Memo 8434.102.01.001

route
28.10.1998

Thema: **Einstellen von CMS-Routings "zu Fuß"**

Verteiler: **ASZ, FAB, FN, HK, MK, MR, SH**
Verfasser: **FR**

Mit folgender Kommandozeile kann man (bei vorher gestartetem VCMS) ein Routing einstellen
(siehe auch unter q_control() im CMS-Manual !) :

`route` *`owner chandle filter`* [*`destination ...`*]

owner CMS-Name: entweder *`queue`* oder *`driver.device`*
chandle Zahl (0 oder von vorherigem route-Kommando ausgegebenes Controlhandle)
filter "*`filterbedingung`*" oder `-f` *`filename`* oder `null` (s.u.)
destination CMS-Queue- oder -Driver-Name(n) (Format wie *owner*)

Ein `route`-Aufruf bewirkt genau einen q/d_control-Aufruf. Das Ergebnis (Returncode) wird ausgegeben (<0 : Fehler, >0: Controlhandle - Es wird in Wirklichkeit Controlhandle-0x40000000 ausgeben, damit man kleine Zahlen erhält.)

Falls keine *destination* angegeben wird, entspricht das der leeren Liste, d.h. niemand bekommt die Nachrichten.

Mit
`route` *`owner chandle`* `null`
kann das Routing *chandle* gelöscht werden.

Es muss eine Filterangabe gemacht werden. Das ist eine einzelne Filterbedingung ('condition'), eine Liste von Filterbedingungen in einer Filterdatei oder `null` (d.i. false - unabhängig vom Nachrichteninhalt). Syntax:

condition ::= [label :] [type] [position compop value] [**true** | **false** | label]

Das Datenelement vom Typ type an der Position position in der Nachricht wird mittels des Vergleichsoparators compop mit value verglichen. Wenn der Vergleich TRUE ergibt, dann wird der hinter value stehende Wert als Ergebnis der Filterbedingung genommen. Steht dort ein Label, dann wird die Verarbeitung an der damit gekennzeichenten Bedingung fortgesetzt.

Ergibt dagegen der Vergleich FALSE, dann wird mit der Filterbedingung in der nächsten Zeile fortgefahren. Wenn es keine nächste Zeile gibt (z.B. bei Direktangabe im route-Kommando), dann das Ergebnis der Filterbedingung **false**, wenn in der vorherigen Zeile **true** oder label steht, ansonsten **true**.

Die Angabe **true** kann weggelassen werden, sie ist der Defaultwert für das Ergebnis. Wenn kein Typ angegeben wir, dann wird **byte** als Typ angenommen. Der ganze Vergleich kann ebenfalls entfallen, er wird dann als TRUE angenommen. Wenn die Bedingung direkt im route-Kommando steht, sind label nicht zulässig.

label ::= **<IDENT>**
type ::= **byte** | **uchar** | **u8** | **word** | **ushort** | **u16** | **dword** | **ulong** | **u32** |
char | **s8** | **short** | **s16** | **long** | **s32**

position ::= offset | **[**[type] indoffset**]** **+** offset

Die Postion ist der Abstand des ersten Bytes des zu testenden Elementes vom Anfang der Nachricht (in Bytes). Die Position kann entweder absolut als offset angegeben werden oder indirekt.

Bei der indirekten Angabe wird die Position aus einer Zahl, die in der Nachricht steht, bestimmt. Zu dieser Zahl kann noch ein Offset addiert werden. Dies entspricht der zweiten angegebenen Schreibweise: In eckigen Klammern steht die (absolute) Position dieser Zahl in der Nachricht, dabei kann noch der Typ angegeben werden - Default ist **byte**.

compop ::= **==** | **!=** | **<** | **<=** | **>** | **>=**
value ::= **<NUMBER>**
offset ::= **<NUMBER>**
indoffset ::= **<NUMBER>**

Zahlen (**<NUMBER>**) können dezimal oder hexadezimal (0x...) geschrieben werden, **-** als Vorzeichen ist möglich.

<IDENT> ist eine Folge von alphanumerischen Zeichen, die mit einem Buchstaben beginnt und nicht gleich einem der (oben fett gedruckten) Schlüsselwörter ist.

Außerdem sind noch folgende Abkürzungen erlaubt:

msgid ist eine Abkürzung für **word 10** .
data ist eine Abkürzung für **[0]** .

Filterdateien bestehen aus mehreren Filterbedingungen, jeweils eine pro Zeile. Außerdem sind noch Kommentarzeilen möglich, diese müssen mit einem **;** beginnen.
Beispiele:

1. Das Filter-Beispiel aus dem CMS-Manual sieht als Filter-Datei so aus:

```
; siehe q_control() im CMS 3 Handbuch
     uchar  0 !=  200 false
     ushort 2 <    22 false
     ushort 2 >    33 false
     ulong  4 == 1234 xyz
     ulong  4 != 4321 false
xyz: ulong  8 == 1234 true
     ulong  8 != 4321 false
```

2. Das folgende Kommando sorgt dafür, dass alle TCFG_TESTCASE-Nachrichten, die an GBFS geschickt werden, auch zu LOG gelangen:

```
route GBFS 0 "msgid == 0x0402" GBFS LOG
```

3. Mit folgendem Kommando und zugehöriger Filterdatei wird verhindert, dass MFS_ERROR-Nachrichten mit dem Errorcode -6 zu TEXE gelangen:

```
route TEXE 0 -f nomfserr.flt
```

nomfserr.flt:

```
       msgid != 0x0199 false
s16 data + 2 == -6     true
```

4. Mit diesem Kommando werden alle Nachrichten an GERR außerdem auch an LOG geschickt:

```
route GERR 0 "" GERR LOG
```