.21.2 Parameter „pid“

Der Parameter *pid* im Unterabschnitt *ca system* legt fest, welche CA System ECM PID in die vom Tabellengenerator erzeugte CAT eingetragen wird.

Beispiel:

```
id = 0x1FFF;
```

Gültige Werte:

kleinster gültiger Wert	0x0000
größter gültiger Wert	0x1FFF
Standardwert	0x1FFF (die ECM PID für BISS)

2.4.22 Parameter „tuner frequency“

Der Parameter *tuner frequency* legt die Empfangsfrequenz für den angeschlossenen Tuner fest. Je nach Tuner sind unterschiedliche Wertebereiche gültig.

Beispiel:

```
tuner frequency = 1234k;
```

Mögliche Werte:

Tuner	gültiger Wertebereich	
	F_{min}	F_{max}
dvb-s		
dvb-c		
dvb-t		

Gültige Werte:

Liste gültiger Werte	siehe angegebene Wertebereiche in Tabelle „Mögliche Werte“
Standardwert	Der Parameter <i>tuner frequency</i> hat keinen Standardwert und muß für jeden Tuner gesetzt werden.

2.4.23 Parameter „tuner symrate“

Der Parameter *tuner symrate* legt die Empfangsfrequenz für den angeschlossenen Tuner fest. Je nach Tuner sind unterschiedliche Wertebereiche gültig.

Für einen DVB-T-Tuner hat der Parameter *tuner symrate* keine Bedeutung.

Beispiel:

```
tuner symrate = 1234k;
```

Mögliche Werte:

<i>Tuner</i>	<i>gültiger Wertebereich</i>	
	<i>SR_{min}</i>	<i>SR_{max}</i>
dvb-s		
dvb-c		

Gültige Werte:

Liste gültiger Werte	siehe angegebene Wertebereiche in Tabelle "Mögliche Werte"
Standardwert	Der Parameter <i>tuner symrate</i> hat keinen Standardwert und muß für DVB-S- und DVB-C-Tuner gesetzt werden.

2.4.24 Parameter „tuner modulation“

Der Parameter *tuner modulation* legt die Konstellation für einen angeschlossenen DVB-C-Tuner fest.

Für DVB-S- oder DVB-T-Tuner hat der Parameter *tuner modulation* keine Bedeutung.

Beispiel:

```
tuner modulation = qam16;
```

Gültige Werte:

Liste der gültigen Werte	qam16, qam32, qam64, qam128, qam256
Standardwert	qam16

2.4.25 Parameter „tuner fec“

Der Parameter *tuner fec* legt den FEC-Modus für einen angeschlossenen DVB-S-Tuner fest. Wird der Parameter *tuner fec* auf *auto* gesetzt, sucht der Tuner automatisch den passenden FEC-Modus. Die Zeit, die der Tuner zum Einrasten auf ein Signal benötigt, verlängert sich dadurch allerdings.

Für DVB-T- oder DVB-C-Tuner hat der Parameter *tuner fec* keine Bedeutung.

Beispiel:

```
tuner fec = auto;
```

Die passende FEC-Rate wird automatisch ermittelt.

Gültige Werte:

Liste der gültigen Werte	1/2, 2/3, 3/4, 5/6, 7/8, auto
Standardwert	Der Parameter <i>fec mode</i> hat keinen Standardwert und muß für DVB-S-Tuner gesetzt werden.

2.4.26 Parameter „tuner polarisation“

Der Parameter *tuner polarisation* legt fest, mit welcher Spannung der DVB-S-Tuner den angeschlossenen LNB versorgt und kann damit zur Auswahl einer Polarisationssebene genutzt werden.

Für DVB-T- oder DVB-C-Tuner hat der Parameter *tuner polarisation* keine Bedeutung.

Beispiel:

```
tuner fec = off;
```

Der Tuner erzeugt keine Spannung.

Mögliche Werte:

Wert	Wirkung
off	Der Tuner erzeugt keine Spannung
h	Der Tuner erzeugt eine Spannung von 18 V (horizontal)
v	Der Tuner erzeugt eine Spannung von 13 V (vertikal)

Gültige Werte:

Liste der gültigen Werte	off, h, v
Standardwert	off

2.4.27 Parameter „tuner band“

Der Parameter tuner band legt fest, ob der DVB-S-Tuner ein 22-kHz-Signal an den LNB sendet und kann damit zur Auswahl des Low- oder High-Bands genutzt werden.

