

Aufbau von Registrationsdateien des TYROS Stand 28.08.2003

Eine Registrationsdatei des Tyros enthält die Daten einer Tyros-Registrations-Bank, bestehend aus acht Registrationen. Eine Bank hat einen Banknamen, jede Registration einen Registrationsnamen.

Der Dateiname hat folgenden Aufbau

<Bankname>.Sxxx.RGT xxx = Icon ID

oder bei fehlendem Icon

<Bankname>.RGT

Eine Registrationsdatei besteht aus einer Folge von Blöcken unterschiedlichen Typs.

Die ersten 6 Bytes jedes Blocks haben eine feste Struktur.

4 Bytes	Typ des Blocks, meist ASCII-Zeichen.
2 Bytes	Länge

Es gibt folgende Block-Typen gefunden:

SpfF	File-Header
BHdx x Hexa 00 oder 01	Sequenz- oder Registrationsblock
GPmx x Hexa 01 bis 2a	Setting-Group
FEnd	File-Trailer

File-Header

Die Datei wird eingeleitet durch den **SpfF-Block**, Länge 22 Bytes. Dieser beginnt mit „SpfF“

Beispiel:

```
00000000 5370 6646 0010 0AD9 5247 5354 0000 0007 SpfF....RGST....
00000010 0000 0B42 1534                               ...B.4
```

Byte 0x00 bis 0x0f anscheinend konstant

Die beiden Bytes 0x04, 0x05 = Anzahl der folgenden Daten, wohl immer 0x0010.

Die vier Bytes 0x10 bis 0x13 ist die Byteanzahl der gesamten Datei, hier 0x00000B42.

Anschließend folgt ein Sequenz-Block.

Sequenz- und Registrationsblock

Diese Blöcke beginnen mit einem Header von 6 Bytes:

„BHd“	3 Bytes ASCII
Typ	1 Byte, 0x00 (Registrationsblock) oder 0x01 (Sequenzblock)
Offset	2 Bytes, Offset (relativ zum darauffolgenden Byte) zeigt auf den Anfang des direkt folgenden Registrationsblocks oder auf den File-Trailer.

Sequenzblock

Der Sequenzblock ist nur einmal vorhanden. Er enthält die Daten für die Registrations-Sequenz. Er hat vermutlich eine feste Länge von 42 Bytes, d.h. File-Header und Sequenzblock belegen zusammen 0x40 Bytes.

Beispiel eines Sequenzblocks

```

00000000 4248 6401 0024 0005 0203 0706 0104 FF00 BHd..$. . . . .
00000010 0000 0000 0000 0000 0000 0000 0000 0000 . . . . .
00000020 0000 0000 0000 0000 0001 . . . . .

```

Ab Byte 0x06 stehen die Sequenznummern (Zählung ab 0). Endebyte FF. Die Sequenz des Beispiels ist demnach 1-6-3-4-8-7-2-5

Im letzten Byte (Adresse 0x29) steht der Code für Sequence End:

```

00 Sequence End Stop
01 Sequence End Top
02 Sequence End Next Bank

```

Der Standard Sequenzblock (jede Registration Stop) ist von Byte 6 bis Blockende mit FF belegt. Das Sequenz End Byte ist bei einer leeren Sequenz (Byte 6 enthält 0xff) irrelevant.

Registrationsblock

Beginn einer Registration, der Offset (relativ zum darauffolgenden Byte) zeigt auf den nächsten Registrationsblock oder auf den File-Trailer.

Beispiel:

```

00000000 4248 6400 0276 . . . . . BHd . . v

```

Eine leere, d.h. unbelegte, Registration ist durch den Offset 0x0000 gekennzeichnet.

File-Trailer

```

00000000 4645 6E64 0000 . . . . . FEnd . .

```

Struktur eines Registrationsblocks

Ein Registrationsblock beginnt mit dem Blockheader „BHd“+0x00. Anschließend folgen GPm-Blöcke unterschiedlichen Typs mit unterschiedlicher, häufig auch variabler Länge. Eine leere Registration enthält keine GPm-Blöcke, d.h. der Offset = 0 zeigt direkt hinter den Blockheader.

Auf den letzten GPm-Block einer Registration folgt entweder der nächste Registrationsblock oder der 6 Bytes lange File-Trailer „FEnd“+0x0000, mit dem jede Registrationsdatei beendet wird.

Eine Registrationsdatei enthält genau 8 leere oder nicht-leere Registrationen der Registrationsbank.

Zusammenfassung: Schematischer Aufbau der Registrationsdatei

File-Header „SPF“...

Sequenz-Block „BHd“+0x01

1. Registrationsblock „BHd“+0x00 (Registration 1), Länge verweist auf den 2. Registrationsblock
Unterschiedlich viele GPm-Blöcke mit aufsteigender Typ-Nummer
 2. Registrationsblock „BHd“+0x00 (Registration 2), Länge verweist auf den 3. Registrationsblock
Unterschiedlich viele GPm-Blöcke mit aufsteigender Typ-Nummer
 - .
 - .
 - .
 8. Registrationsblock „BHd“+0x00 (Registration 8), Länge verweist auf den File-Trailer
Unterschiedlich viele GPm-Blöcke mit aufsteigender Typ-Nummer
- File-Trailer „Fend“+0x0000 (Dateiende)

Globale Struktur der GPM-Blöcke

Art	rel. Byteposition	Byteanzahl	Bemerkung
„GPM“	0000	3 Bytes	Kennung
Typ	0003	1 Byte	es wurden Typen von 01 bis 2a beobachtet
Länge	0004	2 Bytes	Anzahl der folgenden Datenbytes
Daten	0006 und folgende	Variabel	Datenbytes

Funktion der GPM-Blöcke

Typ	Funktion des Blocks
01	Basic, Titel der Registration
02	Basic
03	Song
04	Song
05	Song
06	Noch nicht beobachtet
07	Standard Style
08	Style Attribute
09	Extern Style (Pfad)
0a	Voice global
0b	Voice Right1
0c	Noch nicht beobachtet (Voice R1)
0d	(Voice R1)
0e	Noch nicht beobachtet (Voice R1)
0f	Voice Right2
10	Noch nicht beobachtet (Voice R2)
11	(Voice R2)
12	Noch nicht beobachtet (Voice R2)
13	Voice Left
14	Noch nicht beobachtet (Voice Left)
15	?? (Voice Left)
16	Noch nicht beobachtet (Voice Left)
17	Voice Right3
18	Noch nicht beobachtet (Voice Right3)
19	Voice (Voice Right3)
1a	Noch nicht beobachtet (Voice Right3)
1b	Basic
1c	Basic
1d	Basic
1e	Basic
1f	Basic
20	Basic
21	Basic
22	Basic
23	Noch nicht beobachtet
24	MultiPad
25	MultiPad
26	Harmony
27	Tune Trans
28	Scale
29	Tempo
2a	Pedal
2b	MicSetting
2c	MicSetting

Typ	Funktion des Blocks
2d	MicSetting
2e	Noch nicht beobachtet
2f	LineOut

Setting-Groups und Gpm-Blöcke

Gpm	Style	Voice	TuneTr.	Scale	Harm.	Tempo	Pedal	MPad	Song	MicS	LineO.
01		X	X	X	X	X	X	X	X	X	X
02		X	X	X	X	X	X	X	X	X	X
03									X		
04									X		
05									X		
06											
07	X										
08	X										
09	X										
0a		X									
0b		X									
0c											
0d		X									
0e											
0f		X									
10											
11		X									
12											
13		X									
14											
15											
16											
17		X									
18											
19		X									
1a											
1b	X	X	X	X	X	X	X	X	X	X	X
1c	X	X	X	X	X	X	X	X	X	X	X
1d	X	X	X	X	X	X	X	X	X	X	X
1e	X	X	X	X	X	X	X	X	X	X	X
1f	X	X	X	X	X	X	X	X	X	X	X
20	X	X	X	X	X	X	X	X	X	X	X
21	X	X	X	X	X	X	X	X	X	X	X
22	X	X	X	X	X	X	X	X	X	X	X
23											
24								X			
25								X			
26					X						
27			X								
28				X							
29						X					
2a							X				
2b										X	
2c										X	
2d										X	
2e											
2f											X

Mit der Gruppe „Styles“ wird grundsätzlich auch die Voice „Left“ abgespeichert.