Für DVB-T- oder DVB-C-Tuner hat der Parameter tuner band keine Bedeutung.

Beispiel:

```
tuner band = 1;
```

Mögliche Werte:

Wert	Wirkung
l	Der Tuner erzeugt kein 22-kHz-Signal (Low Band)
h	Der Tuner erzeugt ein 22-kHz-Signal (High Band)

Gültige Werte:

Liste der gültigen Werte	l, h
Standardwert	l

2.4.28 Parameter „tuner sat“

Der Text.

Beispiel:

```
tuner sat = ;
```

 TBD

Mögliche Werte:

Wert	Wirkung

Gültige Werte:

2.4.29 Parameter „tuner inversion“

Der Parameter *tuner inversion* legt fest, ob ein DVB-S- oder DVB-C-Tuner ein invertiertes Signal erwartet.

Wird der Parameter *tuner inversion* auf *auto* gesetzt, rastet der Tuner auf invertierte und nicht invertierte Signale. Die Zeit, die der Tuner zum Einrasten auf ein Signal benötigt, verlängert sich dadurch allerdings.

Für DVB-T-Tuner hat der Parameter *tuner inversion* keine Bedeutung.

Beispiel:

```
tuner inversion = ; TBD
```

Mögliche Werte:

Wert	Wirkung
auto	Automatische Auswahl
off	Der Tuner erwartet ein nicht invertiertes Signal.
on	Der Tuner erwartet ein invertiertes Signal.

Gültige Werte:

Liste der gültigen Werte	off, on, auto
Standardwert	auto

2.4.30 Parameter „tuner ofdm bandwidth“

Der Parameter *tuner ofdm bandwidth* legt die Empfangsbandbreite für einen DVB-T-Tuner fest.

Für DVB-C- und DVB-S-Tuner hat der Parameter *tuner ofdm bandwidth* keine Bedeutung.

Beispiel:

```
tuner ofdm bandwidth = 8; Empfangsbandbreite 8 MHz
```

Gültige Werte:

kleinster gültiger Wert	5
größter gültiger Wert	6
Liste der gültigen Werte	5, 6, 7, 8
Standardwert	8

2.4.31 Parameter „tuner ofdm mode“

Der Parameter *tuner ofdm mode* legt den FFT-Modus für einen DVB-T-Tuner fest.

Für DVB-C- und DVB-S-Tuner hat der Parameter *tuner ofdm mode* keine Bedeutung.

Im DVB-T-Signal wird der verwendete FFT-Modus in den Pilotträgern des Signal (TPS Bits) übertragen. Diese Übertragung funktioniert sehr zuverlässig – der Parameter *tuner ofdm mode* muß also nur in Ausnahmefällen gesetzt werden.

Beispiel:

```
tuner ofdm mode = 2k; 2k-FFT-Modus
```

Mögliche Werte:

Wert	Wirkung
2k	2k-FFT-Modus
8k	8k-FFT-Modus
auto	FFT-Modus wird automatisch eingestellt.

Gültige Werte:

Liste der gültigen Werte	2k, 8k, auto
Standardwert	auto

2.4.32 Parameter „tuner ofdm guard“

Der Parameter *tuner ofdm guard* legt die Länge des Guard-Intervalls für einen DVB-T-Tuner fest.

Für DVB-C- und DVB-S-Tuner hat der Parameter *tuner ofdm guard* keine Bedeutung.

Im DVB-T-Signal wird die Guard-Intervall-Länge in den Pilotträgern des Signals (TPS Bits) übertragen. Diese Übertragung funktioniert sehr zuverlässig – der Parameter *tuner ofdm guard* muß also nur in Ausnahmefällen gesetzt werden.

Beispiel:

```
tuner ofdm guard = 1/4; 1/4 Guard-Intervall
```

Gültige Werte:

kleinster gültiger Wert	1/32
größter gültiger Wert	1/4
Liste der gültigen Werte	1/4, 1/8, 1/16, 1/32, auto
Standardwert	auto

2.4.33 Parameter „pid filter“

Der Text.