Struktur der GPM-Blöcke

Typ 01: Name der Registration

Beispiel:

```
00000000 4750 6D01 0009 4E65 7752 6567 6973 74 Gpm...NewRegist
```

Daten Name der Registration, hier: Voreinstellung

Der Name kann einen Icon-Zusatz haben: <name>.Snnn; z.B. „SpanishEyes.S917“.

Name+Icon sollten nicht weniger als 9 Zeichen haben, da der Platz im Tyros mit „NewRegist“ vorbelegt ist. Ggfs. sollte mit Blanks aufgefüllt werden. Bei Änderung des Namens am Tyros wird grundsätzlich eine Länge von 16 Zeichen eingestellt. Der Name wird mit Blanks auf diese Länge aufgefüllt. Wie Tests zeigen, reichen auch 9 Zeichen.

Die Namenlänge ohne Icon-Zusatz ist möglicherweise mit 13 begrenzt, was aber nur ein Darstellungsproblem auf dem Screen des Tyros ist.

Typ 02: Basic ???

Beispiel:

```
00000000 4750 6D02 0004 0301 0200 Gpm.....
```

Typ 03: Song

Beispiel:

```
00000000 4750 6D03 0001 00 Gpm....
```

Typ 04: Song

Beispiel:

```
00000000 4750 6D04 0012 6402 0202 0202 0202 0202 Gpm...d.....  
00000010 0202 0202 0202 02F0 .....
```

Typ 05: Song

Beispiel:

```
00000000 4750 6D05 0012 0000 413A 2F58 545F 5761 Gpm....A:/XT_Wa  
00000010 6C74 7A4D 2E4D 4944 ltzM.MID
```

Typ 06: ???

Beispiel:

Typ 07: Standard Style

Beispiel:

```
00000000 4750 6D07 000F 0220 7F0B 2A32 3737 030B Gpm.... ..*277..  
00000010 0000 0000 02 .....
```

Zwei Bytes ab Adresse 6: Doppelte Nummer des Styles

0xFFFF bei Nicht-Standard-Styles. In diesem Fall enthält GPM Typ 09 den Pfad der Style-Datei.

Nummern aus Testbeispielen

Pop&Rock - Live8Beat	0000	
Pop&Rock - Cool8Beat	0002	
Pop&Rock - 8BeatModem	0004	
Pop&Rock - 60's8Beat	0006	
Pop&Rock - 70's8Beat	0008	
Pop&Rock - 60'sGutarPop	000A	
Pop&Rock - Bubblegum	000C	
Pop&Rock - BritPop	000E	
Pop&Rock - KoolShuffle	003A	
Ballad - 8BeatBallad	003C	
Dance - Ibiza2000	0070	
World - Flamenco	021C	
World - PopEnka	0256	
Nicht-Standard-Style	FFFF	siehe GPM Typ 09

Ein Byte ab Adresse 0x10: 00 = Sync Start nicht gedrückt, 7f = gedrückt

Typ 08: Styleattribute

Beispiel:

```
00000000 4750 6D08 007D 6440 4040 4040 00FF 0000 Gpm..}d@@@@.....
00000010 0000 1D48 3639 294F 4232 0040 4040 2060 ...H69)OB2.@@@ `
00000020 4020 6000 2818 0020 2014 2222 0000 0000 @ `.(... .".....
00000030 0000 0000 0000 0000 0000 0000 0000 7F7F .....
00000040 7F7F 7F7F 7F7F 007F 0053 7F00 0008 0011 .....S.....
00000050 0073 1800 7318 0800 0108 0001 0800 0200 .s..s.....
00000060 4040 4040 4040 4040 0040 4040 4040 4040 @@@@@@@@@.@@@@@@@@
00000070 4000 4040 4010 1440 4040 0040 4040 7F7F @.@@@..@@@.@@@..
00000080 4040 40      @@@
```

Typ 09: Extern Style (Pfad)

Beispiel:

```
00000000 4750 6D09 001C 0000 433A 2F53 5459 4C45 Gpm....C:/STYLE
00000010 2F44 6973 636F 466F 782E 5337 3234 2E53 /DiscoFox.S724.S
00000020 5459      TY
```

Pfad des externen Styles.

Laufwerksbezeichner C: für UserFlash, E: für Festplatte, A: für Diskette

Typ 0A: Voice

Beispiel:

```
00000000 4750 6D0A 000A 4000 7F5F 0301 0701 0101 Gpm...@.._.....
```

Typ 0B: Voice Right1

Beispiel:

```
00000000 4750 6D0B 002A 7F00 741A 0076 6420 0000 Gpm...*.t..vd ..
00000010 4060 401E 0040 4040 1032 2F50 4040 4002 @`@.@@@.2/P@@@.
00000020 4040 4040 3F40 4009 0002 4040 0600 0000 @@@@?@...@.....
```

Byte 6: 00 nicht aktiviert, 7f aktiviert

Byte 7 MSB, Byte 8 LSB, Byte 9 Prog#

Typ 0C: ???

Beispiel:

Typ 0D: Voice

Beispiel:

00000000 4750 6D0D 000A 7F7F 0009 FFFF 1500 001A Gpm.....

Typ 0E: ???

Beispiel:

Typ 0F: Voice Right2

Beispiel:

00000000 4750 6D0F 002A 0000 **7338** 1A7B 6440 0000 Gpm...*.s8.{d@..
00000010 4061 4000 0040 4040 1046 2F4A 4040 4002 @a@...@@.F/J@@@.
00000020 3E40 4040 4040 400B 001E 404B 0000 0000 >@@@@@...@K....

Byte 6: 00 nicht aktiviert, 7f aktiviert

Byte 7 MSB

Byte 8 LSB

Byte 9 Prog#

Typ 10: ???

Beispiel:

Typ 11: Voice

Beispiel:

00000000 4750 6D11 000A 7F00 000B FFFF 0111 0026 Gpm.....&

Typ 12: ???

Beispiel:

Typ 13: Voice Left

Beispiel:

00000000 4750 6D13 002A 0000 **7234** 1E7F 3C40 0000 Gpm...*.r4..<@..
00000010 4069 4000 0040 4040 1040 2F40 4040 4002 @i@...@@.@/@@@@.
00000020 4040 4040 4140 4009 0005 4048 0000 0000 @@@@A@...@H....

Byte 6: 00 nicht aktiviert, 7f aktiviert

Byte 7 MSB

Byte 8 LSB

Byte 9 Prog#

Typ 14: ???

Beispiel:

Typ 15: ???

Beispiel:

00000000 4750 6D15 000A 7F00 000B FFFF 0111 0028 GPm.....(

Typ 16: ???

Beispiel:

Typ 17: Voice Right3

Beispiel:

00000000 4750 6D17 002A 0000 **7542** 187F 6E40 0000 GPm...*.uB..n@..
00000010 405D 4000 0040 4040 104A 2F46 4040 4002 @]@...@@.J/F@@@.
00000020 4040 4040 3F40 400A 000F 424C 0000 0000 @@@@?@...BL....

Byte 6: 00 nicht aktiviert, 7f aktiviert

Byte 7 MSB

Byte 8 LSB

Byte 9 Prog#

Typ 18: ???

Beispiel:

Typ 19: Voice

Beispiel:

00000000 4750 6D19 000A 7F00 000B FFFF 0111 0027 GPm.....'

Typ 1A: ???

Beispiel:

Typ 1B: Basic

Beispiel:

00000000 4750 6D1B 0007 FFFB FFFF 0111 40 GPm.....@

Typ 1C: Basic

Beispiel:

00000000 4750 6D1C 0007 FFFB FFFF 4210 40 GPm.....B.@

Typ 1D: Basic

Beispiel:

00000000 4750 6D1D 000F FFFB 0000 000A FFFF 4B08 GPm.....K.
00000010 007F 4001 7F ..@..