Beispiel:

```
pid filter = ; TBD
```

Mögliche Werte:

Wert	Wirkung

Gültige Werte:

kleinster gültiger Wert	
größter gültiger Wert	
Liste der gültigen Werte	
Standardwert	

2.5 external program

Sollen zusätzlich zu den von den direkt am Board angeschlossenen Encodern Audio- und Videodatenströme in die vom Modulator bzw. Multiplexer erzeugten Tabellen aufgenommen werden, um z.B. Services von Tunern oder kaskadierten Multiplexern zu übernehmen, können diese als externe Programme in die Konfiguration eingetragen werden.

Dazu muss für jedes solche Programm ein Abschnitt *external program* in der Konfigurationsdatei stehen. Es können bis zu 16 externe Programme konfiguriert werden.

Beispiel:

```
external program {  
    video stream {  
    };  
    audio stream {  
    };  
    teletext stream {  
    };  
}
```

Definiert die verschiedenen Datenströme eines externen Programms.

2.5.1 Parameter „pid“

Der Parameter *pid* legt fest, auf welche PID für den Datenstrom in die PMT des externen Services eingetragen wird. Ein automatisches Remapping wird nicht durchgeführt – die hier angegebene PID muss also mit der PID des externen Datenstroms übereinstimmen.

Beispiel:

```
pid = 0x1234;
```

Gültige Werte:

kleinster gültiger Wert	0x0020
größter gültiger Wert	0x1FFE

2.5.2 Parameter „stream id“

Der Parameter *stream id* legt fest, welche Stream-ID für den Datenstrom in PMT und EIT des externen Services eingetragen wird. Die Stream-ID muss mit der in den PES-Paketen des Datenstroms eingetragenen Stream-ID übereinstimmen.

Für Services die von unseren Encodern erzeugt wurden, sind die Standardwerte korrekt. Der Parameter stream id muss also üblicherweise nicht angegeben werden.

Beispiel:

```
stream id = 0x01;
```

Gültige Werte:

kleinster gültiger Wert	0x01
größter gültiger Wert	0xFF
Standardwert	0x01 in einem Unterabschnitt video stream. 0x02 in einem Unterabschnitt audio stream. 0x03 in einem Unterabschnitt teletext stream. 0x04 in einem Unterabschnitt stream.

2.5.3 Unterabschnitt „video stream“

Beispiel:

```
video stream {  
    pid = 0x1234;  
    stream id = 1;  
    component type = 1;  
    language = "ENG";  
};
```

Definiert einen Videostream auf PID 0x1234

Mögliche Parameter:

Parameter	Beschreibung
pid	siehe Kap. 2.5.1
stream id	siehe Kap. 2.5.2
component type	siehe Kap. 2.5.3.1
language	siehe Kap. 2.5.3.2

2.5.3.1 Parameter „component type“

Der Parameter *component type* gibt an, welcher Component Type für den Datenstrom in der EIT (Event Information Table) des externen Services eingetragen wird. Mögliche Werte liegen zwischen 0x00 und 0xFF. Übliche Werte für Videostreams sind:

Beispiel:

```
component type = 0x01;
```

Mögliche Werte:

Wert	Wirkung
0x01	MPEG-2 video, 4:3 aspect ratio, 25 Hz
0x02	MPEG-2 video, 16:9 aspect ratio with pan vectors, 25 Hz
0x03	MPEG-2 video, 16:9 aspect ratio without pan vectors, 25 Hz
0x04	MPEG-2 video, > 16:9 aspect ratio, 25 Hz
0x05	MPEG-2 video, 4:3 aspect ratio, 30 Hz
0x06	MPEG-2 video, 16:9 aspect ratio with pan vectors, 30 Hz
0x07	MPEG-2 video, 16:9 aspect ratio without pan vectors, 30 Hz
0x08	MPEG-2 video, > 16:9 aspect ratio, 30 Hz
Für weitere Werte siehe ETSI EN 300 468, Tabelle 26.	

Gültige Werte:

Standardwert	0x01
--------------	------

2.5.3.2 Parameter „language“

Der Parameter *language* gibt an, welcher Sprachcode für den Datenstrom in der EIT (Event Information Table) des externen Services eingetragen wird.

Beispiel:

```
language = "deu";
```

Gültige Werte:

Liste der gültigen Werte	Möglich sind alle Zeichenketten mit genau drei Zeichen
Standardwert	"eng"

2.5.4 Unterabschnitt „audio stream“

2.5.4.1 Parameter „component type“

Der Parameter *component type* gibt an, welcher Component Type für den Datenstrom in der EIT (Event Information Table) des externen Services eingetragen wird.

Beispiel:

```
component type = 0x01;
```

Mögliche Werte:

Wert	Wirkung
0x01	MPEG-1 Layer 2 audio, single mono channel
0x02	MPEG-1 Layer 2 audio, dual mono channel
0x03	MPEG-1 Layer 2 audio, stereo (2 channel)

Gültige Werte:

kleinster gültiger Wert	0x00
größter gültiger Wert	0xFF
Liste der gültigen Werte	siehe ETSI EN 300 468, Tabelle 26
Standardwert	0x003

2.5.4.2 Parameter „language“

Der Parameter *language* gibt an, welcher Sprachcode für den Datenstrom in der EIT (Event Information Table) des externen Services eingetragen wird.

Beispiel:

```
language = "deu";
```

Gültige Werte:

Liste der gültigen Werte	Möglich sind alle Zeichenketten mit genau drei Zeichen nach ISO 639-2, z.B. deu für deutsch, eng für englisch, fra für französisch,...
Standardwert	"eng"

2.5.5 teletext stream

2.5.5.1 Parameter „component type“

2.5.5.2 Parameter „language“

Der Parameter *language* gibt an, welcher Sprachcode für den Datenstrom in der EIT (Event Information Table) und im Teletext-Descriptor der PMT des externen Services eingetragen wird.

Beispiel:

```
language = "deu";
```

Gültige Werte:

Liste der gültigen Werte	Möglich sind alle Zeichenketten mit genau drei Zeichen nach ISO 639-2, z.B. <i>deu</i> für deutsch, <i>eng</i> für englisch, <i>fra</i> für französisch,...
Standardwert	"eng"

2.5.6 stream

2.5.6.1 Parameter „stream type“

Der Parameter *stream type* legt fest, welcher Stream Typ für den Datenstrom in die PMT des externen Services eingetragen wird.

Der Parameter *stream type* ist nur in einem Unterabschnitt vom Typ *stream* erlaubt. Für Unterabschnitte der Typen *video stream*, *audio stream* und *teletext stream* wird der Streamtyp automatisch festgelegt.

Beispiel:

```
component type =  
0x01;
```

Mögliche Werte:

Wert	Wirkung
0x01	ISO/IEC 11172 Video
0x03	ISO/IEC 11172 Audio

Gültige Werte:

kleinster gültiger Wert	0x00
größter gültiger Wert	0xFF
Liste der gültigen Werte	Weitere Werte siehe ISO/IEC 13818-1, Tabelle 2-36
Standardwert	0x003