Typ 1E: Basic

Beispiel:

```
00000000 4750 6D1E 000D 007F 7F00 005E FFFF 5010 GPm.....^...P.  
00000010 001A 03                                     ...
```

Typ 1F: Basic

Beispiel:

```
00000000 4750 6D1F 000D 007F 0000 000B FFFF 0111 GPm.....  
00000010 0026 01                                     .&.
```

Typ 20: Basic

Beispiel:

```
00000000 4750 6D20 000D 007F 0000 000F FFFF 0112 GPm .....  
00000010 0028 02                                     .(.
```

Typ 21: Basic

Beispiel:

```
00000000 4750 6D21 000D 007F 007F 0009 FFFF 1500 GPm!.....  
00000010 001A 00                                     ...
```

Typ 22: Basic

Beispiel:

```
00000000 4750 6D22 000D 007F 0000 0009 FFFF 1500 GPm" .....  
00000010 0027 7F                                     .!.
```

Typ 23: ???

Beispiel:

Typ 24: MultiPad

```
00000000 4750 6D24 0008 FFFF 6440 4040 4040      GPm$. . . . d@ @ @ @ @
```

oder

```
00000000 4750 6D24 0008 0016 6440 4040 4040      GPm$. . . . d@ @ @ @ @
```

Ist Byte 6/7 = 0xffff, so ist GPm Typ 25 auszuwerten, anderenfalls enthält Byte 6/7 die MultiPad-Nummer. Die MultiPad-Nummer kann als Index in der MultiPad-Liste des Manuals benutzt werden. Beispiel: 0x0016 = dez. 22 = Index auf 23. Steel Strum1 .

Typ 25: MultiPad

Beispiel:

```
00000000 4750 6D25 0021 0000 433A 2F4D 554C 5449 GPm%. ! . . C : /MULTI  
00000010 2050 4144 2F46 7573 7320 4261 7373 2E53 PAD/Fuss Bass.S  
00000020 3931 302E 5041 44                                     910.PAD
```

Daten(rechtsbündig) Pfad der Multipad-Datei, eingeleitet durch 0000
Typ 25 nur vorhanden, wenn MultiPad-Nummer von GPm-Block Typ 24 = 0xffff

Typ 26: Harmony

Beispiel:

```
00000000 4750 6D26 0006 000B 4800 0000 GPm&....H...
```

Byte 6: 00 nicht aktiviert, 7f aktiviert

Byte 7 enthält den Code der eingestellten Harmony/Echo: 00 für Standard Duet, 01 für Standard Trio,... Hier 0x0B für Octave.

Vgl. die Tabelle im Manual.

Typ 27: TuneTrans

Beispiel:

```
00000000 4750 6D27 0003 4040 40 GPm'..@@@
```

Typ 28: Scale

Beispiel:

```
00000000 4750 6D28 0010 4040 4040 4040 4040 4040 GPm(..@@@@@@@@@@@  
00000010 4040 0000 0038 @@...8
```

Typ 29: Tempo

Beispiel:

```
00000000 4750 6D29 0007 0078 00AB 0078 7F GPm)...x...x.
```

Adresse 0x0A, 2 Bytes Tempo, hier 0x78 = 126

Typ 2A: Pedal

Beispiel:

```
00000000 4750 6D2A 0018 0003 0140 0000 4107 0100 GPm*.....@...A...  
00000010 0000 3A7F 0100 0000 0007 0140 0000 ..:.....@...
```

Typ 2B: MicSetting

Beispiel:

```
00000000 4750 6D2B 0004 5A40 1E00 GPm+..Z@..
```

Typ 2C: MicSetting

Beispiel:

```
00000000 4750 6D2C 0008 0000 001E FFFF 1500 GPm,.....
```

Typ 2D: MicSettingl

Beispiel:

```
00000000 4750 6D2D 000B 0010 0040 0100 7FFF FF59 GPm-.....@.....Y
```

00000010 00

Typ 2E: ???

Beispiel:

Typ 2F: LineOut

Beispiel:

```
00000000 4750 6D2F 002C 0000 0000 0800 0000 0000 Gpm/.,.....  
00000010 0000 0000 0000 0000 0000 0000 0000 0000 .....  
00000020 0000 0000 0000 0000 0000 0000 0000 0000 .....  
00000030 0000 ..
```