2.5.7 ca system

2.5.7.1 id

2.5.7.2 pid

Inhaltsverzeichnis

1	Aufbau	1	2.4.10	Parameter „audio sample rate“	16
1.1	Kommentare	1	2.4.11	Parameter „audio pid“	16
1.2	Selektoren	1	2.4.12	Parameter „video pid“	16
1.3	Attribute und Werte	2	2.4.13	Parameter „pmt pid“	17
2	Gültige Attribute und Wertebereiche	3	2.4.14	Parameter „pcr pid“	17
2.1	Abschnitt „board“	3	2.4.15	Parameter „service id“	18
2.1.1	Parameter „board“	3	2.4.16	Parameter „service provider“	18
2.1.2	Parameter „clock“	3	2.4.17	Parameter „service name“	18
2.2	Abschnitt „modulator“	4	2.4.18	Parameter „event name“	18
2.2.1	für Modulatoren	4	2.4.19	Parameter „event text“	19
2.2.1.1	Parameter „modulation“	4	2.4.20	Parameter „language“	19
2.2.1.2	Parameter „fec“	4	2.4.21	Unterabschnitt „ca system“	19
2.2.1.3	Parameter „constellation“	5	2.4.21.1	Parameter „id“	19
2.2.1.4	Parameter „symbol rate“	5	2.4.21.2	Parameter „pid“	20
2.2.1.5	Parameter „inversion“	6	2.4.22	Parameter „tuner frequency“	20
2.2.1.6	Parameter „frequency“	6	2.4.23	Parameter „tuner symrate“	20
2.2.1.7	Parameter „ptt“	6	2.4.24	Parameter „tuner modulation“	21
2.2.2	für Multiplexer	7	2.4.25	Parameter „tuner fec“	21
2.2.2.1	Parameter „drive ts clock“	7	2.4.26	Parameter „tuner polarisation“	21
2.2.2.2	Parameter „bit rate“	7	2.4.27	Parameter „tuner band“	22
2.3	Abschnitt „sitables“	8	2.4.28	Parameter „tuner sat“	22
2.3.1	Parameter „psi delay“	8	2.4.29	Parameter „tuner inversion“	23
2.3.2	Parameter „generate pat“	8	2.4.30	Parameter „tuner ofdm bandwidth“	23
2.3.3	Parameter „generate nit“	8	2.4.31	Parameter „tuner ofdm mode“	23
2.3.4	Parameter „generate tdt“	9	2.4.32	Parameter „tuner ofdm guard“	24
2.3.5	Parameter „generate sdt“	9	2.4.33	Parameter „pid filter“	24
2.3.6	Parameter „generate eit“	9	2.5	external program	25
2.3.7	Parameter „transport stream id“	9	2.5.1	Parameter „pid“	25
2.3.8	Parameter „network id“	10	2.5.2	Parameter „stream id“	25
2.3.9	Parameter „network name“	10	2.5.3	Unterabschnitt „video stream“	26
2.3.10	Unterabschnitt „ca system“	10	2.5.3.1	Parameter „component type“	26
2.3.10.1	Abschnitt „id“	10	2.5.3.2	Parameter „language“	26
2.3.10.2	Abschnitt „pid“	11	2.5.4	Unterabschnitt „audio stream“	27
2.4	Abschnitt „transportstream“	12	2.5.4.1	Parameter „component type“	27
2.4.1	Parameter „mode“	12	2.5.4.2	Parameter „language“	27
2.4.2	Parameter „clock edge“	13	2.5.5	teletext stream	28
2.4.3	Parameter „clock filter“	13	2.5.5.1	Parameter „component type“	28
2.4.4	Parameter „video input“	13	2.5.5.2	Parameter „language“	28
2.4.5	Parameter „video gop“	14	2.5.6	stream	28
2.4.6	Parameter „spatial filter“	14	2.5.6.1	Parameter „stream type“	28
2.4.7	Parameter „bitrate“	15	2.5.7	ca system	28
2.4.8	Parameter „audio bitrate“	15	2.5.7.1	id	28
2.4.9	Parameter „audio mode“	15	2.5.7.2	pid	28

Errata/corrections:

08.04.2008 Erster Entwurf zur Korrekturlesung.

Die Informationen in diesem Dokument sind mit größter Sorgfalt erstellt worden; dennoch können Fehler nicht vollständig ausgeschlossen werden. Wir übernehmen für möglicherweise daraus folgenden Fehlern und Problemen keinerlei Haftung.

Diese Anleitung ist durch die Urheberrechte geschützt. Alle Rechte, einschließlich Kopie, Übersetzung, Mikrofilm ebenso wie Speicherung und Verarbeitung in elektronischen Systemen, vorbehalten.

Kommentare und Fragen können Sie an uns richten:

The information in this manual was compiled with high care and to our best knowledge; nevertheless there might be some errors left in this document. We do not take legal or any other responsibility for the correctness of any information.

This document is protected by copyright law. All rights including copying, translation, microfilming as well as storage and processing in electronic systems are reserved.

We are happy to receive your comments and questions:

SR-Systems	Tel.: +49 (66 63) 91 88 66
Brüder-Grimm-Straße 130	Fax: +49 (66 63) 91 88 67
36396 Steinau	eMail: DATV@SR-Systems.de